

UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

ISOPLOT200

A PLOTTING AND REGRESSION PROGRAM FOR
ISOTOPE GEOCHEMISTS, FOR USE WITH
HP SERIES 200 COMPUTERS

by

K. R. LUDWIG

OPEN-FILE REPORT 85-513

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards and nomenclature. Any use of trade names is for descriptive purposes only, and does not imply endorsement by the U.S. Geological Survey.

TABLE OF CONTENTS

<u>SUBJECT</u>	<u>PAGE</u>
Introduction -----	1
Hardware required by ISOPLLOT200 -----	1
Preliminary Remarks on Using ISOPLLOT200 -----	2
Getting Started -----	4
Loading the Program -----	4
Setting up the Plot -----	4
Choosing the Axis Names -----	6
Entering the Plotbox Limits -----	7
Defining the Physical Size of the Plot -----	7
Centered Plots -----	8
Uncentered Plots -----	8
Entering the Plot Limits -----	8
Plotting Data Points -----	9
The "SELECT PLOTTING-SYMBOL" display -----	9
Polygon Plotting-Symbol -----	9
Plotting Symbols that Indicate Errors -----	10
Entering Data from the Keyboard -----	11
The ADD Screen -----	12
Labelling Words or Phrases on the Plot -----	13
Storing and Retrieving Data from Data Files -----	13
Accessing Data Files from ISOPLLOT200 -----	14
Using Data from Data Files -----	16
Using Keyboard-Entered Data as a Temporary File --	16
Yorkfits of Data -----	17
Yorkfit "MODELS" -----	17
Obtaining isochron ages from Yorkfit lines -----	19
Re-using Yorkfit points; plotting error-envelopes ----	19
Selecting a Dashed or Solid Yorkfit Line -----	20
The OPTIONS Screen -----	20
Creating Concordia Plots -----	21
Obtaining Growth Curves for Pb-Isotope Plots -----	22
Dumping a CRT Plot to the Plotter or Printer -----	22
References -----	23
Figure Captions -----	25
Figures -----	26-28
Appendix: Program List of ISOPLLOT200 -----	29

ISOPLOT200 - A PLOTTING AND REGRESSION PROGRAM FOR ISOTOPE
GEOCHEMISTS, FOR USE WITH HP SERIES 200 COMPUTERS

by

Kenneth R. Ludwig

INTRODUCTION

ISOPLOT200 is a program designed primarily for use by isotope geologists, but of value for anyone who wants to plot X - Y data in a simple, rapid, and flexible manner. Data can be entered either from the keyboard or from datafiles produced by the commercially-available VISICALC program. Data can be plotted as a wide variety of symbols, including various error-symbols and polygons. ISOPLOT200 can regress data using a modification of the most widely-used two-error regression algorithm, and will calculate isochron ages or concordia-intercept ages directly from the regression lines. U-Pb isotope concordia plots (together with intercepts and errors of regression lines) are a built-in feature, as are single-stage growth-curves for Pb-isotope plots. ISOPLOT200 is intended to be simple to use, and so is provided with HELP screens, extensive error-trapping, and detailed prompts. As a result, most users will seldom need to use this manual.

The pedigree of ISOPLOT200 extends back several years, starting from programs for HP-9830/9831 computers (Ludwig, 1979a, 1979b, 1982). The most recent precursors are a group of programs written for HP-86/87 computers (Ludwig, 1983). Because of the much greater computational speed of the HP Series 200 computers, together with a much more powerful BASIC language superset and greater memory, it seemed worthwhile to develop a much more user-friendly version that would take advantage of the hardware advances. Hence ISOPLOT200.

HARDWARE REQUIRED BY ISOPLOT200

ISOPLOT200 was written for Hewlett-Packard Series 200 computers (Models 216, 217, 236, 9816, 9817, 9836). The required BASIC language version is either BASIC 2.0 or BASIC 3.0 (I've tried only the RAM-based versions), with the necessary language extensions varying according to the version (see below). You should have at least 750 kilobytes of RAM (before loading the BASIC language). The program requires an HP-IB printer (at address 701) and an HP-7475A 6-pen plotter (at address 705). For the

BASIC 2.0 language, you must also load (using the LOAD BIN command) the two binary programs AP2_1 and GRAPH2_1, sold by HP as language extensions. For the BASIC 3.0 language, the following binary language-extension programs are required: GRAPH, GRAPHX, MAT, CLOCK, KBD, ERR, KNBZ_0, CS80, DISC, HPIB, and MS.

The program assumes that the keyboard is the OPTION 805 (ASCII extended keyboard character set) keyboard -- the one that is identical to the standard keyboard of the HP-9836. I assume that the standard Series 200 keyboard would work with little or no modification, but haven't actually tried one myself.

The programs were developed using a dual single-sided 3.5" disk drive (model HP-9121), and should work equally well with the HP82905B 5.25" disk drive. The program itself is compatible with the HP-9122 double-sided 3.5" disk drive: however, HP has not made the VISICALC program available for the double-sided disk drives, nor will they continue selling VISICALC for the Series 200 at all. You can't even get a version of VISICALC for the HP-9121 copied onto disk that can be read by the HP-9122 drive. So for the present, if you use the program with the HP-9122 double-sided 3.5" disk drive, you'll have no way of constructing datafiles that ISOPLOT200 can read - unless you want to write a program to mimic the VISICALC storage format yourself. Data files created by PBDAT200, a Pb-U-Th isotope data-reduction program (Ludwig, in press) are compatible without recourse to VISICALC, however.

I've looked at CONTEXT MBA, a combination spreadsheet/word processing/file management/communications program that's supposed to be compatible with the HP-9122. Unfortunately, the documentation is opaque and the program itself much too complex for easy use, so I'm not sure whether I'll ever have the time to modify ISOPLOT200 to use CONTEXT MBA spreadsheet datafiles. Or whether any users would bother with it either.

PRELIMINARY REMARKS ON USING ISOPLOT200

First, you should understand some of the basic features of the computer -- the keyboard editing keys (left, right, up, and down arrows), what the softkeys (k0 - k9) are for, and how to answer a query by the program. So if you have the time, read the introductory manual to the computer.

GETTING STARTED

LOADING THE PROGRAM

To load the program (the BASIC language and required extensions must already be in memory, of course), just put the ISOPLOT200 disk in the default drive (the one that lights up when you type CAT, then press EXECUTE), type in LOAD "ISOPLOT",1 and wait about a minute. The program will start itself.

SETTING UP THE PLOT (X-Y plots)

The first CRT-display of the program (the one that appears when you press the RUN key) will look something like this:

Rev. July 5, 1985 K.R. Ludwig, U.S. Geol. Survey

<<<<<<<<<< ISOCHRON or X-Y PLOTTER >>>>>>>>>>>>>>

CRT PLOT

CRT OUTPUT

Press CTRL C for concordia plot

PRINTER OUTPUT	PLOTTER PLOT			HELP
			GET DATAFILE	START

In this section, I'll discuss the use of the ISOCHRON or X-Y PLOTTER. If the display says CONCORDIA PLOTTER at the top, press CTRL I. If the display says PLOTTER PLOT at the upper left, press k1, and if the display says PRINTER OUTPUT at the upper right, press k0.

First, select where your plot and your printed output are to be sent. Softkey k0 determines whether the printed output is to be sent to the printer or the CRT, and toggles between the two. Look at the upper right of the CRT to see what the current status is. Notice that the k0 label will always be the opposite. This is because the softkey labels indicate what the softkey will do when pressed, not what the current status is.

Next, select where you want the plot to appear -- on the CRT or the plotter. The current status is indicated at the upper left of the CRT, and is changed (toggles from CRT to plotter) by pressing k1.

If you select the plotter as the plotting device, several more softkeys will be labelled on the bottom of the CRT, as shown below:

```

-----
|PRINTER OUTPUT| CRT PLOT   | CHOOSE PEN | STACKED PLOT|   HELP   |
|-----|-----|-----|-----|-----|
|ADJ. LABELSIZE| ROTATED PLOT | PEN SPEED  | GET DATAFILE| START PLOT |
-----

```

To select one of the 6 pens of the HP-7475A plotter, press k2. One of the bottom lines of the CRT will then ask you to choose a pen.

To select a pen speed (how fast the pen moves while drawing lines), press k7. Use the fastest speed (10) for draft plots, slower for final or publication quality plots. The default speed is the fastest.

To vary the relative sizes of the axis and tick labels of the plot, press k5. The default size (1) is suitable for plots that will not be reduced, but larger sizes are best for plots that will be made into 35 mm transparencies or will be significantly reduced during reproduction.

If you select either a slow pen-speed or large labels, ISOPLOT200 will draw each part of the plot twice (slightly offset the second time to give thicker lines) to improve its clarity.

If you want the plot to be drawn vertically on the paper instead of horizontally, press k6 (ROTATED PLOT).

If you want to stack plots so that they share the same X-axis (and X limits), construct the lower plot as usual, but with specifically-defined limits for the plot-box (see below). Then for the upper plot, press k3 (STACKED PLOT). Position the plot-box precisely over the lower box. Because you specified a STACKED PLOT, the upper plot will be drawn without labels for the X-axis or the X-axis ticks, and without the first Y-axis tick-label.

Choosing the Axis Names:

When you've finished selecting the physical parameters of the plot with the softkeys, as described above, press the START softkey (k9). The CRT display will then be:

...Use ^ to start superscripts or end subscripts,
* to start subscripts or end superscripts

| ENTER X-AXIS NAME, Y-AXIS NAME: |

(press CONTINUE to use PPM Sr and PPM Rb)

Shorthand for common axis-names:

<u>SR</u>	-- Rb-Sr isochron	<u>ND</u>	-- Sm-Nd isochron
<u>PBA</u>	-- Beta-Alpha plot	<u>PBG</u>	-- Gamma-Alpha plot
<u>UA</u>	-- Alpha-Mu isochron	<u>UB</u>	-- Beta-Nu isochron

The axis names are the names that will be labelled under the X-axis and to the left of the Y-axis. You can respond to the query in 3 different ways: (1) by typing in the X-axis name and the Y-axis name (separated by a comma) and pressing the ENTER key, (2) by pressing CONTINUE to re-use whatever names were used in the last plot (PPM Sr and PPM Rb in the above example), or (3) entering one of the "shorthand" codes indicated in the screen: SR to obtain the names for an Rb-Sr isochron plot, PBA for a $^{206}\text{Pb}/^{204}\text{Pb} - ^{207}\text{Pb}/^{204}\text{Pb}$ plot, and so on.

You can specify superscripts or subscripts in the axis names by using the ^ symbol to start superscripts (or cancel subscripts), and the * symbol to start subscripts (or cancel superscripts). So if you wanted the axis names looking like

$$^{87}\text{Sr}/^{86}\text{Sr} \quad \text{and} \quad \text{Al}_2\text{O}_3$$

you would enter the names this way:

$$^87*\text{Sr}/^86*\text{Sr} \quad \text{and} \quad \text{Al}*2^0*3$$

Entering the Plotbox Limits:

After you've entered the axis names for the plot, the CRT will display:

ENTER X AND Y LIMITS ([Xmin,] Xmax, [Ymin,] Ymax)?

(press CONTINUE to re-use 0, 14, .7, 1.2)

Your response to this query defines what the minimum and maximum X and Y values of the plotbox will be. You can enter just 2 values, in which case the X and Y minimum values will be assumed to be zero, or 4 values if at least one of the minimum limits is nonzero. If you just want the same limits as were used for the last plot, though, just press CONTINUE without entering any values. At this point, if your plot is to be drawn on the CRT, the computer will construct the plot box, tick-labels, and axis labels.

The format for entering values in response to a query by ISOPLOT200 is similar to the above example throughout the program. Values for prompts in brackets are optional -- a default value will be used if you don't enter a number for them. To enter several values at once (as indicated in the prompt), just separate the values by commas. And, for many queries, there is a default value or set of values that will be assumed if you just press the CONTINUE key without typing in a specific response.

Defining the Physical Size of the Plot:

If you requested that your plot be drawn on the plotter (rather than the CRT), you'll have to specify the size of the plot. The CRT query will be:

ENTER PLOT-SIZE (3-10)?

(Press CONTINUE for default size of 8)

(Press k4 for nonstandard dimensions/locations)

Centered plots:

If you want a plot with a "normal" height-to-width ratio (about the same as an 8.5" by 11" sheet of paper) and centered in the paper, enter a number from 3 to 10 in response to the above query. A value of 10 gives the largest plot that will fit on a 8.5" by 11" sheet of paper, and a value of 3 a plot only 30% as large. Intermediate values give proportionately intermediate sizes. The default size is 8.

Uncentered plots:

If you want a plot with a different height-to-width ratio (perhaps only 5 cm high by 10 cm wide), or located off-center in the paper, press softkey k4. The CRT query will then be:

PRESS k0 TO DEFINE THE PLOT-LIMITS FOR THE ENTIRE FIGURE AREA,
k4 TO DEFINE LIMITS FOR THE PLOT-BOX ONLY

If you want to define the area that the total plot will occupy, including tick-labels and axis names, press k0. If you want to specify the location and area of the plot in terms of the plot box, however (the plot box is the box outlining the area where the data points are actually plotted), press k4. The latter option is particularly useful for cases where you are putting two or more plots on a single sheet of paper, and essential if you want to stack two plots that share the same X-axis.

Entering the plot limits:

Once you've selected how you want to define the location and size of the plot, the CRT will query:

ENTER PLOT-LIMITS IN mm: X-MIN, X-MAX, Y-MIN, Y-MAX?

(max. limits are 0-249 [X] and 0-179.5 [Y],
 lower-left corner of paper is 0,0)

Enter 4 values, separated by commas, to define the location of the lower-left and lower-right limits of the plot (X-MIN, X-MAX), and the upper-left and upper-right limits of the plot (Y-MIN, Y-MAX). The lower-left corner of the paper is defined as X=0, Y=0. Enter your values in millimeters. The computer will now start constructing your plot.

PLOTTING DATA-POINTSTHE SELECT PLOTTING-SYMBOL DISPLAY

After ISOPLOT200 has drawn the plot box, tick-labels and axis-names, you're ready to start plotting your data points. First, though, you must select a plotting symbol. The CRT display will be:

SELECT A PLOTTING SYMBOL TO PLOT DATA

press CTRL L to draft a phrase on the plot

press CTRL P to dump plot to printer

press CTRL D to dump plot to plotter

press CTRL H for HELP

press any key for keyboard plotting-symbol

```

|-----|-----|-----|-----|-----|
| ERROR BOX | ERROR ELLIPSE | ERROR CROSS | POLYGON | OPTIONS |
|-----|-----|-----|-----|-----|
| SOLID LINE | CHOOSE PEN | YORKFIT | GET DATAFILE | NEW PLOT |
|-----|-----|-----|-----|-----|

```

I'll refer to this display as the "SELECT PLOTTING SYMBOL" display. The bottom rows show the softkey definitions.

You can do several things from this part of the program besides just selecting a plotting symbol, but for now we'll ignore them. The simplest type of plotting symbol to select is just an alphanumeric one - that is, using one of the keyboard symbols, such as +, *, X, O, or A, H, -S, #. To select a keyboard symbol, just press that key, and the computer will go on to ask you for the data itself.

POLYGON PLOTTING-SYMBOL

You can use a polygon for a plotting symbol, with any number of sides, size, and rotational orientation, either open or filled. Press the softkey labelled POLYGON (unshifted for an open polygon, shifted for a filled polygon). The CRT will then query:

POLYGON-SYMBOL: ENTER #SIDES (- for star),
 [,SIZE (mm) [,ROTATION F(0-360)]

Press CONTINUE to re-use last values (20,2.5,0)

As usual, the parameters in brackets represent optional ones, so you can enter 1, 2, or 3 values (or press CONTINUE to re-use the last values entered). The number of sides can vary from 3 (a triangle) to a large number (in effect, giving a circle). The SIZE parameter refers to the diameter of the polygon, in millimeters. Default is 2.5 mm. The ROTATION parameter refers to the angular orientation (counter-clockwise from the X-axis) of one of the vertices of the polygon: 0 gives a vertex at 12 o'clock, 45 a vertex at 1:30 o'clock, 180 a vertex at 9 o'clock, and so on. The default value depends on the number of sides of the polygon, and gives "conventional" orientations (triangle pointing straight up, square with flat side down...).

If you enter the number of sides of the polygon as a negative value, the polygon will be drawn "puckered in" at each side, to give a "star"-shaped figure (figure 1). So if you enter the number of sides as -5 instead of 5, the plotting-symbol will be a 5-pointed star.

PLOTTING-SYMBOLS THAT INDICATE ERRORS

There are 3 plotting symbols that you can use to indicate the errors of the points on the plot: error-box, error-cross, and error-ellipse. If you select one of these symbols, ISOPLLOT200 will need to know estimated errors for each of your points. If you select the error-ellipse symbol, ISOPLLOT200 will also need the correlation between the X- and Y-errors. If you don't know how to estimate the error-correlation, the following relationship may help:

$$\text{Rho} = (E_x^2 + E_y^2 - E_z^2) / (2E_x E_y)$$

where Rho is the correlation between the X and Y errors, E_x , E_y , and E_z are percent errors in X, Y, and Z, respectively, and $Z = Y/X$. Error correlations are zero if the error in X doesn't tend to increase with the error in Y. If the error correlations are much greater than about 0.5, don't use error-box or error-cross symbols, because these symbols will convey a misleading impression as to the actual area of uncertainty on the plot.

To use an open error-box or error-ellipse, just press the labelled softkey. To get filled error-box or error-ellipse symbols, press the labelled softkey with the SHIFT key.

ENTERING DATA FROM THE KEYBOARD

As soon as you select a plotting symbol, the CRT will clear and display:

PLOTTING-SYMBOL IS *

2-SIGMA %-ERRORS

INPUT DATAFILE SET #s AS 1st SET, LAST SET-ASTERISK

(e.g. 2.12* or 2.12:15.18*)

[use *L or *R to label data-file name to (L)eft or (R)ight of point]

SET#	87Rb/86Sr	%err	87Sr/86Sr	%err	RHO
------	-----------	------	-----------	------	-----

#1: 87Rb/86Sr, [%err,] 87Sr/86Sr [,%err] [,err-corr.] (CONT WHEN DONE)

The axis names and plotting symbols are only examples, of course.

To enter the X- and Y-values for your data points, just type in the two numbers, separated by a comma, then press the ENTER key. To include the errors with the points (necessary for error plotting-symbols or for a later Yorkfit), enter 4 values separated by commas, in the format X, X-error, Y, Y-error. The default error-format is at the 2-sigma level and in percent (but can be changed: see a later section). If no error correlations are entered (an optional 5th value), they are assumed to be zero. If the errors for all of the points to be entered are the same, don't bother entering them at this point - you'll be asked to enter them as a group later.

When you've entered all of the points that you want plotted with the particular plotting-symbol or pen color that you chose earlier, press CONTINUE. If you're using an error-symbol (error box, error cross, or error ellipse) as the plotting symbol and you didn't enter values for some of the errors, the computer will query:

ENTER X-%err, Y-%err [,Err.-Corr.] FOR SETS WITH ZERO ERRORS?

Enter the values requested. Note that, again, the error correlation is an optional value, and also that the assigned errors must be nonzero.

The computer will then display a list of the data-point values that you just entered, and query:

SET TO BE CORRECTED? (CONTINUE IF OK)

Check the displayed list for errors, and if you made any, enter the set# (shown to the left of the X- and Y-values) of one of the incorrect data points. Enter the correct values when requested, then continue with the editing process until all of the values are correct. Press CONTINUE, and the data points will be plotted.

THE ADD SCREEN

After the data points are plotted this display will appear on the CRT:

PRESS GRAPHICS TO VIEW PLOT, ALPHA TO VIEW THIS SCREEN

PRESS ADD TO INCLUDE THESE POINTS WITH OTHERS FOR A YORKFIT

PRESS DELETE TO DELETE A POINT FROM THE LAST SET

PRESS NEW POINTS TO PLOT A NEW BATCH OF POINTS

PRESS CTRL L TO DRAFT A PHRASE ON THE PLOT

PRESS YORKFIT FOR YORKFIT

PRESS NEW PLOT TO START A NEW PLOT

(press NEW POINTS if you just want to return to the data-entry screen)

At this point, you can choose to plot more points, do a Yorkfit, or abandon the current plot and start a new one. If your plot is on the CRT, you can toggle between the graphics display and the alphanumeric display (the one above) with the GRAPHICS and ALPHA keys to the upper-right of the keyboard. If you press the NEW POINTS softkey, the program will return to the "SELECT PLOTTING-SYMBOL" display, and any additional data points that you enter will be counted as a new batch for regression-line purposes.

But if you want to plot more points (perhaps with a different plotting-symbol or pen color) that are to be pooled with the just-plotted points for a later Yorkfit (or other regression), press the ADD softkey.

If you don't want to include (for subsequent Yorkfit or other regression) one or more of the points that you just plotted, press the DELETE softkey. The CRT will ask you which point to delete.

LABELLING WORDS OR PHRASES ON THE PLOT

You can have the program draft any phrase you want anywhere on the plot by pressing the CTRL L key from the "SELECT PLOTTING-SYMBOL" display or the "ADD POINTS" display. The CRT will query:

(PRESS CONTINUE TO ESCAPE)

CHAR.-HEIGHT? (axis tick-labels were 4) [, ROTATION]

Enter at least one value: the height of the letters in millimeters. You can specify the rotational orientation (counterclockwise from the X-axis) of the phrase with an optional second value. So an input of

5,90

specifies a phrase whose characters are about 5 millimeters high and oriented vertically (parallel to the Y-axis. If you entered just 5, the phrase would be oriented horizontally.

After selecting the size and rotation of the phrase, the CRT will instruct you to move the cursor (for a CRT plot; the cursor is a cross-shaped symbol) or the pen (for a plotter plot) with the KNOB to indicate where the lower-right corner of the phrase is to be located. Or you can type in a specific X-Y coordinate by pressing the ENTER X-Y softkey. Once you've specified the location of the phrase, just type it in and enter it with the ENTER key. You can specify superscripts or subscripts with the ^ and * keys, just as in the axis labels. You can also specify the \pm symbol with the characters +/-.

STORING AND RETRIEVING DATA FROM DATA FILES

You can use the commercially available VISICALC program to construct, edit, and store datafiles that can be read by ISOPLOT200. The VISICALC data file must conform to a specific format to be compatible with ISOPLOT200, however. This format requires that:

- 1) the column-width be 9 characters (the VISICALC default);

- 2) the column names must appear above any data, and directly overlie a line of repeating equals-signs (=====);
- 3) the column names may occupy up to 2 cells above the repeating equals-signs;
- 4) the first 2 columns (A and B) are used for sample names only;
- 5) no more than 200 rows and 50 columns are allowed;
- 6) the first row of the spreadsheet can be used as a file title (up to 80 characters) to identify the data file;
- 7) the file must be stored as a /PF file, not as a /SS file.
- 8) the file must be stored while the PRINTER WIDTH is defined as 80.
- 9) the file name (as stored on the disk) can include only the characters A through Z (caps or lower case) and numbers 0 through 9.

Because /PF files can't be edited, you should also store the VISICALC file in the normal way - as an /SS file. To make sure that the printer width is 80, type /PN, enter 7 (the printer select-code), enter 1 (the HPiB address), then enter 80.

ACCESSING DATA FILES FROM ISOPLOT200

To get data from a data file, press the softkey labelled GET DATAFILE from either the initial display or the "SELECT PLOTTING SYMBOL" display. The CRT will then display:

Press k0 for a catalog of the disk in the left-hand drive

Press k1 for a catalog of the disk in the right-hand drive

Press k2 or CTRL_H for HELP

Press k3 to print out the datafile-data on the printer

Press k4 to load a Visicalc datafile into memory

Press k9 to escape.

(No datafile currently in memory)

DISPLAY ON CRT ONLY

```

|-----|-----|-----|-----|
|CATALOG Drv #0|CATALOG Drv #1|   HELP   | PRINT DATA |LOAD DATAFILE|
|-----|-----|-----|-----|
|               |               |          | SCRATCH FILE |   ESCAPE   |
|-----|-----|-----|-----|

```

To access data from a data file, if you're sure of its name, press k4. If you're not sure of the name of the file you want, or if you want to see what files are present on a disk, press either k0 or k1. The CRT will then show the names of all of the files on that disk, and indicate those files which are compatible with ISOPLOT200 by highlighting their names. You can then choose one of the compatible files just by entering the number of that file, as shown on the CRT.

Once you've chosen a data file to access, the computer will search for and display a list of all of the column names (headings) for that file. You must then select either 2 columns (X and Y), 4 columns (X, X-error, Y, Y-error), or 5 columns (X, X-error, Y, Y-error, error-correlation) to get data from.

The computer will then get the data from the disk, and printout the values either on the CRT (default) or the printer (if you pressed k3 on the initial GET-DATAFILE display to specify the printer as the datafile output device).

USING DATA FROM DATA FILES

To plot data from a data file, you need only refer to the data by "set numbers", where the set number of a data point corresponds to its row in the VISICALC spreadsheet, and is printed out when the data file is accessed by ISOPLOT200. For example, after you've selected a plotting symbol, the CRT display will be:

PLOTTING SYMBOL IS ERROR-ELLIPSE

2-SIGMA X-ERRORS

INPUT DATAFILE SET #s AS 1st SET, LAST SET-ASTERISK

(e.g. 2,12* or 2,12:15,18*)

[use *L or *R to label data-file name to (L)eft or (R)ight of point]

To plot the data for set number 5, enter 5*; to plot the data for sets 5,6,7,8,9, and 10, enter 5,10*. To plot data for sets 5,6,7,8, and 15,16,17, enter 5,8:15,17*. In other words, indicate continuous sequences of sets with a comma, separate different continuous sequences with a semicolon, and always add an asterisk at the end. It's OK to mix data file and keyboard-entered data, of course.

You can have the computer label each point with the appropriate sample names (from columns A and B in the spreadsheet) by adding an L or an R to the right of the asterisk. An L will result in the names being labelled to the left of the data point, an R to the right of the data point. It's up to you, of course, to beware of any over-writing that this causes.

Using Keyboard-Entered Data as a Temporary Data File:

If you're entering data-points from the keyboard, and would like to temporarily "file" those data-points for subsequent re-plotting (perhaps you didn't get everything perfect the first time?), you can do so in the following way. First, press the GET DATAFILE softkey (present in the initial CRT display and the "SELECT PLOTTING-SYMBOL" display). If the GET DATAFILE display indicates that there is a data file currently in memory, press the SCRATCH FILE softkey. ESCAPE from this screen.

With no data file in memory, each data point that you enter is added to a temporary data file, in the sequence that you enter the points. Only the last batch of data that you enter will be saved, so that the file will consist only of N points, where N is the highest set-number that you entered. In other words, if you type in the X-Y values for 20 data points, press the NEW POINTS key, then type in another 10 X-Y values, the temporary data-file will consist of the second set of 10 points plus the first-entered points 11 through 20.

To re-plot these points, just treat them as points in a data file with sample numbers 1 through 20.

YORKFITS OF DATA

YORKFIT "MODELS"

A "Yorkfit" refers to a linear regression using the general algorithm developed by Derek York (York, 1969), that weights each point according to both its X- and Y-errors and its X-Y error correlation. ISOPLOT200 uses both the original York algorithm and a few modifications, depending on the type of data that you are working with and the amount of scatter that the data shows about a straight line. Each of these modifications, called MODELS by ISOPLOT200, makes different assumptions about the reason for the scatter of the points from a straight line.

The MODEL-1 Yorkfit (York's original algorithm) assumes that the only cause of scatter from a straight line is the errors that you assigned to the points, and weights the points proportional to the inverse square of these errors. A test of this assumption is provided by the PROBABILITY OF FIT value calculated by ISOPLOT200. If this value is reasonably high (more than 0.15 to 0.2), then the MODEL-1 assumptions are probably (but not certainly) justified. If this probability is low (less than 0.2), however, you may choose another model. The MODEL-1 errors are calculated using the maximum-likelihood algorithm of Titterton and Halliday (1979).

A MODEL-2 Yorkfit assigns equal weights and zero error-correlations to each point. This is probably seldom valid in terms of any physical model, but at least avoids the mistake of weighting the points according to analytical errors when it is clear that in fact some other cause of scatter is involved.

A MODEL-3 Yorkfit assumes that the scatter is due to a combination of the assigned errors plus a normally-distributed variation (of unknown magnitude) in the Y-values. This model (similar to the Model 3 algorithm of McIntyre and others, 1966) may be realistic in the case of Rb-Sr isochron data for young rocks, where one may expect that the initial 87/86 ratio may have been different for each sample. For this model, ISOPLOT200 will solve for the unknown Y-variation as well as the best-fit line, so you can judge if the solution is realistic.

A MODEL-4 Yorkfit is used only for U-Pb isotope data on the Concordia diagram. This model assumes that the points scatter due to a combination of the assigned errors plus errors in the 207/206 values that increase linearly from the upper concordia-intercept to the lower concordia-intercept (or vice-versa). This model (see Davis, 1982, for another approach to this problem) may be appropriate for cases where a multi-episodic disturbance is suspected (in which case the closer a point is to the upper intercept the greater its weight), and cases where significant inheritance is suspected (in which case the closer a point is to

the lower intercept the greater its weight). You may choose whether the lower concordia-intercept or the upper concordia-intercept is to be the age of interest.

The program will always attempt a MODEL-1 fit first. In the Yorkfit printout, the A PRIORI errors are the errors in the slope and intercept calculated only from your assigned errors for each point. In other words, no matter how much the points scatter from a straight line, the A PRIORI errors will remain the same. The INCLUDING SCATTER errors, however, are calculated from the actual scatter of the points from the line, so the less the scatter, the lower the INCLUDING SCATTER errors.

How the 95%-confidence limit errors are calculated depends on the probability that the assigned errors for the points can account for the observed scatter. If this probability is greater than 0.15, the 95%-confidence limit errors are simply 1.96 times the A PRIORI errors (1.96 is the Student's-t value for infinite degrees of freedom -- the reason for this is discussed by Brooks and others, 1972). If the probability is less than 0.15, the 95%-confidence limit errors are t times the INCLUDING SCATTER errors, where t is the Student's-t value for N-2 degrees of freedom (N = number of points). If you choose the MODEL-1 Yorkfit when the probability is low, you are in effect assuming that the true analytical errors are greater than your original estimate by some unknown factor, but that the relative X- and Y-errors, and the error-correlations, are still valid.

The MODEL-3 Yorkfit is offered as an option if the program recognizes, from the axis names, that you are plotting isochron data for Rb-Sr, Sm-Nd, U-Pb (238/204 - 206/204 or 235/204 - 207/204), or Th-Pb (232/204 - 208/204). The 95%-confidence limit errors are calculated from t times the INCLUDING SCATTER errors, where the t value is for N-2 degrees of freedom. The MODEL-2 95%-confidence limit errors are also calculated in this way.

The MODEL-4 Yorkfit is offered only for Concordia-diagram data, and only for data-sets with at least 6 points. The 95%-confidence limit errors are calculated by multiplying the INCLUDING SCATTER errors by the t value for N'-2 degrees of freedom, where N' is a number that is in general less than the actual number of points. This approach seems necessary because of the way in which the MODEL-4 algorithm weights the points - the more concordant the point, the greater its weight. In the extreme case, with 2 nearly concordant points plus any number of much more discordant points, the algorithm in effect pays attention only to the 2 most-concordant points and ignores the rest. Obviously in this case, the true degrees of freedom are much less than the apparent degrees of freedom. To take this into account, N' is calculated by assigning each of the 2 most-concordant points a "point value" of exactly 1, and all of the rest of the

points a fractional "point value" based on their weight relative to the second-most concordant point.

The effect of this procedure (which is intuitive rather than mathematical) is to drastically reduce the degrees of freedom assigned to many data sets, especially if 1 or 2 of the points in the data set are much more concordant than the rest. So the data sets which are most appropriate for the MODEL-4 approach are those with a large number of points that are relatively evenly spaced along a linear trend. Data sets with few points will tend to have MODEL-4 errors that are very large, due to the large (perhaps several tens or hundreds) Student's-t multiplier invoked. You can tell the value of N' used by the algorithm from the printout, which refers to the NUMBER OF EQUIVALENT "POINTS".

OBTAINING ISOCHRON AGES FROM YORKFIT LINES

If your plot is an isochron plot (including $^{206}\text{Pb}/^{204}\text{Pb}$ - $^{207}\text{Pb}/^{204}\text{Pb}$ and $^{204}\text{Pb}/^{206}\text{Pb}$ - $^{207}\text{Pb}/^{206}\text{Pb}$), the computer will automatically calculate an isochron age and error from the Yorkfit results. The decay constants used are those recommended by the I.U.G.S. Subcommittee on Geochronology (Steiger and Jager, 1977).

RE-USING YORKFIT POINTS and PLOTTING ERROR-ENVELOPES

After ISOPLOT200 has completed a Yorkfit, the softkeys will be defined as:

HELP			ADD POINTS	CONTINUE
ERROR ENVELOPE			DELETE POINTS	NEW PLOT

If you press CONTINUE, you will return to the "SELECT PLOTTING-SYMBOL" display, and any any new data-points that you enter will count as a new "set" of points for subsequent Yorkfits or other regressions. If you want to either add to or delete from the set of points that was just Yorkfit, press the appropriate softkey. If you want to plot the 95%-confidence limit error-envelope about the Yorkfit line, press the ERROR ENVELOPE softkey. You can request that the error-envelope be drawn, incidentally, by pressing CTRL E from most of the CRT displays that have to do with data entry.

SELECTING A DASHED OR SOLID YORKFIT LINE

The Yorkfit line will be drawn as a dashed pattern (figure 2), unless you specify a solid line by pressing the **SOLID LINE** softkey from the "SELECT PLOTTING-SYMBOL" screen. The dashed pattern will change with each successive Yorkfit that you request for a given plot.

THE OPTIONS SCREEN

The **OPTIONS** screen (available from the "SELECT PLOTTING SYMBOL" screen) allows you to fit a curve to data, change the error-input format, store a CRT plot on a disk, or recall a CRT plot from a disk. The softkeys are labelled as follows:

MISCELLANEOUS OPTIONS

```

|-----|-----|-----|-----|-----|
| LOAD PLOT|ABS ERRORS| % ERRORS | 1-SIG ERRS| 2-SIG ERRS|
|-----|-----|-----|-----|-----|
|STORE PLOT| EXPO FIT | LOG FIT | POLYFIT | ESCAPE |
|-----|-----|-----|-----|-----|

```

The **EXPO FIT**, **LOG FIT**, and **POLYFIT** options will fit a least-squares curve to the last set of plotted data, of the form $Y=A \cdot E^{(B \cdot X)}$ [**EXPO FIT**], $Y=A \cdot \text{LOG}(B \cdot X)$ [**LOG FIT**], or a polynomial of order 1 to 4.

The **% ERRORS**, **1-SIG ERRORS**, and **2-SIG ERRORS** options allow you to change the error-input format from the default one of 2-sigma(%) to the 1-sigma level and/or absolute rather than percent errors.

The **LOAD PLOT** and **STORE PLOT** options allow you to store the image of a CRT plot on a disk, or to retrieve that image from a disk. This way, you can store a plot with data-points for later retrieval and possible additional plotting.

CREATING CONCORDIA PLOTS

If your plot is a concordia plot ($^{206}\text{Pb}/^{238}\text{U} - ^{207}\text{Pb}/^{235}\text{U}$), you'll find a few differences from the normal X-Y plotting routine. First, if the program is in the ISOCHRON or X-Y PLOTTER mode, you must return to the starting screen (NEW PLOT softkey from the "SELECT PLOTTING-SYMBOL" screen). When the initial ISOPLOT200 screen is present, press CTRL C to indicate a concordia plot. Set up the plot in the usual way (CRT or plotter, pen-number, et cetera), then press the START PLOT softkey. The program will then ask you to define the limits of the plot in terms of ages, rather than X-Y values. The display will be:

ENTER AGE LIMITS AS: [MIN. AGE,] MAX. AGE

(or press CONTINUE to define in terms of X and Y limits)

If you enter 1 value only, the minimum age to be shown on the concordia plot is assumed to be 0. If you enter 2 values, the plot will show ages from about the first value to about the second value (generally, a slightly larger age-range so that the axis-ticks will start and end on reasonable numbers).

If you want to define the plot limits in terms of the actual X- and Y-values, though, just press CONTINUE in response to the query, and you can do so.

The concordia curve can be drawn with 2 kinds of ticks and tick-labels: ticks as small circles labelled horizontally to the left of the tick, and ticks as short dashes normal to the concordia curve, with labels above and colinear with the ticks. You can select which style you want from the initial screen, where softkey k3 will be labelled either DASH CTICK or CIRCLE CTICK

An additional feature of both the "SELECT PLOTTING-SYMBOL" and "ADD POINTS" screens for concordia plots is the option to force the next Yorkfit through a given age on the concordia curve. The added prompt is:

press CTRL F to force the Yorkfit line through a point on concordia

After pressing CTRL F, the CRT will ask you the age that you wish to force the Yorkfit line through.

Also, after any Yorkfit, the program will calculate the concordia intercepts of the Yorkfit line and its associated uncertainties, using the algorithm of Ludwig (1980). These intercepts will be labelled on the lower-left corner of the plot (optional if a plotter plot), with the labels for each successive solution for a given plot stacked on the previous one.

OBTAINING GROWTH CURVES FOR Pb-ISOTOPE PLOTS

If your plot is either a $^{207}\text{Pb}/^{204}\text{Pb}$ - $^{206}\text{Pb}/^{204}\text{Pb}$ or a $^{208}\text{Pb}/^{204}\text{Pb}$ - $^{206}\text{Pb}/^{204}\text{Pb}$ plot, after drawing the plotbox and labels the computer will ask:

PRESS k0 TO GET A STACEY-KRAMERS GROWTH-CURVE,

PRESS k1 TO GET SOME OTHER SINGLE-STAGE GROWTH-CURVE.

PRESS k9 TO DECLINE ANY GROWTH-CURVE.

If you press k0, the computer will draw a single-stage Pb-isotope growth-curve on the plot using the constants suggested by Stacey and Kramers (1975). If you press k1, the computer will ask you to define your own single-stage growth curve, in terms of the starting age, the initial isotope ratios, the μ of the source, and (for $^{208}\text{Pb}/^{204}\text{Pb}$ - $^{206}\text{Pb}/^{204}\text{Pb}$ plots) the Th/U of the source.

DUMPING A CRT PLOT TO THE PLOTTER OR PRINTER

You can obtain a hard copy of any CRT plot by requesting that the CRT plot be "dumped" to either the printer or the plotter. Press CTRL P from the "SELECT PLOTTING SYMBOL" screen to dump the plot to the printer. This will only work, incidentally, if the printer is one of the types that can accept direct graphics-dumps, such as the HP-2225 ThinkJet printer. The HP-82905B dot-matrix printer is not such a printer. To dump the CRT plot to the plotter, press CTRL D from the "SELECT PLOTTING SYMBOL" screen. The plot will be transferred to the plotter using plot-size 8 and pen-number 1.

REFERENCES

- Brooks, C., Hart, S.R., and Wendt, I., 1972, Realistic use of two-error regression treatments as applied to rubidium-strontium data: Rev. Geophys. Space Phys., v. 10, p. 551-577.
- Davis, D.W., 1982, Optimum linear regression and error estimation applied to U-Pb data: Canadian Journal of Earth Sciences, v. 198, p. 2141-2149.
- Ludwig, K.R., 1979a, A program in Hewlett-Packard BASIC for X-Y plotting and line-fitting of isotopic and other data: U.S. Geological Survey Open-File Report 79-1641; 33 p.
- , 1979b, A program in Hewlett-Packard BASIC for plotting and processing U-Pb isotope data on "concordia" diagrams: U.S. Geological Survey Open-File Report 79-1692; 32 p.
- , 1980, Calculation of uncertainties of U-Pb isotope data: Earth Planetary Science Letters, v. 46, p. 212-220.
- , 1982, Programs for filing and X-Y plotting of isotopic and other data using an HP-9830 computer and HP-9832 plotter: U.S. Geological Survey Open-File Report 82-385; 36 p.
- , 1982, Programs for filing and plotting U-Pb isotope data for concordia diagrams, using an HP-9830 computer and HP-9862 plotter: U. S. Geological Survey Open-File Report 82-386; 22 p.
- , 1983, Plotting and Regression programs for isotope geochemists, for use with HP-86/87 microcomputers: U.S. Geological Survey Open-File Report File Report 83-849; 89 p.
- , in press, PBDAT200; A Computer Program for Processing Raw Pb-U-Th Isotope Data: U.S. Geological Survey Open-File Report.
- McIntyre, G.A., Brooks, Christopher, Compston, William, and Turek, Andrew, 1966, The statistical assessment of Rb-Sr isochrons: Journal of Geophysical Research, v. 71, p. 5459-5468.
- Stacey, J.S., and Kramers, J.D., 1975, Approximation of terrestrial lead isotope evolution by a two-stage model: Earth Planetary Science Letters, v. 26, p. 207-221.

- Steiger, R.H., and Jager, Emilie, 1977, Subcommittee Geochronology: Convention on the use of decay constants in geo- and cosmochronology: Earth Planetary Science Letters, v. 36, p. 359-362.
- Titterton, D.M., and Halliday, A.N., 1979, On the fitting of parallel isochrons and the method of maximum likelihood: Chemical Geology, v. 26, p. 183-195.
- York, Derek, 1969, Least-squares fitting of a straight line with correlated errors: Earth Planetary Science Letters, v. 5, p. 320-324.

FIGURE CAPTIONS

Figure 1: X-Y plot showing various plotting symbols. Top left row - alphanumeric (keyboard) symbols; top right row - polygon symbols; lower - error symbols. Curve is a polynomial fit through the dots. Plot size is 6.

Figure 2: Concordia plot with Yorkfit line through error-ellipse symbols. Oblique-dash concordia ticks.

Figure 3: Pb-isotope plots created with the "nonstandard" plot-size option and the STACK option for the upper plot. Growth curves are standard Stacey-Kramers (Stacey and Kramers, 1975).

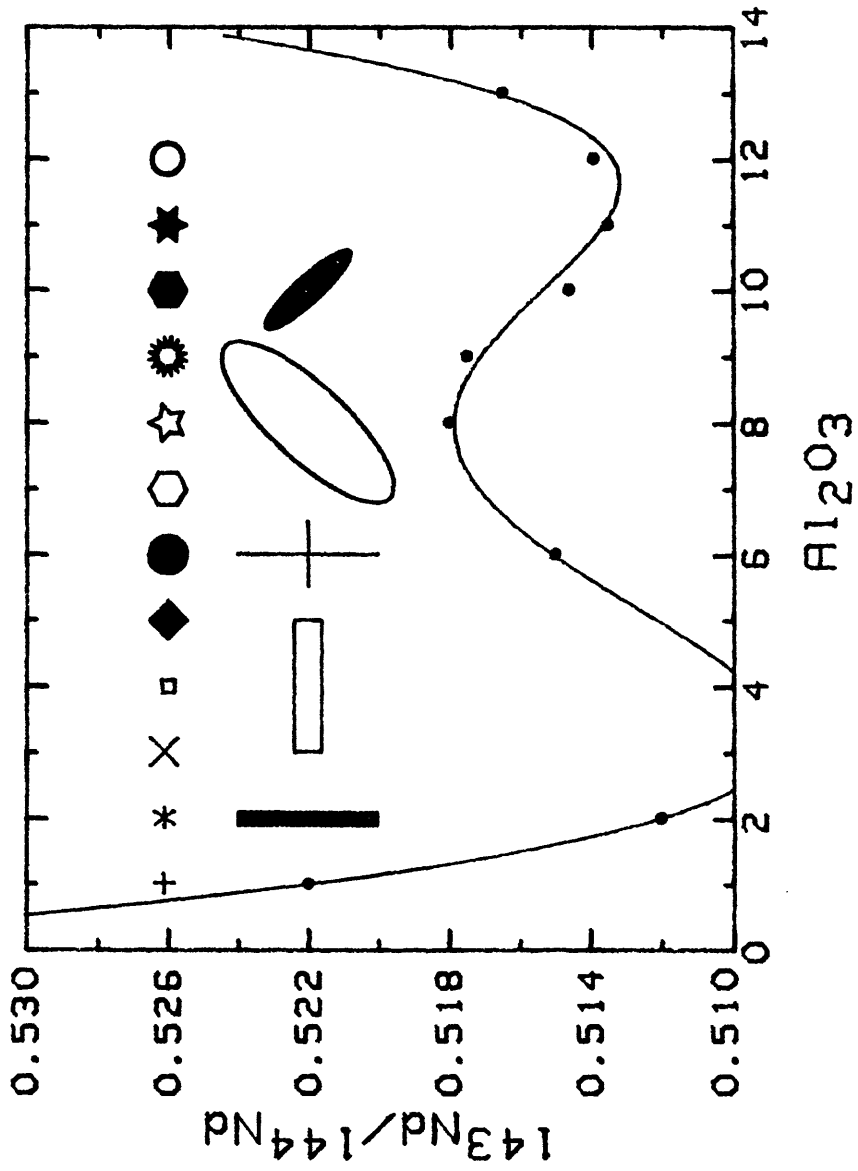


Figure 1

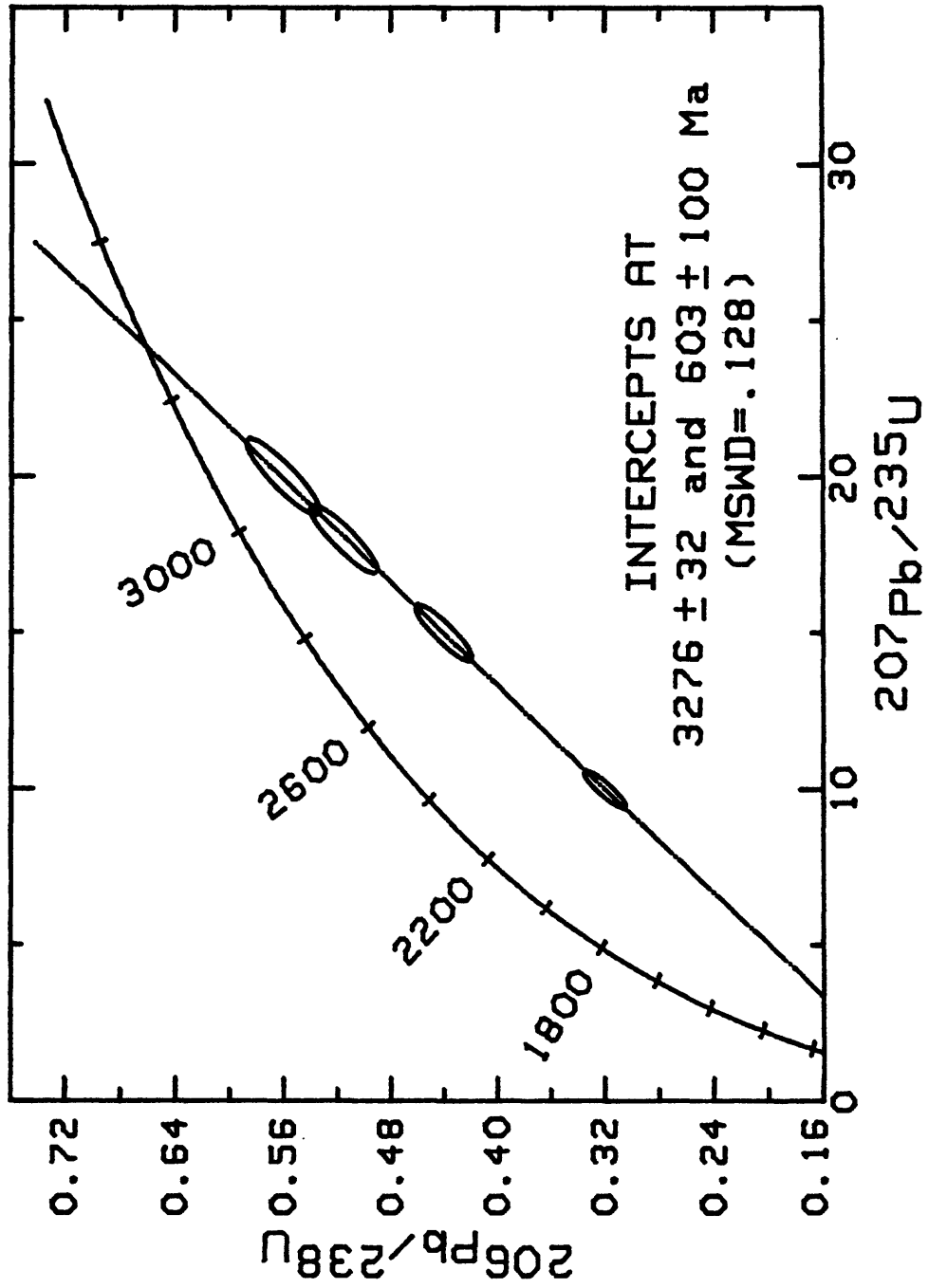


Figure 2

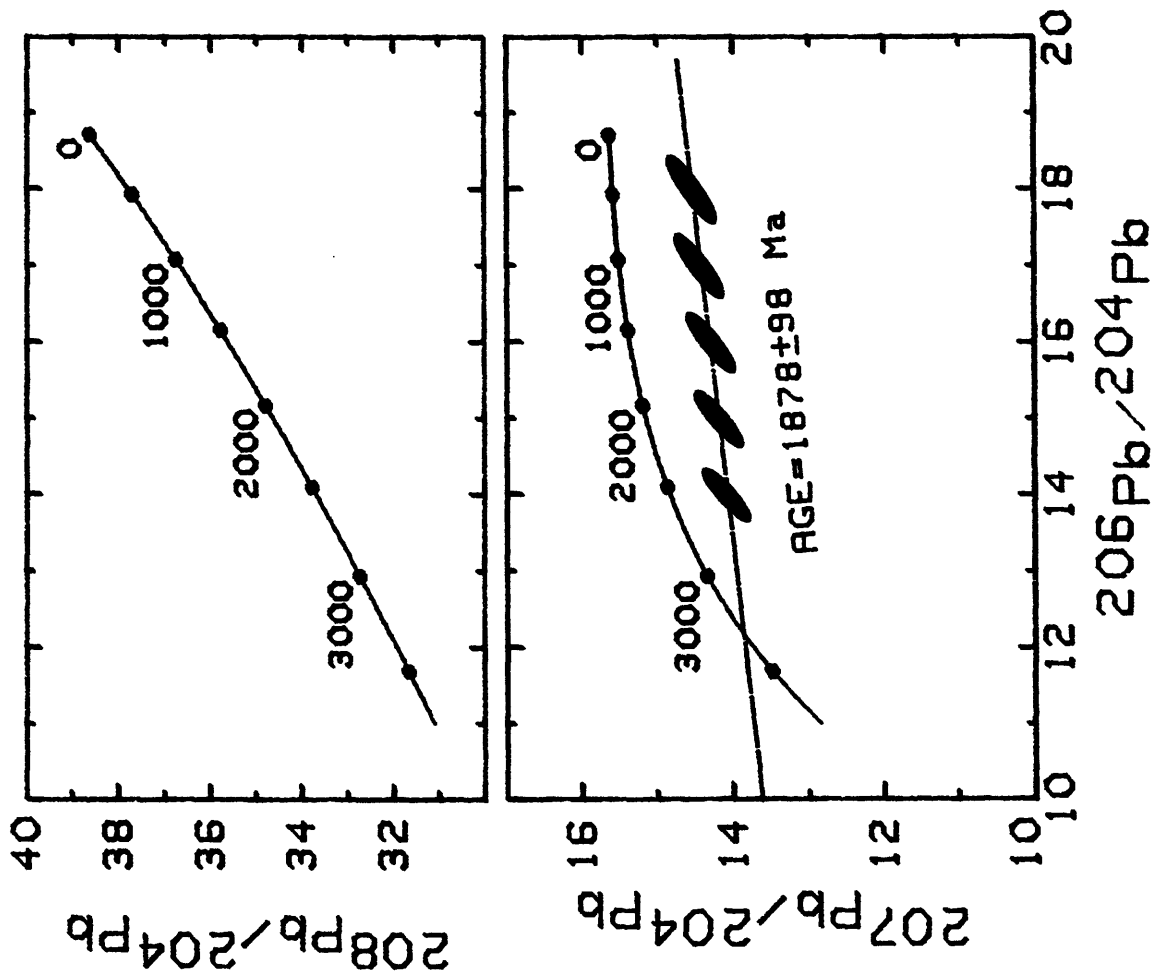


Figure 3

APPENDIX -- PROGRAM LIST OF ISO PLOT 200

```

6    ! ***** ISO PLOT 200 *****
7    !
8    ! K.R. Ludwig, U.S. Geological Survey Open-File Report 85-513.
9    !
12   Rev$=" Aug 16, 1985" ! Version EL
24   ! Program for plotting and regression isotopic and other X-Y data,
30   ! including Concordia data.
36   !
42   ! For a Series 200 Hewlett-Packard computer (9816,9817,9826,9836),
48   ! operating under BASIC 3.0.
54   !
60   OPTION BASE 1
66   !
72   M$(0)=":HP9121,700,0" ! Dual single-sided 3.5" drive.
78   M$(1)=":HP9121,700,1" ! " " "
84   Hp82905=1 ! Assume an HP82905b printer.
90   !
96   OUTPUT KBD;"SCRATCH KEY ";CHR$(255)&CHR$(88); ! undefine any softkey typing-aids
102  TRACE OFF
108  !
114  COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Limx1,Limx2,Limy1,Limy2,Xspred,Yspred,Xloc,Yloc,Xm,Ym,Nx_$[32],My_$[32],Nx_$[32],My_$[32],Nx
    O_$[10],MyO_$[10],Double
120  COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
126  COM /Plotbox2/ Nxr_$[32],Nyr_$[32],Plotdev(2),Pr(2),Plotter,Plotter_dump,Pr,Tht,Lht,Ticksizex1,Y1,Xend,Yend,Speed$[4],Crt,Hardpl
    ot,Cpen,Xgu,Ygu
132  COM /Plotbox3/ Of_xname$[18],Of_yname$[18],Ofile,Stack
138  COM /Concordia/ Age_min,Age_max,Ctick,Oblique_tick
144  COM /Yorkfit1/ Xu(201),Yu(201),Ex(201),Ey(201),RO(201),N,Maxyor,Mn,Prtr,Es$[18],Eb$[18],Er$[18]
150  COM /Yorkfit2/ Xerr(201),Yerr(201),Sig,Line(12,2),Percent,Nlines,N_interlabels,Cint(8),Solid_line,Cache_cint(3,8),Cache_mswd(3)
156  COM /Yorkfit3/ Slope,Errsl,Inter,Errint,X_bar
162  COM /Printplot/ Printer_ok,Plotter_ok
168  COM Cplot,Uc(201,5),N$(201)[18],File$[16],Gfile$[16],Polysides,Polysize,Hp82905,Polyrot,M$(0:1)[13]
174  DIM Key$[2],Crud$[37],Sname$(50)[18],Cache_sname$(50)[18]
180  DIM Cache(201,5),Symb(201,5),Letter$(25)[80],Letter(25,4)
186  INTEGER Crt_plot(7500),L_r(50),Cache_l_r(50)
192  DIM Iso$[81],Cp1$[81],Nu(5),R$[50],Re$[50],Sp$[20]
198  Iso$=CHR$(129)&RPT$(" ",17)&"<<<<<<<<<< ISOCHRON or X-Y PLOTTER >>>>>>>>>>"&RPT$(" ",17)&CHR$(128)
204  Cp1$=CHR$(129)&RPT$(" ",19)&"<<<<<<<<<< CONCORDIA PLOTTER >>>>>>>>>>"&RPT$(" ",19)&CHR$(128)
210  Sp$=RPT$(" ",20)
216  Plotter_model=7475
222  DATA IMPROPER RESPONSE OR NUMBER OF VALUES,Xerr,2-SIGMA," %-ERRORS",1,2,1,705,0
228  DATA 1,1,1,2.5,20,201,200,0,1,0,1,.12,1
234  RESTORE 222
240  READ Crud$,Er$,Es$,Eb$,Percent,Sig,Plotdev(*),Hardplot,Plotter,Pr(1),Prtr,Polysize,Polysides
246  READ Maxnum,Maxyor,Plot_started,Crt,Hardplot,Ticksizex,Offset,Cpen
252  Maxpen=2*(Plotter_model=7470)+6*(Plotter_model=7475) ! # pens in plotter
258  PRINTER IS CRT
264  DEG
270  GINIT
276  !

```

```

282 Redraw:DATA 0,0,0,0,0,0,0,0,0,1,1,0,9,US
288 RESTORE Redraw
294 READ Stack,SizeType,Plotter_dump,Solid_line,Nlines,N_interlabels,Oblique_tick,Nphrases,Bypass,Pr(2),Prtr,C_ct,Penspeed,Speed$
300 OFF DELAY
306 DEG
312 Clear
318 IF Hardplot THEN PEN 0 ! Put plotter-pen away if out
324 !
330 Begin:Clear ! Set up initial screen & options
336 Alpha
342 OFF KEY
348 Newplot=1
354 Slope=0
360 PRINTER IS CRT
366 !
372 PRINT TABXY(1,1);"Rev. ";Rev$;TAB(S1);"K.R. Ludwig, U.S. Geol. Survey"
378 IF Ticksize(>)1 THEN DISP "RELATIVE LABEL-SIZE =" ;Ticksize
384 IF Hardplot THEN
390 PRINT TABXY(1,4);"PLOTTER PLOT"
396 IF Rotate_90 THEN PRINT TABXY(1,7);"ROTATED PLOT"
402 IF NOT Rotate_90 THEN PRINT TABXY(1,7);" "
408 IF Stack THEN PRINT TABXY(1,8);"SUPRESSED X-AXIS & X-TICK LABELS"
414 IF NOT Stack THEN PRINT TABXY(1,8);" "
420 ON KEY 5 LABEL "ADJ. LABELSIZE" GOTO Labelsize
426 ON KEY 1 LABEL " CRT PLOT" GOTO Crtplot
432 ON KEY 2 LABEL " CHOOSE PEN" GOTO Choose_pen
438 ON KEY 7 LABEL " PEN SPEED" GOTO Penspeed
444 IF NOT Cplot AND NOT Stack THEN ON KEY 3 LABEL " STACKED PLOT" GOTO Stack
450 IF NOT Cplot AND Stack THEN ON KEY 3 LABEL "UNSTACKED PLOT" GOTO Stack
456 IF Rotate_90 THEN ON KEY 6 LABEL "UNROTATED PLOT" GOTO Rotate_plot
462 IF NOT Rotate_90 THEN ON KEY 6 LABEL " ROTATED PLOT" GOTO Rotate_plot
468 ELSE
474 PRINT TABXY(1,4);"CRT PLOT"
480 ON KEY 1 LABEL " PLOTTER PLOT" GOTO Hardplot
486 ON KEY 3 LABEL "" CALL Clunk
492 END IF
498 IF Prtr=1 THEN
504 PRINT TABXY(67,4);"CRT OUTPUT"
510 ON KEY 0 LABEL "PRINTER OUTPUT" GOTO Hardcopy
516 ELSE
522 PRINT TABXY(67,4);"PRINTER OUTPUT"
528 ON KEY 0 LABEL " CRT OUTPUT" GOTO Display
534 END IF
540 IF Cplot THEN
546 PRINT TABXY(1,2);Cpl$
552 PRINT TABXY(1,17);FNCenter$("Press "&FNU$("&CTRL I")&" for isochron or other X-Y plot")
558 IF Oblique_tick THEN
564 PRINT TABXY(25,4);"OBLIQUE-DASH CONCORDIA TICKS"
570 ON KEY 3 LABEL " CIRCLE CTICK" GOTO Normaltick
576 ELSE
582 PRINT TABXY(25,4);"HORIZ.-CIRCLE CONCORDIA TICKS"
588 ON KEY 3 LABEL " DASH CTICK" GOTO Obliquetick
594 END IF
600 ELSE
606 PRINT TABXY(1,2);Iso$
612 PRINT TABXY(1,17);FNCenter$("Press "&FNU$("&CTRL C")&" for concordia plot")
618 END IF
624 !
630 ON KEY 4 LABEL " HELP" GOTO Call_help1
636 ON KEY 9 LABEL " START" GOTO Go

```

```

642 ON KEY 8 LABEL " GET DATAFILE" GOTO Get_datafile
648 !
654 React1:ON KBD GOTO Keys1
660 GOTO 660
666 !
672 Keys1:K=FNGet_key          ! trap non-softkey press
678 OFF KBD
684 !
690 SELECT K
696 CASE 8                      ! CTRL-H key pressed
702 Call_help1:Help1(Cplot)
708 GOTO Begin
714 CASE -67                    ! CONTINUE key pressed
720 GOTO 60
726 CASE 3,9                    ! CTRL-C or CTRL-I pressed
732 Cplot=(K=3)
738 BEEP 440,.1
744 GOTO Begin
750 CASE -185,-195              ! CTRL-k9, CTRL-CONTINUE
756 IF Hardplot THEN
762 Bypass=1                    ! (don't draw plotbox)
768 GOTO 60
774 ELSE
780 Clunk
786 END IF
792 CASE ELSE
798 Clunk
804 END SELECT
810 GOTO React1
816 !
822 Go: ! request operator to define plotbox & then draw plotbox
828 OFF KEY
834 OFF KBD
840 IF Crt THEN Ticksize=1
846 Double=Hardplot*(Penspeed<=6 OR Ticksize)=1.5)*(NOT Bypass) ! draw some parts of plot twice if slow penspeed or large label-size
852 CALL Plotbox(Crud$,Cplot,Rotate_90,Bypass,Plot_started,Escape)
858 IF Escape THEN Begin
864 IF Cplot AND NOT Bypass THEN CALL Concordia ! draw & label concordia
870 IF NOT Cplot THEN CALL Pbgrowth
876 IF Plotter_dump THEN RETURN
882 PEN 0
888 GOTO Startplot
894 !
900 Stack:Stack=NOT (Stack) ! suppress X-axis label & X-axis tick-labels?
906 BEEP 440,.1
912 GOTO Begin
918 !
924 Hardcopy=BEEP 440,.1       ! Use printer as printing device
930 Printer_ok=1
936 ON TIMEOUT 7,.1 CALL Printer_timeout
942 PRINTER IS 701
948 PRINT
954 IF Printer_ok THEN
960 OFF TIMEOUT
966 Pr(2)=701
972 Prtr=2
978 END IF
984 PRINTER IS CRT
990 GOTO Begin

```



```

996 !
1002 !
1008 Rotate_plot:Rotate_90=NOT (Rotate_90) ! Rotated plot?
1014 BEEP 440,.1
1020 GOTO Begin
1026 !
1032 Display:BEEP 440,.1 ! Use CRT as printing device
1038 Pr(2)=CRT
1044 Prtr=1
1050 PRINTER IS CRT
1056 GOTO Begin
1062 !
1068 Hardplot:BEEP 440,.1 ! Use HP-7475A as plotter
1074 Plotter=Plotdev(2)
1080 Plotter_ok=1
1086 ON TIMEOUT 7,.1 CALL Plotter_timeout
1092 PLOTTER IS Plotter,"HPGL"
1098 IF Plotter_ok THEN
1104 OFF TIMEOUT
1110 Hardplot=1
1116 Crt=0
1122 PEN 0
1128 END IF
1134 GOTO Begin
1140 !
1146 Crtplot:DISP FNH$("CRT GRAPHICS")&Sp$ ! Use CRT as plotter
1152 BEEP 440,.1
1158 Plotter=1
1164 Hardplot=0
1170 Crt=1
1176 GOTO Begin
1182 !
1188 Normaltick:Oblique_tick=0 ! for a concordia plot, use an "o" for the concordia tick & label the ticks horizontally
1194 BEEP 440,.1
1200 GOTO Begin
1206 !
1212 Obliquetick:Oblique_tick=1 ! for a concordia plot, use a short dash for a tick, perpendicular to the concordia curve, labelled
parallel to the dash
1218 BEEP 440,.1
1224 GOTO Begin
1230 !
1236 Labelsize:Ticksize=1 ! adjust the relative size of the axis labels, tick labels, & alphanumeric plotting symbols
1242 OFF KBD
1248 OFF KEY
1254 Clear
1260 PRINT USING "6/,K,2/,K";"ENTER A NUMBER TO REDEFINE THE RELATIVE SIZE OF THE PLOT LABELS","(0.3 - 3; 1=default)..."
1266 LOOP
1272 INPUT Ticksize
1278 EXIT IF Ticksize>=3 AND Ticksize<=3
1284 Bad_input("ILLEGAL SIZE")
1290 END LOOP
1296 GOTO Begin
1302 !
1308 Get_datafile:GOSUB Datafile ! get data from a datafile
1314 GOTO Begin
1320 G_datafile:GOSUB Datafile ! ditto, called after plotbox drawn
1326 GOTO More_points
1332 !
1338 Penspeed:GOSUB Pen_speed
1344 GOTO React1

```

```

1350 !
1356 Pen_speed: ! Define pen-speed for plotter
1362 DISP "PRESS KEY (0-9) TO INDICATE THE PLOTTER PEN-SPEED..."&Sp$
1368 REPEAT
1374 ON KBD GOTO 1392
1380 GOTO 1380
1386 !
1392 Penspeed=FNGet_key-48
1398 OFF KBD
1404 UNTIL Penspeed=0 AND Penspeed=9
1410 !
1416 IF Penspeed=9 THEN Speed$="US" ! fast + max. acceleration
1422 IF Penspeed<9 THEN Speed$="US"&URL$(INT(38.1*(Penspeed+1)/10))
1428 BEEP 440,.1
1434 DISP FNH$("PEN-SPEED "&URL$(Penspeed))
1440 RETURN
1446 !
1452 Kretrn:OFF KBD
1458 RETURN
1464 !
1470 Startplot:Nn=0
1476 LDIR 0
1482 IF Cplot=0 THEN Line$="ISOCHRON"
1488 IF Cplot THEN Line$="CHORD"
1494 PRINTER IS CRT
1500 !
1506 Plotdata:OFF KEY
1512 ALPHA ON
1518 DATA 0,1,1,0,0
1524 RESTORE 1518
1530 READ N,Ii,Kk,Name_label,Forced_inter
1536 !
1542 ! N = number of data-points last input
1548 ! Kk = number of first data-point in current set
1554 ! Ii = counter for data-point in current set
1560 ! Nn = number of points in current set
1566 !
1572 More_points:Clear
1578 Mpts:OFF KBD
1584 OFF KEY
1590 Alpha
1596 PRINTER IS CRT
1602 ON DELAY 3600 GOTO Redraw ! If no user action within an hour, return to
1608 ! starting display.
1614 IF Somepoints THEN GRAPHICS OFF
1620 !
1626 ! Set up the plot-points options screen
1632 PRINT TABXY(1,6);FNCenter$(FNH$("SELECT A PLOTTING-SYMBOL TO PLOT DATA"))
1638 IF Crt THEN
1644 PRINT TABXY(20,12);"press "&FNH$("CTRL O")&" to dump plot to plotter"
1650 PRINT TABXY(20,14);"press "&FNH$("CTRL P")&" to dump plot to printer"
1656 END IF
1662 PRINT TABXY(20,10);"press "&FNH$("CTRL L")&" to draft a phrase on the plot"
1668 IF Cplot AND Nn THEN PRINT TABXY(20,10);"press "&FNH$("CTRL-F")&" to force Yorkfit through point on concordia"
1674 PRINT TABXY(20,16);"press "&FNH$("CTRL H")&" for help"
1680 PRINT TABXY(20,18);"press any key for keyboard plotting-symbol"
1686 !
1692 IF Hardplot THEN
1698 PEN 0
1704 MOVE Xmin,Ymax

```

```

1710 END IF
1716 Keyboard=0
1722 Newplot=0
1728 !
1734 ! Set up the softkey branches
1740 Reenter=PRINTER IS CRT
1746 IF NOT Cplot THEN
1752   ON KEY 0 LABEL " ERROR BOX" GOTO Oebox
1758   ON KEY 10 GOTO Sebox
1764   ON KEY 2 LABEL " ERROR CROSS" GOTO Errcross
1770 END IF
1776 ON KEY 1 LABEL "ERROR ELLIPSE" GOTO Deell
1782 ON KEY 11 GOTO Seell
1788 ON KEY 3 LABEL " POLY60N" GOTO Open_poly
1794 ON KEY 13 GOTO Solid_poly
1800 ON KEY 4 LABEL " OPTIONS" GOTO Options
1806 IF Hardplot THEN ON KEY 6 LABEL " CHOOSE PEN" GOTO Choose_pen
1812 IF Crt THEN ON KEY 6 LABEL " ERASE PEN?" GOSUB Erase_pen
1818 ON KEY 8 LABEL " GET DATAFILE" GOTO G_datafile
1824 IF Mn>1 THEN ON KEY 7 LABEL " YORKFIT" GOTO Call_yorkfit
1830 ON KEY 9 LABEL " NEW PLOT" GOTO Redraw
1836 ! IF Crt THEN ON KEY 5 LABEL " PRINT PLOT" GOTO Print_plot
1842 IF Hardplot AND Solid_line THEN ON KEY 5 LABEL " DASH "&Line$ GOTO Solid_line
1848 IF Hardplot AND NOT Solid_line THEN ON KEY 5 LABEL "SOLID "&Line$ GOTO Solid_line
1854 !
1860 React2=ON KBD GOTO Kybrd
1866 GOTO 1866
1872 !
1878 ! Select pen#1 for hard-plotter
1884 Choose_pen=DISP FNH$("PRESS A NUMBER-KEY (1 - "&VAL$(Maxpen)&") TO SELECT A PEN FOR PLOTTING...")
1890 REPEAT
1896   ON KBD GOTO 1908
1902   GOTO 1902
1908   K=FNGet_key
1914   OFF KBD
1920   Cpen=K-48
1926 UNTIL Cpen=1 AND Cpen<=Maxpen
1932 BEEP 440, .1
1938 DISP FNH$("PEN# "&VAL$(Cpen))
1944 ON 1+Newplot GOTO React2,React1
1950 !
1956 Erase_pen: ! Use an "erasing pen" (CRT-plot only)
1962 IF Cpen<0 THEN
1968   Cpen=1
1974   DISP FNH$(" WHITE PEN ")&Sp$
1980 ELSE
1986   Cpen=-1
1992   DISP FNH$(" ERASING PEN ")&Sp$
1998 END IF
2004 BEEP 440, .1
2010 RETURN
2016 !
2022 Solid_line:Solid_line=NOT (Solid_line) ! use a solid, rather than a dashed line for Yorkfit lines
2028 OFF KEY
2034 OFF KBD
2040 IF NOT Solid_line THEN DISP FNH$(" USE DASHED LINE FOR YORKFIT LINES ")&Sp$
2046 IF Solid_line THEN DISP FNH$(" USE SOLID LINE FOR YORKFIT LINES ")&Sp$
2052 BEEP 440, .1
2058 GOTO Reenter
2064 !

```

```

2070 Oebox:Symbol=1
2076 Clear
2082 PRINT FNH$(" OPEN ERROR-BOX SYMBOL ")
2088 GOTO Input_data
2094 !
2100 Sebox:Symbol=-1
2106 Clear
2112 PRINT FNH$(" SOLID ERROR-BOX SYMBOL ")
2118 GOTO Input_data
2124 !
2130 Oeell:Symbol=2
2136 Clear
2142 PRINT FNH$(" OPEN ERROR-ELLIPSE SYMBOL ")
2148 GOTO Input_data
2154 !
2160 Seell:Symbol=-2
2166 Clear
2172 PRINT FNH$(" SOLID ERROR-ELLIPSE SYMBOL ")
2178 GOTO Input_data
2184 !
2190 Errcross:Symbol=4
2196 Clear
2202 PRINT FNH$(" ERROR-CROSS SYMBOL ")
2208 GOTO Input_data
2214 !
2220 Open_poly:Solid=0
2226 GOTO Poly
2232 Solid_poly:Solid=1
2238 GOTO Poly
2244 !
2250 Poly:Symbol=3 ! define a polygon as the plotting symbol
2256 Clear
2262 OFF KEY
2268 GRAPHICS OFF
2274 IF Solid THEN PRINT TABXY(1,1);CHR$(129)&" SOLID";
2280 IF NOT Solid THEN PRINT TABXY(1,1);CHR$(129)&" OPEN";
2286 PRINT " POLYGON SYMBOL "&CHR$(128)
2292 PRINT TABXY(1,14);"POLYGON-SYMBOL: ENTER #SIDES (- for star), [,SIZE (mm) [,ROTATION (0-360)]]]"
2298 PRINT TABXY(1,16);"Press"&FNCT$&"to re-use last values (&VAL$(Polysides)&",&VAL$(Polysize)&",&VAL$(Polyrot)&")"
2304 P_kbd=ON KBD GOTO 2328
2310 ON KEY 9 LABEL " ESCAPE" GOTO More_points
2316 GOTO 2316
2322 !
2328 K=FNGet_key
2334 OFF KBD
2340 OFF KEY
2346 SELECT K
2352 CASE NUM("1") TO NUM("9"),NUM("-")
2358 OUTPUT KBD;CHR$(K);
2364 PRINT TABXY(1,16);RPT$(" ",80)
2370 DISP FNast$;
2376 Retrieve(R$,Nv(*),Minputs,1)
2382 IF R$="*" THEN More_points
2388 IF Minputs THEN
2394 Polysides=Nv(1)
2400 Polysize=Nv(2)*(Minputs+1)+3*(Minputs+2)
2406 IF Polysize>40 THEN
2412 Bad_input("A POLYGON OF "&VAL$(Polysize)&" mm IS TOO LARGE!")
2418 GOTO Poly
2424 END IF

```

```

2430 Polyrot=Nu(3)*(Ninputs/2)+(90*(Polysides/2=INT(Polysides/2))*(180/Polysides-90))*(Ninputs/3)
2436 IF Ninputs<3 AND Polysides>10 THEN Polyrot=0
2442 END IF
2448 CASE -67 ! CONTINUE key: do nothing
2454 CASE ELSE ! invalid
2460 GOTO P_kbd
2466 END SELECT
2472 !
2478 PRINT TABXY(1,14);RPT$(" ",3*80)
2484 GOTO Input_data
2490 !
2496 Kybrd:Keycode=FNGet_key ! react from a non-softkey press
2502 OFF KBD
2508 SELECT Keycode
2514 CASE 8 ! CTRL-H key
2520 CALL Help2(Cplot)
2526 CASE 12 ! CTRL-L
2532 GOTO Draft
2538 CASE 4 ! CTRL-D key
2544 GOTO Dump_to_plotter
2550 CASE 16,-78 ! CTRL-P, DUMP GRAPHICS
2556 GOTO Print_plot
2562 CASE 19 ! CTRL-S
2568 GOSUB Pen_speed
2574 GOTO React2
2580 CASE 5 ! CTRL-E
2586 IF Slope THEN Error_envelope
2592 CASE 6 ! CTRL-F
2598 GOSUB Force_inter
2604 IF R$(">") THEN Mpts
2610 CASE 32 TO 122,-67 ! Valid alphanumeric key for plotting
2616 Clear ! or CONTINUE key
2622 Symbol=5
2628 IF Keycode=-67 THEN ! (CONTINUE key- use + as symbol)
2634 Key$="+ "
2640 ELSE
2646 Key$=UPC$(CHR$(Keycode)) ! valid alphanumeric key
2652 END IF
2658 PRINT FNH$(" PLOTTING-SYMBOL IS ")&FNH$(Key$&" ")
2664 GOTO Input_data
2670 CASE ELSE ! invalid key
2676 GOTO 1752
2682 END SELECT
2688 GOTO More_points
2694 !
2700 Force_inter:LOOP ! force the next Yorkfit through an arbitrary age on the concordia curve
2706 Clear
2712 OFF KEY
2718 PRINT TABXY(4,12);"What age (in Ma) do you want to force the Yorkfit through?"
2724 ON KEY 9 LABEL " ESCAPE" GOTO 2880
2730 ON KBD GOTO 2748
2736 DISP "?"
2742 GOTO 2742
2748 K=FNGet_key
2754 OFF KBD
2760 OFF KEY
2766 IF K<NUM("0") OR K>NUM("9") THEN 2724
2772 OUTPUT KBD;CHR$(K);
2778 DISP FNst$;
2784 INPUT R$

```

```

2790 IF R$="*" THEN RETURN
2796 ON ERROR GOTO 2832
2802 Forced_age=VAL(R$)
2808 IF Forced_age=0 THEN Forced_age=1.E-8 ! else get divide-by-zero errors
2814 OFF ERROR
2820 EXIT IF Forced_age<6000 OR Forced_age>2000
2826 OFF ERROR
2832 Bad_input("INVALID FORCED-AGE ("&R$&")")
2838 END LOOP
2844 !
2850 Clear
2856 DISP FNM$("FORCE NEXT YORKFIT THROUGH "&R$& " ma")
2862 BEEP
2868 WAIT 1
2874 Forced_inter=1
2880 RETURN
2886 !
2892 Input_data=OFF KEY ! request user to enter his data-points for plotting
2898 OFF KBD
2904 PRINT TABXY(1,3);Ea$&Eb$
2910 IF Dfile THEN
2916 PRINT "INPUT DATAFILE SET #s AS 1st SET, LAST SET-ASTERISK (e.g. 2,12* or 2,12;15,18*)"
2922 PRINT USING "K,/";"Luse *L or *R to label data-file name to (L)eft or (R)ight of point]"
2928 END IF
2934 Data_request=60SUB Input_points
2940 ON I+(I)Maxnum) GOTO Data_request,Edit
2946 !
2952 Input_points=DATA 0,0,0,0,0
2958 RESTORE Input_points
2964 READ Xv(Ii),Yv(Ii),Ex(Ii),Ey(Ii),RO(Ii)
2970 Alpha
2976 OFF KEY
2982 IF Ii=1 THEN
2988 PRINT "SET#";TAB(27);Nx0$;TAB(38);Er$;TAB(51);Ny0$;TAB(62);Er$;TAB(73);"RHO"
2994 PRINT
3000 END IF
3006 DISP "#&VAL$(Ii)&": "&Nx0$& ", ["&Er$& ",] ";Ny0$;" [, "&Er$& "] [, "&err-corr." (&FNCT$&"WHEN DONE)";
3012 D_kbd=ON KBD GOTO 3036
3018 ON KEY 9 LABEL " ESCAPE" GOTO More_points
3024 GOTO 3024
3030 !
3036 K=FNGet_key
3042 OFF KBD
3048 OFF KEY
3054 SELECT K
3060 CASE -63 ! RECALL key
3066 OUTPUT KBD;CHR$(255)&CHR$(63);
3072 GOTO 3090
3078 CASE NUM("0") TO NUM("9"),NUM("-"),NUM("+"),NUM(".")
3084 OUTPUT KBD;CHR$(K);
3090 LINPUT R$
3096 IF R$="*" THEN More_points
3102 CASE -67 ! CONTINUE
3108 GOTO Edit
3114 CASE ELSE
3120 GOTO D_kbd
3126 END SELECT
3132 !
3138 ! If datafile input, use commas to separate data-sets in a continuous series, semicolons to separate individual sets or different series.

```

```

3144 ! So 2,5;7;9,14;17,29;6* indicates sets 2 thru 5 plus set 7 plus sets 9 thru 14 plus sets 17 thru 29 plus set 6.
3150 !
3156 Process_dstring=Semicolon=POS(Re$,";") ! Semicolon in data-input string?
3162 IF Semicolon THEN
3168 R$=Re$[1,Semicolon-1]
3174 ELSE
3180 R$=Re$
3186 END IF
3192 Retrieve(R$,Nv(*),Ninputs,0)
3198 IF Ninputs=0 THEN Edit
3204 !
3210 IF NOT POS(Re$,"*") THEN
3216 Df=0
3222 GOSUB Keyboard_input
3228 RETURN
3234 ELSE
3240 Name_label=(POS(UPC$(Re$),"*R")<>0)-(POS(UPC$(Re$),"*L")<>0)
3246 ! label sample-names to (R)ight or (L)eft of point
3252 END IF
3258 !
3264 IF NOT Dfile AND (Uc(1,1)+Uc(1,3)=0) THEN
3270 Bad_input("SORRY- YOU HAVEN'T LOADED A DATAFILE INTO MEMORY")
3276 END IF
3282 !
3288 Df=1
3294 S1=Nv(1) ! 1st, last datafile set #s in a continuous series
3300 S2=Nv(1+(Ninputs)1)
3306 !
3312 FOR Id=S1 TO S2 ! Transfer datafile sets to plotting array & display
3318 ON ERROR GOTO Nextid
3324 Xu(Ii)=Uc(Id,1)
3330 Ex(Ii)=Uc(Id,2)
3336 Yu(Ii)=Uc(Id,3)
3342 Ey(Ii)=Uc(Id,4)
3348 RO(Ii)=Uc(Id,5)
3354 IF Xu(Ii)*Yu(Ii) THEN ! bypass datafile sets that aren't nonzero
3360 FOR P=1 TO Prtr
3366 PRINTER IS Pr(P)
3372 PRINT Ii;TAB(6);N$(Id);TAB(26);Xu(Ii);TAB(37);Ex(Ii);TAB(50);Yu(Ii);TAB(61);Ey(Ii);TAB(69);RO(Ii)
3378 NEXT P
3384 PRINTER IS CRT
3390 IF Ii<=50 THEN L_r(Ii)=Name_label
3396 IF Ii<=50 AND Name_label THEN Sname$(Ii)=N$(Id)
3402 GOSUB Process_data
3408 IF Ii=Maxnum THEN Edit
3414 END IF
3420 Nextid=NEXT Id
3426 OFF ERROR
3432 PRINT
3438 IF Semicolon=0 THEN RETURN
3444 !
3450 R$=Re$[1+Semicolon] ! get ready to process next element of data-string
3456 GOTO Process_dstring
3462 !
3468 Keyboard_input=Keyboard=1 ! Get typed-in data from keyboard
3474 IF Ii<=50 THEN L_r(Ii)=0
3480 SELECT Ninputs
3486 CASE (2,3),5
3492 CALL Bad_input(Crud$)
3498 GOTO Input_points

```

```

3504 CASE 2 ! 2 values entered: => x-value and y-value
3510 Xv(Ii)=Nu(1)
3516 Yv(Ii)=Nu(2)
3522 GOTO Process_data
3528 CASE 4,5 ! 4 or values entered: => x,x-error,y,y-error[,rho]
3534 Xv(Ii)=Nu(1)
3540 Ex(Ii)=Nu(2)
3546 Yv(Ii)=Nu(3)
3552 Ey(Ii)=Nu(4)
3558 IF Minputs=5 THEN RO(Ii)=Nu(5)
3564 END SELECT
3570 !
3576 Process_data: ! check data for nonzero X & Y values, for permissible
3582 ! error correlations. Convert errors to abs-2-sigma
3588 IF Xv(Ii)*Yv(Ii)=0 THEN
3594 Clunk
3600 PRINT USING "/,K,2/";FNHS(" **** SET# "&VAL$(Ii)&" -NONZERO INPUT REQUIRED **** ")
3606 RETURN
3612 END IF
3618 IF Percent THEN
3624 Ex_=Ex(Ii) ! percent or fractional errors for rho-testing
3630 Ey_=Ey(Ii) ! ditto
3636 ELSE
3642 Ex_=Ex(Ii)/Xv(Ii)*100 ! ditto
3648 Ey_=Ey(Ii)/Yv(Ii)*100 ! ditto
3654 END IF
3660 Check_rho(RO(Ii),Ex(Ii),Ey(Ii),Xv(Ii),Yv(Ii),Percent,Ok)
3666 IF Ok THEN
3672 IF NOT Df THEN PRINT Ii;TAB(26);Xv(Ii);TAB(37);Ex(Ii);TAB(50);Yv(Ii);TAB(61);Ey(Ii);TAB(69);RO(Ii)
3678 Ii=Ii+1
3684 N=N+1
3690 END IF
3696 RETURN
3702 !
3708 !
3714 Edit: ! allow user to edit/correct keyboard-entered data
3720 IF Keyboard THEN
3726 Clear
3732 PRINT "SET#";TAB(27);Nx0$;TAB(38);Er$;TAB(51);Ny0$;TAB(62);Er$;TAB(73);"RHO"
3738 PRINT
3744 FOR I=Kk TO N
3750 PRINT I;TAB(26);Xv(I);TAB(37);Ex(I);TAB(50);Yv(I);TAB(61);Ey(I);TAB(69);RO(I)
3756 NEXT I
3762 PRINT
3768 END IF
3774 !
3780 IF NOT Keyboard AND Df AND Semicolon THEN Plot_points
3786 IF ABS(Symbol)<>3 AND Symbol<>5 THEN
3792 ! insist on assigned errors for error-ellipse/box/cross plot-symbols
3798 Missing_errors=Missing_errors=0
3804 FOR I=Kk TO N
3810 Missing_errors=Missing_errors+(Ex(I)*Ey(I)=0)
3816 NEXT I
3822 IF Missing_errors THEN
3828 LOOP
3834 DISP
3840 DISP "ENTER X-"&Er$&" , Y-"&Er$&" [, Err.-Corr.] FOR SETS WITH UNASSIGNED OR ZERO ERRORS";
3846 Retrieve(R$,Nu(*),Minputs)
3852 EXIT IF Minputs>1
3858 CALL Bad_input(Crud$)

```



```

3864   END LOOP
3870   FOR I=Kk TO N
3876     IF Ninputs=3 AND RO(I)=0 THEN RO(I)=Nu(3)
3882     IF Ex(I)=0 THEN Ex(I)=Nu(1)
3888     IF Ey(I)=0 THEN Ey(I)=Nu(2)
3894     Check_rho(RO(I),Xerr(I),Yerr(I),Xv(I),Yv(I),Percent,Ok)
3900     IF NOT Ok THEN
3906       RO(I)=0
3912       Xerr(I)=0
3918       Yerr(I)=0
3924       GOTO Edit
3930     END IF
3936   NEXT I
3942   GOTO Edit
3948   END IF
3954   END IF
3960   !
3966   IF NOT Keyboard THEN Plot_points ! Don't ask for corrections if only datafile data were entered
3972   DISP USING "#,/,K";"SET TO BE CORRECTED? (&FNCT$&"IF OK)"
3978   Retrieve(R$,Nu(*),Ninputs)
3984   IF Ninputs=0 THEN Plot_points
3990   I=Nu(1)
3996   Ii=I
4002   GOSUB Input_points
4008   M=M-1
4014   GOTO Edit
4020   !
4026   !
4032   Error_box=Draw_errbox(Xerr(I),Yerr(I),Xv(I),Yv(I),Xmm,Ymm,Crt,Symbol)           ! draw an error-box about the point
4038   RETURN
4044   !
4050   Error_cross:      ! Draw error-cross through data-point
4056   IPLDT 0,Yerr(I)
4062   IPLDT 0,-(2*Yerr(I))
4068   IMOVE -Xerr(I),Yerr(I)
4074   IPLDT 2*Xerr(I),0,-1
4080   RETURN
4086   !
4092   Error_ellipse=Draw_ellipse(Xerr(I),Yerr(I),RO(I),Symbol,Xspred,Yspred)       !Draw error-ellipse about data-point
4098   RETURN
4104   !
4110   Polygon=Draw_polygon(Polysides,Polysize,Polyrot,Solid,Symb(*),Jj)           ! Draw open or filled polygon or star symbol
4116   RETURN
4122   !
4128   Keyboard_symbol:IF Plotter_dump THEN Key$=CHR$(Symb(Jj,2))
4134   IF Key$="." THEN ! just put the plotter-pen at the point
4140   DRAW Xv(I),Yv(I)
4146   ELSE
4152   LORG 5
4158   As=.6+.2*(Key$="X" OR Key$="O") ! use aspect-ratio of 0.6 except for X or O symbols, in which case use 0.8
4164   Cs=2.5*Ticksize ! size of alphanumeric plotting-symbol
4170   CSIZE Cs,As
4176   FOR Dx=0 TO Double
4182     IF Hardplot AND Dx THEN MOVE Xv(I),Yv(I)
4188     IMOVE FNCwidth(Cs,Xmm,0,As,0)/15+Dx*Offset/Xmm,FNCheight(Cs,0,Ymm,0)/15+Dx*Offset/Ymm
4194     LABEL Key$
4200   NEXT Dx
4206   END IF
4212   RETURN
4218   !

```

```

4224 Plot_points=PEN Cpen      ! Draw data-point symbols
4230 IF Crt THEN CALL Graph
4236 FOR I=Kk TO N
4242  IF Hardplot THEN
4248    DISP I;TAB(10);DROUND(Xv(I),5);TAB(20);DROUND(Yv(I),5)
4254  ELSE
4260    C_ct=C_ct+(C_ct<200)!"Cache" CRT-plotted points in case of plotter-dump
4266    Cache(C_ct,1)=Xv(I)
4272    Cache(C_ct,2)=Ex(I)
4278    Cache(C_ct,3)=Yv(I)
4284    Cache(C_ct,4)=Ey(I)
4290    Cache(C_ct,5)=RO(I)
4296    Symb(C_ct,1)=Symbol
4302    IF C_ct<=50 THEN
4308      IF L_r(I) THEN
4314        Cache_sname$(C_ct)=Sname$(I)
4320        Cache_l_r(C_ct)=L_r(I)
4326      ELSE
4332        Cache_sname$(C_ct)="
4338        Cache_l_r(C_ct)=0
4344      END IF
4350    END IF
4356    IF Symbol=3 THEN
4362      Symb(C_ct,2)=Polysides
4368      Symb(C_ct,3)=Solid
4374      Symb(C_ct,4)=Polysize
4380      Symb(C_ct,5)=Polyrot
4386    END IF
4392    IF Symbol(>)3 AND Symbol(>)5 THEN Symb(C_ct,1)=Symbol
4398    IF Symbol=5 THEN Symb(C_ct,2)=Keycode
4404  END IF
4410  IF Percent THEN ! convert to 1 or 2 -sigma absolute errors
4416    Xerr(I)=Xv(I)*Ex(I)/100
4422    Yerr(I)=Yv(I)*Ey(I)/100
4428  ELSE
4434    Xerr(I)=Ex(I)
4440    Yerr(I)=Ey(I)
4446  END IF
4452  IF Xv(I)>=Xmin AND Yv(I)>=Ymin AND Xv(I)<=Xmax AND Yv(I)<=Ymax THEN
4458    IF NOT Keyboard_symbol OR Key$="" THEN MOVE Xv(I),Yv(I)
4464    ON ABS(Symbol) GOSUB Error_box,Error_ellipse,Polygon,Error_cross,Keyboard_symbol
4470    IF I<=50 THEN
4476      IF L_r(I) THEN
4482        IF LEN(Sname$(I)) THEN ! Label the sample-name next to the point
4488          LOG 2*(L_r(I)=1)+8*(L_r(I)=-1)
4494          MOVE Xv(I),Yv(I)
4500          CSIZE Tht*.9
4506          IF L_r(I)=1 THEN LABEL " "&Sname$(I)
4512          IF L_r(I)=-1 THEN LABEL Sname$(I)&" "
4518        END IF ! name exists?
4524      END IF ! label the name?
4530    END IF ! I<50?
4536  END IF ! within plotting-limits?
4542  IF NOT Dfile AND NOT Df THEN ! put the point in a "file" in memory
4548    Vc(I,1)=Xv(I)
4554    Vc(I,3)=Yv(I)
4560    Vc(I,2)=Ex(I)
4566    Vc(I,4)=Ey(I)
4572    Vc(I,5)=RO(I)
4578  END IF

```

```

4584 NEXT I
4590 Somepoints=1 ! indicates that at least 1 point has been plotted
4596 IF NOT Plotter_dump THEN
4602 MOVE Xmin,Ymax
4608 IF Hardplot THEN PEN 0
4614 ELSE
4620 RETURN
4626 END IF
4632 !
4638 !
4644 Add:=Clear ! Yorkfit, or plot more points?
4650 PRINTER IS CRT ! Just in case
4656 IF Crt THEN
4662 Alpha
4668 PRINT USING "K,2/";"PRESS "&FNH$("&GRAPHICS "&)" TO VIEW PLOT, "&FNH$("&ALPHA "&)" TO VIEW THIS SCREEN"
4674 END IF
4680 Nn=N
4686 ON DELAY 3600 GOTO Redraw ! If no user action within an hour, restart.
4692 PRINT USING "K,/";"PRESS "&FNH$("&ADD")&" TO INCLUDE THESE POINTS WITH OTHERS FOR A YORKFIT."
4698 PRINT USING "K,/";"PRESS "&FNH$("&NEW POINTS")&" TO PLOT A NEW BATCH OF POINTS."
4704 IF N>1 THEN PRINT USING "K,/";"PRESS "&FNH$("&YORKFIT")&" FOR YORKFIT."
4710 PRINT USING "K,/";"PRESS "&FNH$("&NEW PLOT")&" TO START A NEW PLOT."
4716 IF Cplot AND N THEN PRINT USING "/,K";"press "&FNH$("&CTRL-F")&" to force the Yorkfit line through a point on concordia"
4722 PRINT USING "K,/";"PRESS "&FNH$("&CTRL L")&" TO DRAFT A PHRASE ON THE PLOT"
4728 DISP "(press "&FNH$("&NEW POINTS")&" if you just want to return to the data-entry screen)"
4734 !
4740 Readd:=ON KEY 0 LABEL " HELP" GOTO Call_help4
4746 IF Nn>1 THEN ON KEY 2 LABEL " YORKFIT" GOTO Call_yorkfit
4752 ON KEY 3 LABEL " ADD POINTS" GOTO Retain
4758 ON KEY 8 LABEL "DELETE POINTS" GOTO 4950
4764 ON KEY 4 LABEL " NEW POINTS" GOTO Plotdata
4770 ON KEY 9 LABEL " NEW PLOT" GOTO Redraw
4776 Add_kbd:=ON KBD GOTO 4812
4782 GOTO 4782
4788 !
4794 Adlet:=Letter(Letter$(*),Letter(*),Nphrases,0)
4800 GOTO Add
4806 !
4812 K=FNGet_key
4818 OFF KBD
4824 OFF KEY
4830 SELECT K
4836 CASE 8 ! CTRL-H
4842 GOTO Call_help4
4843 CASE 5 ! CTRL-E
4844 IF Slope THEN Error_envelope
4848 CASE -67 ! CONTINUE key
4854 GOTO Plotdata
4860 CASE 12 ! CTRL-L
4866 GOTO Adlet
4872 CASE 6 ! CTRL-F
4878 GOSUB Force_inter
4884 IF R$="*" THEN Add
4890 GOTO Call_yorkfit
4896 END SELECT
4902 GOTO Readd
4908 !
4914 Retain:=OFF KEY ! Add next points to existing sequence
4920 OFF KBD
4926 Ii=N+1

```

```

4932 Kk=N+1
4938 GOTO More_points
4944 !
4950 GOSUB Delete_points
4956 GOTO Add
4962 !
4968 Options:Clear
4974 OFF KEY
4980 OFF KBD
4986 Alpha
4992 Reopt:PRINT TABXY(1,17);FNCenter$(FNH$(MISCELLANEOUS OPTIONS:))
4998 DISP FNCenter$("Press "&FNH$("CTRL-H")&" for HELP)")
5004 ON KEY 0 LABEL " LOAD PLOT" GOTO Load_plot
5010 ON KEY 1 LABEL " ABS ERRORS" GOSUB Abs_errs
5016 ON KEY 2 LABEL " % ERRORS" GOSUB Percent_errs
5022 ON KEY 3 LABEL " 1-SIG ERRORS" GOSUB One_sig
5028 ON KEY 4 LABEL " 2-SIG ERRORS" GOSUB Two_sig
5034 ON KEY 9 LABEL " ESCAPE" GOTO More_points
5040 IF Crt THEN ON KEY 5 LABEL " STORE PLOT" GOTO Store_plot
5046 IF Cplot=0 THEN
5052 ON KEY 6 LABEL " EXPO FIT" GOTO Exp_fit
5058 ON KEY 7 LABEL " LOG FIT" GOTO Log_fit
5064 ON KEY 8 LABEL " POLYFIT" GOTO Regress_poly
5070 END IF
5076 ON KBD GOTO 5094
5082 GOTO 5082
5088 !
5094 K=FNGet_key
5100 OFF KBD
5106 SELECT K
5112 CASE 8 ! CTRL-H
5118 GOTO Call_help3
5124 CASE -67 ! CONTINUE
5130 GOTO More_points
5136 END SELECT
5142 GOTO Reopt
5148 !
5154 Abs_errs:Percent=0
5160 Eb$=" ABS.-ERRORS"
5166 PRINT Eb$
5172 BEEP
5178 Er$="err"
5184 GOTO Kretrn
5190 !
5196 Percent_errs:Percent=1
5202 Eb$=" %-ERRORS"
5208 PRINT Eb$
5214 BEEP
5220 Er$="%err"
5226 GOTO Kretrn
5232 !
5238 One_sig:Sig=1
5244 BEEP
5250 Ea$="1-SIGMA"
5256 PRINT Ea$&" ERRORS"
5262 GOTO Kretrn
5268 !
5274 Two_sig:Sig=2
5280 BEEP
5286 Ea$="2-SIGMA"

```

```

5292 PRINT Ea&&" ERRORS"
5298 GOTO Kretrn
5304 !
5310 Store_plot:OFF KEY
5316 OFF KBD
5322 Alpha
5328 Clear
5334 INPUT "ENTER A FILE-NAME FOR THIS PLOT (* to escape)?",Gfile$
5340 IF Gfile$("<")"*" THEN
5346 DISP "Storing CRT plot on disk as file "&CHR$(34)&Gfile&&CHR$(34)&"..."
5352 GSTORE Crt_plot(*)
5358 ON ERROR GOTO Bad_gfile
5364 CREATE BDATA Gfile$,7500/256+100
5370 ASSIGN @Path1 TO Gfile$
5376 OUTPUT @Path1;Crt_plot(*)
5382 ASSIGN @Path1 TO *
5388 DISP "GRAPHICS-FILE "&CHR$(34)&Gfile&&CHR$(34)&" STORED."
5394 OFF ERROR
5400 END IF
5406 GOTO More_points
5412 !
5418 Bad_gfile:OFF ERROR
5424 DISP FNCenter$(FNM$("*** UNABLE TO ACCESS GRAPHICS-FILE "&Gfile&&" ***"))
5430 CAT
5436 Clunk
5442 WAIT 3
5448 GOTO More_points
5454 !
5460 Load_plot:OFF KEY
5466 OFF KBD
5472 Clear
5478 Alpha
5484 PRINT TABXY(1,12);"WHAT IS THE NAME OF THE GRAPHICS-FILE THAT YOU WANT TO LOAD (* to escape)?"
5490 INPUT Gfile$
5496 IF Gfile$("<")"*" THEN
5502 Clear
5508 DISP "Loading graphics file "&CHR$(34)&Gfile&&CHR$(34)&"..."
5514 ON ERROR GOTO Bad_gfile
5520 ASSIGN @Path1 TO Gfile$
5526 ENTER @Path1;Crt_plot(*)
5532 Graph
5538 GLOAD Crt_plot(*)
5544 OFF ERROR
5550 DISP
5556 WAIT 2
5562 END IF
5568 GOTO More_points
5574 !
5580 Call_yorkfit:CALL Yorkfit(Cplot,Forced_inter,Forced_age)
5586 ON DELAY 3600 GOTO Redraw ! If no user action within an hour, restart.
5592 Keys2:=IF Crt THEN DISP "PRESS "&FNM$("<" GRAPHICS ">")&" TO VIEW PLOT "
5598 OFF KEY
5604 ON KEY 9 LABEL " NEW PLOT" GOTO Redraw
5610 ON KEY 3 LABEL " ADD POINTS" GOTO Retain
5616 ON KEY 8 LABEL "DELETE POINTS" GOTO 5814
5622 ON KEY 0 LABEL " HELP" GOTO Call_help6
5628 ON KEY 4 LABEL " CONTINUE" GOTO Plotdata
5634 ON KEY 5 LABEL "ERROR ENVELOPE" GOTO Error_envelope
5640 ON KBD GOTO 5658
5646 GOTO 5646

```

```

5652 !
5658 K=FNGet_key
5664 OFF KBD
5670 SELECT K
5676 CASE 8          ! CTRL-H
5682 GOTO Call_help6
5688 CASE -67       ! CONTINUE key
5694 GOTO Plotdata
5695 CASE 5         ! CTRL-E
5696 GOTO Error_envelope
5700 END SELECT
5706 GOTO 5604
5712 !
5718 Error_envelope=PEN Cpen
5724 IF Crt THEN CALL Graph
5730 Err_env(Slope,Errsl,Inter,Errint,X_bar,Xmin,Xmax,Ymin,Ymax)
5736 GOTO Plotdata
5742 !
5748 Print_plot=OFF KEY
5754 Clear
5760 OFF KBD
5790 DUMP GRAPHICS
5802 GOTO More_points
5808 !
5814 GOSUB Delete_points
5820 GOTO Keys2
5826 !
5832 Delete_points=OFF KEY
5838 OFF KBD
5844 Clear
5850 PRINTER IS CRT
5856 PRINT "SET#";TAB(20);Nx0$;TAB(35);Ny0$
5862 FOR I=1 TO Nn
5868 PRINT I;TAB(19);Xv(I);TAB(34);Yv(I)
5874 NEXT I
5880 I=0
5886 DISP "WHICH SET# DO YOU WANT TO DELETE? (&FNCT$&"to escape)";
5892 INPUT I
5898 DISP
5904 IF I<1 OR I>Nn THEN Keys2
5910 Nn=Nn-I
5916 N=N-1
5922 FOR J=I TO N
5928 Xv(J)=Xv(J+1)
5934 Yv(J)=Yv(J+1)
5940 Ex(J)=Ex(J+1)
5946 Ey(J)=Ey(J+1)
5952 Rv(J)=Rv(J+1)
5958 NEXT J
5964 RETURN
5970 !
5976 Draft:Letter(Letter$(*),Letter(*),Nphrases,0)
5982 GOTO More_points
5988 !
5994 Datafile=Convert_vc(Dfile,Uc(*),Pr(*),Cplot,N$(*),Df_xname$,Df_yname$,File$,Ms$(*))
6000 Nx_$=Df_xname$
6006 Ny_$=Df_yname$
6012 RETURN
6018 !
6024 Regress_poly=IF Nn>2 THEN

```

```

6030 CALL Polynomial(Xv(*),Yu(*),Nn,Maxnum,Xmin,Xmax,Xspped,Cpen,Crt,Pr(*),Prtr)
6036 ELSE
6042 GRAPHICS OFF
6048 CALL Bad_input("NEED AT LEAST 3 POINTS FOR POLYNOMIAL REGRESSION")
6054 END IF
6060 GOTO More_points
6066 !
6072 Dump_to_plotter=Alpha      ! Transfer CRT-plot to plotter
6078 Plotter_ok=1
6084 ON TIMEOUT 7,.1 CALL Plotter_timeout
6090 PLOTTER IS Plotdev(2),"HPGL"
6096 IF NOT Plotter_ok THEN More_points
6102 !
6108 OFF TIMEOUT
6114 DATA 1,US,1,1,0,705
6120 RESTORE 6114
6126 READ Cpen,Speed$,Plotter_dump,Hardplot,Crt,Plotter
6132 CALL Plotbox(Crud$,Cplot,0,0,0,0)
6138 IF Cplot THEN CALL Concordia ! draw & label concordia
6144 Clear
6150 FOR Jj=1 TO C_ct
6156   K=1
6162   N=1
6168   Xv(K)=Cache(Jj,1)
6174   Yu(K)=Cache(Jj,3)
6180   Ex(K)=Cache(Jj,2)
6186   Ey(K)=Cache(Jj,4)
6192   RO(K)=Cache(Jj,5)
6198   Symbol=Symb(Jj,1)
6204   Sname$(K)=Cache_sname$(Jj)
6210   IF Jj<50 THEN
6216     Sname$(K)=Cache_sname$(Jj)
6222     L_r(K)=Cache_l_r(Jj)
6228   ELSE
6234     Sname$(K)=""
6240     L_r(K)=0
6246   END IF
6252 GOSUB Plot_points
6258 NEXT Jj
6264 FOR I=1 TO Nlines
6270   Slope=Line(I,1)
6276   Inter=Line(I,2)
6282   CALL Drawline(Slope,Inter,(I),Solid_line,Line(*))
6288   IF Cplot AND I<4 THEN
6294     FOR K=1 TO 8
6300       Cint(K)=Cache_cint(I,K)
6306     NEXT K
6312     CALL Label_inters(Cint*),(I),Cache_msud(I))
6318   END IF
6324 NEXT I
6330 !
6336 FOR Jj=1 TO Nphrases
6342   CALL Letter(Letter$(*),Letter*),(Nphrases,Jj)
6348 NEXT Jj
6354 !
6360 Plotter_dump=0
6366 GOTO More_points
6372 !
6378 Call_help3:CALL Help3
6384 GOTO Options

```

```

6390 !
6396 Call_help4=CALL Help4(Cplot)
6402 GOTO Add
6408 !
6414 Call_help6=CALL Help6
6420 GOTO Keys2
6426 !
6432 Exp_fit=CALL Expo_fit(Nn,Xv(*),Yv(*),Xmin,Xmax,Ymin,Ymax,Crt,Cpen,Pr(*),Prtr)
6438 GOTO More_points
6444 Log_fit=CALL Logar_fit(Xv(*),Yv(*),Nn,Xmin,Xmax,Ymin,Ymax,Crt,Cpen,Pr(*),Prtr)
6450 GOTO More_points
6456 !
6462 !
6468 ! *****
6474 Endmain=END ! end of main program *****
6480 !
6486 !
6492 Z=DEF FNZ(X) ! utility function for misc. numeric formatting needs
6498 RETURN INT<LGT<ABS<DROUND(X,9)+<X=0>>>>
6504 FNEND
6510 !
6516 !
6522 Tick=SUB Tick(Range,Tick_interval) ! Returns best tick-interval
6528 Tick_interval=10^FNZ(Range)/8
6534 WHILE ABS(Range/Tick_interval)>12
6540 Tick_interval=2*Tick_interval
6546 END WHILE
6552 B=ABS(Tick_interval)/10^FNZ(Tick_interval)
6558 IF B<INT(B) THEN Tick_interval=INT(B)*10^FNZ(Tick_interval)
6564 Tick_interval=DROUND(Tick_interval,8)
6570 SUBEND
6576 !
6582 !
6588 Axis=SUB Axis(P,Spred,Tik,Ticksizex,xy) ! Draw axis & ticks
6594 ! xy=1 for X-axis, xy=0 for Y-axis
6600 C=1-2*(xy=0)
6606 Ntix=DROUND(Spred/Tik,3)
6612 Etix=FNEven(Ntix)
6618 FOR I=1 TO Ntix
6624 Etik=FNEven(I)
6630 IF (P=1)*Etik OR (P=-1)*(<Etik EXOR Etix>=0) THEN
6636 M=Ticksizex
6642 ELSE
6648 M=Ticksizex/2
6654 END IF
6660 IDRAW P*Tik*xy,P*Tik*(NOT xy)
6666 IDRAW C*P*M*(NOT xy),P*M*xy
6672 IDRAW P*M*(NOT xy),-P*M*xy
6678 NEXT I
6684 SUBEND
6690 !
6696 !
6702 De=DEF FNDe(N,D)
6708 ! Round value N to D significant figs.(to lowest least-sf,unlike DROUND)
6714 A=10^(D-FNZ(N)-1)
6720 B=DROUND(ABS(N*A),9)
6726 RETURN SGN(N)*INT(B)/A
6732 FNEND
6738 !
6744 !

```



```

6750 Decimal=DEF FNDecimal(X) ! Returns #digits beyond decimal pt.
6756 X=DROUND(X,9)
6762 A=0
6768 N=ABS(X)
6774 WHILE N
6780 N=DROUND(10*(ABS(N)-INT(ABS(N))),8)
6786 A=A+1
6792 END WHILE
6798 RETURN A-1+(NOT A)
6804 FNEND
6810 !
6816 !
6822 Char=DEF FNChar(X,Lzero) ! #Characters required for the number X.
6828 RETURN 1+(ABS(X)>1)*(FNZ(X)+(X<>INT(X)))+(X<0)+FNDecimal(X)+(ABS(X)<1)*Lzero
6834 FNEND
6840 !
6846 !
6852 H=DEF FNH$(String$) ! put string in inverse-video, padded with one leading
6858 ! and one trailing blank
6864 RETURN CHR$(129)&" "&String&" " &CHR$(128)
6870 FNEND
6876 !
6882 !
6888 Retrieve=SUB Retrieve(Input_string$,Numeric_value(*),Ninputs,OPTIONAL Enterstring)
6894 !From a string input, extract 1 or more numeric values, separated by commas
6900 ! Ninputs is the # of numeric values in the string
6906 IF NPAR=3 THEN
6912 Estring=1
6918 ELSE
6924 Estring=Enterstring
6930 END IF
6936 IF Estring THEN LINPUT Input_string$ ! otherwise string supplied by calling context
6942 MAT Numeric_value= (0)
6948 Comma=1
6954 Ninputs=0
6960 ON ERROR GOTO Done
6966 WHILE Comma>0
6972 Numeric_value(1+Ninputs)=VAL(Input_string$)! extract number
6978 Ninputs=1+Ninputs ! increment #-of-values counter
6984 Comma=POS(Input_string$,"") ! position of next comma
6990 Input_string$=Input_string$[1+Comma] ! strip segment of string already extracted
6996 END WHILE
7002 Done=SUBEND
7008 !
7014 !
7020 Retrieve_str=SUB Retrieve_str(Input_string$,Output_string$(*),Ninputs)
7026 ! extract 1 or more strings from a compound string, where the strings are separated by commas
7032 LINPUT Input_string$
7038 MAT Output_string$= ("")
7044 Ninputs=0
7050 IF Input_string$="" THEN SUBEXIT
7056 ON ERROR GOTO Done
7062 LOOP
7068 Comma=POS(Input_string$,"") ! position of nearest comma
7074 End_part=Comma*(Comma<>0)+(LEN(Input_string$)+1)*(Comma=0) ! position of last character of this section of the input string
7080 Output_string$(1+Ninputs)=Input_string$[1,End_part-1] ! put this section of the input string as an element in the output-string array
7086 Ninputs=1+Ninputs ! increment #-of-strings counter
7092 EXIT IF NOT POS(Input_string$,"") ! done if no more commas present
7098 Input_string$=Input_string$[1+Comma] ! strip part of input string already extracted

```

```

7104 END LOOP
7110 Done:SUBEND
7116 !
7122 !
7128 Remove:SUB Remove(Input_string$,Output_string$,Length)
7134 ! remove super- (CTRL ^) and sub- (CTRL *) script indicators from label
7140 Output_string$=""
7146 FOR J=1 TO LEN(Input_string$)
7152 N=NUM(Input_string$[J,J])
7158 SELECT N
7164 CASE 10,30
7170 ! do nothing
7176 CASE ELSE
7182 Length=1+Length
7188 Output_string$[Length,Length]=Input_string$[J,J]
7194 END SELECT
7200 NEXT J
7206 SUBEND
7212 !
7218 !
7224 Draw_ellipse:SUB Draw_ellipse(Xerr,Yerr,R0,Symbol,Xspred,Yspred)
7230 Xe=Xerr*(1+.0001*(Xerr=Yerr))
7236 Corr=R0*(R0<1)+.999999*(R0=1) ! Guard against a rho of 1
7242 Angle=.5*ATN(2*Corr*Xe*Yerr/(Xe^2-Yerr^2))
7248 ! ANGLE is an angle that the axis of the ellipse makes with the X-axis
7254 C1=2*(1-Corr^2)*1.224^2
7260 C2=1/COS(2*Angle)
7266 Ux=Xe^2
7272 Uy=Yerr^2
7278 A=SQR(C1/((1+C2)/Ux+(1-C2)/Uy)) ! Length of major axis of ellipse
7284 B=SQR(C1/((1-C2)/Ux+(1+C2)/Uy)) ! Length of minor axis of ellipse
7290 St=A/INT(A/(Xspred/400)+1) ! Step-size along X-axis
7296 St=A/INT(A/St+.0001) ! force step to divide evenly
7302 Sin_angle=SIN(Angle)
7308 Cos_angle=COS(Angle)
7314 FOR Seq=1 TO 1+(Symbol<0)
7320 FOR K=1 TO -1 STEP -2
7326 Start_x=K*A
7332 End_x=-K*(A+St)
7338 FOR X=Start_x TO End_x STEP -(K*St)
7344 IF (K=1)*(X<-A*1.001) OR (K=-1)*(X)*A*1.001 THEN 7428
7350 Z=1-(X/A)^2
7356 IF Z>=0 THEN
7362 Y=K*B*SQR(Z)
7368 ELSE
7374 Y=0
7380 END IF
7386 IF Symbol>0 OR K=1 OR DROUND(X,9)<>DROUND(End_x,9) THEN PLOT X*Cos_angle-Y*Sin_angle,X*Sin_angle+Y*Cos_angle
7392 IF Seq=2 THEN ! fill in the ellipse
7398 K=-K
7404 IF DROUND(X,9)<>DROUND(End_x,9) THEN PLOT X*Cos_angle+Y*Sin_angle,X*Sin_angle-Y*Cos_angle
7410 END IF
7416 NEXT X
7422 IF Seq=2 THEN SUBEXIT
7428 NEXT K
7434 NEXT Seq
7440 SUBEND
7446 !
7452 !
7458 Draw_polygon:SUB Draw_polygon(Polysides,Polysize,Polyrot,Solid,Symb(*),Jj)

```

```

7464 ! draw a polygon of arbitrary size, #sides, orientation,infill
7470 COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Linx1,Linx2,Liny1,Liny2,Xspred,Yspred,Xloc,Yloc,Xmm,Ymm,Nx$,Ny$,NxD$,NyD$,Double
7476 COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
7482 COM /Plotbox2/ Nxr$,Nyr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Icksize,Xl,Yl,Xend,Yend,Speed$,Crt,Hardplot,Cpen,X
gu,Ygu
7488 !
7494 IF Plotter_dump THEN ! cache polygon-specs
7500   Polysides=Symb(Jj,2)
7506   Solid=Symb(Jj,3)
7512   Polysize=Symb(Jj,4)
7518   Polyrot=Symb(Jj,5)
7524 END IF
7530 !
7536 Star=(Polysides<0) ! is this to be a "star"-shaped figure?
7542 S7475=(Hardplot AND Solid AND NOT Star AND Polysides<7)
7548 IF S7475 THEN ! use HP-7475 HPGL command to draw solid poly
7554   S=0
7560   DATA 0,0,120,90,72,60,25.714,0 ! starting=position, in degrees
7566   RESTORE 7560
7572   IF Polysides<9 THEN
7578     FOR I=1 TO Polysides
7584       READ S
7590     NEXT I
7596   END IF
7602   P=Polysides*(Polysides<10)+50*(Polysides=10) ! use 50 sides if polysides>9
7608   OUTPUT Plotdev(2);"W6"UVAL$(Polysize*17)&","&UVAL$(S+Polyrot)&","360,"&UVAL$(360/P)&";"
7614 END IF
7620 Xp=Polysize/Xmm/2.3 ! make diameter equal to polysize in millimeters
7626 Yp=Polysize/Ymm/2.3
7632 D=.2+.1*Crt ! effective pen-thickness in mm
7638 M_polys=(S7475=0)*((Hardplot AND (Ticksize(1.4) AND Polysides>0 AND Polysize>3)+50*Solid)
7644 IF M_polys=0 AND Double AND S7475=0 THEN M_polys=1
7650 ! M_polys is the number of (increasingly smaller) polygons to be drawn
7656 FOR I=0 TO M_polys
7662   X=Xp-I*D/Xmm
7668   Y=Yp-I*D/Ymm
7674   S=360/ABS(Polysides)
7680   E=.5-(ABS(Polysides)-3)/5 ! degree of pucker-indent for star
7686   IF X<0 OR Y<0 THEN SUBEXIT ! stop when radius of zero is passed
7692   !
7698   FOR J=Polyrot TO Polyrot+360 STEP S
7704     RPLOT X*COS(J),Y*SIN(J)
7710     IF Star THEN ! do a pucker
7716       C=J+S/2
7722       RPLOT E*X*COS(C),E*Y*SIN(C)
7728     END IF
7734   NEXT J
7740 NEXT I
7746 SUBEND
7752 !
7758 !
7764 Draw_errbox=SUB Draw_errbox(Xerr,Yerr,X,Y,Xmm,Ymm,Crt,Symbol)
7770 ! draw a rectangle whose dimensions correspond to the uncertainties of the point
7776 IMOVE -Xerr,-Yerr ! start at the lower-left corner of the box
7782 W=2*Xerr ! box height
7788 H=2*Yerr ! box width
7794 !
7800 ! use RECTANGLE command for CRT-plots or open boxes
7806 IF NOT Crt OR (Crt AND Symbol<0) THEN RECTANGLE W,H
7812 IF Crt AND Symbol<0 THEN RECTANGLE W,H,FILL

```

```

7818 !
7824 IF NOT Crt AND Symbol<D THEN
7830   Penthick=.36/Ymm/(1+Crt)!  UU's per .36 mm in Y-direction
7836   D=1
7842   CLIP X-Xerr,X+Xerr,Y-Yerr,Y+Yerr
7848   REPEAT ! fill in the box (plotter only)
7854     IDRAW D*U,0
7860     IDRAW 0,Penthick
7866     Height=Height+Penthick
7872     D=-D
7878   UNTIL Height>H*2*Penthick ! quit when filled in box (+ abt 1-2 lines)
7884   CLIP Xmin,Xmax,Ymin,Ymax
7890 END IF
7896 SUBEND
7902 !
7908 !
7914 Yorkfit:SUB Yorkfit(Cplot,Forced_inter,Forced_age)
7920 ! line-fitting algorithm modified from D. York, EPSL v. 5, p. 320-324, 1969
7926 OPTION BASE 1
7932 COM /Yorkfit1/ Xv(*),Yv(*),Ex(*),Ey(*),RD(*),N,Maxyor,Nn,Prtr,Ea$,Eb$,Er$
7938 COM /Yorkfit2/ Xerr(*),Yerr(*),Sig,Line(*),Percent,Nlines,N_interlabels,Cint(*),Solid_line,Cache_cint(*),Cache_msud(*)
7944 COM /Yorkfit3/ Slope,Errsl,Inter,Errint,X_bar
7950 DIM X_wt(200),Y_wt(200),R(200),Weight(200),Mean_wt(200),Yorkslope(5),Yorkinter(5)
7956 COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Linx1,Linx2,Liny1,Liny2,Xsprd,Ysprd,Xloc,Yloc,Xmm,Ymm,Nx$,Ny$,Nx$,Ny$,NxD$,NyD$,Double
7962 COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
7968 COM /Plotbox2/ Nxr$,Nyr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Ticksizex,Xl,Yl,Xend,Yend,Speed$,Crt,Hardplot,Cpen,X
gu,Ygu
7974 DIM Y_errint(5),Y_errsl(5),Name$(70),Isotope$(4)[3],Lambda(4),Xwt(200),Ywt(200),Interxy(2,2)
7980 DIM E_x0(200),E_y0(200),E_76(200),E_x(200),E_y(200),Xbar(5),Ybar(5),Cint_r(8),E_760(200)
7986 !
7992 !   Rb-87, Sm-147, U-238,   U-235,   Th-232 decay constants (/my)
7998 DATA 1.42E-5,6.54E-6,1.55125E-4,9.8485E-4,4.9475E-5
8004 READ Lambda(*)
8010 !
8016 IF N>Maxyor THEN
8022   Bad_input(FMH$( "*** CAN ONLY YORKFIT UP TO "BUAL$(Maxyor)&" POINTS ***"))
8028   SUBEXIT
8034 END IF
8040 !
8046 OFF KEY
8052 OFF KBD
8058 Alpha
8064 N=Nn
8070 !
8076 ! Determine if isochron data by looking for isotope #s in X & Y names
8082 DATA 87,86,87,86,143,144,147,144,206,204,238,204,207,204,235,204,208,204,232,204,207,204,206,204,207,206,204,206
8088 RESTORE 8082
8094 FOR I=1 TO 7
8100   READ Isotope$(*)
8106   A=POS(Ny$,Isotope$(1))*POS(Ny$,Isotope$(2))*POS(Nx$,Isotope$(3))*POS(Nx$,Isotope$(4))
8112   IF A AND I<=5 THEN Isochron=I ! Type of std isochron (Rb-Sr,Sm/Nd...)
8118   IF A AND I>=6 THEN Pbpb=I-5 ! Normal or inverse Pb-Pb isochron
8124 NEXT I
8130 !
8136 LOOP
8142 Clear
8148 PRINT USING "K,/": "YORKFIT:"
8154 ! Check for unassigned errors and error-correlations
8160 Missing_errors=0
8166 Zero_rhos=0

```

```

8172 FOR I=1 TO N
8178     Missing_errors=Missing_errors+(Ex(I)*Ey(I)=0)
8184     IF Isochron(3 OR Cplot THEN Zero_rhos=Zero_rhos+(RO(I)=0)
8190     ! Don't ask for error-correlations for Rb/Sr or Sm/Md isochrons
8196     NEXT I
8202 !
8208 IF Missing_errors AND NOT Zero_rhos THEN
8214     PRINT "X-ERRS, Y-ERRS FOR SAMPLES WITH UNASSIGNED OR ZERO ERRORS?"
8220     LOOP
8226     Retrieve(R$,Nu(*),Ninputs)
8232     IF R$="*" THEN SUBEXIT
8238     EXIT IF Nu(1) AND Nu(2)
8244     Bad_input("NEED AT LEAST 2 NONZERO RESPONSES")
8250     END LOOP
8256     X_error=Nu(1)
8262     Y_error=Nu(2)
8268     END IF
8274 IF Missing_errors AND Zero_rhos THEN
8280     IF Percent THEN PRINT "X-ERRS, Y-ERRS";
8286     IF NOT Percent THEN PRINT "X-ERRS, Y-ERRS";
8292     IF NOT Cplot THEN PRINT " [, ERR-CORRS]";
8298     IF Cplot THEN PRINT ", ERR-CORRS";
8304     PRINT " FOR SAMPLES WITH UNASSIGNED OR ZERO VALUES?"
8310     LOOP
8316     Retrieve(R$,Nu(*),Ninputs)
8322     IF R$="*" THEN SUBEXIT
8328     EXIT IF Nu(1) AND Nu(2)
8334     Bad_input("NEED AT LEAST 2 NONZERO RESPONSES")
8340     END LOOP
8346     X_error=Nu(1)
8352     Y_error=Nu(2)
8358     Rho=Nu(3)
8364     END IF
8370 IF Zero_rhos AND NOT Missing_errors THEN
8376     PRINT "ERR-CORRS FOR SAMPLES WITH UNASSIGNED OR ZERO VALUES (&fnc%$IF ZERO)?"
8382     LOOP
8388     Retrieve(R$,Nu(*),Ninputs)
8394     IF R$="*" THEN SUBEXIT
8400     Rho=Nu(1)
8406     EXIT IF ABS(Rho)<=1
8412     Bad_input("ERROR-CORRELATIONS MUST BE BETWEEN -1 AND +1")
8418     END LOOP
8424     END IF
8430 EXIT IF NOT Zero_rhos OR NOT Cplot OR Rho
8436     Bad_input("CONCORDIA X- & Y-VALUES ARE ALWAYS CORRELATED! (usually 0.9 to 0.995)")
8442     END LOOP
8448 FOR I=1 TO N
8454     IF Ex(I)=0 THEN
8460         Ex(I)=X_error
8466         Xerr(I)=X_error*(Percent*Nu(I)/100+(Percent=0))
8472     END IF
8478     IF Ey(I)=0 THEN
8484         Ey(I)=Y_error
8490         Yerr(I)=Y_error*(Percent*Nu(I)/100+(Percent=0))
8496     END IF
8502     IF RO(I)=0 THEN RO(I)=Rho
8508     NEXT I
8514 !
8520 IF Cplot AND Forced_inter AND N(Maxyor THEN
8526     N=N+1

```

```

8532   Xv(N)=FNA(Forced_age)
8538   Yv(N)=FNB(Forced_age)
8544   Xerr(N)=Xv(N)/1.E+6
8550   Yerr(N)=Yv(N)/1.E+6
8556   RO(N)=0
8562 END IF
8568 !
8574 Alpha
8580 IF Prtr=2 THEN ! ask for a name for the yorkfit if printer output
8586   PRINT "NAME FOR THIS YORKFIT?";
8592   LINPUT Name$(1,70)
8598 END IF
8604 !
8610 PRINTER IS Pr(2)
8616 Clear
8622 PRINT RPT$("*",30)&"YORKFIT"&RPT$("*",30)
8628 IF Prtr=2 THEN
8634   PRINT Name$
8640 ELSE
8646   PRINT
8652 END IF
8658 PRINT USING "/,25X,K";Ea$&Eb$ ! error-type heading
8664 IF Cplot THEN PRINT "SET#";TAB(8);Nx0$;TAB(21);Er$;TAB(31);Ny0$;TAB(43);Er$;TAB(56);"Err-Correl.";TAB(72);"X7/6 err"
8670 IF NOT Cplot THEN PRINT "SET#";TAB(9);Nx0$;TAB(23);Er$;TAB(34);Ny0$;TAB(47);Er$;TAB(61);"Err-Correl."
8676 PRINT
8682 !
8688 !
8694 Ok=0 ! Ok Remains zero until a model-1 fit is obtained
8700 Model5=0 ! Don't use Model-5 - gives funny convergence
8706 Slope=.1 ! Trial slope to start with
8712 Model=1 ! Model number
8718 Itermax=15 ! Maximum# of iterations before giving up
8724 M3_itermax=30 ! Maximum# of Model-3 passes before giving up
8730 M4_itermax=15 ! ditto, Model-4
8736 Toler=.02 ! Model-4 tolerance on MSWD convergence
8742 Model4_mult=1 ! for Model-4 fit: starting error-additive
8748 M4_count=0 ! Model-4 pass-counter
8754 M3_count=0 ! Model-3 iteration counter
8760 !
8766 !
8772 Start_model:Slope0=Slope
8778 !
8784 Iterate:Iteration=1 ! Iteration-counter for slope-determination
8790 ON KEY 9 LABEL " ABORT FIT" GOTO Yorkescape
8796 REPEAT
8802   Slope=Slope0
8808   IF Iteration=1 OR Model=2 OR Model=3 THEN
8814     FOR I=1 TO N
8820       SELECT Model
8826       CASE 1
8832         ! Model-1 weighting - analytical errors only (York's original)
8838         IF NOT Cplot THEN PRINT I;TAB(8);Xv(I);TAB(22);Ex(I);TAB(33);Yv(I);TAB(46);Ey(I);TAB(60);RO(I)
8844         Y_wt(I)=(Sig/Yerr(I))^2
8850         Ywt(I)=Y_wt(I)
8856         X_wt(I)=(Sig/Xerr(I))^2
8862         Xwt(I)=X_wt(I)
8868         R(I)=RO(I)
8874         IF R(I)=1 THEN R(I)=1-1.E-9 ! correlations of exactly 1 dangerous
8880         IF Cplot THEN ! calculate 207/206 error to show user
8886           E_x0(I)=Xerr(I)/Xv(I)! fractional error in X

```

```

8892      E_y0(I)=Yerr(I)/Yv(I)!      "      "      Y
8898      !      "      "      207/206
8904      E_760(I)=SQR(E_x0(I)^2+E_y0(I)^2-2*RO(I)*E_x0(I)*E_y0(I))
8910      IF NOT Forced_inter OR I<N THEN PRINT I;TAB(6);Xv(I),TAB(21);Ex(I);TAB(30);Yv(I);TAB(42);Ey(I);TAB(55);RO(I);TAB(72
);DROUND<100*E_760(I),3)
8916      END IF
8922      !
8928      CASE 2 ! Model-2 wting: Equal wts, uncorrell. error (implies
8934      ! knowing nothing about the actual cause of scatter)
8940      IF I<N OR NOT Cplot OR NOT Forced_inter THEN
8946      X_wt(I)=1
8952      Y_wt(I)=1/Slope^2
8958      R(I)=0
8964      END IF
8970      !
8976      CASE 3 ! Model-3 weighting: analytical errors plus a normally-dis-
8982      ! tributed variation in the initial-ratio. The initial-
8988      ! ratio variation is unknown & must be estimated by the
8994      ! program.
9000      T=(Yerr(I)/Sig)^2
9006      R(I)=RO(I)*SQR(T/(T+Sig_y_init^2))
9012      Y_wt(I)=1/(T+Sig_y_init^2)
9018      !
9024      CASE 4
9030      ! MODEL-4 weights points according to their degree of discordance along
9036      ! the best-fit line. The errors of each point are calculated from the
9042      ! the sum of the variances due to analytical error and to an error-
9048      ! ellipse for equal X and Y errors and an error-correlation of -1
9054      ! (equivalent to increasing 207/206 errors). The size of the added
9060      ! error-ellipse goes from zero for completely concordant to infinite
9066      ! for completely discordant points. Student's-t for MODEL-4 is calcu-
9072      ! lated by assigning the 2 highest-weighted points a "point" value of
9078      ! 1, and the remaining points some fractional "point" value based on
9084      ! their MODEL-4 weight. MODEL-4 only allowed for >5 points.
9090      ! Discord is the fractional discordance of a point.
9096      ! Errmult is multiplier for added errors according to discordance
9102      !
9108      IF Old_apex THEN Discord=SQR((Interxy(2,1)-Xv(I))^2+(Interxy(2,2)-Yv(I))^2)/Chordlength
9114      IF Young_apex THEN Discord=SQR((Interxy(1,1)-Xv(I))^2+(Interxy(1,2)-Yv(I))^2)/Chordlength
9120      Errmult=Discord/(1-Discord)*Model4_mult/100
9126      ! add variances proportional to discordance
9132      E_x(I)=SQR(E_x0(I)^2+Errmult^2) ! new fract. X-error
9138      E_y(I)=SQR(E_y0(I)^2+Errmult^2) ! new fract. Y-error
9144      U76=E_76(I)^2+(Errmult*2)^2 ! new 7/6 variance
9150      R(I)=(E_x(I)^2+E_y(I)^2-U76)/(2*E_x(I)*E_y(I)) ! new rho
9156      Y_wt(I)=(Sig/(Yv(I)*E_y(I)))^2 ! 1/Y-variance
9162      X_wt(I)=(Sig/(Xv(I)*E_x(I)))^2 ! 1/X-variance
9168      CASE 5
9174      ! Model-5 weighting - use observed X- & Y-residuals to estimate X
9180      ! & Y-residuals to estimate X- and Y-errors of each point.
9186      ! Assume zero error-correlation.
9192      Xres=-(Slope*Weight(I)/X_wt(I)*(Inter+Slope*Xv(I)-Yv(I)))
9198      Yres=Weight(I)/Y_wt(I)*(Inter+Slope*Xv(I)-Yv(I))
9204      ! if resid is w.in .3-sigma(analytical), use the anal. error
9210      A=1/Xres^2
9216      IF A<11*Xwt(I) THEN
9222      X_wt(I)=A
9228      ELSE
9234      X_wt(I)=Xwt(I)
9240      END IF

```

```

9246      A=1/Yres^2
9252      IF A<11*Ywt(I) THEN
9258          Y_wt(I)=A
9264      ELSE
9270          Y_wt(I)=Ywt(I)
9276      END IF
9282      R(I)=0
9288      END SELECT
9294      Mean_wt(I)=SQR(X_wt(I)*Y_wt(I))
9300      NEXT I
9306  END IF
9312  !
9318  DATA 0,0,0,0,0,0
9324  RESTORE 9318
9330  READ X_s,Y_s,Weightsum,C,D,E
9336  ON ERROR GOTO Fail
9342  !
9348  FOR I=1 TO N
9354      Weight(I)=X_wt(I)*Y_wt(I)/(Slope^2*Y_wt(I)+X_wt(I)-2*Slope*R(I))*Mean_wt(I))
9360      Weightsum=Weightsum+Weight(I)
9366      X_s=X_s+Weight(I)*Xv(I)
9372      Y_s=Y_s+Weight(I)*Yv(I)
9378  NEXT I
9384  Xbar(Model)=X_s/Weightsum
9390  Ybar(Model)=Y_s/Weightsum
9396  FOR I=1 TO N
9402      Wt_sqr=Weight(I)^2
9408      U=Xv(I)-Xbar(Model)
9414      V=Yv(I)-Ybar(Model)
9420      C=C+(U^2/Y_wt(I)-U^2/X_wt(I))*Wt_sqr
9426      D=D+(U*U/X_wt(I)-R(I)*U^2/Mean_wt(I))*Wt_sqr
9432      E=E+(U*U/Y_wt(I)-R(I)*U^2/Mean_wt(I))*Wt_sqr
9438  NEXT I
9444  !
9450  Slope0=C^2+4*D*E
9456  IF Slope0<0 THEN Fail ! test for square-root of negative number
9462  Slope0=(SQR(Slope0)-C)/(2*D)
9468  IF Slope0=0 THEN Slope0=1.E-99
9474  DISP "MODEL "&VAL$(Model)&":   Iteration "&VAL$(Iteration)&"   Slope= "&VAL$(Slope0)
9480  Iteration=1+Iteration
9486  IF Iteration>Itermax THEN Fail
9492  UNTIL ABS((Slope0-Slope)/Slope0)<.0001 !.01% tolerance on slope-convergence
9498  !
9504  Converged=OFF ERROR
9510  Slope=Slope0
9516  Yorkslope(Model)=Slope
9522  Slope=Slope0
9528  Inter=Ybar(Model)-Slope*Xbar(Model)
9534  Yorkinter(Model)=Inter
9540  !
9546  IF Model=3 THEN
9552      M3_count=1+M3_count
9558      IF M3_count>M3_itermax THEN Fail
9564  END IF
9570  !
9576  ! Error algorithm of Titterton & Halliday, Chem Geol v 26, p 183
9582  D=0
9588  E=0
9594  Sums=0
9600  FOR I=1 TO N

```



```

9606   Y_resid=Yv(I)-(Slope*Xv(I)+Inter) ! unweighted Y-residuals
9612   Sums=Sums+Weight(I)*Y_resid^2 ! same as York's SUMS
9618   X=Xv(I)-Weight(I)*Y_resid*(R(I)*Mean_wt(I)-Slope*Y_wt(I))/(Y_wt(I)*X_wt(I))
9624   D=D+X*Weight(I)
9630   E=E+X^2*Weight(I)
9636   NEXT I
9642   !
9648   Nu=N-2 ! Degrees of freedom
9654   Nu1=Nu*(Nu>0)+(Nu<=0) ! in case fitting only 23 points
9660   Mswd=(Nu>0)*Sums/Nu1 ! Mean square of weighted deviates
9666   IF Model=4 AND ABS(Mswd-1)>>Toler THEN
9672     M4_count=1+M4_count
9678     IF M4_count>M4_itermax THEN Fail
9684     Model4_mult=Model4_mult*SQR(Mswd) ! increase error-multiplier
9690     DISP "Model 4: Pass ";M4_count," MSWD: ";Mswd
9696     CALL Model4_calc(Slope,Inter,Cint(*),Interxy(*),Chordlength)
9702     GOTO Start_model
9708   END IF
9714   !
9720   IF Model=3 AND ABS(Mswd-1)>>.01 THEN ! increase estimated initial-ratio variation
9726     Sig_y_init=Sig_y_init*SQR(Mswd) ! est. var. in init. ratio to account for scatter
9732     DISP "Try "&VAL$(M3_count)&": MSWD= "&VAL$(DROUND(Mswd,3))&" for init-Y var. of "&VAL$(DROUND(2*Sig_y_init,3))
9738     GOTO Iterate
9744   END IF
9750   !
9756           ! Calculate degrees of freedom by weighting number of
9762   IF Model=4 THEN ! points by assigning max-wtd point & next-to-max wtd
9768     Ncount=1 ! point a "point" value of 1, and all others a lesser
9774           ! value according to their yorkfit weight.
9780     Maxweight=MAX(Weight(*))
9786     Max2weight=0 ! Find second-highest weighted point
9792     FOR I=1 TO N
9798       IF Weight(I)>Max2weight AND Weight(I)<Maxweight THEN Max2weight=Weight(I)
9804     NEXT I
9810     FOR I=1 TO N
9816       IF Weight(I)<Maxweight THEN Ncount=Ncount+Weight(I)/Max2weight
9822     NEXT I
9828     Nu=Ncount-2 ! can be non-integral or <1
9834   END IF
9840   !
9846   Students_t=FNStudents_t(Nu)
9852   Errsl_apr=Weightsum/(E*Weightsum-D^2) ! a priori (from analytical errors only) 1-sigma
9858           ! error on slope
9864   IF Errsl_apr>0 THEN ! guard against sqrt(neg. #)
9870     Errsl_apr=SQR(Errsl_apr)
9876   ELSE
9882     Errsl_apr=0
9888   END IF
9894   Errint_apr=Errsl_apr*SQR(E/Weightsum) ! a priori 1-sigma error on intercept
9900   Errint_incsc=Errint_apr*SQR(Mswd) ! 1-sigma error on intercept calculated
9906           ! from actual scatter
9912   IF Model=1 THEN Sig_y_init=Errint_incsc ! in case do a model-3 fit later
9918   Errsl_incsc=Errsl_apr*SQR(Mswd) ! 1-sigma "including scatter" slope-err
9924   PRINT
9930   !
9936   IF Model=1 THEN ! calculate chi-square probability
9942     SELECT Mswd
9948     CASE >2D ! obviously zero-probability
9954       Prob=0
9960     CASE <.0001 ! obviously near-1 probability

```

```

9966     Mswd=0
9972     Prob=1
9978     CASE ELSE      ! calculate probability from cumulative chi-square distr.
9984     ON ERROR GOTO Underflow
9990     D=1
9996     A=1
10002    IF NOT FNEven(Nu) THEN ! Nu odd
10008    I=0
10014    REPEAT
10020    I=I+2
10026    D=D*Sums/(Nu+I)
10032    A=A*D
10038    UNTIL D<1.E-7
10044    D=SQR(PI)
10050    FOR I=.5 TO Nu/2
10056    D=D*I
10062    NEXT I
10068    Prob=1-EXP(-(Sums/2))*(Sums/2)^(Nu/2)*A/D
10074    ELSE! Nu even
10080    FOR I=1 TO Nu/2-1
10086    D=D*Sums/(2*I)
10092    A=A*D
10098    NEXT I
10104    Prob=A*EXP(-(Sums/2))
10110    END IF
10116    END SELECT
10122    OFF ERROR
10128    END IF
10134    !
10140    Yorkprint:IF Prob<.0001 THEN Prob=0 ! don't print out low probabilities
10146    DISP
10152    FOR Printer=1 TO Prtr
10158    PRINTER IS Pr(Printer)
10164    SELECT Model
10170    CASE 1
10176    IF Printer=2 THEN PRINT RPT$(" ",80)
10182    IF NOT Cplot OR NOT Forced_inter THEN
10188    PRINT "***MODEL 1 SOLUTION -- ASSUMES ALL SCATTER DUE TO ANALYTICAL ERROR**"
10194    ELSE
10200    PRINT "      **** PSEUDO MODEL-1 SOLUTION ... FORCED THROUGH "&VAL$(DROUND(Forced_age,3))&" Ma ****"
10206    END IF
10212    Ok=1          ! successful model-1 fit
10218    Mswd0=Mswd    ! model-1 mswd - need to remember
10224    IF Prob>.15 THEN ! use A PRIORI errors of prob. of fit is >15%
10230    Y_errs1(1)=1.96*Errs1_apr
10236    Y_errint(1)=1.96*Errint_apr
10242    END IF
10248    CASE 3
10254    PRINT USING "/,K";"*****MODEL 3 SOLUTION -- ASSUMES SCATTER IS DUE TO ANALYTICAL ERROR"
10260    PRINT USING "K,/" ;"*****PLUS NORMALLY-DISTRIBUTED ERROR IN INITIAL "&Ny$&"*****"
10266    PRINT "CALCULATED VARIATION IN INITIAL ";Ny$;"=";DROUND(2*Sig_y_init,3);" (2-SIGMA)"
10272    CASE 2
10278    PRINT USING "/,K";"*****MODEL 2 SOLUTION -- EQUALLY WEIGHTED POINTS WITH UNCORRELATED"
10284    PRINT "*****ERRORS; ASSUMES NORMAL DISTRIBUTION OF RESIDUALS*****"
10290    CASE 4
10296    PRINT "***** MODEL 4 SOLUTION -- POINTS WEIGHTED ACCORDING TO DEGREE *****"
10302    PRINT RPT$(" ",30)&" OF CONCORDANCE TOWARDS ";
10308    IF Old_apex THEN PRINT "UPPER INTERCEPT ***"
10314    IF Young_apex THEN PRINT "LOWER INTERCEPT ***"
10320    CASE 5

```

```

10326 PRINT USING "/,K";"**MODEL 5 SOLUTION -- POINTS WEIGHTED BY OBSERVED SCATTER FROM BEST-FIT LINE**"
10332 END SELECT
10338 IF Model>1 OR Prob<=.15 THEN ! use students't-t for Mu d.f. if prob>15%
10344 Y_errsl(Model)=Students_t*Errsl_incsc
10350 Y_errint(Model)=Students_t*Errint_incsc
10356 END IF
10362 IF Prtr=1 THEN PRINT USING "/,32X,K,19X,K,/" ; "SLOPE", "INTERCEPT"
10368 IF Prtr=2 THEN PRINT USING "/,33X,K,19X,K,/,33X,K,19X,K,/" ; "SLOPE", "INTERCEPT", "-----", "-----"
10374 PRINT "BEST-FIT=" ; TAB(31); DROUND(Slope,6); TAB(56); DROUND(Inter,6)
10380 IF Printer=2 THEN PRINT RPT$(" ",80)
10386 IF Model=1 THEN
10392 PRINT "ERROR (1-SIGMA, A PRIORI) " ; TAB(31); DROUND(Errsl_apr,4); TAB(56); DROUND(Errint_apr,4)
10398 IF Printer=2 THEN PRINT
10404 END IF
10410 PRINT "ERROR (1-SIGMA, FROM SCATTER) " ; TAB(31); DROUND(Errsl_incsc,4); TAB(56); DROUND(Errint_incsc,4)
10416 IF Printer=2 THEN PRINT
10422 PRINT "ERROR (95% CONFIDENCE LIMITS)"; TAB(31); DROUND(Y_errsl(Model),4); TAB(56); DROUND(Y_errint(Model),4)
10428 IF Model=4 THEN PRINT USING "/,3(K)"; "(number of equivalent "&CHR$(34)&"points"&CHR$(34)&" = ", DROUND(Ncount,3), ")
10434 PRINT
10440 PRINT "COORDINATES OF CENTROID: X=" ; DROUND(Xbar(Model),6), "Y=" ; DROUND(Ybar(Model),6)
10446 IF Model=1 THEN
10452 PRINT USING "/,K"; "ANALYTICAL ERRORS ALONE WILL CAUSE THE OBSERVED AMOUNT OF SCATTER"
10458 PRINT "OR MORE " ; VAL$(DROUND(100*Prob,2)); "% OF THE TIME, AS INDICATED BY THE M.S.U.D. OF"; DROUND(Msrd,3)
10464 END IF
10470 IF Printer=2 THEN PRINT USING "K,/" ; RPT$(" ",80)
10476 NEXT Printer
10482 PRINTER IS CRT
10488 IF (Model=1 AND Prob<=.2) THEN ! may want to try another model
10494 IF Cplot=0 OR Forced_inter THEN
10500 ! Try another model, since Model-1 doesn't seem too probable
10506 Model=2+(Isochron)0 ! use model-2 if non-isochron, 3 if isochron
10512 Sig_y_init=Errint_incsc ! initial est. Y-variability=intercept-error
10518 PRINT RPT$(" ",80)
10524 DISP "MODEL "&VAL$(Model)&" YORKFIT..."
10530 GOTO Iterate
10536 ELSE ! give user a choice of not trying another model
10542 DISP "ACCEPT MODEL-1 FIT OR FIT WITH ANOTHER MODEL?"
10548 OFF KEY
10554 ON KEY 1 LABEL "ACCEPT MODEL-1" GOTO Model_chosen
10560 ON KEY 2 LABEL "DO MODEL-2" GOTO 10698
10566 IF N>5 THEN
10572 ON KEY 3 LABEL "DO MODEL-4 UI" GOTO 10716
10578 ON KEY 4 LABEL "DO MODEL-4 LI" GOTO 10740
10584 END IF
10590 ON KEY 7 LABEL " HELP" GOTO Call_help7
10596 ON KEY 9 LABEL " ESCAPE" GOTO Yorkexit
10602 ON KBD GOTO 10620
10608 GOTO 10608
10614 !
10620 K=FNGet_key
10626 OFF KBD
10632 SELECT K
10638 CASE -67 ! CONTINUE
10644 GOTO Model_chosen
10650 CASE 8 ! CTRL-H
10656 GOTO Call_help7
10662 END SELECT
10668 GOTO 10542
10674 !
10680 Call_help7:CALL Help7

```

```

10686     GOTO 10542
10692     !
10698     Model=2
10704     GOTO Start_model
10710     !
10716     Old_apex=1
10722     Young_apex=0
10728     GOTO 10752
10734     !
10740     Young_apex=1
10746     Old_apex=0
10752     OFF KEY
10758     DISP
10764     Model=4
10770     CALL Model4_calc(Slope,Inter,Cint(*),Interxy(*),Chordlength)
10776     GOTO Start_model
10782     END IF
10788     !
10794     END IF
10800     !
10806     Choose_model:IF Model>1 THEN ! select which solution to use
10812     DISP "PRESS KEY TO CHOOSE MODEL-1 OR MODEL-"&VAL$(Model)&" SOLUTION."
10818     OFF KEY
10824     ON KEY 1 LABEL " MODEL-1" GOTO 10926
10830     IF Model=2 THEN ON KEY 2 LABEL " MODEL-2" GOTO 10938
10836     IF Model=3 THEN ON KEY 3 LABEL " MODEL-3" GOTO 10938
10842     IF Model=4 THEN ON KEY 4 LABEL " MODEL-4" GOTO 10938
10848     ON KEY ? LABEL " HELP" GOTO Call_help5
10854     ON KBD GOTO 10872
10860     GOTO 10860
10866     !
10872     K=FNGet_key
10878     SELECT K
10884     CASE 8           ! CTRL H
10890     GOTO Call_help5
10896     CASE -67       ! CONTINUE
10902     GOTO 10938
10908     END SELECT
10914     GOTO 10824
10920     !
10926     Model=1
10932     !
10938     PRINT RPT$( " ",80)
10944     END IF
10950     !
10956     Model_chosen=Slope=Yorkslope(Model)
10962     ! draw line, solve for isochron age & label if relevant
10968     DISP
10974     BEEP
10980     OFF KEY
10986     OFF KBD
10992     Inter=Yorkinter(Model)
10998     Errsl=Y_errsl(Model)
11004     Errint=Y_errint(Model)
11010     X_bar=Xbar(Model)
11016     IF Crt AND NOT Cplot THEN CALL Graph
11022     Drawline(Slope,Inter,Nlines,Solid_line,Line(*))
11028     IF Crt AND NOT Cplot THEN WAIT 1.5
11034     IF NOT Cplot THEN
11040     SELECT Pbpb

```

```

11046 CASE 0! not a Pb-Pb isochron
11052 IF Isochron THEN
11058 Age=LOG(1+Slope)/Lambda(Isochron)! standard-isochron age
11064 Age_err=Errsl/(Lambda(Isochron)*(1+Slope))
11070 END IF
11076 CASE 1 ! a normal (beta-alpha) Pb isochron
11082 CALL Pb_pb_age(Slope,Age,Errsl,Age_err,Hi_age,Lo_age)
11088 CASE 2 ! a 207/206 vs 204/206 Pb isochron
11094 CALL Pb_pb_age(Inter,Age,Errint,Age_err,Hi_age,Lo_age)
11100 END SELECT
11106 !
11112 IF Isochron OR Pbbp THEN
11118 Print_age=FOR Printer=1 TO Prtr
11124 PRINTER IS Pr(Printer)
11130 IF Age THEN PRINT "MODEL";Model;" ISOCHRON AGE =";ROUND(Age,5);
11136 IF Age_err THEN PRINT " +/-";ROUND(Age_err,3);
11142 IF Age THEN PRINT USING "K, #"; " Ma"
11148 IF Pbbp THEN PRINT USING "4X,5(K), #"; "<",Lo_age," (<,Hi_age,)"
11154 PRINT USING ",K";RPT$( "*",80)
11160 NEXT Printer
11166 PRINT USING "3/"
11172 END IF
11178 ELSE
11184 ! Calculate, printout & label the concordia intercepts
11190 MAT Cint= <0>
11196 GRAPHICS OFF
11202 DISP "Calculating concordia intercepts and errors..."
11208 Conc_inters(Slope,Inter,Cint(*))
11214 Cint_errs(Model,Slope,Inter,Y_errsl(*),Y_errint(*),Xbar(*),Cint(*))
11220 IF Cplot AND Forced_inter THEN
11226 IF Cint(7)<.01 THEN ! Avoid too many significant figures here
11232 Cint(5)=Cint(2)
11238 Cint(6)=Cint(2)
11244 Cint(7)=0
11250 END IF
11256 IF Cint(8)<.01 THEN
11262 Cint(3)=Cint(1)
11268 Cint(4)=Cint(1)
11274 Cint(8)=0
11280 END IF
11286 END IF
11292 DISP
11298 Print_inters(Cint_r(*),Cint(*),Pr(Printer))
11304 Label_inters(Cint(*),N_interlabels,Mswd0)
11310 IF Crt AND N_interlabels<3 THEN ! Cache results for later plotter-dump
11316 Cache_mswd(N_interlabels)=Mswd0
11322 FOR I=1 TO 8
11328 Cache_cint(N_interlabels,I)=Cint(I)
11334 NEXT I
11340 END IF
11346 IF Crt THEN CALL Graph
11352 WAIT 2
11358 END IF
11364 Yorkexit:IF Crt THEN CALL Alpha
11370 SUBEXIT
11376 !
11382 Call_help5:CALL Help5
11388 GOTO Choose_model
11394 !
11400 Underflow: ! numeric under/overflow in chi-square prob. calculation

```

```

11406 OFF ERROR
11412 Prob=0
11418 GOTO Yorkprint
11424 !
11430 Yorkescape:OFF ERROR
11436 OFF KEY
11442 DISP FNN$("ABORTING MODEL-"&VAL$(Model)&" YORKFIT...")
11448 BEEP
11454 WAIT 1
11460 IF Model=1 OR Ok=0 THEN
11466   Clear
11472   SUBEXIT
11478 ELSE
11484   Model=1
11490   GOTO Model_chosen
11496 END IF
11502 !
11508 Fail:OFF ERROR
11514 DISP
11520 PRINTER IS Pr(Prtr)
11526 PRINT USING "/,K,K,2/";"DATA CAN'T BE FIT TO A STRAIGHT LINE USING MODEL ",Model
11532 PRINTER IS CRT
11538 IF Model=1 THEN
11544   Model=2+(Isochron>0)*(Cplot=0)
11550   GOTO Start_model
11556 ELSE ! model>1
11562   IF Ok=0 THEN
11568     Clunk
11574     SUBEXIT
11580   ELSE
11586     Model=1
11592     GOTO Model_chosen
11598   END IF
11604 END IF
11610 !
11616 SUBEND
11622 !
11628 !
11634 Bad_input:SUB Bad_input(Crud$) ! generic error-message & sound routine
11640 Clunk
11646 DISP FNN$(" *** "&Crud$&" *** ")
11652 WAIT 2
11658 SUBEND
11664 !
11670 !
11676 Plotbox:SUB Plotbox(Crud$,Cplot,Rotate_90,Bypass,Plot_started,Escape)
11682 ! draw plotbox, ticks, tick & axis labels
11688 OPTION BASE 1
11694 COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Limx1,Limx2,Limy1,Limy2,Xspred,Yspred,Xloc,Yloc,Xmm,Ymm,Mx$,My$,Mx0$,My0$,Double
11700 COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
11706 COM /Plotbox2/ Mxr$,Myr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Ticksizex,Xl,Yl,Xend,Yend,Speed$,Crt,Hardplot,Cpen,X
gu,Ygu
11712 COM /Plotbox3/ Of_xname$,Of_yname$,Dfile,Stack
11718 COM /Concordia/ Age_min,Age_max,Ctick,Oblique_tick
11724 DIM R$(80),Nu(5),Xtik(9),Ytik(9),Xt_val$(9)[12],Yt_val$(9)[12],Xdec(9),Ydec(9),Zero$[9],Str$(8)[40]
11730 !
11736 Plotter_xmax=249   ! valid for HP-7475 plotter
11742 Plotter_ymax=179.5 ! " " " "
11748 Zero$="000000000"
11754 Escape=0

```

```

11760 Alpha
11766 Clear
11772 OFF KBD
11778 OFF KEY
11784 IF NOT Plotter_dump THEN
11790 !
11796 Enter_axisnames=IF Cplot THEN
11802   Nx_$="^207*Pb/^235*U"
11808   Ny_$="^206*Pb/^238*U"
11814 ELSE
11820   Clear
11826   PRINT USING "K,/,K,2/";"... Use ^ to start superscripts or end subscripts,"," * to start subscripts or end superscripts."
11832   PRINT FNH$("ENTER X-AXIS NAME, Y-AXIS NAME:")
11838   IF Nx_$("<")" AND Ny_$("<")" THEN PRINT USING "/,K";"(press"&FNCT$&"to use "&Nx_$&" and "&Ny_$
11844   PRINT
11850   PRINT USING "/,K,2/,K,7X,K,/,K,#";"Shorthand for common axis-names:","SR -- Rb-Sr isochron","ND -- Sm-Nd isochron","P
BA -- Beta-Alpha plot"
11856   PRINT USING "6X,K,/,K,4X,K,/,K";"PB6 -- Gamma-Alpha plot","UA -- Alpha-Mu isochron","UB -- Beta-Mu isochron"
11862   ON KBD GOTO 11892
11868   ON KEY 9 LABEL " ESCAPE" GOTO Escape
11874   DISP "?"
11880   GOTO 11880
11886 !
11892   K=FMGet_key
11898   OFF KBD
11904   OFF KEY
11910   IF K("<")-67 THEN ! not CONTINUE
11916     OUTPUT KBD;CHR$(K);
11922     PRINT TABXY(1,18);RPT$(" ",80)
11928     DISP FNAst$;
11934     Retrieve_str(R$,Str$(*),Ninputs)
11940     SELECT Ninputs
11946     CASE 0
11952       IF LEN(Nx_$)=0 THEN Nx_$="X"! lazy bastard
11958       IF LEN(Ny_$)=0 THEN Ny_$="Y"
11964     CASE 1
11970       SELECT TRIM$(UPC$(Str$(1)))! shortcut axis-names
11976       CASE "*"
11982         GOTO Escape
11988       CASE "SR"
11994         Ny_$="^87*Sr/^86*Sr"
12000         Nx_$="^87*Rb/^86*Sr"
12006       CASE "ND"
12012         Ny_$="^143*Nd/^144*Nd"
12018         Nx_$="^147*Sm/^144*Nd"
12024       CASE "PB", "PBA"
12030         Nx_$="^206*Pb/^204*Pb"
12036         Ny_$="^207*Pb/^204*Pb"
12042       CASE "PB6"
12048         Nx_$="^206*Pb/^204*Pb"
12054         Ny_$="^208*Pb/^204*Pb"
12060       CASE "U", "UA"
12066         Nx_$="^238*U/^204*Pb"
12072         Ny_$="^206*Pb/^204*Pb"
12078       CASE "UB"
12084         Nx_$="^235*U/^204*Pb"
12090         Ny_$="^207*Pb/^204*Pb"
12096       CASE ELSE
12102         Bad_input("SINGLE RESPONSE: UNRECOGNIZED SHORTHAND FOR AXIS-NAMES")

```

```

12108         GOTO Enter_axisnames
12114         END SELECT
12120     !
12126         CASE 2
12132             Nx_$=Str$(1)
12138             Ny_$=Str$(2)
12144         CASE ELSE
12150             Bad_input("INVALID AXIS-NAME RESPONSE (NEED 2 NAMES ONLY)")
12156             GOTO Enter_axisnames
12162         END SELECT
12168     END IF
12174 END IF
12180     !
12186 . IF NOT Cplot THEN PRINT Nx_$, Ny_$
12192 Nx_$=FNZcon$(0, TRIM$(Nx_$))
12198 Ny_$=FNZcon$(0, TRIM$(Ny_$))
12204 !
12210 Agelimits=IF Cplot THEN
12216     Clear
12222     PRINT USING "2/,K,2/,K"; "ENTER AGE LIMITS AS: EMIN. AGE,] MAX. AGE", "(or press"&FNCT$&"to define by X- and Y-limits)"
12228     ON KBD GOTO 12258
12234     ON KEY 9 LABEL "    ESCAPE" GOTO Escape
12240     DISP "?"
12246     GOTO 12246
12252 !
12258     K=FNGet_key
12264     OFF KBD
12270     OFF KEY
12276     IF K(<)-67 THEN ! Not CONTINUE
12282         OUTPUT KBD;CHR$(K);
12288         PRINT TABXY(1,5);RPT$(" ",80)
12294         DISP FNAst$;
12300         Retrieve(R$,Nu(*),Ninputs)
12306         IF TRIM$(R$)="*" THEN
12312             GOTO Escape
12318         END IF
12324         IF Ninputs>1 THEN PRINT Nu(1);Nu(2)
12330         IF Ninputs=1 THEN PRINT Nu(1)
12336         IF Ninputs>2 THEN
12342             Bad_input(Crud$)
12348             GOTO Agelimits
12354         END IF
12360 !
12366     SELECT Ninputs
12372     CASE 0
12378         GOTO Enter_limits
12384     CASE 1
12390         Age_min=0
12396         Age_max=Nu(1)
12402     CASE 2
12408         Age_min=Nu(1)
12414         Age_max=Nu(2)
12420     END SELECT
12426     !
12432     Agelimited=! plotbox limits defined by concordia ages
12438     CALL Limits(Xmin,Xmax,Ymin,Ymax,Age_min,Age_max,Xtik_inter,Ytik_inter)
12444     GOTO Hardplot_limits
12450     END IF
12456 END IF
12462     !

```



```

12468 Enter_limits=Clear
12474 PRINT USING "2/,K,/";"ENTER X and Y LIMITS (Xmin,] Xmax, [Ymin,] Ymax)?"
12480 IF (Xmax-Xmin) AND (Ymax-Ymin) THEN
12486 PRINT "(press"&FNCT$&"to re-use "&UAL$(Xmin)&", "&UAL$(Xmax)&", "&UAL$(Ymin)&", "&UAL$(Ymax)&")"
12492 END IF
12498 ON KBD GOTO 12528
12504 ON KEY 9 LABEL " ESCAPE" GOTO Escape
12510 DISP "?"
12516 GOTO 12516
12522 !
12528 K=FNGet_key
12534 OFF KBD
12540 OFF KEY
12546 IF K(<)-6? THEN ! Not CONTINUE
12552 OUTPUT KBD;CHR$(K);
12558 PRINT TABXY(1,5);RPT$(" ",80)
12564 DISP FNAst$;
12570 Retrieve(R$,Nu(*),Ninputs)
12576 IF TRIM$(R$)="*" THEN Escape
12582 PRINT R$
12588 IF Ninputs(>0) AND Ninputs(<2) AND Ninputs(<4) THEN
12594 CALL Bad_input(Crud$)
12600 GOTO Enter_limits
12606 END IF
12612 SELECT Ninputs
12618 CASE 2
12624 Xmin=0
12630 Ymin=0
12636 Xmax=Nu(1)
12642 Ymax=Nu(2)
12648 CASE 4
12654 Xmin=Nu(1)
12660 Ymin=Nu(3)
12666 Xmax=Nu(2)
12672 Ymax=Nu(4)
12678 END SELECT
12684 END IF
12690 PRINT Xmin;Xmax,Ymin;Ymax
12696 IF Xmax<Xmin OR Ymax<Ymin THEN
12702 CALL Bad_input("MAX. LIMITS MUST BE MORE THAN MIN. LIMITS")
12708 GOTO Enter_limits
12714 END IF
12720 IF Cplot THEN
12726 Age_min=MAX(FND(Xmin),FNE(Ymin))
12732 Age_max=MIN(FND(Xmax),FNE(Ymax))
12738 . Age_limited=0
12744 END IF
12750 END IF ! plotter-dump endif
12756 !
12762 Hardplot_limits=OFF KBD
12768 OFF KEY
12774 IF NOT Plotter_dump THEN
12780 IF Hardplot THEN
12786 Clear
12792 PRINT USING "8/,K,2/";"ENTER PLOT-SIZE (3-10)?"
12798 PRINT USING "K,/";"(Press"&FNCT$&"for default size of 8)"
12804 PRINT USING "K,/";"(Press "&FNH$("k4")&" for nonstandard dimensions/locations.)"
12810 Hs_kbd=ON KBD GOTO 12846
12816 ON KEY 9 LABEL " ESCAPE" GOTO Escape
12822 ON KEY 4 LABEL " NONSTANDARD" GOTO Nonstd

```

```

12828 DISP "?"
12834 GOTO 12834
12840 !
12846 K=FNGet_key
12852 OFF KBD
12858 OFF KEY
12864 SELECT K
12870 CASE -67 ! CONTINUE
12876 Plotsize=8
12882 CASE NUM("1") TO NUM("9")
12888 OUTPUT KBD;CHR$(K);
12894 PRINT TABXY(1,12);RPT$(" ",3*80)
12900 DISP FNAst$;
12906 LOOP
12912 Retrieve(R$,Nu(*),Ninputs)
12918 IF TRIM$(R$)="*" THEN Escape
12924 EXIT IF Ninputs=1 AND (Nu(1))=3 AND Nu(1)<=10)
12930 Clunk
12936 END LOOP
12942 Plotsize=Nu(1)
12948 CASE ELSE
12954 GOTO Hs_kbd
12960 END SELECT
12966 Std: Oddsize=0
12972 Sizetype=0
12978 GOTO Startbox
12984 !
12990 Nonstd:OFF KEY
12996 OFF KBD
13002 Oddsize=1
13008 Clear
13014 PRINT TABXY(8,9);"PRESS "&FNH$("k0")&" TO DEFINE THE PLOT-LIMITS FOR THE ENTIRE FIGURE-AREA,"
13020 PRINT TABXY(14,11);FNH$("k4")&" TO DEFINE LIMITS FOR THE PLOT-BOX ONLY"
13026 OFF KEY
13032 ON KEY 0 LABEL " ENTIRE AREA" GOTO 13062
13038 ON KEY 4 LABEL "PLOT-BOX ONLY" GOTO 13074
13044 ON KEY 9 LABEL " ESCAPE" GOTO Escape
13050 GOTO 13050
13056 !
13062 Sizetype=1
13068 GOTO 13086
13074 Sizetype=2
13080 !
13086 Hardlimits:Clear
13092 OFF KEY
13098 PRINT TABXY(10,10);"ENTER PLOT-LIMITS IN mm: X-MIN, X-MAX, Y-MIN, Y-MAX?"
13104 PRINT TABXY(14,12);"<max. limits are 0- "&VAL$(Plotter_xmax)&" [X] and 0- "&VAL$(Plotter_ymax)&" [Y],"
13110 PRINT TABXY(18,13);"lower-left corner of paper is 0,0)"
13116 PRINT TABXY(14,16);"Press "&FNCT$&" to re-use " &VAL$(DROUND(Linx1,4))& ", "&VAL$(DROUND(Linx2,4))& ", "&VAL$(DROUND(Liny1,4))& ", "&VAL$(DROUND(Liny2,4))& "
13122 Hl_kbd:ON KBD GOTO 13152
13128 ON KEY 9 LABEL " ESCAPE" GOTO Escape
13134 DISP "?"
13140 GOTO 13140
13146 !
13152 K=FNGet_key
13158 OFF KBD
13164 OFF KEY
13170 SELECT K
13176 CASE -67 ! CONTINUE

```

```

13182     CASE NUM("0") TO NUM("9"),NUM(".". )
13188     OUTPUT KBD;CHR$(K);
13194     PRINT TABXY(1,16);RPT$( " ",80)
13200     DISP FNAst$;
13206     LOOP
13212     INPUT Limx1,Limx2,Limy1,Limy2
13218     EXIT IF Limx1*Limy1=0 AND Limx2<=Plotter_xmax AND Limy2<=Plotter_ymax AND Limx2-Limx1=30 AND Limy2-Limy1=30
13224     CALL Bad_input(Crud$)
13230     END LOOP
13236     CASE ELSE
13242     GOTO H1_kbd
13248     END SELECT
13254     Plotsize=10*((Limx2-Limx1)/Plotter_xmax*(Limy2-Limy1)/Plotter_ymax)/2
13260     ELSE
13266     Oddsize=0
13272     Sizetype=0
13278     Plotsize=Nv(1)
13284     IF Plotsize=10 THEN Plotsize=9.9
13290     END IF      ! hardplot endif
13296     ELSE      ! Plotter-dump
13302     Oddsize=0
13308     Sizetype=0
13314     Plotsize=8
13320     END IF      ! Plotter-dump endif
13326     !
13332     Startbox=Clear
13338     OFF KEY
13344     OFF KBD
13350     Nx$=Nx_$
13356     Ny$=Ny_$
13362     IF Crt THEN
13368     PLOTTER IS 3,"INTERNAL"
13374     GRAPHICS ON
13380     END IF
13386     !
13392     IF NOT Cplot OR NOT Agelimited THEN ! draw box with ticks
13398     CALL Tick(Xmax-Xmin,Xtik_inter)
13404     CALL Tick(Ymax-Ymin,Ytik_inter)
13410     END IF
13416     !
13422     ! Force lower-bound X & Y values to have a minimum # of sign. figures.
13428     I=0
13434     REPEAT
13440     I=I+1
13446     X=FNDc(ABS(Xmin),I)
13452     UNTIL I>7 OR ABS(Xmin)-X<Xtik_inter
13458     IF Xmin=0 OR ABS(Xmin)=X THEN
13464     Xmin=SGN(Xmin)*X
13470     ELSE
13476     Xmin=-X-Xtik_inter
13482     END IF
13488     !
13494     I=0
13500     REPEAT
13506     I=I+1
13512     Y=FNDc(ABS(Ymin),I)
13518     UNTIL I>7 OR ABS(Ymin)-Y<Ytik_inter
13524     IF Ymin=0 OR ABS(Ymin)=Y THEN
13530     Ymin=SGN(Ymin)*Y
13536     ELSE

```

```

13542 Ymin=Y-Ytik_inter
13548 END IF
13554 !
13560 X=Xmin
13566 REPEAT ! force Xmax to lie on a tick
13572 X=X+Xtik_inter
13578 UNTIL DROUND(X,9)=DROUND(Xmax,9)
13584 Xmax=X
13590 !
13596 Y=Ymin
13602 REPEAT ! force Ymax to lie on a tick
13608 Y=Y+Ytik_inter
13614 UNTIL DROUND(Y,9)=DROUND(Ymax,9)
13620 Ymax=Y
13626 !
13632 Xspred=Xmax-Xmin
13638 Yspred=Ymax-Ymin
13644 Xdec_max=0
13650 Ydec_max=0
13656 Ychar=0
13662 Xchar=0
13668 Lzero=Hardplot ! include leading zeroes for tick-labels if not a CRT-plot
13674 !
13680 ! Find max. # characters in X & Y ticks & ticks w. max # decimal places
13686 Nx_majortix=DROUND(Xspred/Xtik_inter/2+1,9)
13692 Ny_majortix=DROUND(Yspred/Ytik_inter/2+1,9)
13698 !
13704 FOR I=1 TO Nx_majortix
13710 Xtik(I)=Xmin+2*Xtik_inter*(I-1)
13716 IF ABS(Xtik(I))<1.E-12 THEN Xtik(I)=0 ! guard against roundoff
13722 A=FNChar(Xtik(I),Lzero) ! max # of characters
13728 IF A>Xchar THEN Xchar=A
13734 Xdec(I)=FNDecimal(Xtik(I))
13740 IF Xdec(I)>Xdec_max THEN Xdec_max=Xdec(I)
13746 NEXT I
13752 !
13758 FOR I=1 TO Ny_majortix
13764 Ytik(I)=Ymin+2*Ytik_inter*(I-1)
13770 IF ABS(Ytik(I))<1.E-12 THEN Ytik(I)=0 ! guard against roundoff
13776 A=FNChar(Ytik(I),Lzero) ! max # of characters
13782 IF A>Ychar THEN Ychar=A
13788 Ydec(I)=FNDecimal(Ytik(I))
13794 IF Ydec(I)>Ydec_max THEN Ydec_max=Ydec(I)
13800 NEXT I
13806 !
13812 IF Crt THEN
13818 GCLEAR
13824 Limx1=0
13830 Limy1=10
13836 Limx2=100*MAX(1,RATIO)
13842 Limy2=100*MAX(1,1/RATIO)
13848 Tht=4
13854 Lht=5
13860 ELSE
13866 IF NOT Oddsize THEN
13872 Limx1=Plotter_xmax*(1-.1*Plotsize)/2 ! For centered plots only
13878 Limx2=Plotter_xmax-Limx1
13884 Limy1=Plotter_ymax*(1-.1*Plotsize)/2
13890 Limy2=Plotter_ymax-Limy1
13896 END IF

```

```

13902 END IF
13908 Ar=.6 ! Aspect ratio, tick label char. ht.
13914 IF Hardplot THEN
13920 Tht=.45*Plotsize ! character height of tick-labels, in mm
13926 M=.8*Ticksize*(1+2.7*(Plotsize<6)/Plotsize)! For plotsizes<6, somewhat buffer label-character size.
13932 Tht=M*Tht! Tick-label and axis-label character heights
13938 IF Tht<2.0 THEN Tht=2.0
13944 Lht=1.25*Tht
13950 END IF
13956 !
13962 Pu=POS(Nx$, "")
13968 Pd=POS(Nx$, "*")
13974 Qu=POS(Ny$, "")
13980 Qd=POS(Ny$, "*")
13986 P=Pu OR Pd
13992 Q=Qu OR Qd
13998 N_xtix=2*Nx_majortix
14004 N_ytix=2*Ny_majortix
14010 Ss=(Qu AND Qd AND Qu<Qd) ! =1 if Y-axis label has superscripts
14016 Xdsc=(POS(Nx$, "g") OR POS(Nx$, "q") OR POS(Nx$, "p") OR POS(Nx$, "y"))
14022 ! Xdsc=1 if the X-axis label contains a character with descenders
14028 Ydsc=(POS(Ny$, "g") OR POS(Ny$, "q") OR POS(Ny$, "p") OR POS(Ny$, "y"))
14034 ! ditto Ydsc for Y-axis label
14040 !
14046 Define_box= !
14052 Yend=Tht/2*FNEven(N_ytix) ! 6U's, mm above top of box
14058 IF Crt THEN
14064 Xend=(Xchar*Ar*Tht/2)*FNEven(N_xtix) ! 6U's, mm to right of X-axis
14070 Xl=Ar*Tht*Ychar+Lht*(17/15+4/15*Xdsc+5/15*Q)! 6U's,mm to left of Y-axis
14076 Yl=Tht*15/15+Lht*(1-Crt/15+4/15*Xdsc+4/15*P)! 6U's,mm below X-axis
14082 ELSE ! plotter limits
14088! Xend=FNWidth(Tht,Ymm,D,Ar,0)*Xchar/2*FNEven(N_xtix) ! 6U's to R. of X-axis
14094 Xend=Xchar*Ar*Tht/2*FNEven(N_xtix) ! 6U's to R. of X-axis
14100 Xl=Ar*1.183/.67*Tht*Ychar+Lht*(1+.8+4/15*Xdsc+.5*Q) ! 6U's to L of Y-axis
14106 Yl=Tht*16/15+Lht*(1+.8+4/15*Xdsc+.5*P) ! 6U's below X-axis
14112 END IF
14118 !
14124 IF Hardplot AND Sizetype<2 THEN
14130 Limx1=Limx1+Xl
14136 Limx2=Limx2-Xend
14142 Limy1=Limy1+Yl
14148 Limy2=Limy2-Yend
14154 END IF
14160 !
14166 Rat=RATIO*Crt*(Limx2-Limx1)/(Limy2-Limy1)*Hardplot
14172 Xgu=100*MAX(1,Rat) ! total 6U's of plot (X)
14178 Ygu=100*MAX(1,1/Rat) ! " " " " (Y)
14184! Xgu_mm=(Limx2-Limx1)/Xgu ! mm per 6U (X),CRT
14190! Ygu_mm=(Limy2-Limy1)/Ygu ! mm per 6U (Y),CRT
14196 !
14202 Xloc=(Xgu-Xend-Xl)/Xspred ! 6U'S per UU (X-direction)
14208 Yloc=(Ygu-Yend-Yl)/Yspred ! 6U'S per UU (Y-direction)
14214 IF Crt THEN
14220 B=1
14226 ELSE
14232 B=Plotter_ymax/100 ! mm to 6U conversion for 7475 (8.5X11" paper)
14238 END IF
14244 Lx1=Limx1/D
14250 Lx2=Limx2/D
14256 Ly1=Limy1/D

```

```

14262 Ly2=Limy2/D
14268 IF Crt THEN
14274 Xmm=((Limx2-Limx1-(X1+Xend))*160/133.44)/Xspred ! mm/UU (X)
14280 Ymm=((Limy2-Limy1-(Y1+Yend))*120/100)/Yspred ! mm/UU (Y)
14286 ELSE
14292 Xmm=(Limx2-Limx1)/Xspred
14298 Ymm=(Limy2-Limy1)/Yspred
14304 END IF
14310 !
14316 FOR Dx=0 TO Double
14322 VIEWPORT Lx1+Dx*Offset,Lx2+Dx*Offset,Ly1+Dx*Offset,Ly2+Dx*Offset
14328 IF Hardplot THEN
14334 WINDOW Xmin,Xmax,Ymin,Ymax
14340 OUTPUT Plotter;Speed$
14346 IF Rotate_90 THEN OUTPUT Plotter;"R090;"
14352 IF NOT Rotate_90 THEN OUTPUT Plotter;"R0;"
14358 ELSE
14364 WINDOW Xmin-X1/Xloc,Xmax+Xend/Xloc,Ymin-Y1/Yloc,Ymax+Yend/Yloc
14370 END IF
14376 CSIZE Tht,Ar
14382 ! Label every 4th tick, not every other tick if little room available
14388 Xtmm=2*Xtik_inter*Xmm ! mm between every other tick on X-axis
14394 IF Crt THEN Xtmm=Tht*Ar*Xchar*Xmm/Xloc ! mm taken up by largest X-tick label
14400 IF Hardplot THEN Xtmm=Tht*Ar*1.183/.67 ! mm taken up by largest X-tick label
14406 IF Xtmm/Xtmm<.5 THEN ! Avoid crowding X-tick labels
14412 Mtik=2
14418 ELSE
14424 Mtik=4
14430 END IF
14436 !
14442 IF NOT Bypass THEN
14448 TickLabel(Xt_val$(*),Xtik(*),Xdec(*),Xdec_max,Lzero,Nx_majortix)
14454 TickLabel(Yt_val$(*),Ytik(*),Ydec(*),Ydec_max,Lzero,Ny_majortix)
14460 END IF
14466 !
14472 Xticksize=2.5/Yloc
14478 Yticksize=Xticksize*Ymm/Xmm
14484 IF Crt THEN Cpen=1
14490 IF NOT Bypass THEN
14496 PEN Cpen
14502 IF DROUND(Xmin,9)=0 THEN Xmin=0 ! roundoff problems?
14508 IF DROUND(Ymin,9)=0 THEN Ymin=0 ! roundoff problems?
14514 MOVE Xmin,Ymin
14520 !
14526 ! draw axes and ticks
14532 Axis(1,Xspred,Xtik_inter,Xticksize,1)
14538 Axis(1,Yspred,Ytik_inter,Yticksize,0)
14544 Axis(-1,Xspred,Xtik_inter,Xticksize,1)
14550 Axis(-1,Yspred,Ytik_inter,Yticksize,0)
14556 CLIP OFF
14562 !
14568 IF NOT Stack THEN
14574 LORG 6
14580 FOR I=1 TO Nx_majortix STEP Mtik/2! Label X-ticks
14586 MOVE Xtik(I),Ymin
14592 LABEL Xt_val$(I)
14598 NEXT I
14604 END IF
14610 !
14616 LORG 8

```

```

14622     IF CrT THEN B=Tht*Ar/2/Xloc
14628     IF Hardplot THEN B=FNCwidth(Tht,Xmm,0,Ar,0)/2
14634     FOR I=i TO Ny_majortix
14640 ! Slightly offset 1st y-tick to avoid overlap w. 1st x-tick
14646     IF NOT Stack OR I>1 THEN
14652         IF CrT THEN MOVE Xmin-B,Ytik(I)+(I=1)*Tht/3/Yloc
14658         IF Hardplot THEN MOVE Xmin-B,Ytik(I)+(I=1)*Tht*.36/Ymm
14664         LABEL Yt_val$(I)
14670     END IF
14676     NEXT I
14682     END IF
14688 !
14694     IF Dx=0 THEN
14700         Stripscript(Ny$)! convert super/subscript tokens
14706         Stripscript(Nx$)
14712         Remove(Nx$,Hxr$,Lx)! remove super/subscript tokens
14718         Remove(Ny$,Hyr$,Ly)
14724     END IF
14730 !
14736     CSIZE Lht,Ar
14742     CLIP OFF
14748     LORG 1
14754     IF NOT Bypass THEN
14760         IF NOT Stack THEN
14766             IF CrT THEN
14772                 MOVE Xmin+Xspred/2,Ymin-(Y1+Lht*4/15)/Yloc
14778                 IMOVE -LEN(Hxr$)/2*Lht*Ar/Xloc,0
14784             ELSE
14790                 MOVE Xmin+Xspred/2,Ymin-(Y1+Lht*9/15)/Ymm
14796                 IMOVE -LEN(Hxr$)/2*FNCwidth(Lht,Xmm,0,Ar,0),0
14802             END IF
14808             FOR I=i TO LEN(Nx$)
14814                 S=(Nx$[I,I]=CHR$(30))-(Nx$[I,I]=CHR$(10))
14820                 IF S THEN
14826                     IF CrT THEN IMOVE 0,S*5/15*Lht/Yloc
14832                     IF Hardplot THEN IMOVE 0,S*.5*Lht/Ymm
14838                 ELSE
14844                     LABEL Nx$[I,I];
14850                 END IF
14856             NEXT I
14862             END IF
14868 !
14874             LDIR 90
14880             IF CrT THEN
14886                 MOVE Xmin-Xl/Xloc,Ymin+Yspred/2-Ar/2*Lht*Ly/Yloc
14892                 IMOVE Lht*(14/15+Ss*5/15)/Xloc,0
14898             ELSE
14904                 MOVE Xmin-Xl/Xmm,Ymin+Yspred/2-FNCwidth(Lht,Ymm,0,Ar,0)/2*Ly
14910                 IMOVE Lht*(21/15+Ss*5/15)/Xmm,0
14916             END IF
14922             FOR I=1 TO LEN(Ny$)
14928                 S=(Ny$[I,I]=CHR$(30))-(Ny$[I,I]=CHR$(10))
14934                 IF S THEN
14940                     IF CrT THEN IMOVE -5/15*S*Lht/Xloc,0
14946                     IF Hardplot THEN IMOVE -.5*S*Lht/Xmm,0
14952                 ELSE
14958                     LABEL Ny$[I,I];
14964                 END IF
14970             NEXT I
14976             END IF

```

```

14982 !
14988   LDIR 0
14994   NEXT Dx
15000   Nx$=Nxr$ ! Stripped of ^ and * characters
15006   Ny$=Nyr$
15012   Nx0$=Nx$[1,MIN(10,LEN(Nx$))]
15018   Ny0$=Ny$[1,MIN(10,LEN(Ny$))]
15024   IF Hardplot AND NOT Cplot THEN PEN 0
15030   CLIP Xmin,Xmax,Ymin,Ymax
15036   Plot_started=1
15042   SUBEXIT
15048   !
15054   Escape=Escape+1
15060   SUBEND
15066   !
15072   !
15078   Drawline:SUB Drawline(Slope,Inter,Nlines,Solid_line,Line(*) ) ! plot the best-fit line
15084   OPTION BASE 1
15090   COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Linx1,Linx2,Liny1,Liny2,Xspread,Yspread,Xloc,Yloc,Xmm,Ymm,Nx$,Ny$,Nx0$,Ny0$,Double
15096   COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
15102   COM /Plotbox2/ Nxr$,Nyr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Ticksizex1,V1,Xend,Yend,Speed$,Crt,Hardplot,Cpen,X
gu,Ygu
15108   Nlines=Nlines+(Nlines/10)
15114   IF Crt THEN
15120     Line(Nlines,1)=Slope
15126     Line(Nlines,2)=Inter
15132   END IF
15138   LINE TYPE 1+(Solid_line=0)*(4+Nlines)
15144   CLIP Xmin,Xmin+.97*Xspread,Ymin,Ymin+.97*Yspread
15150   PEN Cpen
15156   FOR Dx=0 TO Double
15162     MOVE Xmin,Xmin*Slope+Inter-Dx*Offset/Ymm
15168     DRAW Xmax,Xmax*Slope+Inter-Dx*Offset/Ymm
15174   NEXT Dx
15180   MOVE Xmax,Ymax
15186   LINE TYPE 1
15192   PEN Crt
15198   CLIP Xmin,Xmax,Ymin,Ymax
15204   SUBEND
15210   !
15216   !
15222   Letter:SUB Letter(Letter$(*),Letter(*),Nphrases,Phrase)
15228   OPTION BASE 1
15234   COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Linx1,Linx2,Liny1,Liny2,Xspread,Yspread,Xloc,Yloc,Xmm,Ymm,Nx$,Ny$,Nx0$,Ny0$,Double
15240   COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
15246   COM /Plotbox2/ Nxr$,Nyr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Ticksizex1,V1,Xend,Yend,Speed$,Crt,Hardplot,Cpen,X
gu,Ygu
15252   DIM R$(32),Nu(4),P$(80),Pm(7,3)
15258   DATA 1,10,-2,7,10,-1,4,13,-2,4,7,-1,1,4,-2,7,4,-1,9,1,-2
15264   READ Pm(*) ! plus-or-minus symbol
15270   !
15276   First_letter=1
15282   Clear
15288   GRAPHICS OFF
15294   CLIP OFF
15300   OFF KEY
15306   OFF KBD
15312   !
15318   IF NOT Plotter_dump THEN
15324   Phrase_specs=PRINT USING ",/k,/";"(PRESS"&FNCT$&"TO ESCAPE)"

```



```

15330 PRINT "CHAR.-HEIGHT? (axis-tick labels were";DROUND(Tht,2);">) [ , ROTATION]]"
15336 Retrieve(R$,Nu(*),Ninputs)
15342 Clear
15348 IF Ninputs THEN
15354   H=Nv(1)
15360   Rot=Nv(2)
15366   PRINT H,Rot
15372   IF H<.1 THEN
15378     Bad_input("CHARACTER-HEIGHT MUST BE >0.1")
15384     GOTO Phrase_specs
15390   END IF
15396 ELSE
15402   CLIP Xmin,Xmax,Ymin,Ymax
15408   SUBEXIT
15414   END IF
15420 ELSE
15426   P$=Letter$(Phrase)
15432   X=Letter(Phrase,1)
15438   Y=Letter(Phrase,2)
15444   H=Letter(Phrase,3)
15450   Rot=Letter(Phrase,4)
15456 END IF
15462 !
15468 PEN Cpen
15474 Cs=H
15480 U=Ar*Cs*(1-Hardplot/3.1)
15486 CSIZE Cs,Ar
15492 LDIR Rot
15498 Clear
15504 !
15510 IF Hardplot AND NOT Plotter_dump THEN
15516 PRINT USING "2/,K,2/";"POSITION PEN WITH "&FNH$("KNOB")&" (X-direction) OR "&FNH$("SHIFT-KNOB")&" (Y-direction)..."
15522 PRINT "WHEN PEN IS POSITIONED, PRESS "&FNH$("k4")&", THEN TYPE IN PHRASE."
15528 END IF
15534 !
15540 ON KEY 4 LABEL "   READY" GOTO 15900
15546 ON KEY 0 LABEL "   ENTER X-Y" GOTO 15714
15552 ON KEY 9 LABEL "   ESCAPE" GOTO L_end
15558 !
15564 IF Crt THEN CALL Graph
15570 IF NOT Plotter_dump THEN
15576   Drx=3+(Xspred/Ymax(.1)+(Xspred/Xmax(.01) ! for adequate # sig. figs
15582   Dry=3+(Yspred/Ymax(.1)+(Yspred/Ymax(.01) ! "
15588   A=KNOBK
15594   IF Hardplot THEN WHERE X,Y
15600   IF X(<=Xmin OR Y(<=Ymin OR X(>=Xmax OR Y(>=Ymax THEN
15606     X=Xmax-Xspred/2*Crt
15612     Y=Ymax-Yspred/2*Crt
15618   END IF
15624   MOVE X,Y
15630 END IF
15636 !
15642 IF Crt THEN
15648   SET ECHO X,Y
15654   PRINT TABXY(13,15);"MOVE CURSOR WITH "&FNH$(" KNOB ")&"(X-dir.) or "&FNH$(" SHIFT-KNOB ")&" (Y-dir.),"
15660   PRINT TABXY(13,16);"THEN PRESS "&FNH$("k4")&" AND ENTER PHRASE"
15666   ALPHA ON
15672   GRAPHICS ON
15678   END IF
15684 QISP TAB(30);"X: ";DROUND(X,Drx);TAB(46);"Y: ";DROUND(Y,Dry)

```

```

15690 IF Plotter_dump THEN 15936
15696 ON KNOB .1 GOSUB 15798
15702 GOTO 15702
15708 !
15714 ! X-Y locating
15720 LOOP
15726 DISP "ENTER COORDINATES (X, Y) OF LOWER-LEFT CORNER OF PHRASE";
15732 Retrieve(R$,Nu(*),Ninputs)
15738 EXIT IF Ninputs>1
15744 Clunk
15750 GOTO 15900
15756 END LOOP
15762 X=Nu(1)
15768 Y=Nu(2)
15774 PRINT X,Y
15780 PEN Cpen
15786 GOTO 15900
15792 !
15798 STATUS 2,10:Dir
15804 Tot=Yspred*Dir+Xspred*(NOT Dir)
15810 Delta=KNOBK/600*Tot/(1+Hardplot)
15816 IF Dir=0 THEN X=X+Delta
15822 IF Dir THEN Y=Y+Delta
15828 IF Crt THEN
15834 IF X>Xmax THEN X=Xmin
15840 IF Y>Ymax THEN Y=Ymin
15846 IF X<Xmin THEN X=Xmax
15852 IF Y<Ymin THEN Y=Ymax
15858 SET ECHO X,Y
15864 ELSE
15870 MOVE X,Y
15876 END IF
15882 DISP TAB(3D);"X: ";DROUND(X,Drx);TAB(46);"Y: ";DROUND(Y,Drx)
15888 RETURN
15894 !
15900 OFF KBD
15906 OFF KNOB
15912 Clear
15918 MOVE X,Y
15924 IF Crt THEN SET ECHO 2*Xmax,2*Ymax
15930 !
15936 LORG 1
15942 IF Crt THEN CALL Alpha
15948 IF NOT Plotter_dump THEN
15954 LINPUT "PHRASE TO BE DRAFTED?",P$
15960 END IF
15966 Clear
15972 PRINT P$
15978 Stripscrip(P$)
15984 P$=FNZcon$(0,P$)
15990 L=LEN(P$)
15996 !
16002 Charwidth=FNWidth(Cs,Xm,Xloc,Ar,Crt)
16008 Charheight=FNHeight(Cs,Yloc,Ym,Crt)
16014 !
16020 FOR Dx=0 TO Double
16026 IF Dx THEN
16032 MOVE X+Offset/Xm,Y+Offset/Ym
16038 ELSE
16044 IF Crt THEN

```

```

16050      MOVE 0,-Charheight/4! adjust for fact that LOG 1 starts
16056      Graph          ! character CELL location, not CHAR.
16062      Nphrases=Nphrases+(Nphrases/25)
16068      Letter$(Nphrases)=P$
16074      Letter(Nphrases,1)=X
16080      Letter(Nphrases,2)=Y
16086      Letter(Nphrases,3)=H
16092      Letter(Nphrases,4)=Rot
16098      IF Cpen=0 THEN Nphrases=Nphrases-2
16104      END IF
16110      END IF
16116      !
16122      IF Plotter_dump THEN
16128          PEN Cpen
16134          MOVE X,Y
16140          END IF
16146      IF Hardplot THEN MOVE -2*Charwidth/15,-6*Charheight/15! to compensate for char.-cell discrepancy
16152      IF POS(P$,CHR$(30))=0 AND POS(P$,CHR$(10))=0 AND POS(P$,"+/-")=0 THEN
16158          LABEL P$
16164      ELSE
16170          FOR I=1 TO L
16176              IF I<=L-2 THEN
16182                  IF P$(I,I+2)="+/-" THEN
16188                      SYMBOL P$(*)
16194                      I=I+2
16200                      GOTO 16254
16206                  END IF
16212              END IF
16218              A=(P$(I,I)=CHR$(30))-P$(I,I)=CHR$(10))* .6
16224              IF A THEN
16230                  MOVE -A*Charwidth*SIN(Rot),-A*Charheight*SIN(Rot-90)
16236              ELSE
16242                  LABEL P$(I,I);
16248              END IF
16254          NEXT I
16260      END IF
16266      NEXT Dx
16272      WHERE X,Y
16278      PENUP
16284      PEN Crt
16290      LDIR 0
16296      IF Crt THEN
16302          OFF KEY
16308          WAIT 2
16314          Alpha
16320      END IF
16326      IF NOT Plotter_dump THEN Phrase_specs
16332      L_end=SUBEND
16338      !
16344      !
16350      Polynomial=SUB Polynomial(Xu(*),Yv(*),N,Maxnum,Xmin,Xmax,Xspred,Cpen,Crt,Pr(*),Prtr)
16356      OPTION BASE 1
16362      ! Up to 5th order polynomial regression on up to 200 points.
16368      ! Unweighted for analytical errors.
16374      ! A-Array is the X Power-Matrix,
16380      ! Coef-Array Contains the P+1 Coefficients of the Pth-Order Polynomial.
16386      !
16392      DIM Coef(5,1),Suffix$(5)[2]
16398      DATA st,nd,rd,th,th
16404      READ Suffix$(*)

```

```

16410 Clear
16416 OFF KEY
16422 OFF KBD
16428 IF M>200 THEN
16434   Bad_input("**** CAN'T FIT MORE THAN 200 POINTS ****")
16440   SUBEXIT
16446 END IF
16452 Max_order=M-1
16458 IF Max_order>4 THEN Max_order=4
16464 PRINT USING "0/,K,2/";"PRESS A NUMBER-KEY (1-&VAL$(Max_order)&) TO INDICATING THE ORDER OF THE POLYNOMIAL TO BE FIT..."
16470 PRINT "(Press"&FNCT$&"to escape)"
16476 ON KBD GOTO 16488
16482 GOTO 16482
16488 K=FNGet_key
16494 P=K-48
16500 BEEP
16506 OFF KBD
16512 Clear
16518 IF K=-67 THEN SUBEXIT ! CONTINUE key pressed
16524 IF P<0 OR P>9 THEN 16410 ! illegal key
16530 IF P<1 OR P>Max_order THEN
16536   Bad_input("POLYNOMIAL ORDER MUST BE BETWEEN 1 AND &VAL$(Max_order)")
16542   GOTO 16410
16548 END IF
16554 !
16560 Polycalc(Xv(*),Yv(*),Coef(*),P,M)
16566 !
16572 FOR J=1 TO Prtr
16578   PRINTER IS Pr(J)
16584   PRINT TABXY(1,3);"BEST-FIT "&VAL$(P)&Suffix$(P)&"-ORDER POLYNOMIAL IS:"
16590   PRINT USING "2/,K,0#";"Y = "&VAL$(DROUND(Coef(1,1),6))
16596   FOR I=1 TO P
16602     IF Coef(I+1,1)<0 THEN PRINT " -";
16608     IF Coef(I+1,1)>=0 THEN PRINT " +";
16614     PRINT VAL$(DROUND(ABS(Coef(I+1,1)),6))&"*";
16620     IF I=1 THEN PRINT "X";
16626     IF I>1 THEN PRINT "X"&VAL$(I);
16632   NEXT I
16638   PRINT
16644 NEXT J
16650 OISP
16656 PRINTER IS CRT
16662 PEN Cpen
16668 IF Crt THEN CALL Graph
16674 Xstart=Xmin*(Xmin<>0)+Xspred/100*(Xmin=0)
16680 FOR I=Xstart TO Xmax STEP (Xspred)/100
16686   Y=0
16692   IF I=0 THEN GOTO 16752
16698   FOR J=1 TO P+1
16704     Y=Y+Coef(J,1)*I^(J-1)
16710   NEXT J
16716   IF I<>Xstart THEN
16722     ON ERROR GOTO 16752
16728     DRAW I,Y
16734   ELSE
16740     MOVE I,Y
16746   END IF
16752 NEXT I
16758 PEN 0
16764 Alpha

```

```

16770 DISP "PRESS "&FNCT&&" TO CONTINUE"
16776 PAUSE
16782 SUBEXIT
16788 !
16794 RETURN
16800 SUBEND
16806 !
16812 !
16818 Expo_fit=SUB Expo_fit(N,Xv(*),Yv(*),Xmin,Xmax,Ymin,Ymax,Crt,Cpen,Pr(*),Prtr)
16824 OPTION BASE 1
16830 OFF KEY
16836 Clear
16842 Z=N
16848 AO=1
16854 ON ERROR GOSUB Next_maxpt
16860 BO=LOG(Yv(Z)/AO)/Xv(Z)
16866 OFF ERROR
16872 LOOP
16878 DATA 0,0,0,0,0,0
16884 RESTORE 16878
16890 READ FO,A1,B1,B2,C1,C2
16896 FOR I=1 TO N
16902   FO=Yv(I)-AO*EXP(BO*Xv(I))
16908   A1=A1+Yv(I)^2
16914   B2=B2+(Xv(I)*Yv(I))^2
16920   B1=B1+Xv(I)*Yv(I)^2
16926   C1=C1-Yv(I)*FO
16932   C2=C2-Xv(I)*Yv(I)*FO
16938 NEXT I
16944 Count=1+Count
16950 IF Count>50 THEN
16956   Cant_fit
16962   SUBEXIT
16968 END IF
16974 ON ERROR GOTO 16956
16980 A1=A1/AO^2
16986 B1=B1/AO
16992 C1=C1/AO
16998 A2=B1
17004 !
17010 A=(C1-B1*C2/B2)/(A1-A2*B1/B2)
17016 B=(C2-A2*A)/B2
17022 A=AO-A
17028 B=BO-B
17034 Aa=DROUND(A,5)
17040 Bb=DROUND(B,5)
17046 DISP Aa;TAB(20);Bb
17052 EXIT IF Aa=DROUND(AO,5) OR Bb=DROUND(BO,5)
17058 AO=A
17064 BO=B
17070 END LOOP
17076 !
17082 OFF ERROR
17088 FOR I=1 TO Prtr
17094   PRINTER IS Pr(I)
17100   PRINT TABX(1,3);"BEST-FIT EXPONENTIAL CURVE IS  Y = "&VAL$(Aa)&"*e^("&VAL$(Bb)&"*X)"
17106 NEXT I
17112 PRINTER IS CRT
17118 PEN Cpen
17124 IF Crt THEN GRAPHICS ON

```

```

17130 FOR X=Xmin TO Xmax-(Xmax-Xmin)/70 STEP (Xmax-Xmin)/70
17136   Y=A*EXP(B*X)
17142   IF X>Xmin AND Y>Ymin AND Y<Ymax THEN
17148     DRAW X,Y
17154   ELSE
17160     MOVE X,Y
17166   END IF
17172 NEXT X
17178 PEN 0
17184 GRAPHICS OFF
17190 DISP "PRESS "&FNCT$&" TO CONTINUE"
17196 PAUSE
17202 SUBEXIT
17208 !
17214 Next_maxpt=Z=Z-1
17220 IF Z<3 THEN
17226   Cant_fit
17232   SUBEXIT
17238 END IF
17244 RETURN
17250 SUBEND
17256 !
17262 !
17268 Logar_fit=SUB Logar_fit(Xv(*),Yv(*),N,Xmin,Xmax,Ymin,Ymax,Crt,Cpen,Pr(*),Prtr)
17274 OPTION BASE 1
17280 OFF KEY
17286 Clear
17292 !
17298 Z=N
17304 AO=1
17310 ON ERROR GOSUB Next_maxpt
17316 BO=EXP(Yv(Z)/Xv(Z))
17322 ON ERROR GOTO Bad_logfit
17328 LOOP
17334 DATA 0,0,0,0,0
17340 RESTORE 17334
17346 READ FO,A1,B1,C1,C2
17352 FOR I=1 TO N
17358   IF Xv(I)<=0 THEN
17364     Cant_fit
17370     SUBEXIT
17376   END IF
17382   FO=Yv(I)-AO*LOG(BO*Xv(I))
17388   A1=A1+Yv(I)^2
17394   B1=B1+Yv(I)
17400   C1=C1-Yv(I)*FO
17406   C2=C2-FO
17412 NEXT I
17418 Count=1+Count
17424 IF Count>50 THEN
17430   Cant_fit
17436   SUBEXIT
17442 END IF
17448 ON ERROR GOTO 17430
17454 A1=A1/AO^2
17460 B2=N*(AO/BO)^2
17466 A2=B1/BO
17472 B1=B1/BO
17478 C1=C1/AO
17484 C2=C2*AO/BO

```

```

17490 !
17496 A=(C1-B1*C2/B2)/(A1-A2*B1/B2)
17502 B=(C2-A2*A)/B2
17508 A=A0-A
17514 B=B0-B
17520 IF B<=0 THEN B=B0/2 ! guard against negative b-estimate
17526 Aa=DROUND(A,6)
17532 Bb=DROUND(B,6)
17538 DISP Aa;TAB(20);Bb
17544 EXIT IF Aa=DROUND(A0,6) OR Bb=DROUND(B0,6)
17550 A0=A
17556 B0=B
17562 END LOOP
17568 !
17574 OFF ERROR
17580 DISP
17586 FOR I=1 TO Prtr
17592 PRINTER IS Pr(I)
17598 PRINT TABXY(1,3);"BEST FIT LOGARITHMIC CURVE IS Y = "&URL$(Aa)&"*LN("&URL$(Bb)&"*X)"
17604 NEXT I
17610 PRINTER IS CRT
17616 PEN Cpen
17622 IF Crt THEN GRAPHICS ON
17628 ON ERROR GOTO 17688
17634 FOR X=Xmin TO Xmax-(Xmax-Xmin)/70 STEP (Xmax-Xmin)/70
17640 IF X>0 THEN
17646 Y=A*LOG(B*X)
17652 IF X>Xmin AND Y>Ymin AND Y<Ymax THEN
17658 DRAW X,Y
17664 ELSE
17670 MOVE X,Y
17676 END IF
17682 END IF
17688 NEXT X
17694 PEN 0
17700 GRAPHICS OFF
17706 DISP "PRESS "&FNct&" TO CONTINUE"
17712 PAUSE
17718 SUBEXIT
17724 !
17730 Next_maxpt=Z-1
17736 IF Z<3 THEN
17742 Cant_fit
17748 SUBEXIT
17754 END IF
17760 RETURN
17766 !
17772 Bad_logfit=OFF ERROR
17778 Cant_fit
17784 SUBEND
17790 !
17796 !
17802 Cant_fit:SUB Cant_fit
17808 Clunk
17814 PRINT USING "3/,K,/";"*** SORRY- CAN'T FIT REQUESTED CURVE TO DATA ***"
17820 WAIT 2
17826 SUBEND
17832 !
17838 !
17844 Get_key=DEF FNGet_key

```

```

17850 DIM Key$(3)
17856 ON ERROR GOTO Error
17862 Key$=KBD$
17868 L=LEN(Key$)
17874 SELECT L
17880 CASE 1
17886   Keycode=NUM(Key$)
17892 CASE 2
17898   Keycode=-NUM(Key$[2,2])
17904 CASE 3
17910   Keycode=-NUM(Key$[3,3])-128
17916 END SELECT
17922 IF Keycode=-77 THEN CALL Alpha
17928 IF Keycode=-79 THEN DUMP ALPHA
17934 IF Keycode=-76 THEN CALL Graph
17940 RETURN Keycode
17946 !
17952 Error=RETURN 0
17958 FNEND
17964 !
17970 !
17976 Stripscript=SUB Stripscript(S$) ! change * to subscript token, CTL-* to *
17982 DIM T$(80)
17988 FOR P=1 TO LEN(S$)
17994   N=NUM(S$[P,P])
18000   SELECT N
18006   CASE 94 ! ^
18012     T$[P,P]=CHR$(30) ! CTRL ^
18018   CASE 42 ! *
18024     T$[P,P]=CHR$(10) ! CTRL *
18030   CASE 30 ! CTRL ^
18036     T$[P,P]="^"
18042   CASE 10 ! CTRL *
18048     T$[P,P]="*"
18054   CASE ELSE
18060     T$[P,P]=S$[P,P]
18066   END SELECT
18072 NEXT P
18078 S$=T$
18084 IF S$="" THEN S$=" "
18090 SUBEND
18096 !
18102 !
18108 Even=DEF FNEven(N) ! determine whether N is an even integer
18114 IF ABS(N/2-INT(N/2))<1.E-10*N THEN
18120   RETURN 1
18126 ELSE
18132   RETURN 0
18138 END IF
18144 FNEND
18150 !
18156 !
18162 Help1=SUB Help1(Cplot) ! helpscreen for initial ("Begin:") display
18168 OPTION BASE 1
18174 DIM S$(18)[160]
18180 DATA 0: PRINTER OUTPUT Use the external printer as the output device (toggles with CRT OUTPUT).$
18186 DATA 0: CRT OUTPUT Use the CRT as the printer (toggles with PRINTER OUTPUT).$
18192 DATA 1: PLOTTER PLOT Use the external plotter as the plotting device (toggles with CRT PLOT).$
18198 DATA 1: CRT PLOT Use the CRT as the plotting device (toggles with PLOTTER PLOT).$
18204 DATA 2: CHOOSE PEN (Plotter-plot only) Select a pen of the external plotter.$

```


18210 DATA 3: STACKED PLOT (Plotter plot only) Suppress the X-axis label and tick-labels so that it can be stacked on top of a nother plot that

18216 DATA has the same X-axis and X-axis limits.\$4: HELP Invoke this HELP screen.\$

18222 DATA"5: ADJ. LABELSIZE (Plotter-plot only) Expand or contract the relative size of the axis- and tick-labels, as well as al pha-character plotting"

18228 DATA symbols.\$6: ROTATE PLOT (Plotter-plot only) Rotate the plot 90 degrees (toggles with UNROTATED PLOT).\$

18234 DATA 6: UNROTATED PLOT (Plotter-plot only) Use the normal (unrotated) orientation of the plot (toggles with ROTATE PLOT).\$

18240 DATA 7: PEN SPEED (Plotter-plot only) Select a pen-speed for the external plotter.\$

18246 DATA 8: GET DATAFILE Transfer data from a VISICALC file into memory.

18252 DATA \$CTRL-C Toggle from Isochron plotter to Concordia Plotter.\$CTRL-I Toggle from Concordia Plotte r to Isochron Plotter.\$

18258 DATA"3: DASH CTICK (Concordia plots only) Use a dash-symbol for the concordia ticks, oriented (with its labels) normal to the"

18264 DATA"concordia curve (toggles with CIRCLE CTICK).\$3: CIRCLE CTICK (Concordia-plots only) Use a small circle for the conco rdia ticks,"

18270 DATA"with tick-labels oriented horizontal (toggles with DASH CTICK)."

18276 DATA "\$9: START Enter the axis-names and limits, then start drawing the plot (for a plotter-plot, if you want"

18282 DATA "to bypass the drawing of the plot-box and labels, press CTRL-k9 instead)."

18288 READ S\$(*)

18294 CALL Helpscreen(S\$(*),21)

18300 SUBEND

18306 !

18312 !

18318 Help2:SUB Help2(Cplot)

18324 OPTION BASE 1

18330 DIM S\$(17)C160J

18336 DATA 1: ERROR ELLIPSE Use an open (unshifted) or solid (shifted) error-ellipse as the plotting symbol.\$

18342 DATA 3: POLYGON Use an open (unshifted) or solid (shifted) regular polygon as the plotting symbol. The polygon can have any number of sides

18348 DATA"and an arbitrary orientation. Star-shaped polygons are also permitted.\$4: OPTIONS Select error-input as percent or absolute,"

18354 DATA"1-sigma or 2-sigma; do a polynomial, linear, or log fit to data (ISOPLOT only); store the CRT plot on a disk...\$"

18366 DATA 5: SOLID ISOCHRON (Plotter-plot only) Use solid rather than dashed

18372 DATA 5: PRINT PLOT (CRT-plot only) Dump the CRT plot to the printer.\$5: SOLID ISOCHRON (Plotter-plot only) Use solid rather than dashed

18378 DATA lines to draw isochrons or chords\$6: ERASE PEN? (CRT-plot only) toggle between white (normal)

18384 DATA and erasing pen.\$8: GET DATAFILE Transfer data from a VISICALC file into memory.\$

18390 DATA 6: CHOOSE PEN (Plotter plot ony) Select a different pen from the external plotter.\$

18396 DATA 7: YORKFIT Do a Yorkfit (2-error linear regression) to the plotted data (must have at least 2 points).\$

18402 DATA 9: NEW PLOT Start a completely new plot.\$

18414 DATA CTRL-L Draft a phrase anywhere on the plot.\$CTRL-P Dump the CRT plot to the printer.\$

18420 DATA CTRL-D Dump the CRT plot to the external plotter.\$*OK.... Use any keyboard-symbol as a plotting -symbol.\$

18426 DATA CTRL-E Draw the 95%-confidence limit error-envelope about the last Yorkfit line.\$

18432 DATA CTRL F: (Concordia-plotter only) Force the next-invoked Yorkfit through an arbitrary age on the concordia-curve.\$

18438 DATA 0: ERROR BOX Use an open (unshifted) or solid (shifted) error-box as the plotting symbol.\$

18444 DATA 2: ERROR CROSS Use an error-cross as the plotting symbol.\$

18450 !

18456 IF Cplot THEN REDIM S\$(15)

18462 READ S\$(*)

18468 CALL Helpscreen(S\$(*),21)

18474 SUBEND

18480 !

18486 !

18492 Help3:SUB Help3

18498 OPTION BASE 1

18504 DIM S\$(7)C160J

18510 DATA 1: ABS ERRORS Input errors as their ABSOLUTE values (rather than as PERCENT).\$2: % ERRORS Input errors in PERCENT(rather than

18516 DATA their ABSOLUTE values).\$3: 1-SIG ERRORS Input errors at the 1-SIGMA level rather than the 2-SIGMA level.\$4: 2-SIG ERRORS Input errors

18522 DATA at the 2-SIGMA level rather than the 1-SIGMA level.\$6: EXPO FIT Fit an exponential curve

18528 DATA to the data of the form $Y=a \cdot E^{(b \cdot X)}$.\$7: LOG FIT Fit a logarithmic curve to the data of the form $Y=a \cdot \ln(b \cdot X)$.\$

18534 DATA 8: POLYFIT Fit a polynomial curve to the data.\$

18540 DATA 0: LOAD PLOT Put a CRT plot stored on a disk onto the CRT.\$5: STORE PLOT (CRT-plot only) Store the current CRT graphics on a disk.\$

18546 DATA 9: ESCAPE Return to the calling screen.

18552 READ S\$(*)

18558 CALL Helpscreen(S\$(*),21)

18564 SUBEND

18570 !

18576 !

18582 Help6:SUB Help6

18588 OPTION BASE 1

18594 DIM S\$(5)[160]

18600 DATA 1: PRINT PLOT Dump the CRT plot to the printer.\$9: NEW PLOT Abandon this plot and start a new one.\$3: ADD POINTS Add

18612 DATA more data-points to the plot and pool them with the previously-plotted points for any line- or curve-fitting.\$4: CONTINUE Return

18618 DATA "to the 'Select Plotting-Symbol' screen, and don't pool any additional data-points with previously-input points for line- or curve-fitting."

18624 DATA "\$5: ERROR ENVELOPE Draw the 95%-confidence limit error-envelope about the Yorkfit line."

18630 DATA "\$8: DELETE POINTS Delete a point from the next line- or curve-fit."

18636 READ S\$(*)

18642 CALL Helpscreen(S\$(*),21)

18648 SUBEND

18654 !

18660 !

18666 Help5:SUB Help5

18672 OPTION BASE 1

18678 DIM S\$(5)[160]

18684 DATA "1: MODEL-1 Choose the MODEL-1 Yorkfit as the best-fit line to the data (assumes that analytical error alone is the cause of the"

18690 DATA "observed scatter, & so weights the points according to their analytical errors).\$"

18696 DATA "2: MODEL-2 Choose the MODEL-2 Yorkfit as the best-fit line to the data (assumes nothing about the cause of the scatter, and weights"

18702 DATA "each point equally).\$3: MODEL-3 Choose the MODEL-3 Yorkfit as the best-fit line to the data (assumes that the points scatter because"

18708 DATA "of a combination of analytical error and a normally-distributed variation in the initial-ratios)."

18714 READ S\$(*)

18720 CALL Helpscreen(S\$(*),21)

18726 SUBEND

18732 !

18738 !

18744 Help4:SUB Help4(Cplot)

18750 OPTION BASE 1

18756 DIM S\$(8)[160]

18762 DATA 2: YORKFIT Do a York-type linear regression on the data-points and calculate isochron ages if appropriate.\$3: ADD POINTS DO

18768 DATA include the last-plotted data-points with subsequently-input points for any line- or curve-fitting.\$4: NEW POINTS DON'T include the

18774 DATA last-plotted points with subsequently-input points for any line- or curve-fitting.\$

18780 DATA 8: DELETE POINT Delete a point for the next line- or curve-fit.\$9: NEW PLOT Abandon this plot and start a completely new one.\$

18786 DATA CTRL L: Draft a phrase anywhere on the plot.\$

18787 DATA CTRL E: Draw the 95%-confidence limit error-envelope about the last Yorkfit line.

18792 DATA "\$CTRL F: (Concordia-plotter only) Do a York-type linear regression on the data points, forcing one of the concordia- intercepts"

18798 DATA through an arbitrary age on the concordia curve. Calculate the intercepts and errors.

```

18804 IF NOT Cplot THEN REQIM S$(6)
18810 READ S$(*)
18816 CALL Helpscreen(S$(*),21)
18822 SUBEND
18828 !
18834 !
18840 Pb_pb_age=SUB Pb_pb_age(R76,Age,OPTIONAL R76_err,Age_err,Hi_age,Lo_age)
18846 ! calculate Pb-207/206 age and (optional) error
18852 OPTION BASE 1
18858 DIM Pb_pb_age(3)
18864 U238_235=137.88
18870 Lambda238=1.55125E-4
18876 Lambda235=9.8485E-4
18882 Age=0
18888 IF NPAR=6 THEN Calc_errs=1
18894 MAT Pb_pb_age= (0)
18900 !
18906 FOR I=2-Calc_errs TO 2+Calc_errs
18912 R=R76
18918 IF Calc_errs THEN
18924 R=R+(I-2)*R76_err
18930 END IF
18936 IF R>.0156 AND R<1.9 THEN
18942 Trial_t=4500*(SGM(-(1/2)+(R>Lambda235/(Lambda238*U238_235)))+(R>.7))
18948 !
18954 Change=1
18960 REPEAT
18966 F=Lambda238*EXP(Trial_t*(Lambda238-Lambda235))/Lambda235
18972 T=LOG(1+(EXP(Trial_t*Lambda238)-1-F*(EXP(Trial_t*Lambda235)-1)))/(1/(U238_235*R)-F))/Lambda235
18978 Change=ABS(T-Trial_t)
18984 Trial_t=T
18990 UNTIL Change<.01
18996 Pb_pb_age(I)=T
19002 END IF
19008 NEXT I
19014 Age=Pb_pb_age(2)
19020 IF NPAR=6 THEN
19026 Hi_age=DROUND(Pb_pb_age(3),3)
19032 Lo_age=DROUND(Pb_pb_age(1),3)
19038 IF Hi_age AND Lo_age THEN Age_err=(Hi_age-Lo_age)/2
19044 END IF
19050 !
19056 SUBEND
19062 !
19068 !
19074 Concordia=SUB Concordia
19080 OPTION BASE 1
19086 COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Linx1,Linx2,Liny1,Liny2,Xspred,Yspred,Xloc,Yloc,Xm,Ym,Nx$,Ny$,Nx$,Ny$,Nx0$,Ny0$,Double
19092 COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Dffset
19098 COM /Plotbox2/ Nxr$,Nyr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Ticksizex1,Y1,Xend,Yend,Speed$,Crt,Hardplot,Cpen,X
gu,Ygu
19104 COM /Concordia/ Age_min,Age_max,Ctick,Oblique_tick
19110 OFF KBD
19116 OFF KEY
19122 !
19128 IF NOT Oblique_tick THEN Ct$="0"
19134 IF Oblique_tick THEN Ct$="-"
19140 Ctht=Tht/2
19146 CALL Tick(Age_max-Age_min,Ctick)
19152 IF Age_max-Age_min>3000 THEN Ctick=Ctick/2 ! more ticks for large age-spans

```

```

19158 IF (Age_max-Age_min)/Ctick<8 THEN Ctick=Ctick/2
19164 Plot_concordia=K=8*Ctick
19170 Cht=1.05*Tht ! character-size for concordia-tick labels
19176 !
19182 ! Find where to put first tick for concordia curve
19188 C=0
19194 FOR Pass=1 TO 2
19200 Age_min=C
19206 WHILE Age_min<Age_max AND (FNA(Age_min)<Xmin OR FNB(Age_min)<Ymin)
19212 Age_min=Age_min+K
19218 END WHILE
19224 !
19230 IF Pass=1 THEN
19236 C=Age_min-K
19242 K=Ctick
19248 END IF
19254 NEXT Pass
19260 !
19266 FOR Dx=0 TO Double
19272 IF Hardplot AND Dx THEN
19278 VIEWPORT Lx1,Lx2,Ly1,Ly2
19284 WINDOW Xmin,Xmax,Ymin,Ymax
19290 END IF
19296 CLIP Xmin,Xmax,Ymin,Ymax
19302 D=0
19308 DEG
19314 FOR T=Age_min-Ctick TO Age_max STEP Ctick/8! draw concordia curve
19320 X=FNA(T)
19326 Y=FNB(T)
19332 IF (X>Xmin) AND (Y>Ymin) THEN
19338 DRAW X,Y
19344 ELSE
19350 MOVE X,Y
19356 END IF
19362 !
19368 IF T/Ctick=INT(T/Ctick) THEN
19374 IF T>=Age_min THEN
19380 !
19386 ! ----- Concordia ticks & tick-labels -----
19392 !
19398 IF T THEN
19404 IF Oblique_tick THEN
19410 ! tangent to concordia for T, corrected for plot distortion
19416 Slope_angle=ATN(FNCSlope(T)*Ymm/Xmm)
19422 Tick_angle=Slope_angle-90! tick perpendicular to curve
19428 LDIR Tick_angle
19434 END IF
19440 LORG 5
19446 FOR L=1 TO 1+0*(Oblique_tick=0) !DISABLE LOOP
19452 IF Hardplot THEN IMOVE +FNCwidth(Cht,Xmm,0,1,D)/15,-FNCheight(Cht,0,Ymm,0)/15
19458 CSIZE Cht*(4-L)/3,.8+.2*Oblique_tick
19464 IF L=2 THEN MOVE X,Y
19470 LABEL Cht$
19476 NEXT L
19482 IF D THEN
19488 ! Check if tick-labels lie within plot-box
19494 L=Cht*(1.5+INT(LGT(T)))
19500 Nchars_tick=FNZ(T)+1
19506 IF Hardplot THEN Length_tlabel=(1.3*Oblique_tick+1+Nchars_tick)*FNCwidth(Cht,Xmm,0,Ar,0)! in UU
19512 IF Crt THEN Length_tlabel=(1.3*Oblique_tick+1+Nchars_tick)*FNCwidth(Cht,0,Xloc,Ar,1)! in UU's

```

```

19518      IF Hardplot THEN Height_tlabel=FNHeight<Cht,0,Ymm,0>*1.2
19524      IF Crt THEN Height_tlabel=Cht/Yloc*11/15*1.2! in 6U's,mm
19530      IF NOT Oblique_tick THEN
19536          X_label=X-Length_tlabel
19542          Y_label=Y+Height_tlabel
19548      ELSE
19554          X_label=X-COS(Tick_angle)*Length_tlabel
19560          Y_label=Y-SIN(Tick_angle)*Length_tlabel*Xmm/Ymm+COS(Tick_angle+90)*Height_tlabel
19566      END IF
19572      IF X_label>Xmin AND Y_label<Ymax THEN
19578          MOVE X-(Oblique_tick=0)*Cht*Ar/Xloc/2.,Y
19584          LORG 7+Oblique_tick
19590          CSIZE Cht,Ar
19596          IF NOT Oblique_tick THEN LABEL FNZcon$(T)
19602          IF Oblique_tick THEN LABEL FNZcon$(T)&" "
19608      END IF
19614      END IF
19620      MOVE X,Y
19626      END IF
19632      END IF
19638      D=NOT <D>
19644      END IF
19650      NEXT T
19656      NEXT Dx
19662      !
19668      IF File$="" THEN Errtype=0
19674      LDIR 0
19680      !
19686      SUBEND
19692      !
19698      !
19704!
19710 A=DEF FNA(T) ! 207/235 ratio from age
19716 RETURN EXP(9.8485E-4*T)-1
19722 FNEND
19728!
19734 B=DEF FNB(T) ! 206/238 ratio from age
19740 RETURN EXP(1.55125E-4*T)-1
19746 FNEND
19752!
19758 Cslope=DEF FNCslope(T) ! Slope of concordia curve for time T
19764 L8=1.55125E-4
19770 L5=9.8485E-4
19776 RETURN L8*EXP((L8-L5)*T)/L5
19782 FNEND
19788!
19794 D=DEF FND(X) ! 207/235 age from ratio
19800 RETURN LOG(1+X)/9.8485E-4
19806 FNEND
19812!
19818 E=DEF FNE(Y) ! 206/238 age from ratio
19824 RETURN LOG(1+Y)/1.55125E-4
19830 FNEND
19836!
19842 !
19848 Limits=SUB Limits(Xmin,Xmax,Ymin,Ymax,Age_min,Age_max,Xtik_inter,Ytik_inter)
19854 ! Calculate X- and Y-limits for age-defined concordia plot
19860 !
19866 Xmax=FNA(Age_max)
19872 Ymax=FNB(Age_max)

```

```

19878 CALL Tick(Xmax-FNA(Age_min),Xtik_inter)
19884 CALL Tick(Ymax-FNB(Age_min),Ytik_inter)
19890 !
19896 Xmin=0
19902 LOOP ! calculate minimum X-value for minimum age
19908 EXIT IF Xmin>Xmax OR FND(Xmin+Xtik_inter)>Age_min
19914 Xmin=Xmin+Xtik_inter
19920 END LOOP
19926 !
19932 Ymin=0
19938 LOOP ! calculate minimum Y-value for minimum age
19944 EXIT IF Ymin>Ymax OR FNC(Ymin+2*Ytik_inter)>Age_min
19950 Ymin=Ymin+Ytik_inter
19956 END LOOP
19962 !
19968 Tick(Xmax-Xmin,Xtik_inter)
19974 Tick(Ymax-Ymin,Ytik_inter)
19980 Tick((Age_max-Age_min)/(1+(Age_max-Age_min)3000)),Ctick)
19986 !
19992 SUBEND
19998 !
20004 !
20010 Conc_inters=SUB Conc_inters(Slope,Intercept,Cint(*))
20016 OPTION BASE 1
20022 ! Calculate intercepts of a line with the concordia curve
20028 DIM I$(0:1)[5]
20034!
20040 FOR J=0 TO 1
20046 ! 1st Trial-T is -500 Ma (lower-int) or 5500 Ma (upper)
20052 Trial_t=-500+5500*J
20058 LOOP
20064 C=FNCslope(Trial_t)
20070 X=(Intercept+C*FNA(Trial_t)-FNB(Trial_t))/(C-Slope)
20076 IF X<-1 THEN No_inter
20082 !
20088 T=FND(X)
20094 EXIT IF ABS(T-Trial_t)<.01
20100 Trial_t=T
20106 END LOOP
20112 !
20118 Cint(J+1)=DROUND(T,6)
20124 NEXT J
20130 SUBEXIT
20136 !
20142 No_inter:I$(0)="LOWER"
20148 I$(1)="UPPER"
20154 PRINT "YORKFIT LINE HAS NO "&I$(J)&" INTERCEPT WITH CONCORDIA..."
20160 Cint(J+1)=PI
20166 GOTO 20124
20172!
20178 SUBEND
20184 !
20190 !
20196 Cint_errs=SUB Cint_errs(Model,Slope,Inter,Errsl(*),Errrint(*),Xbar(*),Cint(*))
20202 ! Calculate concordia-intercept errors -----
20208 !
20214 Slope_err=Errsl(Model)
20220 Inter_err=Errrint(Model)
20226 X_bar=Xbar(Model)
20232 !

```

```

20238 FOR J=3 TO 6
20244 IF J<5 AND Cint(1)=PI THEN No_env_inter
20250 IF J>4 AND Cint(2)=PI THEN No_env_inter
20256 SELECT J
20262 CASE 3 ! Lower intercept, younger age-limit
20268 Trial_t=-1000
20274 CASE 4 ! Lower intercept, older age-limit
20280 Trial_t=Cint(1)
20286 CASE 5 ! Upper intercept, younger age-limit
20292 Trial_t=Cint(2)
20298 CASE 6 ! Upper intercept, older age-limit
20304 Trial_t=6000
20310 END SELECT
20316 !
20322 A=2*FNEven(J)-1
20328 I=9.E+99
20334 LOOP
20340 S=FNCslope(Trial_t)
20346 B=FNB(Trial_t)-S*FNA(Trial_t)-Inter
20352 D=2*(B*(S-Slope)+K_bar*Slope_err^2)
20358 E=(S-Slope)^2-Slope_err^2
20364 G=D^2-4*E*(B^2-Inter_err^2)
20370 IF G<0 THEN No_env_inter
20376 X=(A*SQR(G)-D)/(2*E)
20382 IF X<-1 THEN No_env_inter
20388 T=FND(X)
20394 IF ABS(T-Trial_t)>I THEN No_env_inter
20400 EXIT IF ABS(T-Trial_t)<.01
20406 I=ABS(T-Trial_t)
20412 Trial_t=T
20418 END LOOP
20424 IF Cint(J)<>PI THEN Cint(J)=DRROUND(T,5)
20430 NEXT J
20436 Cint(7)=DRROUND(ABS(Cint(4)-Cint(3))/2,2) ! average lower-int. error
20442 Cint(8)=DRROUND(ABS(Cint(6)-Cint(5))/2,2) ! " upper-int. "
20448 IF Cint(3)=PI OR Cint(4)=PI THEN Cint(7)=PI
20454 IF Cint(5)=PI OR Cint(6)=PI THEN Cint(8)=PI
20460 SUBEXIT
20466 !
20472 No_env_inter=PRINT "ARM";J-2;"OF ERROR-ENVELOPE HAS NO CONCORDIA INTERCEPT..."
20478 Cint(J)=PI
20484 GOTO 20430
20490!
20496 SUBEND
20502 !
20508 !
20514 Model4_calc=SUB Model4_calc(Slope,Inter,Cint(*),Interxy(*),Chordlength)
20520 CALL Conc_inters(Slope,Inter,Cint(*))
20526 Interxy(1,1)=FNA(Cint(1)) ! X value of lower intercept
20532 Interxy(1,2)=FNB(Cint(1)) ! Y " " " "
20538 Interxy(2,1)=FNA(Cint(2)) ! X " " upper "
20544 Interxy(2,2)=FNB(Cint(2)) ! Y " " " "
20550 Chordlength=SQR((Interxy(2,1)-Interxy(1,1))^2+(Interxy(2,2)-Interxy(1,2))^2)
20556 SUBEND
20562 !
20568 !
20574 Help?=SUB Help?
20580 OPTION BASE 1
20586 DIM S$(13)E160]
20592 DATA*1: ACCEP MODEL-1 Accept the MODEL-1 fit as the best-fit line, so don't do any other model Yorkfits.*"

```

```

20598 DATA*2: DD MODEL-2      Do a MODEL-2 fit to the data. A MODEL-2 fit weights each point equally, regardless of assigned errors. This model is useful"
20604 DATA "if the points are obviously scattered due to some non-analytical cause, but you have no idea what that cause might be.$
"
20610 DATA"                GENERAL MODEL-4 COMMENTS: A MODEL-4 fit assumes that the degree of scatter will increase proportional to the degree"
20616 DATA of discordance either away from the Upper Intercept (MODEL-4UI) or the Lower Intercept (MODEL-4LI). So the MODEL-4UI fit weights the points
20622 DATA "that are closest to the UPPER intercept the most, and MODEL-4LI weights the points closest to the LOWER intercept the most. These models are"
20628 DATA"useful if you can assume that the most-concordant points are the least affected by complexities either in multi-episodic disturbance (4UI), or"
20634 DATA "inheritance of zircons with different 207/206 ages (4LI).$The penalty you pay for choosing a MODEL-4 fit is that the Student's-t multiplier"
20640 DATA "on the errors tends to be significantly higher, since except for the 2 most-concordant points, the points only count as 'fractional' points"
20646 DATA "towards increasing the degrees of freedom of your data set."
20652 DATA "$<Note: MODEL-4 fits are only allowed with 6 or more points>$"
20658 DATA 3: DD MODEL-4UI      Do a MODEL-4UI fit to the data.$
20664 DATA 4: DD MODEL-4LI      Do a MODEL-4LI fit to the data.
20670 READ S$(*)
20676 CALL Helpscreen(S$(*),21)
20682 SUBEND
20688 !
20694 !
20700 Print_inters:SUB Print_inters(Cint_r(*),Cint(*),Prntr)
20706 ! print out the concordia intercepts, making sure that no extra significant figures are printed out.
20712 DATA 1,1,1,1
20718 READ A1,A2,D1,D2
20724 MAT Cint_r= Cint
20730 IF Cint_r(7)<>PI THEN A1=10*FNZ(Cint_r(7))
20736 IF Cint_r(8)<>PI THEN A2=10*FNZ(Cint_r(8))
20742 IF Cint_r(1)<>PI THEN D1=1-FNZ(A1/Cint_r(1))
20748 IF Cint_r(2)<>PI THEN D2=1-FNZ(A2/Cint_r(2))
20754 IF Cint_r(1)<>PI THEN Cint(1)=DROUND(Cint_r(1),D1)
20760 IF Cint_r(2)<>PI THEN Cint(2)=DROUND(Cint_r(2),D2)
20766 IF Cint_r(1)-Cint(1)>1 THEN Cint(1)=INT(Cint_r(1))
20772 IF Cint_r(2)-Cint(2)>1 THEN Cint(2)=INT(Cint_r(2))
20778 PRINTER IS Prntr
20784 PRINT
20790 IF Cint_r(1)<>PI THEN
20796   PRINT "LOWER INTERCEPT: ";
20802   IF Cint_r(1)<>PI THEN PRINT Cint(1);
20808   IF Cint_r(1)<>PI AND Cint_r(7)<>PI THEN PRINT "+/-";Cint(7);TAB(36);"Ma";
20814   IF Cint_r(1)=PI OR Cint_r(7)=PI THEN PRINT TAB(36);"Ma";
20820   IF Cint_r(3)=PI AND Cint_r(4)=PI THEN
20826     PRINT
20832   ELSE
20838     PRINT TAB(53);"< ";
20844   END IF
20850   IF Cint_r(3)<>PI THEN PRINT ">";DROUND(Cint(3),D1);
20856   IF Cint_r(4)<>PI THEN PRINT " <";DROUND(Cint(4),D1);">"
20862   IF Cint_r(4)=PI THEN PRINT ">"
20868 END IF
20874 IF Cint_r(2)<>PI THEN
20880   PRINT "UPPER INTERCEPT: ";
20886   IF Cint_r(2)<>PI THEN PRINT Cint(2);
20892   IF Cint_r(2)<>PI AND Cint_r(8)<>PI THEN PRINT "+/-";Cint(8);TAB(36);"Ma";
20898   IF Cint_r(2)=PI OR Cint_r(8)=PI THEN PRINT TAB(36);"Ma";
20904   IF Cint_r(5)=PI AND Cint_r(6)=PI THEN

```



```

20910 PRINT
20916 ELSE
20922 PRINT TAB(53);"(<";
20928 END IF
20934 IF Cint_r(5)<>PI THEN PRINT ">";DROUND(Cint(5),02);
20940 IF Cint_r(6)<>PI THEN PRINT " (<";DROUND(Cint(6),02);)"
20946 IF Cint_r(6)=PI THEN PRINT ">)"
20952 PRINT USING "2(/,K)";"(UNCERTAINTIES ARE 95%-CONFIDENCE LIMITS)",RPT$("- ",80)
20958 END IF
20964 SUBEND
20970 !
20976 !
20982 Label_inters:SUB Label_inters(Cint(*),N_interlabels,Mswd)
20988 ! label the concordia-intercepts on the lower-right corner of the plot
20994 OPTION BASE 1
21000 COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Limx1,Limx2,Limy1,Limy2,Xspred,Yspred,Kloc,Yloc,Xmm,Ymm,Hx$,Ny$,Hx0$,Ny0$,Double
21006 COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
21012 COM /Plotbox2/ Mxr$,Myr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Ticksizex,Xl,Yl,Xend,Yend,Speed$[4],Crt,Hardplot,Cpen,
Xgu,Ygu
21018 DIM I$(8)[8],Pm(7,3)
21024 DATA 1,10,-2,7,10,-1,4,13,-2,4,7,-1,1,4,-2,7,4,-1,9,1,-2
21030 OFF KEY
21036 OFF KBD
21042 IF Hardplot AND NOT Plotter_dump THEN
21048 DISP "PRESS "&FNH$("&k4")& " TO LABEL INTERCEPTS ON PLOT, "&FNH$("&ko")& " TO DECLINE."
21054 ON KEY 0 LABEL " NO LABEL" GOTO 21342
21060 ON KEY 4 LABEL "LABEL INTERS" GOTO 21084
21066 GOTO 21066
21072 END IF
21078 !
21084 READ Pm(*) ! plus-or-minus symbol
21090 OFF KEY
21096 N_interlabels=1+N_interlabels
21102 IF Hardplot THEN Pm(7,2)=0
21108 IF Crt THEN CALL Graph
21114 Iht=1.1*Tht
21120 Charwidth=FNWidth(Iht,Xmm,Kloc,Ar,Crt)
21126 Charheight=FNHeight(Iht,Yloc,Ymm,Crt)
21132 MAT I$= ("***")
21138 FOR J=1 TO 8
21144 IF J=1 OR J=2 OR J=7 OR J=8 THEN
21150 IF Cint(J)<>PI THEN I$(J)=VAL$(Cint(J))
21156 I$(J)=FNZcon$(0,I$(J))
21162 END IF
21168 NEXT J
21174 L1=LEN(I$(2))
21180 L2=LEN(I$(8))
21186 L3=LEN(I$(1))
21192 L4=LEN(I$(7))
21198 Nchar=14+L1+L2+L3+L4
21204 CSIZE Iht,Ar
21210 LORG 1
21216 PEN Cpen
21222 FOR Dx=0 TO Double
21228 MOVE Xmax+Dx*Offset/Xmm,Ymin+Dx*Offset/Ymm
21234 IMOVE -(Nchar+14+2.4*Hardplot)/2*Charwidth,Charheight*(3.7+2.6*Hardplot)*N_interlabels-1/2+.2*Hardplot
21240 LABEL "INTERCEPTS AT"
21246 IMOVE -(Nchar-15+0*Hardplot)/2*Charwidth,0
21252 LABEL I$(2)
21258 IMOVE (.5+0*Hardplot+L1)*Charwidth,Charheight*(16/15+Hardplot*9/15)

```

```

21264 SYMBOL Pn(*)
21270 IMOVE (1+0*Hardplot)*Charwidth/2,0
21276 LABEL I$(8)& and "BI$(1)
21282 IMOVE (L2+L3+5.5+0*Hardplot)*Charwidth,Charheight*(16/15+Hardplot*9/15)
21288 SYMBOL Pn(*)
21294 IMOVE (1+0*Hardplot)*Charwidth/2,0
21300 LABEL I$(7)& " Ma"
21306 MOVE Xmax+Dx*Offset/Xm, Ymin+Dx*Offset/Ym
21312 IMOVE ((-Nchar-12-2.9*Hardplot)/2*Charwidth),Charheight*((3.7+2.6*Hardplot)*N_interlabels-2.8-1.3*Hardplot)
21318 LABEL "MSWD="&FNZcon$(DROUND(Ms wd,3))&"
21324 NEXT Dx
21330 MOVE Xmin,Ymax
21336 PEN 0
21342 DISP
21348 SUBEND
21354 !
21360 !
21366 Graph:SUB Graph
21372 GRAPHICS ON
21378 ALPHA OFF
21384 SUBEND
21390 !
21396 !
21402 Alpha:SUB Alpha
21408 GRAPHICS OFF
21414 ALPHA ON
21420 SUBEND
21426 !
21432 !
21438 Un:DEF FNU$(String$)
21444 RETURN CHR$(132)&String$&CHR$(128)
21450 FNEND
21456 !
21462 !
21468 Zcon:DEF FNZcon$(U,OPTIONAL S$) ! change variable to string, zeroes to 0's
21474 DIM N$(80)
21480 IF NPAR=1 THEN
21486 ON ERROR GOTO 21558
21492 N$=VAL$(U)
21498 OFF ERROR
21504 ELSE
21510 N$=S$
21516 END IF
21522 FOR I=1 TO LEN(N$)
21528 IF N$(I,I)="" THEN N$(I,I)="0"
21534 NEXT I
21540 !
21546 RETURN N$
21552 !
21558 RETURN "0"
21564 FNEND
21570 !
21576 !
21582 Ticklabel:SUB Ticklabel(Tick$(*),Tickval(*),Dec(*),Dec_max,Lzero,N_majortix)
21588 OPTION BASE 1
21594 ! convert the tick-value array into a string array, with zeroes replaced by 0h's
21600 !
21606 DIM Zero$(10)
21612 Zero$="0000000000"
21618 FOR I=1 TO N_majortix

```

```

21624 Tick$(I)=FNZcon$(Tickval(I))
21630 IF Dec_max AND Dec_max>Dec(I) THEN
21636 IF Dec(I)=0 THEN Tick$(I)=Tick$(I)&"."
21642 Tick$(I)=Tick$(I)&Zero$[1,Dec_max-Dec(I)]
21648 END IF
21654 IF Lzero THEN
21660 IF Tickval(I)>0 AND Tickval(I)<1 THEN Tick$(I)="0"&Tick$(I)
21666 IF Tickval(I)<0 AND Tickval(I)>-1 THEN Tick$(I)="-0"&Tick$(I)[2]
21672 END IF
21678 NEXT I
21684 SUBEND
21690 !
21696 !
21702 Cwidth=DEF FNCwidth(Charsize,Wm,Xloc,Ar,Crt)
21708 ! return character width, in UU's
21714 Xfactor=Xloc*Crt+Wm*(NOT Crt)
21720 IF Crt THEN RETURN Charsize*Ar/Xfactor
21726 IF NOT Crt THEN RETURN Charsize*Ar*1.183/.67/Xfactor
21732 FNEND
21738 !
21744 !
21750 Cheight=DEF FNCheight(Charsize,Yloc,Ym,Crt)
21756 ! return character height in UU's
21762 Yfactor=Yloc*Crt+Ym*(NOT Crt)
21768 RETURN Charsize/Yfactor
21774 FNEND
21780 !
21786 !
21792 Printer_timeout=SUB Printer_timeout
21798 COM /Printplot/ Printer_ok,Plotter_ok
21804 OFF TIMEOUT
21810 OFF KEY
21816 OFF KBD
21822 Printer_ok=0
21828 PRINTER IS CRT
21834 Clear
21840 PRINT TABXY(1,12);FNCenter$(FNH$("*** PRINTER DOES NOT RESPOND ***"))
21846 Beeper(50,1200,.03,.01)
21852 SUBEND
21858!
21864!
21870 Beeper=SUB Beeper(Ncycles,Hertz,Beep_time,Wait_time)
21876 FOR I=1 TO Ncycles
21882 BEEP Hertz,Beep_time
21888 WAIT Wait_time
21894 NEXT I
21900 SUBEND
21906 !
21912 !
21918 Plotter_timeout=SUB Plotter_timeout
21924 COM /Printplot/ Printer_ok,Plotter_ok
21930 OFF TIMEOUT
21936 OFF KEY
21942 OFF KBD
21948 Clear
21954 Plotter_ok=0
21960 PRINT TABXY(1,12);FNCenter$(FNH$("**** PLOTTER DOES NOT RESPOND ****"))
21966 Beeper(50,1200,.03,.01)
21972 SUBEND
21978 !

```

```

21984 !
21990 Center:=DEF FNCenter$(S$) ! create a centered string
21996 DIM Pad$[80]
22002 L=LEN(S$)
22008 IF L<80 THEN
22014 Pad$=RPT$(" ",(80-L)/2)
22020 RETURN Pad$&S$&Pad$
22026 ELSE
22032 RETURN S$
22038 END IF
22044 FNEND
22050 !
22056 !
22062 Check_rho:=SUB Check_rho(Rho,Xerr,Yerr,X,Y,Percent,Ok)
22068 IF Percent THEN ! convert to either percent or fractional errors
22074 Ex=Xerr
22080 Ey=Yerr
22086 ELSE
22092 Ex=Xerr/X
22098 Ey=Yerr/Y
22104 END IF
22110 !
22116 IF ABS(Rho<=1 AND (Ex^2+Ey^2))=(2*Rho)*Ex*Ey THEN
22122 Ok=1
22128 ELSE
22134 Ok=0
22140 PRINT USING "/,K";FNH$(" **** ERROR-CORREL. OF "&RHO$(Rho)&" IS NOT POSSIBLE **** ")
22146 Clunk
22152 WAIT 2
22158 END IF
22164 SUBEND
22170 !
22176 !
22182 Col:=DEF FNCol$(N) ! return column-letter from column-number
22188 IF N<25 THEN RETURN CHR$(N+2+64)
22194 IF N=25 THEN RETURN "A"&CHR$(N+2+64-26)
22200 FNEND
22206 !
22212 !
22218 Convert_vc:=SUB Convert_vc(Dfile,Uc(*),Pr(*),Cplot,Name$(*),Df_xname$,Df_yname$,File$,Ms$(**))
22224 !
22230 ! Subprogram to access UISICALC /PF files. The UISICALC files must have
22236 ! the following attributes:
22242 !
22248 ! The file title, if any, must occupy the first row of the file, and
22254 ! will be printed out by this program above the data.
22260 !
22266 ! The file must have been set for a column-width of 9, and stored
22272 ! with a printer-width of 80 (not 132).
22278 !
22284 ! The column-names or headings can occupy 2 rows, but must appear dir-
22290 ! ectly above a line of "equals" signs (=====) which serves as
22296 ! both an indicator of the column-heading rows and as a visual double-
22302 ! underscore sign.
22308 !
22314 ! The sample or row names are assumed to be in the first two columns
22320 ! (A and B) of the file.
22326 !
22332 ! The file can contain up to 101 rows of sample-data and up to
22338 ! 50 columns.

```

```

22344 !
22350 OPTION BASE 1
22356 OFF KEY
22362 OFF KBD
22368 Alpha
22374 FileD$=File$
22380 !
22386 COM /Printplot/ Printer_ok,Plotter_ok
22392 DIM U$(500),Temp$(80),R$(160),Str$(8)[16],Crud$(24),Cnum(50),C$(5)[6],Cat$(80)[80],Row$(500),Ufile$(80)[10],FileD$(12),Cplot_
col(5),Tc(5),Cpl$(5)[9]
22398 DIM Title$(80),L1$(500),L2$(500),Ifile$(25),Top_cname$(50)[9],Bottom_cname$(50)[9],Cname$(50)[18],Bh$(5)[9],Th$(5)[9],Drive_s
tring$(15),T$(18)
22404 !
22410 Print=0
22416 Q$=CHR$(34)
22422 Start=Clear
22428 OFF KEY
22434 ON KBD GOTO 22632
22440 PRINTER IS CRT
22446 !
22452 PRINT TABXY(1,4);"Press "&FNH$( " k0 "&" for a catalog of the disk in the "&FNUn$("left")&"-hand drive."
22458 PRINT TABXY(1,6);"Press "&FNH$( " k1 "&" for a catalog of the disk in the "&FNUn$("right")&"-hand drive."
22464 PRINT TABXY(1,8);"Press "&FNH$( " k2 "&" or "&FNH$( " CTRL H "&" for HELP."
22470 PRINT TABXY(1,10);"Press "&FNH$( " k4 "&" to load a Visicalc datafile into memory."
22476 PRINT TABXY(1,14);"Press "&FNH$( " k9 "&" to escape."
22482 !
22488 ON KEY 9 LABEL " ESCAPE" GOTO 25140 ! SUBEND line
22494 IF Dfile THEN
22500 ON KEY 8 LABEL " SCRATCH FILE" GOTO Scratchfile
22506 PRINT TABXY(1,18);"<File "&FNUn$(File$)&" currently in memory)"
22512 ELSE
22518 ON KEY 8 CALL Clunk
22524 PRINT TABXY(1,18);"<No datafile currently in memory)"
22530 END IF
22536 IF Dfile THEN ON KEY 8 LABEL " SCRATCH FILE" GOTO Scratchfile
22542 ON KEY 4 LABEL "LOAD DATAFILE" GOTO Whichfile
22548 ON KEY 0 LABEL "CATALOG Drv #0" GOTO 22860
22554 ON KEY 1 LABEL "CATALOG Drv #1" GOTO 22884
22560 ON KEY 2 LABEL " HELP" GOTO 22674
22566 IF Print THEN
22572 PRINT TABXY(1,12);"Press "&FNH$( " k3 "&" to print out the datafile-data on the CRT."
22578 ON KEY 3 LABEL " DISPLAY DATA" GOTO Use_crt
22584 DISP FNH$( " PRINTOUT DATA ON THE PRINTER (make sure printer is ON & ON LINE)")
22590 ELSE
22596 PRINT TABXY(1,12);"Press "&FNH$( " k3 "&" to print out the datafile-data on the printer."
22602 ON KEY 3 LABEL " PRINT DATA" GOTO Use_printer
22608 DISP FNH$( " DISPLAY DATA ON CRT ONLY ")
22614 END IF
22620 GOTO 22620
22626 !
22632 K=FNGet_key
22638 OFF KBD
22644 IF K=-79 THEN DUMP ALPHA
22650 IF K<>8 THEN
22656 Clunk
22662 GOTO 22428
22668 ELSE
22674 CALL Help8
22680 GOTO 22422
22686 END IF

```

```

22692 !
22698 Scratchfile:MAT Uc= (0)
22704 MAT Name$= ("")
22710 Dfile=0
22716 OFF KEY
22722 Clear
22728 IF File$(<)" THEN
22734 DISP FNM$(" FILE "&CHR$(34)&File$&CHR$(34)&" SCRATCHED FROM MEMORY")
22740 File$=""
22746 WAIT 2
22752 END IF
22758 SUBEXIT
22764 !
22770 Use_printer:Printer_ok=1
22776 ON TIMEOUT 7,.1 CALL Printer_timeout
22782 PRINTER IS 701
22788 PRINT
22794 IF Printer_ok THEN
22800 OFF TIMEOUT
22806 Print=1
22812 BEEP
22818 END IF
22824 GOTO Start
22830 !
22836 Use_crt:Print=0
22842 BEEP
22848 GOTO Start
22854 !
22860 Drive_string$=Ms$(0)
22866 Drive_number=0
22872 GOTO 22896
22878 !
22884 Drive_string$=Ms$(1)
22890 Drive_number=1
22896 OFF KEY
22902 OFF KBD
22908 Clear
22914 DISP "Getting disk catalog..."
22920 FOR P=1 TO 1 !+Print
22926 IF P=1 THEN PRINTER IS CRT
22932 IF P=2 THEN PRINTER IS 701
22938 ON ERROR GOTO Bad_cat
22944 S=8 ! CAT strings containing filenames start on 8th CAT string
22950 Nfiles=0
22956 CAT Drive_string$ TO Cat$(*)
22962 DISP
22968 OFF ERROR
22974 PRINT TABXY(15,2);FNM$("FILES ON DRIVE %"&Drive_string$(13));TAB(40);FNM$("("&TRIM$(Cat$(2)[8])&")")
22980 PRINT
22986 LOOP ! determine if file is a VISICALC /PF file (must end in A and be padded with underscore characters
22992 Tac$=TRIM$(REUS$(Cat$(S)[1,10]))
22998 EXIT IF LEN(Tac$)=0
23004 LOOP
23010 EXIT IF Tac$(1,1)(<)"_"
23016 Tac$=Tac$(2)
23022 END LOOP
23028 IF Tac$(1,1)="A" AND Nfiles<=80 THEN
23034 Nfiles=1+Nfiles
23040 Ufile$(Nfiles)=REUS$(Tac$(2))
23046 ! display valid Visicalc files in inverse-video, non-Visicalc files normal & in parentheses

```

```

23052     PRINT TAB(15);CHR$(129)&" #"&VAL$(Nfiles);TAB(21);"-- ";Ufile$(Nfiles)&" "&CHR$(128)
23058     ELSE
23064     PRINT TAB(15);("&TRIM$(Cat$(S)[1,10])&")"
23070     END IF
23076     S=S+1
23082     EXIT IF LEN(TRIM$(Cat$(S)))=0 OR S=80
23088     END LOOP
23094     OFF ERROR
23100     NEXT P
23106     PRINTER IS CRT
23112     IF Nfiles=0 THEN
23118     PRINT USING "/,5X,K";FNH$("THERE AREN'T ANY /PF VISICALC FILES ON THE DISK IN DRIVE# "&Drive_string$(L13))
23124     DISP "Press"&FNCT&"when ready..."
23130     PAUSE
23136     GOTO 22422
23142     ELSE
23148     DISP "ENTER THE FILE-"&FNUn$("NUMBER")&" THAT YOU WANT TO LOAD (press"&FNCT&"to escape)";
23154     Filenum=0
23160     INPUT Filenum
23166     IF Filenum(1 OR Filenum)>Nfiles THEN 22422
23172     Temp%=Ufile$(Filenum)
23178     GOTO Catfile
23184     END IF
23190     !
23196     Bad_cat=OFF ERROR
23202     DISP FNH$(ERRM%)
23208     Clunk
23214     WAIT 2
23220     GOTO 22422
23226     !
23232     Whichfile=Clear
23238     OFF KBD
23244     OFF KEY
23250     PRINT USING "2/,K,/,K,/";"WHICH VISICALC ASCII-FILE [DRIVE] DO YOU WANT TO BRING INTO MEMORY?";"(example: Trialfile:1)"
23256     IF LEN(File$) THEN
23262     PRINT TABXY(1,7);"Press"&FNCT&"for more data from "&Q&FileO&Q&
23268     Temp%=File$
23274     END IF
23280     ON KBD GOTO 23310
23286     ON KEY 9 LABEL "   ESCAPE" GOTO 22422
23292     DISP "?"
23298     GOTO 23298
23304     !
23310     K=FNGet_key
23316     OFF KBD
23322     OFF KEY
23328     SELECT K
23334     CASE 65 TO 90,97 TO 122
23340     OUTPUT KBD;CHR$(K);
23346     PRINT TABXY(1,7);RPT$(" ",80)
23352     DISP FNst%;
23358     INPUT Temp%
23364     IF Temp%="*" THEN 22422
23370     CASE -67     ! CONTINUE
23376     CASE ELSE
23382     GOTO 23280
23388     END SELECT
23394     Catfile=Clear
23400     L=LEN(Temp%)
23406     IF L=0 THEN Whichfile ! ENTER pressed with no input

```

```

23412 IF L<3 THEN
23418 N1=NUM(Temp$[1,1])
23424 N2=NUM(Temp$[L,L])
23430 IF N1>48 AND N1<58 AND N2>47 AND N2<58 THEN Temp$=Ufile$(VAL(Temp$))
23436 END IF
23442 LOOP
23448 Colon=POS(Temp$,":") ! Use colon to separate file-name from drive#
23454 EXIT IF Colon
23460 Temp$=TRIM(Temp$)&"&VAL$(Drive_number)
23466 END LOOP
23472 ON ERROR GOTO Badspec
23478 File$=TRIM(Temp$[1,Colon-1])
23484 FileD$=File$
23490 L=LEN(File$)
23496 IF L<10 THEN File$=File$&"A"
23502 Drive_number=VAL(Temp$[1+Colon])
23508 Drive_string$=Ms$(Drive_number)
23514 OFF ERROR
23520 !
23526 L=LEN(File$)
23532 IF L<10 THEN File$[L+1,10]=RPT$("_",10-L) ! add underscore characters as fill to be compatible with the Visicalc format
23538 !
23544 Badcount=0
23550 Recoup:ON ERROR GOTO Badfile
23556 DISP "Accessing file "&Q$&fileD$&Q$&"..."
23562 Ifile$=file$&Drive_string$
23568 ASSIGN @P1 TO Ifile$
23574 ENTER @P1;U$
23580 L=LEN(U$)
23586 L=MIN(80,L)
23592 Title$=U$[1,L] ! first line in the file assumed to be a title-line
23598 File$=Temp$
23604 OFF ERROR
23610 !
23616 Clear
23622 !
23628 Control1=POS(Title$,CHR$(27)) ! Escape-character location -
23634 IF Control1 THEN ! trim printer-control code if present
23640 Control2=POS(Title$[Control1],"")! end of control-sequence
23646 IF Control2 THEN Title$=Title$[1,Control1-1]&Title$[Control2]
23652 END IF
23658 PRINT USING "K,/";Title$
23664 OFF KBD
23670 GOSUB Search_colnames
23676 !
23682 IF NOT Ccol THEN
23688 ! Enter the column-identifiers (alphabetic) and convert to column-numbers
23694 Enter_columns=DISP "ENTER THE COLUMNS (D,G,AF...) CONTAINING X [, Xerr], Y [, Yerr [, Rho(XY)]]";
23700 Entercol1:R$=""
23706 MAT Cnum= (0)
23712 C_kbd:ON KBD GOTO 23742
23718 ON KEY 9 LABEL " ESCAPE" GOTO Exit_sub
23724 DISP "?"
23730 GOTO 23730
23736 !
23742 K=FNGet_key
23748 OFF KBD
23754 OFF KEY
23760 IF K<65 OR (K>90 AND K<97) OR K>122 THEN C_kbd
23766 OUTPUT KBD;CHR$(K);

```



```

23772 DISP FNrst$;
23778 Retrieve_str(R$,Str$(*),Ninputs)
23784 IF Str$(1)="*" THEN Exit_sub
23790 MAT C$= (**)
23796 !
23802 SELECT Ninputs
23808 CASE 2 ! X- and Y-columns only
23814 C$(1)=Str$(1)
23820 C$(3)=Str$(2)
23826 CASE 4,5 ! X, X-error, Y, Y-error [,Rho] columns
23832 C$(1)=Str$(1)
23838 C$(2)=Str$(2)
23844 C$(3)=Str$(3)
23850 C$(4)=Str$(4)
23856 C$(5)=Str$(5)
23862 CASE ELSE
23868 CALL Bad_input("INVALID COLUMN-INPUT")
23874 GOTO Entercol1
23880 END SELECT
23886 END IF
23892 !
23898 !
23904 FOR I=1 TO 5 ! figure out where in the row-strings to find the numeric
23910 ! data for the specified columns
23916 L=LEN(C$(I))
23922 IF L THEN
23928 C$(I)=UPC$(C$(I)) ! convert column-letters to uppercase
23934 IF L=1 THEN Cnum(I)=NUM(C$(I))-64
23940 ! Cnum(I) is the column-number for column-letter C$(I)
23946 IF L=2 THEN Cnum(I)=26*(NUM(C$(I)[1,1])-64)+NUM(C$(I)[2,2])-64
23952 IF L>2 THEN
23958 Bad_input(Crud$)
23964 GOTO Entercol1
23970 END IF
23976 IF ((I=1 OR I=3) AND Cnum(I)=0) OR Cnum(I)<3 OR Cnum(I)-2>Num_datacolumns THEN
23982 Bad_input("INVALID COLUMN-SPECIFIER RESPONSE")
23988 GOTO Entercol1
23994 END IF
24000 END IF
24006 NEXT I
24012 !
24018 MAT Uc= (0)
24024 Row=0
24030 LOOP ! get sample names and numeric data from file
24036 ON ERROR GOTO Done
24042 Row$="" ! this row-string is the entire row of the file
24048 FOR I=1 TO Nsux ! build row-string from "Nsux" calls from file
24054 ENTER @P1;U$
24060 Row$=Row$U$[1+3*(I)1]
24066 NEXT I
24072 OFF ERROR
24078 Row=i+Row
24084 Name$(Row)=TRIM$(Row$[1,18]) ! row-names in 1st 2 columns (A and B)
24090 FOR Col=1 TO 5
24096 C=Cnum(Col)
24102 IF C THEN
24108 ON ERROR GOTO 24132 !**
24114 Uc(Row,Col)=VAL(Row$[9*C-8,9*C]) ! extract numeric value
24120 DISP Name$(Row),Row;Col,Uc(Row,Col)
24126 END IF

```

```

24132 OFF ERROR !**
24138 NEXT Col
24144 END LOOP
24150 !
24156 Done:OFF ERROR ! got all data, so printout or display values
24162 ASSIGN @P1 TO *
24168 Clear
24174 FOR P=1 TO 1+Print
24180 IF P=1 THEN PRINTER IS CRT
24186 IF P=2 THEN PRINTER IS 701
24192 PRINT USING "K,/";Title$
24198 DATA "X", "X-ERR", "Y", "Y-ERR", "RHO"
24204 RESTORE 24198
24210 READ Bh$(*) ! default column-names
24216 MAT Th$= (**)
24222 FOR I=1 TO 5 ! build string-arrays containing upper column-names
24228 ! (Top_cname$) and lower column-names (Bottom_cname$)
24234 IF Cnum(I)>0 AND Cnum(I)<=50 THEN
24240 IF Bottom_cname$(Cnum(I)-2)<>" OR Top_cname$(Cnum(I)-2)<>" THEN
24246 Th$(I)=Top_cname$(Cnum(I)-2)
24252 Bh$(I)=Bottom_cname$(Cnum(I)-2)
24258 END IF
24264 END IF
24270 NEXT I
24276 PRINT TAB(27);Th$(1);TAB(37);Th$(2);TAB(47);Th$(3);TAB(59);Th$(4);TAB(70);Th$(5)
24282 PRINT "SET#";TAB(9);"NAME";TAB(27);Bh$(1);TAB(37);Bh$(2);TAB(47);Bh$(3);TAB(59);Bh$(4);TAB(70);Bh$(5)
24288 PRINT
24294 FOR I=1 TO Row
24300 PRINT I;TAB(6);Name$(I);
24306 IF Uc(I,1) THEN PRINT TAB(26);DROUND(Uc(I,1),6);
24312 IF Uc(I,2) THEN PRINT TAB(36);DROUND(Uc(I,2),3);
24318 IF Uc(I,3) THEN PRINT TAB(46);DROUND(Uc(I,3),6);
24324 IF Uc(I,4) THEN PRINT TAB(58);DROUND(Uc(I,4),3);
24330 IF Uc(I,5) THEN PRINT TAB(70);DROUND(Uc(I,5),6);
24336 PRINT
24342 NEXT I
24348 IF P=2 THEN PRINT USING "0/"
24354 Dfile=(Row)0
24360 Df_xname$=Cname$(Cnum(1)-2)
24366 Df_yname$=Cname$(Cnum(3)-2)
24372 NEXT P
24378 PRINTER IS CRT
24384 DISP "PRESS"&FNCT0&"WHEN READY TO CONTINUE..."
24390 PAUSE
24396 Exit_sub:File$=File0$
24402 SUBEXIT
24408 !
24414 Search_colnames: ! Search for column-name strings in the datafile.
24420 Nsux=1
24426 Row$=U$
24432 GOTO 24486
24438 IF LEN(U$)>75 THEN ! Evidently not stored with a printer-width of 80
24444 PRINT USING "18/,K,3/";FNCenter$(FNH$(File0$0) IS NOT A COMPATIBLE VISICALC /PF FILE")
24450 PRINT USING "K,/";"PLEASE, "&FNUn$("PLEASE")&", "&FNH$("PLEASE")&", "&FNB1$("PLEASE")&" TRY TO REMEMBER NOT TO STORE YOUR /
PF FILES"
24456 PRINT USING "K,4/";"WHILE THE PRINTER-WIDTH IS DEFINED AS ANYTHING BESIDES 80 CHARACTERS."
24462 Clunk
24468 WAIT 7
24474 GOTO Convert_vc
24480 END IF

```

```

24486 IF LEN(U$)=72 THEN ! Length of 1st string is 72 only if the real row
24492 LOOP ! is longer than 72 characters.
24498 ENTER @P1;U$
24504 IF LEN(U$)>72 THEN ! Sorry, really only N*8 rows if still =72
24510 Nsux=1+Nsux
24516 Row$=Row$&U$[4]! The operating system adds 3 space-characters after
24522 END IF
24528 EXIT IF LEN(U$)<75 ! each 72 characters in the row.
24534 END LOOP
24540 END IF
24546 Ncols=INT(LEN(Row$)/9) ! Assume a column-width of 9 characters.
24552 !
24558 ON ERROR GOTO No_colnames ! Look for a row-string of repeating-equals
24564 L1$="" ! signs extending across the whole file-width.
24570 L2$="" ! This row separates the column-names from the
24576 LOOP ! actual data.
24582 Row$=""
24588 FOR I=1 TO Nsux
24594 ENTER @P1;U$
24600 Row$=Row$&U$[1+3*(I-1)]
24606 NEXT I
24612 EXIT IF POS(Row$,RPT$("=",9*Ncols))
24618 L2$=L1$
24624 L1$=Row$
24630 END LOOP
24636 OFF ERROR
24642 !
24648 ! extract column-names from the 2 rows above the "====..." row.
24654 Num_datacolumns=Ncols-2
24660 IF Num_datacolumns<2 THEN
24666 Bad_input("FILE CONTAINS LESS THAN TWO DATA-COLUMNS")
24672 GOTO Convert_vc
24678 END IF
24684 PRINT USING "K,7X,K,#";CHR$(132)&"COLUMN"&CHR$(128),CHR$(132)&"COLUMN-NAME"&CHR$(128)
24690 PRINT USING "16X,K,7X,K,/" ;CHR$(132)&"COLUMN"&CHR$(128),CHR$(132)&"COLUMN-NAME"&CHR$(128)
24696 FOR I=1 TO Num_datacolumns
24702 ON ERROR GOTO 24714
24708 Top_cname$(I)=TRIM$(L2$[9*(I+1)+1,9*(I+2)])
24714 ON ERROR GOTO 24732
24720 Bottom_cname$(I)=TRIM$(L1$[9*(I+1)+1,9*(I+2)])
24726 Cname$(I)=Top_cname$(I)&" "&Bottom_cname$(I)
24732 OFF ERROR
24738 NEXT I
24744 !
24750 IF Cplot THEN ! Locate the correct columns if possible.
24756 MAT Cplot_col= (0)
24762 Ccol=0
24768 FOR I=1 TO Num_datacolumns
24774 MAT Tc= (0)
24780 T$=TRIM$(UPC$(Cname$(I)))
24786 IF POS(T$,"207") AND POS(T$,"235") AND NOT POS(T$,"ERR") AND NOT POS(T$,"X") AND NOT POS(T$,"PERCENT") THEN Tc(1)=1
24792 IF POS(T$,"206") AND POS(T$,"238") AND NOT POS(T$,"ERR") AND NOT POS(T$,"X") AND NOT POS(T$,"PERCENT") THEN Tc(3)=1
24798 IF POS(T$,"207") AND POS(T$,"235") AND (POS(T$,"ERR") OR POS(T$,"X") OR POS(T$,"PERCENT")) THEN Tc(2)=1
24804 IF POS(T$,"206") AND POS(T$,"238") AND (POS(T$,"ERR") OR POS(T$,"X") OR POS(T$,"PERCENT")) THEN Tc(4)=1
24810 IF POS(T$,"6/") AND POS(T$,"8") AND POS(T$,"7/") AND POS(T$,"5") AND POS(T$,"RHO") THEN Cplot_col(5)=I
24816 FOR J=1 TO 4
24822 IF Tc(J) AND NOT POS(T$,"MA") AND NOT POS(T$,"AGE") AND NOT POS(T$,"M.Y.") THEN Cplot_col(J)=I
24828 NEXT J
24834 NEXT I
24840 IF Cplot_col(1)*Cplot_col(2)*Cplot_col(3)*Cplot_col(4)*Cplot_col(5) THEN

```

```

24846 DATA "207/235", "7/5 ERROR", "206/238", "6/8 ERROR", "RHD"
24852 RESTORE 24846
24858 READ Cpl$(*)
24864 FOR I=1 TO 5
24870 IF Cplot_col(I)<=24 THEN C$(I)=CHR$(Cplot_col(I)+66)
24876 IF Cplot_col(I)>24 THEN C$(I)="A"&CHR$(Cplot_col(I)+40)
24882 PRINT TAB(3);C$(I);TAB(14);Cpl$(I)
24888 NEXT I
24894 Ccol=1
24900 END IF
24906 END IF
24912 !
24918 ! Printout the column-names in two vertical columns to avoid scrolling
24924 IF NOT Ccol THEN
24930 Halfcols=INT(Num_datacolumns/2)
24936 FOR I=1 TO Halfcols
24942 PRINT TAB(3);FNCol$(I);TAB(14);Cname$(I);TAB(43);FNCol$(I+Halfcols);TAB(54);Cname$(I+Halfcols)
24948 NEXT I
24954 IF Num_datacolumns>2*Halfcols THEN PRINT TAB(3);FNCol$(Num_datacolumns);TAB(14);Cname$(Num_datacolumns)
24960 END IF
24966 RETURN
24972 !
24978 !
24984 Badfile=Badcount+1+Badcount ! Assume that either the case or the drive-
24990 IF Badcount<7 THEN ! specifier is wrong, & keep trying until all
24996 ! permutations of wrong drives/case are tried.
25002 IF Badcount=3 THEN File$=UPC$(File$)
25008 IF Badcount=5 THEN File$=LWC$(File$)
25014 Drive_string$(13,13)=VAL$(HOT (VAL(Drive_string$(13,13))))
25020 GOTO Recoup
25026 END IF
25032 DISP USING "12X,K";CHR$(131)&" **** CAN'T ACCESS FILE "&Q$&File0$&Q$&" **** "&CHR$(128)
25038 File$=""
25044 Clunk
25050 WAIT 3
25056 GOTO Convert_vc
25062 !
25068 Badspec=Bad_input("INVALID FILE OR MASS-STORAGE SPECIFIER")
25074 File$=""
25080 GOTO Whichfile
25086 !
25092 No_colnames=Clear
25098 OFF ERROR
25104 PRINT USING "6/,K,/,K,/" ;FNH$("CAN'T FIND REPEATING-EQUALS LINE (=====...) THAT MUST UNDERLIE THE "),FNCenter$(FNH$(" COLUMN-NAMES "))
25110 Clunk
25116 WAIT 5
25122 Clear
25128 GOTO Convert_vc
25134 !
25140 SUBEND
25146 !
25152 !
25158 Help8=SUB Help8
25164 OPTION BASE 1
25170 DIM S$(11)[160]
25176 DATA*4: LOAD DATAFILE Bring data from a VISICALC file on either drive into memory. The VISICALC file must conform to the
following format: 1) It"
25182 DATA "must be a /PF file, 2) the column-width must be 9 characters, 3) sample-names must appear in the first two columns,
4) a name for the file"

```

```

25188 DATA"should occupy the first row of the file, 5) the column names must appear above the numeric data and be separated from the data by a"
25194 DATA"directly-underlying row of repeating equals- characters (=), and (6) the file must have been stored with a printer-width of 80, NOT 132."
25200 DATA "So if you printed the file out on the printer with compressed type, be sure and restore the printer-width to 80 before storing the file."
25206 DATA" The column names can occupy either 1 or 2 rows.$Q: CATALOG Dru #0 Display a catalog of the files on the disk in the left-hand drive.$"
25212 DATA"1: CATALOG Dru #1 Display a catalog of the files on the disk in the right-hand drive.$"
25218 DATA "6: PRINT DATA Use the printer, rather than the CRT to print out the listing of the data obtained from the LOAD DATAFILE operation.$"
25224 DATA"7: DISPLAY DATA Use the CRT to show the data obtained from the LOAD DATAFILE operation, rather than printing it out on the printer.$"
25230 DATA"8: SCRATCH FILE Delete data from any data-file currently in memory (but don't delete the data from the disk). Any keyboard- entered data will"
25236 DATA "then be temporarily stored in memory as a set of datafile-data, and so can be re-used without typing the data in again.$"
25242 DATA"9: ESCAPE Leave this function.
25248 !
25254 READ S$(*)
25260 CALL Helpscreen(S$(*),21)
25266 SUBEND
25272 !
25278 !
25284 Bl=DEF FNBl$(S$)
25290 RETURN CHR$(131)&" "&S&" "&CHR$(128)
25296 FNEND
25302 !
25308 Ct=DEF FNct$
25314 RETURN " "&FNct$("&CONTINUE")&" "
25320 FNEND
25326 !
25332 !
25338 Helpscreen:SUB Helpscreen(Input_string$(*),Indent)
25344 ! prints out a message on the CRT with linefeeds at appropriate breaks
25350 ! between words. Also inserts one linefeed for each $ symbol, and in-
25356 ! dents each line after a linefeed by Indent spaces. Tricky, huh?
25362 !
25368 DIM Largestring$(2400),Linestring$(80),Substring$(2400),Revstring$(80)
25374 !
25380 Clear
25386 OFF KEY
25392 OFF KBD
25398 OFF KNOB
25404 PRINTER IS CRT
25410 GRAPHICS OFF
25416 Largestring$=""
25422 ! build a single large string out of the input-string array
25428 FOR I=1 TO SIZE(Input_string$,1)
25434 Largestring$=Largestring$&TRIM$(Input_string$(I))&" "
25440 NEXT I
25446 Maxlen=80
25452 !
25458 PRINT USING " 16X,K,/:":FNH$("&FUNCTIONS OF KEYS DEFINED DURING LAST DISPLAY: ")
25464 PRINT USING "K,/:":FNUn$("&KEY")&" "&FNUn$("&LABEL")&" "&FNUn$("&FUNCTION")
25470 Nlines=5 ! starting-line of HELP minus one.
25476 Firstscreen=1 ! first screen of this HELP?
25482 Firstline=1 ! Indicates not an indented line
25488 !
25494 LOOP

```

```

25500 EXIT IF LEN(TRIM$(Largestring$))=0
25506 Maxlen=80-Indent*(Firstline=0) ! Max. permissible length, not
25512 Maxlen=MIN(Maxlen,LEN(Largestring$)) ! including indentation.
25518 Lf=POS(Largestring$,"$")
25524 IF Lf=1 THEN ! If first character is a linefeed token, then
25530 PRINT ! print a blank line & strip the token.
25536 Nlines=1+Nlines ! Increment line# counter.
25542 Firstline=1 ! Next line not to be indented.
25548 IF Largestring$=" $" THEN Done
25554 Largestring$=Largestring$[2] ! Strip linefeed token.
25560 ELSE ! First character not a linefeed token.
25566 Next_lf=POS(Largestring$,"$") ! Position of next linefeed token.
25572 IF Next_lf THEN ! Substring to next linefeed token.
25578 Substring$=Largestring$[1,Next_lf-1]
25584 ELSE
25590 Substring$=Largestring$
25596 END IF
25602 IF LEN(Substring$)<Maxlen THEN ! Don't need to find wordbreak.
25608 IF Indent AND NOT Firstline THEN PRINT TAB(Indent); ! Indent line.
25614 PRINT TRIM$(Substring$)
25620 Firstline=0
25626 Nlines=1+Nlines
25632 ! Strip last printed line from total string.
25638 IF LEN(Substring$)>=LEN(Largestring$) THEN Largestring$=Largestring$[1+LEN(Substring$)]
25644 ELSE
25650 Linestring$=Largestring$[1,Maxlen] ! Fragment that can fit on line.
25656 Revstring$=REUS$(Linestring$) ! Reverse-string.
25662 REPEAT ! Don't include any linefeed tokens
25668 Lf=POS(Revstring$,"$")
25674 IF Lf THEN Revstring$=Revstring$[2]
25680 UNTIL Lf=0
25686 S=POS(Revstring$," ") ! Find wordbreak position
25692 IF S THEN Revstring$=Revstring$[1+S] ! Goto wordbreak position.
25698 Lrev=LEN(Revstring$)
25704 IF Indent AND NOT Firstline THEN PRINT TAB(Indent); ! Indent line.
25710 PRINT TRIM$(REUS$(Revstring$))
25716 Nlines=1+Nlines ! Increment line#-counter
25722 IF LEN(Largestring$)=Lrev THEN Largestring$=Largestring$[Lrev+1]
25728 Firstline=0 ! Next line will be indented.
25734 END IF
25740 IF Nlines>=16 THEN ! Pause for user to read.
25746 DISP "Press "&FNCT98;" for more HELP...";
25752 IF NOT Firstscreen THEN
25758 DISP RPT$(CHR$(8),3)&" , Roll screen with arrows"
25764 ELSE
25770 DISP
25776 END IF
25782 Firstscreen=0
25788 Nlines=0
25794 PAUSE
25800 END IF
25806 END IF
25812 END LOOP
25818 !
25824 Done=DISP "Roll screen with arrows, press "&FNCT98;" to return to main program..."
25830 PAUSE
25836 SUBEND
25842 !

```

```

25848 !
25854 Students_t=DEF FNStudents_t(Nu)
25860 ! Student's-t approximation (from Andy Turek for Nu)=1, KRL for less)
25866 IF Nu=0 THEN Nu=1.E-9
25872 SELECT Nu
25878 CASE 1
25884 RETURN 12.7
25890 CASE >1
25896 W=Mu-.85
25902 RETURN 1.96-.00554568/SQR(W)+2.4059333/W+.6711777/W^2-.3831214/W^3
25908 CASE <1
25914 RETURN 1/(-.00765*Mu+.11085*Mu^2-.02446*Mu^3)
25920 END SELECT
25926 FNEND
25932 Ast=DEF FNAst$
25938 RETURN "(enter "&FNAst$("*"&" to escape)"
25944 FNEND
25950 !
25956 !
25962 Polycalc=SUB Polycalc(X(*),Y_in(*),Coef(*),Order,N)
25968 OPTION BASE 1
25974 ! Do actual calculations for a least-squares polynomial regression
25980 DIM Ka(5,200),Kb(5,5),Kd(5,200),Poly_array(200,5),Kc(5,5),Y(200,1)
25986 REDIM Ka(Order+1,N),Kb(Order+1,Order+1),Kc(Order+1,Order+1),Kd(Order+1,N),Poly_array(N,Order+1),Y(N,1)
25992 ! Create Power-matrix from X-values
25998 FOR I=1 TO N
26004 Y(I,1)=Y_in(I)
26010 FOR J=1 TO Order+1
26016 Poly_array(I,J)=K(I)^(J-1)
26022 NEXT J
26028 NEXT I
26034 !
26040 ! Solve matrix equations for best-fit polynomial coefficients
26046 MAT Ka= TRN(Poly_array)
26052 MAT Kb= Ka*Poly_array
26058 MAT Kc= INV(Kb)
26064 MAT Kd= Kc*Ka
26070 MAT Coef= Kd*Y
26076 SUBEND
26082 !
26088 Clear=SUB Clear
26094 OUTPUT KBD;CHR$(255)&CHR$(75);
26100 SUBEND
26106 !
26112 !
26118 Pbgrowth=SUB Pbgrowth
26124 ! Draw a single-stage Pb-isotope growth-curve if appropriate.
26130 OPTION BASE 1
26136 COM /Plotbox1/ Xmin,Xmax,Ymin,Ymax,Linx1,Linx2,Liny1,Liny2,Xspred,Yspred,Xloc,Yloc,Xm,Ym,Hx$,Hy$,Hx0$,Hy0$,Double
26142 COM /Plotbox1a/ Lx1,Lx2,Ly1,Ly2,Offset
26148 COM /Plotbox2/ Hxr$,Hyr$,Plotdev(*),Pr(*),Plotter,Plotter_dump,Ar,Tht,Lht,Ticksizex1,Y1,Xend,Yend,Speed$,Crt,Hardplot,Cpen,X
gu,Ygu
26154 COM /Plotbox3/ Df_xname$,Df_yname$,Dfile,Stack
26160 COM /Growth/ Lambda(0=2),RO(0=2),P(0=2),E(0=2)
26166 DIM Z(0=2),Y$(2)[7],Input$[80],Nu(4)
26172 Hx$=FNZcon$(0,Hx$)
26178 Hy$=FNZcon$(0,Hy$)
26184 IF POS(Hx$,"206") AND POS(Hx$,"204") AND POS(Hy$,"207") AND POS(Hy$,"204") THEN Pdtype=1
26190 IF POS(Hx$,"206") AND POS(Hx$,"204") AND POS(Hy$,"208") AND POS(Hy$,"204") THEN Pdtype=2
26196 IF NOT Pdtype THEN SUBEXIT

```

```

26202 !
26208 DATA .155125E-3, .98485E-3, .049475E-3, 11.152, 12.998, 31.230, 1, 137.88, .26455, 3700, 9.74, "207/204", "208/204"
26214 RESTORE 26208
26220 READ Lambda(*), RO(*), Z(*), TO, Mu0, Y>(* )
26226 !
26232 Clear
26238 Alpha
26244 PRINT TABXY(1,11); "DO YOU WANT TO INCLUDE A SINGLE-STAGE GROWTH-CURVE IN THE PLOT?"
26250 PRINT USING "3/,K,2/"; "press "&FNH$("k0")&" for a Stacey-Kramers growth-curve, "&FNH$("k1")&" for some other growth-curve,"
26256 PRINT "press "&FNH$("k9")&" to decline growth-curve."
26262 ON KEY 0 LABEL "STACEY-KRAMERS" GOTO S_k
26268 ON KEY 1 LABEL " OTHER" GOTO Other
26274 ON KEY 9 LABEL " NO CURVE" GOTO 26892
26280 GOTO 26280
26286 !
26292 Other=OFF KEY
26298 Clear
26304 PRINT TABXY(1,10); "ENTER 4 VALUES TO DEFINE A SINGLE-STAGE Pb-ISOTOPE GROWTH CURVE:"
26310 PRINT TABXY(1,12); "206/204 at start of growth, "&Y$("Pbtype")&" at start of growth,"
26316 PRINT TABXY(1,13); "Time before present to start growth, and Mu (238/204) of source..."
26322 PRINT TABXY(1,15); "enter "&FNH$("*")&" to escape)"
26328 Retrieve(Input$,Nu(*),Minputs)
26334 IF TRIM$(Input$)="*" THEN Pbgrowth
26340 IF Minputs(>)4 THEN
26346 Bad_input("NEED EXACTLY 4 NUMERIC VALUES FOR RESPONSE")
26352 GOTO Other
26358 END IF
26364 IF Nu(1)<=0 OR (Pbtype=1 AND Nu(2)<5) OR (Pbtype=2 AND Nu(2)<20) OR Nu(3)>6000 OR Nu(4)<=0 THEN
26370 Bad_input("INVALID INPUT -- CHECK NUMBERS & TRY AGAIN.")
26376 GOTO Other
26382 END IF
26388 RO(0)=Nu(1)
26394 RO(Pbtype)=Nu(2)
26400 TO=Nu(3)
26406 Mu0=Nu(4)
26412 IF Pbtype=2 THEN
26418 Clear
26424 INPUT "Enter the present-day 232/238 of the source...",Thu
26430 IF Thu<=0 THEN SUBEXIT
26436 Z(2)=1/Thu
26442 END IF
26448 !
26454 S_k=OFF KEY
26460 Draw_growthcurv=Clear
26466 PEN Cpen
26472 FOR I=0 TO 2
26478 E(I)=EXP(Lambda(I)*TO)
26484 P(I)=Mu0/Z(I)
26490 NEXT I
26496 Age_min=0
26502 Age_max=FNpbt(Xmin*(Xmin=Alpha0)+Alpha0*(Xmin<Alpha0),0)
26508 Tick(Age_max-Age_min,Tik)
26514 K=8*Tik
26520 Phi=Tht*(1-Hardplot/5)
26526 D=0
26532 Pht=Tht/2.5
26538 C=10*FNZ(TO)
26544 FOR Age_max=C*(1+INT(TO/C)) TO Age_min STEP -Tik
26550 IF FNpbr(Age_max,0)>Xmin AND FNpbr(Age_max,Pbtype)>Ymin THEN 26560
26556 NEXT Age_max

```



```

26562 ! Avoid crowding of tick-labels
26568 Curvex_min=FNpbr(MIN(Age_max,T0),0)
26574 Curvex_max=FNpbr(MAX(Age_min,0),0)
26580 F=(Curvex_max-Curvex_min)/Kspread
26586 IF Pdtype=1 THEN Tik=Tik*(1+(F<.5)*2*(F<.25))
26592 !
26598 Graph
26604 FOR Dx=0 TO Double
26610 IF Hardplot AND Dx THEN
26616 VIEWPORT Lx1,Lx2,Ly1,Ly2
26622 WINDOW Xmin,Xmax,Ymin,Ymax
26628 END IF
26634 CLIP Xmin,Xmax,Ymin,Ymax
26640 O=0
26646 FOR T=Age_max+2*Tik TO Age_min STEP -Tik/8 ! Draw growth-curve
26652 X=FNpbr(T,0) ! 206/204
26658 Y=FNpbr(T,Pdtype) ! 207/204 or 208/204
26664 IF (X>Xmin) AND (Y>Ymin) AND T<=T0 THEN
26670 DRAW X,Y
26676 ELSE
26682 MOVE X,Y
26688 END IF
26694 !
26700 IF DROUND(T/Tik,9)=INT(T/Tik) AND T<=T0 THEN
26706 IF T=Age_min THEN
26712 !
26718 ! ----- growth-curve ticks & tick-labels -----
26724 !
26730 LORG 5
26736 IF Hardplot THEN IMOVE +FNCwidth(Pht,Xmm,0,1,0)/15,-FNCheight(Pht,0,Ymm,0)/15
26742 CSIZE Pht,.8
26748 LABEL "0"
26754 IF 0 THEN
26760 ! Check if tick-labels lie within plot-box
26766 L=Pht*(1.5+INT(LGT(T+T=0)))
26772 Nchars_tick=FNZ(T)+1
26778 IF Hardplot THEN Length_tlabel=(1+Nchars_tick)*FNCwidth(Pht,Xmm,0,Ar,0) ! in UU
26784 IF CrT THEN Length_tlabel=(1+Nchars_tick)*FNCwidth(Pht,0,Kloc,Ar,1) ! in UU's
26790 IF Hardplot THEN Height_tlabel=FNCheight(Pht,0,Ymm,0)*1.2
26796 IF CrT THEN Height_tlabel=Pht/Yloc*11/15*1.2 ! in GU's,mm
26802 X_label=X-Length_tlabel
26808 Y_label=Y+Height_tlabel
26814 IF X_label>Xmin AND Y_label<Ymax THEN
26820 MOVE X-Pht*Ar/Kloc/7.,Y
26826 LORG 7
26832 CSIZE Pht,Ar
26838 LABEL FNZcon$(T)
26844 END IF
26850 END IF
26856 MOVE X,Y
26862 END IF
26868 D=NOT (D)
26874 END IF
26880 NEXT T
26886 NEXT Dx
26892 SUBEND
26898 !
26904 Pbr=DEF FNpbr(T,I) ! Pb-isotope ratio from age, assuming single-stage
26910 COM /Growth/ Lambda(*),RO(*),P(*),E(*)
26916 RETURN RO(I)+P(I)*(E(I)-EXP(Lambda(I)*T))

```

```

26922 FNEND
26928 !
26934 Pbt:=DEF FNPbt(R,I) ! Age from Pb-isotope ratio, assuming single-stage
26940 COM /Growth/ Lambda(*),RO(*),P(*),E(*)
26946 RETURN LOG(E<I>-(R-RO<I>)/P<I>)/Lambda<I>
26952 FNEND
26958 !
26964 !
26970 Err_env:=SUB Err_env<Slope,Slope_err,Inter,Inter_err,Xbar,Xmin,Xmax,Ymin,Ymax>
26976 ! Plot error-envelope for a least-squares line
26982 OFF KEY
26988 OFF KBD
26994 FOR Arn=-1 TO 1 STEP 2
27000 Y=1.E-99
27006 X=1.E-99
27012 Started=0
27018 WHILE X<Xmax AND Y<Ymax
27024 FOR X=Xmin TO Xmax STEP (Xmax-Xmin)/200
27030 Ysqr_incr=Inter_err^2+Slope_err^2*X*(X-2*Xbar)
27036 Ysqr_incr=Ysqr_incr*(Ysqr_incr>0)
27042 IF Ysqr_incr THEN
27048 Y=Slope*X+Inter+Arn*SQR(Ysqr_incr)
27054 IF Y>Ymin AND X>Xmin THEN Started=1
27060 IF Started THEN
27066 DRAW X,Y
27072 ELSE
27078 MOVE X,Y
27084 END IF
27090 END IF
27096 NEXT X
27102 END WHILE
27108 NEXT Arn
27114 PEN 0
27120 SUBEND
27126 Clunk:=SUB Clunk ! Clunky sound to indicate error
27132! BEEP 250,.1 ! for quiet environments
27138! BEEP 100,.2
27144 FOR J=0 TO 2000 STEP 200
27150 FOR I=1 TO 1 ! Better for a noisy environment
27156 BEEP 3500-J,.001
27162 NEXT I
27168 NEXT J
27174 SUBEND

```