UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Plotting and Regression Programs for Isotope Geochemists, for Use with HP-86/87 Microcomputers

by

K. R. Ludwig

Open-File Report 83-849

1983

TABLE OF CONTENTS

INTRODUCTION

This package of programs includes three plotting-regression programs (ISOPLOT, CPLOT, and PBPLOT), and a short program for conversion of VISICALC data-files to simple numeric data-files (NUMVERT). They are intended primarily for geochemists and geochronologists, though ISOPLOT can be used for a variety of X-Y plotting tasks. The programs are self-guiding, insensitive to user-errors, and can produce anything from rough, very rapidly-created plots to camera-ready, publication-quality figures.

ISOPLOT, a general X-Y plotting/regression program, is especially useful for isochron plots and age-calculations. ISOPLOT also permits polynomial regressions of X-Y data. CPLOT plots U-Pb isotope data on the familiar concordia diagram ($^{206}Pb/^{238}U$ versus $^{207}Pb/^{235}U$), calculates the best-fit line to an array of data, and calculates the concordia-intercepts and intercept errors of this line. PBPLOT plots either 207/204 or 208/204 versus 206/204, constructs a single-stage Pb-growth curve, and calculates the intercepts and errors of a best-fit line with the growth curve.

The required equipment are an HP-87 or HP-86 microcomputer with at least 128 kbytes of RAM, a disk drive, and the Plotter, Matrix, and Advanced Programming ROMs (the latter two ROMs can be deleted at the cost of some program modification). For permanent copies, an HP-7470A plotter and/or an HP-82905B or HP-2673 printer are necessary. If data-file input of data is desired, the Hewlett-Packard VISICALC PLUS program is strongly recommended, as all programs are designed to interact with VISICALC files (though simple numeric files can also be accessed).

ISOPLOT

ISOPLOT is the most straightforward of the three plotting programs, so its characteristics can be used as examples for the other plotting programs. The program lising for ISOPLOT is given in Appendix 1.

Conventions:

Queries displayed on the CRT will often request several values, separated by commas. You must enter one value for each parameter requested that is not enclosed in brackets. Enter by typing in the values (separated by commas), and pressing the ENDLINE key. Parameters displayed enclosed in square brackets are optional, and need be entered only if relevant, or if you do not wish default values for these parameters to be used.

If you accidentally touch one of the keyboard keys during program execution (but not while the program is waiting for you to respond to a query), the computer will beep and stop execution of the program. To restore operation, simply press the CONT key. To view any CRT-graphics during a pause in the program, press the A/G key, which will toggle between the alphanumeric and graphics CRT display.

For most of the "select-option" screens of the program, functions of the 14 special-function keys at the top of the keyboard are labeled at the bottom of the CRT - the lower label corresponding to the unshifted keys (k1-k7), and the upper to the shifted keys (k8-k14). To obtain more information on the functions of the keys, there is usually a key defined as "HELP". When you press the "help" key, more detailed information about your options will appear on the CRT.

Setting Up the Plot Format:

Load the program (insert the disk into the disk-drive, type LOAD ISOPLOT:D700 (assuming the disk is drive 0), and press the ENDLINE key). The CRT will clear, and display the definitions of the 14 special-function keys at the bottom of the CRT. When you press one of these keys, the computer will respond immediately in a manner suggested by the key definitions. For most plotting operations, you can begin plotting after only a few keystrokes (for example, just pressing the START key if all you want is a straightforward CRT plot), and using the HELP keys when necessary. The following instructions need only be referred to if you wish detailed information about the operation and capabilities of the programs, or if you wish to plot data from a data file instead of by typing

in data from the keyboard.

The actual functions of the keys are as follows:

KEY#	KEY LABEL	FUNCTION
k1	PRINT-DAT	Use printer instead of CRT for printout.
k2	CRT-DAT	Use CRT instead of printer for printout (default).
k3	HARDPLOT	Use HP-7470A plotter for graphics.
k4	CRT PLOT	Use CRT for graphics (default).
k5	PEN 1	Use pen# 1 (HP-7470A) or solid-line pen (CRT) for plotting (default).
k6	PEN 2	Use pen# 2 (HP-7470A only) for plotting.
k7	START	Start input of data to define plot type, size, etc.
k8	PEN SPEED	Define pen-speed (HP-7470A only); default is the fastest speed.
k9	LDG ZEROES	Include a leading zero in tick labels of absolute values less than 1 and with a decimal point (e.g. 0.14 instead of .14). 0.14 instead of .14).
k10	STACK	Suppress X-axis label and X-axis tick labels so that plots sharing the same X-axis can be vertically stacked.
k11	HELP	Display a list of explanations of the functions of the keys for this display.
k12	NORM	Use normal-size plot-labels and symbols (default).
k13	SLIDE	Use "slide-size" plot-labels and symbols (somewhat larger than normal size; useful for plots that will be significantly reduced, or for plots to

be used as 35mm transparencies).

k14 DATAFILE — Bring data from a VISICALC or numeric data- file into memory.

Using the defined keys (note that the keys specific to the HP-7470A plotter are only defined once the HARDPLOT key has been pressed), select which plotter and output device are to be used (the CRT is the default device in both cases). If you are using the HP-7470A plotter, turn it on and insert the paper. Select the pen# and speed, if different from the default values. You can bring data from a data file into memory at this point (by pressing the DATAFILE key), or you can wait until the frame of the plot has been drawn. Press the START key (k7). The CRT will clear, then query:

X-AXIS NAME, Y-AXIS NAME?

Enter the names (axis-labels) for each axis, separated by a comma. The names cannot exceed 18 characters each. If you wish to include superscripts or subscripts in the axis names, use the ^ character to indicate that the characters following are to be written up 1/2 line, and use the \ character to indicate that the characters following are to be written down 1/2 line. Thus $^{206}Pb/^{204}Pb$ would be indicated by the characters ^206\Pb/^204\Pb. The next query will be:

X and Y LIMITS ([Xmin,] Xmax, [Ymin,] Ymax?

Enter either 2 or 4 values (the brackets indicate optional inputs). If the plot is to start at the origin (X-minimum and Y-minimum = 0), just enter the maximum limits for the plot-box X and Y values, separated by a comma. Otherwise, enter the X-minimum, X-maximum, Y-minimum, and Y-maximum values for the plot-box, separated by commas. The plot-box limits that will actually appear on the plot may be slightly larger than the ones you specified, so that they will fall on logical tick-locations. For example, if you specified X-axis limits of 2.13 to 15.74, the plot box would actually be drawn from 2.00 to 16.00, with a tick interval of 2.

If you chose the HP-7470A plotter instead of the CRT as the plotting device, the next query will be:

PLOT-SIZE (3-10)
(press ENDLINE for nonstandard dimensions/locations)?

The largest size that will fit on standard 8.5"X11" paper is 10, whereas a size of 3 will give a centered plot only a few inches across. Any plot defined with a plot-size of 3 to 10 will have the same, rectangular X-Y dimensions. If, however, you want your plot to be uncentered in the paper, or to have some arbitrary X-Y dimensions, press the ENDLINE key.

If you pressed ENDLINE in response to the PLOT-SIZE query, the CRT will then display:

PRESS 1 TO DEFINE THE PLOT-LIMITS FOR THE ENTIRE FIGURE-AREA,

PRESS 2 TO DEFINE LIMITS FOR THE PLOT-BOX ONLY

Press the 1 key if you wish to define the plot-limits in terms of the smallest box that will enclose the entire figure, including tick labels and axis labels. Press the 2 key if you wish to define the plot-limits in terms of only the plot-box itself - that is, the box (containing the axis-ticks) that is actually drawn. The CRT will then ask you to enter the minimum and maximum values of these plot limits, defined in millimeters from the lower-left corner of the actual plotting-paper. Maximum values that can be used are 257 mm along the X-direction (the long dimension of the paper), and 191 mm along the Y-direction (the short dimension of the paper). This feature is particularly useful for stacking plots with the same X-axis and X-axis limits.

After a few seconds of calculations, the computer will draw the plot-box, defined by the X and Y axes, draw the axis ticks, label the ticks, and label the axes. The display will clear, and a new set of definitions for the special-function keys will appear. I will refer to this point in the program as the "select plotting-symbol" location.

The functions of these keys are as follows:

KEY#	KEY LABEL	FUNCTION
k1	OPEN Ebox	Use open (unfilled) error-boxes as plotting symbols.
k2	OPEN Eell	Use open error-ellipses as plotting symbols.
k3	POLYGON	Use a polygon of arbitrary type as plotting symbols.
k4	ERRCROSS	Use error-crosses as plotting-symbols.
k6	YORKFIT	Calculate a regression line to the last set of plotted data (appears only after 2 or more points have been plotted).
k7	OPTIONS	Temporarily redefine the special-function keys to permit some additional functions, including changing the error-input format,dumping the CRT plot to the printer, storing the CRT plot on a disk, and fitting a polynomial curve to data. Details are given in a later section.
k8	SOLID Ebox	Same as OPEN Ebox, but solid (filled in).
k9	SOLID Eell	Same as OPEN Eell, but solid (filled in).
k10	LETTER	Draft a phrase at some arbitrary location on the plot.
k11	PEN 1	(Default) Select pen# 1 (HP-7470A) or the solid-line "pen" (CRT-plot).
k12	PEN 2	Select pen# 2 (HP-7470A plots only).
k12	ERASE PEN	Use the "erasing pen" (CRT plots only). "Erases" any previously-plotted line that it encounters - useful for "unplotting" symbols or lettered phrases.
k13	DATAFILE	Bring data from a VISICALC or

numeric da file into memory.

k14	REDRAW	Start defining and drawing a completely new plot. Better than pressing PAUSE and then RUN, because the program is much faster once it has gone through its functons at least once.
CTL D	---	Dump the CRT plot to an HP-7470A plotter. The dumped plot will be drawn using pen# 1 at the fastest pen-speed, at PLOTSIZE 8.
CTL S	---	Use a solid line instead of a dashed line for drawing Yorkfit lines.
+ O X * ...		Any upper-case keyboard-symbol can be used as a plotting symbols. Use the space-bar if you don't actually want the points plotted.
CTL H	[HELP]	Display a list of explanations of the functions of the keys for this display.

Plotting data-points:

Keep in mind that once you have reached the "select plotting-symbol" part of the program, you can always re-start the program at this location by pressing PAUSE, typing in CONT 1, and then pressing ENDLINE. This re-entry procedure can be very useful in recovering from errors, or just as an immediate re-start at the data-input part of the program.

Make sure that the current pen is the one that you wish to plot with. If not, select the appropriate pen by pressing either the PEN 1 (k11) or PEN 2 (k12) keys (HP-7470A plotter only). Then, press the appropriate key for the plotting-symbol that you wish to use. If you chose the POLYGON key (k3), the query,

```
#SIDES (- for star) [,OPEN/SOLID(0/1) [,SIZE(1-5)
                                [,ROTATION(0-360)]]]?
```

will appear. You must choose the number of sides (for example, 3 for a triangle, 10 or more for a circle), but

the other parameters are optional. If you enter the number of sides as a negative number, the plotting symbol will be a star-shaped polygon with the number points of the star equal to the #SIDES value. OPEN/SOLID defines whether the polygon is to be left open (0) or filled in (1), and has a default value of 0. SIZE defines the size of the polygon, with 1 being a very small polygon and 10 a very large one. The default SIZE is 2.5. ROTATION defines the angular orientation of the vertices of the polygon, in degrees counterclockwise from the horizontal. Thus a ROTATION of 45 for a 4-sided polygon will give an upright appearing square, and a value of 270 for a 3-sided polygon will give a downward pointing triangle. If a ROTATION value is not specified, even-sided polygons will be drawn with one side parallel to the X-axis, and odd-sided polygons with an upward-pointing vertex.

After answering the POLYGON query, or immediately after pressing any other of the plotting-symbol keys, the program will request you to enter the data-points by the query,

INPUT DATAFILE SET #s AS 1st SET, LAST SET-ASTERISK
(e.g. 2,12* or 2,12;15,18*)

#1: Xname, [%err,] Yname [,%err] [,err-corr.]
(ENDLINE when done)

The first line, incidentally, appears only if you have previously loaded a data file into memory. Also, "Xname" and "Yname" are just examples of labels for the axes of the plots; obviously, your axis-labels would be different.

To enter data from the keyboard, you may use any of three types of data input. You can simply type in the X-value and the Y-value of the data-point, separated by a comma, then press the ENDLINE key. Or, you can type in the data in the form X-value, %X-error, Y-value, %Y-error, ENDLINE. Or you can enter the full error and error correlation information in the form X-value, %X-error, Y-value, %Y-error, X-Y error correlation. Note that in the standard (default) error-entry mode, the errors must be entered at the 2-sigma (or 95% confidence) level and in percent. You can, however, choose to enter your errors at the 1-sigma level and/or in absolute rather than percent values, using the

OPTIONS key mentioned above and discussed below.

To enter data from a data file (assuming that you have already loaded data from a data file into memory using the DATAFILE key), the following input format is used. To enter a single set, enter the number of the data-set to be plotted, followed by an asterisk; for example, 17*. To enter several data sets in sequential order, enter the first and last sets of the series, followed by an asterisk; for example, 17,32* for sets 17 through 32, inclusive. To enter a mixture of different series and individual sets, separate broken series with a semicolon; for example, 2,5;11,13;15;20,22* for sets 2 through 5, 11 through 13, 15, and 20 through 22.

After entering the first data-point, continue the process (up to 300 points for plotting or polynomial regression, and up to 100 points for plotting with Yorkfit) until all of the data-points that you wish plotted with the particular symbol and pen chosen earlier have been entered. Then, to start plotting the data, press the ENDLINE key without typing in any data. If any data-points have X or Y values of zero, they will not be accepted (because the Yorkfit algorithm involves dividing by the X and Y values). Also, the internal consistency of the errors and error-correlations will be checked.

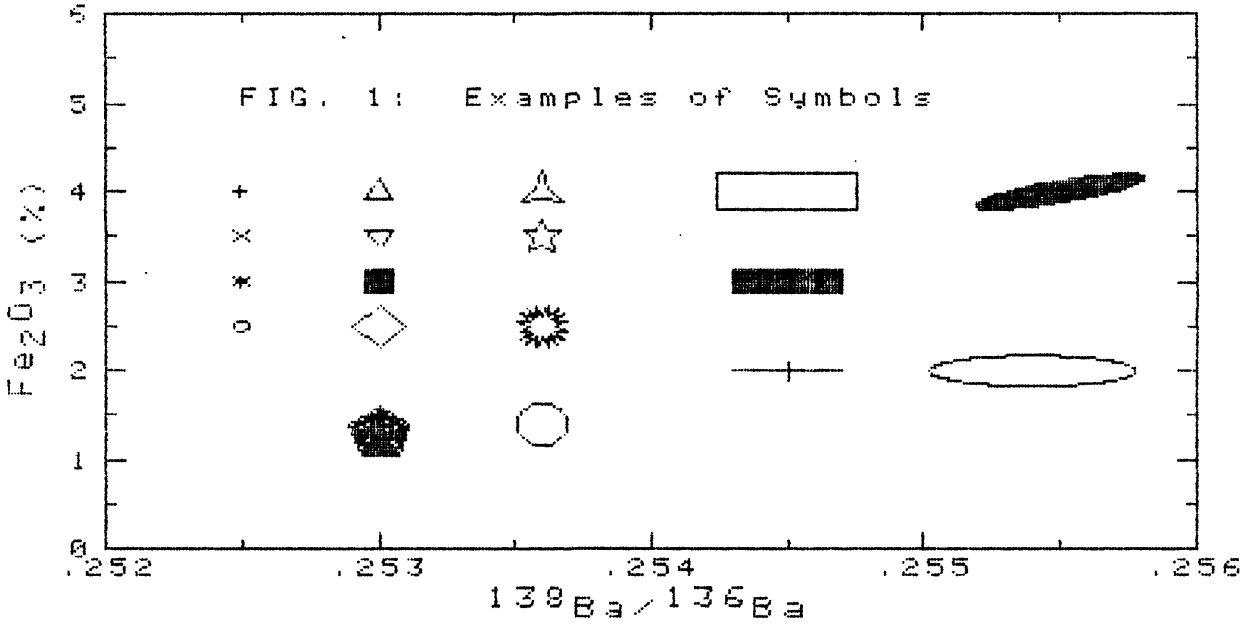
If you chose a plotting symbol that requires the errors of the data-points (for example, error-box, error-cross), you will be asked to provide blanket errors for each of the points for which no errors were assigned. Similarly, if the plotting symbol were an error-ellipse, errors and error correlations will be requested for each point with unassigned errors and/or error-correlations.

If any of the data points to be plotted were typed in from the keyboard, the values of all the entered points will be displayed, and the CRT will query,

SET TO BE CORRECTED? (ENDLINE IF OK)

If you wish to correct any of the input, enter the set# (as indicated on the CRT) of the incorrect data, then enter the correct values. Press ENDLINE when no more sets require correction. The data-points will then be plotted using the plotting symbol chosen earlier.

Examples of plots and plotting-symbols are shown in figs. 1 and 2.



After plotting the data-points, the CRT will display,

PRESS RETAIN TO INCLUDE THESE POINTS WITH OTHERS FOR A YORKFIT.

PRESS CONTINUE TO CONTINUE.

PRESS YORKFIT FOR A YORKFIT.

If you wish to add to the data-points that were just plotted with other data-points using a different plotting-symbol or pen color, and then calculate a best-fit line for all of these points, press the RETAIN special-function key (k6). If you wish to calculate a best-fit line for the last group of points (defined as the points numbered from 1 to the last-point plotted), press the YORKFIT special-function key (k5). Otherwise, press the CONTINUE special-function key (k7). Both the RETAIN and CONTINUE keys will return you to the "select

plotting-symbol" part of the program.

Yorkfit:

The YORKFIT procedure is a least-squares line-fitting method that adjusts for errors in both X and Y values, as well as correlations between those errors, and is based on the algorithm devised by York (1969). Errors of the slope and intercept of the best-fit line are calculated using the maximum likelihood approach given by Titterington and Halliday (1979). If you aren't sure what the correlations between the X and Y errors are, the relationship

$$Rho = (Ex^2+Ey^2-Ez^2)/(2ExEy)$$

where Ex, Ey, and Ez are the percent errors in X, Y, and Z, respectively, Z=Y/X, and Rho is the correlation between the X and Y errors. Rho can range from -1 to +1.

The best-fit line may be calculated in three ways (models), depending on the axis-labels and the scatter of the data. The first line-fit (Model-1) is always calculated using the original York (1969) algorithm, wherein all scatter of the data points about a straight line is assumed to be due only to the assigned analytical errors. For this model, the calculated 95%-confidence limit errors of the slope and intercept are propagated only from the assigned analytical errors and with a t-multiplier of 1.96, so long as the probability of fit is greater than 15%. If the probability is less than 15%, the actual scatter of the points and a t-multipler for N-2 degrees of freedom (N = number of data points) are used to estimate the errors. In the Yorkfit printout, the a priori errors refer to errors calculated from the assigned analytical errors alone, and thus are insensitive to the actual amount of scatter of the points. The including scatter errors, on the other hand, are calculated from the actual amount of scatter of the points from the best-fit line.

The probability that the actual amount of scatter about the best-fit line would be met or exceeded from the assigned analytical errors alone will be printed out with the Model-1 line-fit. If this calculated probability is low (say less than about 10%), then either the points scatter for some reason besides

analytical error alone, or you have underestimated the analytical error. If you suspect the latter, you may estimate the degree of analytical error underestimation by multiplying each data-point error by the square root of the M.S.W.D. value. If these increased errors (but the same error correlations) were used for the Yorkfit, the excess scatter would precisely disappear.

If the probablity calculated for the Model-1 Yorkfit is less than 20%, the best-fit line will be recalculated using one of two assumptions. First, if the axis-labels indicate that the plot is a conventional Rb-Sr, Sm-Nd, U-Pb, or Th-Pb isochron diagram, then all excess scatter will be assumed to be due to a normally-distributed variation in the initial Y-axis value. In this case, the points will be weighted according to their assigned analytical errors plus this unknown Y-axis error (calculated iteratively). The magnitude of the calculated initial Y-axis variation can be used as a guide to the reasonableness of the assumption. This is a Model-3 fit, and is similar to the Model-3 fit of McIntyre and others (1966), except that in ISOPLOT the analytical errors are also taken into account.

If the axis labels do not indicate that the plot is an isochron plot of the types mentioned above, then it is assumed that nothing is known about the cause of the scatter, and each of the points will be assigned equal weights and zero error-correlations (Model-2).

In both Model-2 and Model-3 fits, because the actual scatter of the points is used for calculating the slope and intercept errors, a t-multiplier for N-2 degrees of freedom is used for calculating the 95% confidence-limit errors.

If the plot is recognized as an isochron diagram of the usual Rb-Sr, Sm-Nd, U-Pb (both Alpha versus Mu and Beta versus Nu), Th-Pb, Pb-Pb (both 207/204 versus 206/204 and 207/206 versus 204/206) plots, the isochron age and errors will be calculated using the decay constants recommended by the I.U.G.S. (Steiger and Jäger, 1977). The program recognizes an isochron plot by looking for the appropriate isotope numbers in the X- and Y-axis labels.

After completing the Yorkfit regression and any isochron-age calculation, a dashed line corresponding to the best-fit line will be drawn on the plot. If more than one line is drawn on the plot, the dash pattern will be slightly different for each line. If you wish a solid line to be used instead, you must press CTL S during the select-plotting-symbol display. the CRT will clear, and the special-function keys will be defined as:

KEY#	KEY LABEL	FUNCTION
k1	RESTART	Construct a competely new plot.
k2	ADD POINTS	Add more points to the ones that were just used for a Yorkfit, with the intention of requesting a Yorkfit for this larger assemblage of points.
k3	PRINT PLOT	Dump the CRT plot onto the printer (appears only if the CRT is used as the plotting device).
k7	CONTINUE	Return to the "select plotting-symbol" display and keys (see above), for plotting of more data, lettering on the plot, etcetera.
k11	HELP	Display a list of explanations of the functions of the keys of this display.

To Draft a Phrase on the Plot:

From the "select plotting-symbol" display, press the LETTER key (k10). The CRT will clear, and display:

```
          (press ENDLINE to escape)
CHAR.-HEIGHT (tick labels were 3.2) [,SLANT(0-30)
                                    [,ROTATION]]?
```

Select a character size, using the information on the size of the tick-label characters as a guide. If you do not wish the characters of the lettered phrase to be slanted or rotated (not horizontal), just type in the character size and press ENDLINE. Otherwise, also type in the slant of the characters in degrees, and the angular rotation of the phrase (counterclockwise from

the X-direction), also in degrees.

If the CRT is the plotting device, you will be requested to enter the X and Y coordinates of the lower-left corner of the phrase.

If the HP-7470A plotter is the plotting device, the display will be:

POSITION PEN WITH PLOTTER CONTROLS, THEN PRESS CONT

Move the plotter pen, using the four arrow-marked buttons at the lower right of the HP-7470A plotter, to the desired location of the lower-left corner of the phrase to be lettered, and then press the CONT key. If you wish to start the lettering at a specific X-Y location, type in MOVE 20,42 (20 and 42 being the X and Y coordinates in this example), then press the CONT key.

For either plotting device (but for horizontal angles only), you can use the ^ and \ keys to indicate super- and sub-scripts in the same way as for the axis names, and you can use the +/- characters to obtain the superimposed plus-or-minus symbol (±).

The OPTIONS Key:

The OPTIONS key (k7), defined during the "select plotting-symbol" display permits you to redefine the format of the data-point errors, to dump a CRT plot to the printer, and to store a CRT plot on a flexible disk. When the OPTIONS key is pressed, the special-function keys are defined as:

KEY#	KEY LABEL	FUNCTION
1	PRINT PLOT	Dump the CRT graphics to an HP-82905B or HP-2673 printer.
2	ABS ERRORS	Permits entry of the data-point errors in absolute values, rather than in percent.
3	% ERRORS	(Default) Permits entry of the data point errors in percent, rather than as absolute values.

4	1-SIG ERRORS	Permits entry of the data-point errors at the 1-sigma level rather than 2-sigma.
5	2-SIG ERRORS	(Default) Permits entry of the data point errors at the 2-sigma level rather than 1-sigma.
6	STORE PLOT	Allows you to store the CRT graphics on a flexible disk for later retrieval and display, using the GLOAD command.
11	HELP	Display a list of explanations of the functions of the keys of this display.
14	POLYFIT	Do a polynomial regression (up to 5th-order) and draw the polynomial curve for the last block of data.

Retrieving Data from Numeric Data-Files:

Two types of data-file structures can be accessed by the ISOPLOT program. The simplest structure (and more rapid by far) is data stored as simple numeric arrays, plus simple string arrays for brief names for each data set. These data-file structures can be created from existing VISICALC data-files with the NUMVERT program, or quite simply by a user-written program. Each data-set, together with an 18-character name for the set, is stored on the flexible disk as an individual record which contains the 18-character name-string followed by a 45-element, full-precision numeric vector. Thus each record contains up to 45 different values (parameters), from which the user can select the X-Y data points, together with (optionally) the errors and error-correlations of the points.

The first query after pressing the DATAFILE special-function key is:

NAME OF DATA-FILE TO BE RETRIEVED? (to get a disk catalog, enter C for drive 0, c for drive 1).

Type in the file-name of the data-file, followed by the mass-storage specifier (for example, TRIALFILE:D701 or TRIALFILE.PLOTDISK), then press ENDLINE. The program automatically determines if the file is a numeric or VISICALC file. Then, assuming that the data file is a numeric file and not a VISICALC file, the CRT will query;

COLUMN#s FOR X, [,%Xerr,] Y [,%Yerr,] [,Err-Correl]"?
(press ENDLINE for column-name listing)

You must know which columns of the numeric array contain the desired X and Y values (and errors - error correlations) to answer this query. To obtain a listing of the column-names for the VISICALC file, just press the ENDLINE key, and after searching the disk for the column names, the list of column names will be printed out. There are up to 45 columns available. The response format is similar to that of the keyboard data-entry format, in that 2, 4, or 5 values can be entered, except that the inputs are column-numbers of the appropriate data, not the values themselves. For example, if no error-values were stored in the array, you might enter 6,19 to indicate that the X-values are in column 6 and the Y-values in column 19. Or, an input of 6,8,19,22,25 would indicate that the X and Y values are in columns 6 and 19 respectively, that the X-error and Y-error values are in columns 8 and 22 respectively, and the X-Y error-correlations are in column 25.

The data will then be retrieved from the disk and printed out either on the CRT or the printer, depending on which was chosen during the plot set-up part of the program. You can then plot this data by simply referring to the set numbers printed out with each data set.

Retrieving Data from VISICALC Data Files:

VISICALC is an extremely powerful and flexible, commercially-available program for creating, manipulating, displaying, and storing data. The program is available for many small computers, and is sold by Hewlett-Packard for the HP-86/87 as the VISICALC PLUS program. In brief, the program allows either numeric or alphanumeric data to be displayed as a 2-dimensional array of rows and columns of arbitrary size, and stored

as individual character-strings for each element in the array. The advantage of this data-format is the flexibility of display format and ease of manipulating the format (such as moving, adding, or deleting data-elements, rows, columns, or blocks). The disadvantage, compared to less-flexible formats, is that such data files are slower to load into the memory of the computer than simple numeric data-files, as the elements of the two-dimensional array are stored individually as strings, and must be converted to numeric values by the computer program.

Enter the file name of the VISICALC data-file when requested (see numeric data-file section). The CRT will locate the file, then query,

INPUT BEGINNING, ENDING ROW-NUMBERS CONTAINING SAMPLE-DATA?

You do not have to know precisely which row numbers of the VISICALC file contain the data of interest, but data access can be speeded up somewhat if no more rows are specified than are actually needed. For most VISICALC file formats, for example, the first 5 or 6 rows are taken up by titles and column-names, so that the beginning row-number is generally no less than 5 or 6. If you are very unsure as to the row numbers which include the data of interest, enter 1 as the beginning row-number and some safely large number (for example, 60 if you know that only 40 or so data-sets exist in the file) as the ending row-number.
PP The next query will be:

(enter ALL to convert all columns)

INPUT COLUMNS (e.g. C,G,AK...) FOR: X, [%Xerr] Y [,%Yerr] [,Rho(X-Y)]?

VISICALC file columns are indicated by alphabetic symbols - A through Z, followed by AA through AZ and BA through BK. Enter the letters for the columns containing the X and Y values, and their errors and error correlations, if available (Rho indicates X-Y error correlation). Again, you may enter two values only (X and Y columns, without errors), 4 values (X, X-error, Y, Y-error), or 5 values (X, X-error, Y, Y-error, X-Y error correlation).

Or, if you intend to make more than one plot from the data in this VISICALC file, where different columns will be used for the X and Y values for different lots, you can save time by entering ALL in response to the column-numbers query. A response of ALL commands the program to convert all of the VISICALC columns to numeric values at this time, thus avoiding a second conversion of the same file later during the plotting session. Once you have requested that all columns from a VISICALC file be converted, the query that appears when the DATAFILE key is pressed will be,

NAME OF DATA-FILE TO BE RETRIEVED?
(press ENDLINE for more data from TRIALFILE:D701)

assuming that all columns from VISICALC file TRIALFILE:D701 had previously been converted.

During loading of the VISICALC data-file into memory, the CRT will show the values of each data-element of interest as it is recognized (found to be in the appropriate column and row) and converted. For small VISICALC files, this process will take one or two minutes; for large files, up to 5-10 minutes.

If only some (i.e., not all) of the VISICALC columns were converted, The converted data will be immediately printed out on the CRT or printer, depending on which was selected during the plot set-up part of the program. The set numbers indicated on this printout will then be useable as input for the plotting segment of the program. If all of the VISICALC columns were converted, the CRT will first query which columns are to be used for the plotting data.

Note that VISICALC cells containing formulas will not be evaluated as numeric data by the plotting programs.

CPLOT

The main differences between CPLOT (see Appendix 2 for program listing) and ISOPLOT are that,

1) The X and Y axes are defined for the conventional concordia plot; 207/235 and 206/238, respectively;

2) CPLOT draws the concordia curve and age-ticks on the plot, and the plot-limits can be defined in terms of concordia ages;

3) CPLOT does not permit error-box or error-cross plotting-symbols, as the concordia-plot errors are generally so highly-correlated that such symbols would give a misleading impression of the actual error limits;

4) CPLOT permits only 2-sigma, percent errors to be used, but

5) Errors can be entered either in the conventional format (X-err, Y-err, err-correl.), or in a simplified format (206/238-error, 207/206-error) which is approximately valid for data with small common-Pb corrections.

6) CPLOT permits a "Model-4" algorithm for calculation of best-fit lines, based on a model of increasing scatter with discordance; and

7) CPLOT calculates the concordia intercepts and intercept errors of best-fit lines.

The first display after pressing the RUN key is the definition of the special-function keys, as follows:

KEY#	KEY LABEL	FUNCTION
k1	O CTICKS	Use a small circle to indicate concordia-curve age-ticks, and label the age-ticks with horizontal labels to the left of the ticks (default).
k2	CRT-PRINT	Use the CRT as the printing device (default).
k3	CRT-PLOT	Use the CRT as the plotting device (default).
k4	PEN 1	Use a solid white line (CRT) or pen #1 (HP-7470A) as the plotting pen (default).
k5	DATAFILE	Load data from either a VISICALC or numeric data-file into memory.
k6	NORM	Use normal-size labels and symbols (default).
k7	START	Start constructing the concordia-plot.
k8	\ CTICKS	Use a dash constructed perpendicular to the concordia curve to indicate age ticks, and label these ticks along the same line as the ticks but upwards and to the left.
k9	PRINTER	Use an HP-82905B or HP-2673 printer as the printing device.
k10	HARDPLOT	Use an HP-7470A plotter as the plotting device.
k11	PEN 2	Use pen #2 as the plotting pen (HP-7470A plotter only).
k12	PENSPEED	Select pen-speed of HP-7470A (default is fastest speed).
k13	SLIDE	Use slide-size (proportionally larger) labels and plotting-symbols.
k14	HELP	Display a list of explanations of the

functions of the keys of this display.

L --- Include leading zeroes for any tick label besides zero) of less than one (for example, 0.14 instead of .14).

After you have selected the printer, plotter, pen, pen-speed, letter-size, and concordia-tick style, press the START key (k7). Remember, of course, that you need only pay attention to these parameters if you wish them to be different than their default conditions. In other words, if you wished to construct a plot using the CRT only, with horizontal concordia-ticks, you would simply press the START key without any other input.

After pressing START, the CRT will query:

[MIN. AGE,] MAX. AGE (Ma)?
(press ENDLINE to define by 6/8-7/5 limits)

If you enter only 1 number in response (by typing in the number, then pressing the ENDLINE key), the plot will begin at the origin (zero age) and include the entered value as the maximum age, in millions of years, of the concordia curve. If you enter 2 numbers, separated by a comma, the first number will be the minimum concordia age of the curve, the second number the maximum concordia age. For example, if you entered "800,2950" in response to the above query, you would obtain a plot where 800 Ma (or slightly less, so that the axis ticks would not start at some value such as 0.511937845) were the youngest age included in the plot-box, and 2950 Ma were the oldest age.

To define the limits of the plot-box by the actual 207/235 and 206/238 minimum and maximum values, press the ENDLINE key in response to the age-limits query, and the CRT will query,

207/235, 206/238 LIMITS ([Min.,] Max., [Min.], Max.)?

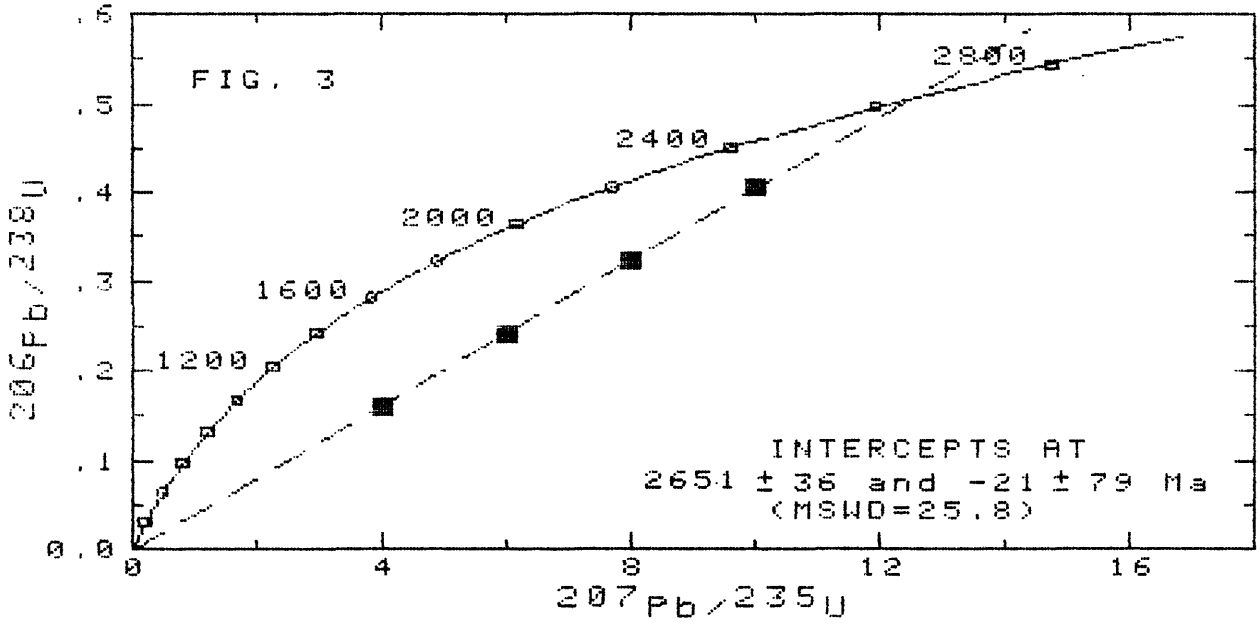
Enter two values separated by commas to define the maximum 207/235 and 206/238 limits of the plot-box, and to start the plot-box at the origin. Enter four values separated by commas to define the mimimum and maximum values of both the 207/235 and 206/238 axes.

Plotting and data-input are done in a similar way to the ISOPLOT program, with the following exceptions:

1) to dump a CRT plot to the HP-7470A plotter, press the key labled HPLOT DUMP (k3); and

2) to permit a data-point error-format using 206-Pb/238-U and 207-Pb/206-Pb errors instead of 206-Pb/238-U and 207-Pb/235-U errors plus their error correlations, press the key labeled ?ERRORS? (k13). The program can calculate approximately valid 207-Pb/235-U errors and the X and Y error-correlations from this input (206/238 and 207/206 errors only), provided that the data points were not calculated with large common-Pb corrections.

Examples of CPLOT plots are shown in figs. 3 and 4.



The Model-1 YORKFIT algorithm of CPLOT is identical to the algorithm used in ISOPLOT, and if the probability of the scatter for the Model-1 fit is sufficiently high (more than 20%), the intercepts and intercept-errors with the concordia curve of the best-fit line are immediately calculated. The algorithm used for calculation of the concordia-intercepts and intercept-errors is that of Ludwig (1980).

If the probability of the scatter for the Model-1 fit is less than 20%, you must choose whether the subsequent attempt at line fitting uses Model-2 assumptions (see ISOPLOT discussion of Model-2), or Model-4 assumptions. Model-4 assumes that the scatter of the points must increase with the discordance of the points, and so weights the points according to their degree of concordance. This may be done in two ways, depending on whether one wishes to consider the lower intercept of the chord with concordia as concordant (Model 4-LI) or the upper intercept as concordant (Model 4-UI). Model 4-UI in effect assumes that the excess scatter arises from the effect of more than one episodic disturbance, and that this effect increases with increasing discordance. Thus Model 4-UI attempts to model a suite of points that lie within a triangular wedge, with its apex at the true primary age of the zircons, and its lower concordia-intercepts lying on or between the youngest and and oldest subsequent disturbances. The basis for this model comes from Davis (1982). Model 4-LI, on the other hand, assumes that the true age of the suite is at the lower intercept of the best-fit line, and that the excess scatter about this line is due to the presence of one or more older (inherited) components.

To deal with these assumptions, the Model-4 algorithm in effect expands the $^{207}Pb/^{206}Pb$ errors of each data-point proportional to the discordance of that point, until all of the observed scatter is accounted for. The effect of this algorithm is to assign more weight to the relatively concordant points and less to the relatively discordant points. Several iterative cycles of line-fitting and concordia-intercept calculation are required. Whether Model-2 or Model-4 is chosen depends on what you believe to be the reason for the excess scatter of the points. If either the multiple-disturbance or simple-inheritance model seems likely, and especially if you have several points that

are only slightly discordant, Model-4 may be reasonable choice. However, if you consider that the Model-4 assumptions are simplistic, perhaps the "know-nothing" approach of Model-2 would be more reasonable.

After completing the Yorkfit, if you are using the CRT as the plotting device, the intercepts and intercept-uncertainties will be printed out on the plot. If the HP-7470A is the plotting device, the CRT will display,

PRESS LABEL INT. FOR PLOT-LABEL OF INTERCEPTS.
PRESS CONTINUE TO DECLINE.

If you press k6 (LABEL INT.), the intercepts, intercept errors, and MSWD will be lettered in the lower-right hand corner of the plot-box. If more than one such Yorkfit and intercepts have been calculated, succeeding intercept-solution phrases will be stepped up in the plot box so as not to overwrite each other.

After lettering the intercept solution, or if you pressed k7 (CONTINUE), the program will return to the RETAIN/CONTINUE/YORKFIT display (see above).

PBPLOT

The PBPLOT program (see Appendix 3 for listing) is similar to the ISOPLOT program except for the following differences:

1) The X-axis is always defined as 206-Pb/204-Pb, and the Y-axis can only be defined as 207-Pb/204-Pb or 208-Pb/204-Pb;

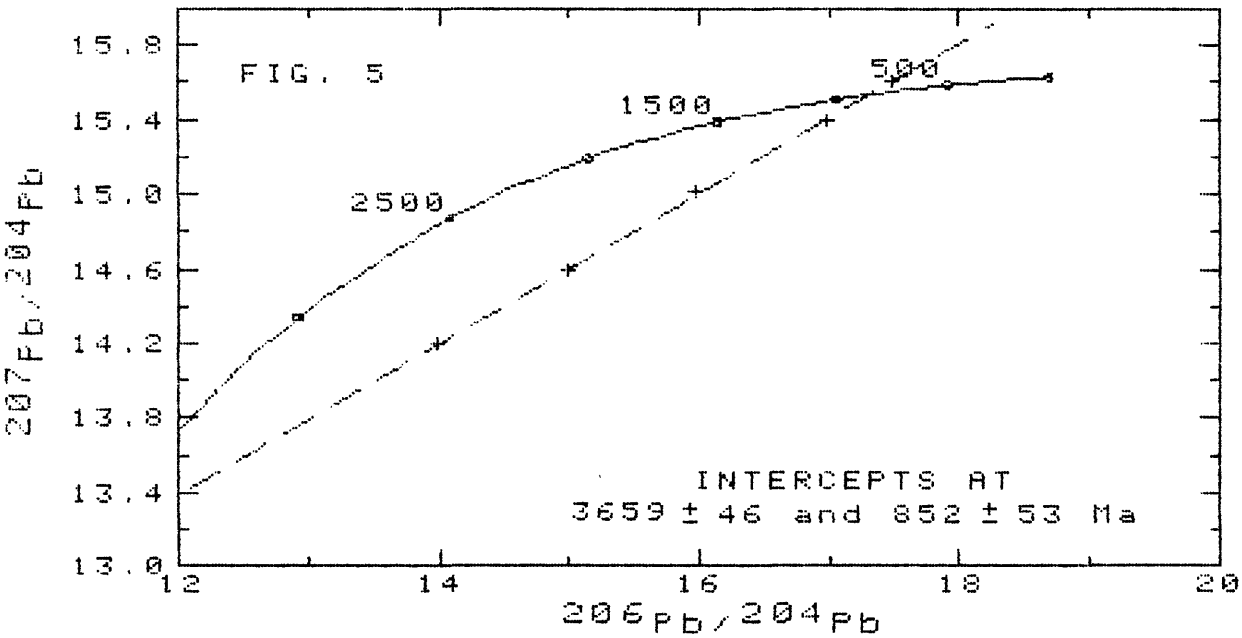
2) A single-stage Pb growth-curve can be drawn in the plot;

3) The error-box and error-cross plotting-symbols are not permitted, as the errors in the X and Y axes are always highly correlated;

4) Only Model-1 and Model-2 Yorkfits are permitted;

5) After each Yorkfit, a 207/206 (or 208/206) age and age- error is calculated, and the intercepts and intercept- errors of the best-fit line with the single-stage growth- curve are calculated.

Examples of PBPLOT are shown in figs. 5 and 6.



The first CRT display after pressing the RUN key defines the special-function keys as follows:

KEY#	KEY LABEL	FUNCTION
k1	O GTICKS	Use a small circle to indicate growth curve ticks, and label these ticks horizontally, above and to the left of the ticks (default).
k2	CRT PRINT	Use the CRT as the printer (default).
k3	CRT PLOT	Use the CRT as the plotter (default).
k4	PEN 1	Use pen #1 of the HP-7470A plotter (default).
k5	STACK	Suppress the X-axis label and the X-axis tick-labels, so that a 208/204 plot can be stacked on a 207/204 plot.
k5	DATAFILE	Load data from a VISICALC or numeric data file into memory (appears only

		after the Y-axis has been selected).
k6	NORM	Use normal-size labels and symbols (default).
k7	207/204	Define $^{207}Pb/^{204}Pb$ as the Y-axis.
k8	\ GTICKS	Use a dash which would intersect the starting age on the Pb-growth curve for the growth-curve ticks (so the dash is a portion of an isochron), with parallel tick-labels.
k9	PRINTER	Use an HP-82905B or HP-2673 printer as the printer.
k10	HARDPLOT	Use an HP-7470A as the plotter.
k11	PEN 2	Use pen #2 of the HP-7470A plotter.
k12	PENSPEED	Select the pen speed of the HP-7470A plotter (fastest speed is default).
k13	SLIDE	Use larger-size labels and symbols.
k14	208/204	Define $^{208}Pb/^{204}Pb$ as the Y-axis.
k7	START	Enter the plot-limits and draw the plot-box (appears only after the Yaxis has been selected).
N		Press the N key to suppress the Pb-isotope growth-curve.
T		Press the T key to suppress just the growth-curve ticks and tick-labels.
H	[HELP]	Press the H key to obtain a list of explanations of the functions of the keys of this display.

After selecting the Y-axis, and changing any required printer, plotter, pen, or pen-speed values from their default values as required, press the START key (k7). The CRT will clear and display,

PRESS ENDLINE FOR STACEY-KRAMERS GROWTH-CURVE,
PRESS * FOR ARBITRARY GROWTH-CURVE.

If you wish the growth-curve for the plot to be the standard, single-stage growth curve of Stacey and Kramers (1977), press the ENDLINE key. This growth curve has a starting age of 3700 Ma, initial-Pb 206/204 = 11.152, 207/204 = 12.998, and 208/204 = 31.230, and assumes $^{238}U/^{204}Pb$ = 9.78 for the Pb-source (present-day value), and $^{238}U/^{232}Th$ = .26455. If, however, you wish to define the parameters for some other Pb-growth curve, press the asterisk key. Upon pressing the asterisk key, the CRT will query,

INITIAL 206/204 and 207/204, STARTING AGE (Ma), SOURCE 238/204?

Enter the starting 206/204 and 207/204 (or 208/204, if a 208/204-206/204 plot) for the growth curve, the age at which these starting values apply, and the present-day $^{238}U/^{204}Pb$ of this isotopic reservoir. If the plot is a 208/204-206/204 plot, the CRT will also query its present-day $^{238}U/^{232}Th$.

Once the growth-curve parameters are entered (either specifically or by default), the CRT will query:

206/204 and 207/204 LIMITS (min., max., min., max.)?

Enter 4 values, separated by commas, to define the maximum and minimum limits of the plot box: minimum-206/204, maximum 206/204, minimum 207/204, maximum 207/204 (for a 207/204-206/204 plot). If the plot is to be drawn with the HP-7470A plotter, the CRT will request the plot size, as in the ISOPLOT and CPLOT programs. The labeled plot-box and Pb-growth curve will then be constructed, and the CRT will show the "select plotting-symbol" display, which is similar to that of the CPLOT program.

The plotting and line-fitting portion of PBPLOT is similar to that of CPLOT, except that,

1) Only Yorkfit models 1 and 2 are permitted,

2) Once a best-fit line is obtained, the intercepts and intercept errors of this line with the Pb-growth curve are calculated, and

3) An age and age-error are calculated from just the slope of the 207/204-206/204 or 208/204-206/204 best-fit line.

For PBPLOT, data from VISICALC files can be recovered in a slightly different way than ISOPLOT or CPLOT, in that the user can choose to bring all three of the relevant Pb-isotope ratios (206/204, 207/204, and 208/204) into memory at the same time. This is especially convenenient if 206/204 - 208/204 and 206/204 - 207/204 plots are to be constructed from the same data-set. When the query for the columns for the Pb-isotope data appears, the CRT will also prompt:

6/4, %err, 7/4, %err, Rho(6/4-7/4),
8/4, %err, Rho(6/4-8/4)

or: 6/4, %err, 7/4, %err, Rho(6/4-7/4)

or: 6/4, 7/4, 8/4

or: 6/4, 7/4 ?

The number of values in your response defines which of the above formats that the computer assumes. For example, suppose that you requested all three of the 206/204, 207/204, and 208/204 ratios to be recovered from the VISICALC file. Then, when you request a REDRAW of the plot with, for example, 208/204 as the Y-axis rather than 207/204, to use the 208/204 data in the data file just press the DATAFILE key, and the CRT will show the message,

NAME OF DATA-FILE TO BE RETRIEVED?
(press ENDLINE to re-use data from file TRIALFILE)

Press the ENDLINE key, and the previously-recovered 208/204 data will automatically replace the earlier 207/204 data.

NUMVERT

NUMVERT is a short program to convert VISICALC-created data files to a more-rapidly accessible form of numeric plus string files. Such a file takes up less space on the floppy disks, and is of most value for large files that will not require editing and that will be accessed repeatedly by one or more of the plotting programs.

NUMVERT does make some assumptions about the structure of the VISICALC file. First, the VISICALC file must be assigned a column width of 9 characters in order for the row names (sample names) to be recovered accurately. Row names (sample names) are assumed to occupy the first two columns (A and B) of the VISICALC array. Second, no more than 45 columns (C through AU) of numeric data in the VISICALC file will be converted. And third, no more than 100 rows of numeric data in the VISICALC file will be converted. Note also that cells with formulae instead of numeric data will not be converted to numeric data.

Each row of VISICALC data is converted to a SHORT precision (6 digit accuracy), 45 element vector, with each element in the vector corresponding to the numeric value in the equivalent column of the VISICALC array (less the two columns used for the row names (sample names). The alphanumeric characters in the columns A and B of the VISICALC array are combined to form a row name (sample name) of up to 18 characters. Each converted row is stored as an individual record in the form

```
N$,X()
```

where N$ is dimensioned to a length of 18 characters, and X is dimensioned as X(45) (option base 1).

Running NUMVERT:

After loading NUMVERT and pressing the RUN key, the CRT will display,

```
NAME OF VISICALC FILE TO BE CONVERTED TO NUMERIC FILE?
```

Enter the name of the VISICALC file, complete with appropriate mass storage specifier, if necessary. The

program will confirm that such a file exists, and immediately start converting the VISICALC cells to numeric and (for row names) string values. The CRT will show the values of the VISICALC strings, row names, rows, columns, and numeric values as the conversion proceeds. When the conversion is complete, the CRT will query,

NAME OF NUMERIC DATA-FILE?

Enter the name of the numeric file that you wish stored, complete with mass storage specifier if necessary. If sufficient space exists on the specified disk, the file will be created, stored, and a catalog of the disk will appear on the CRT. If sufficient space does not exist on the disk, the CRT will display,

DISK FULL: REPLACE/PACK DISK, PRESS CONT

Either choose a new disk, or pack the existing disk to create more useable storage space and then press the CONT key.

REFERENCES

Davis, D.W., 1982, Optimum linear regression and error estimation applied to U-Pb data: Canadian Jour. Earth Sciences, v. 19, p. 2141-2149.

Ludwig, K.R., 1980, Calculation of uncertainties of U-Pb isotope data: Earth Planetary Science Letters, v. 46, p. 212-220.

McIntyre, G.A., Brooks, Christopher, Compston, William, and Turek, Andrew, The statistical assessment of Rb-Sr isochrons: Jour. Geophys. 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 Jäger, Emilie, 1977, Subcommission geochronology: Convention on the use of decay constants in geo- and cosmochronology: Earth Planetary Science Letters, v. 36, p. 359-362.

Titterington, 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: Example of CRT plot (dumped to HP-82905B printer) from ISOPLOT.

Figure 2: Example of HP-7470A plot from ISOPLOT. Dashed line is best-fit line to data points with error-ellipse symbols. Examples of some of the other possible symbols are shown to right. The isochron age is calculated automatically by the program.

Figure 3: Example of CRT plot (dumped to HP-82905B printer) from CPLOT, using horizontal concordia-tick option.

Figure 4: Example of HP-7470A plot from CPLOT, using oblique concordia-tick option.

Figure 5: Example of CRT plot (dumped to HP-82905b printer) from PBPLOT, using horizontal growth-curve tick option.

Figure 6: Example of HP-7470A plot from PBPLOT, using both growth-curve tick options. The 208/204 plot was precisely aligned with the 207/204 plot using the nonstandard plot-size and stack options.

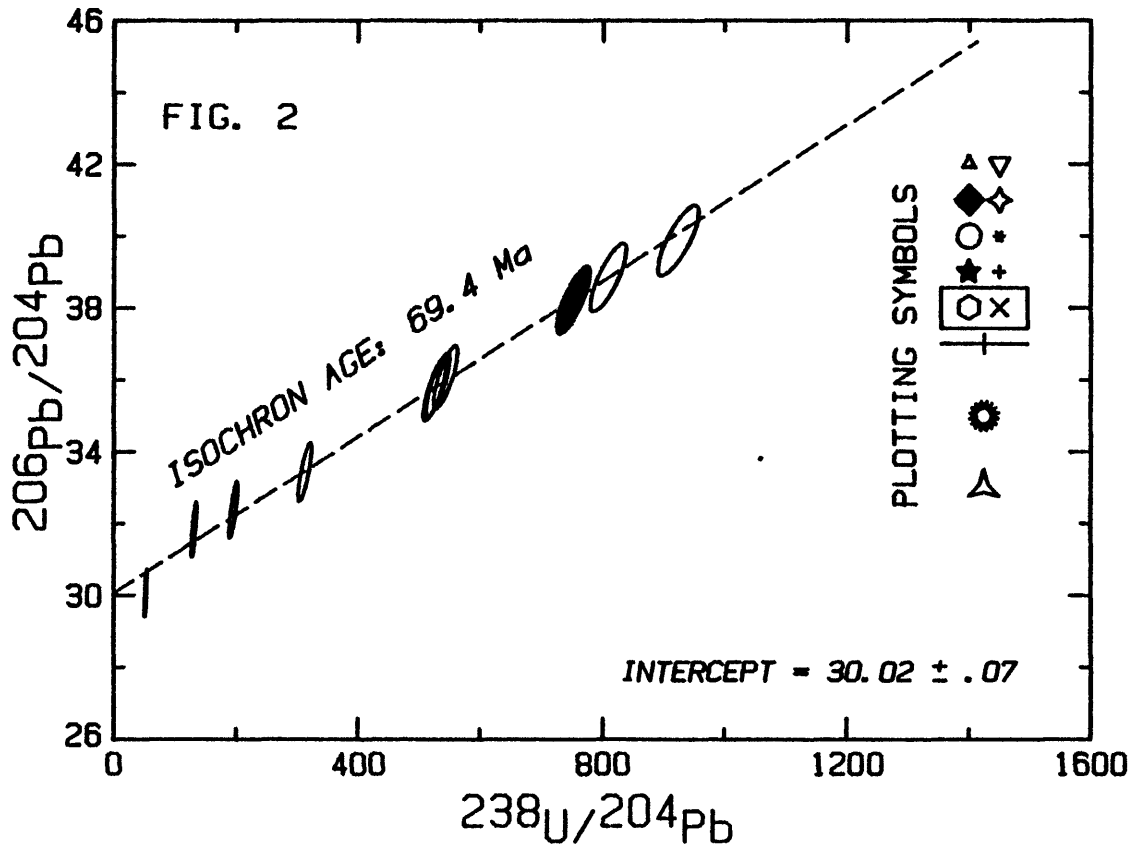
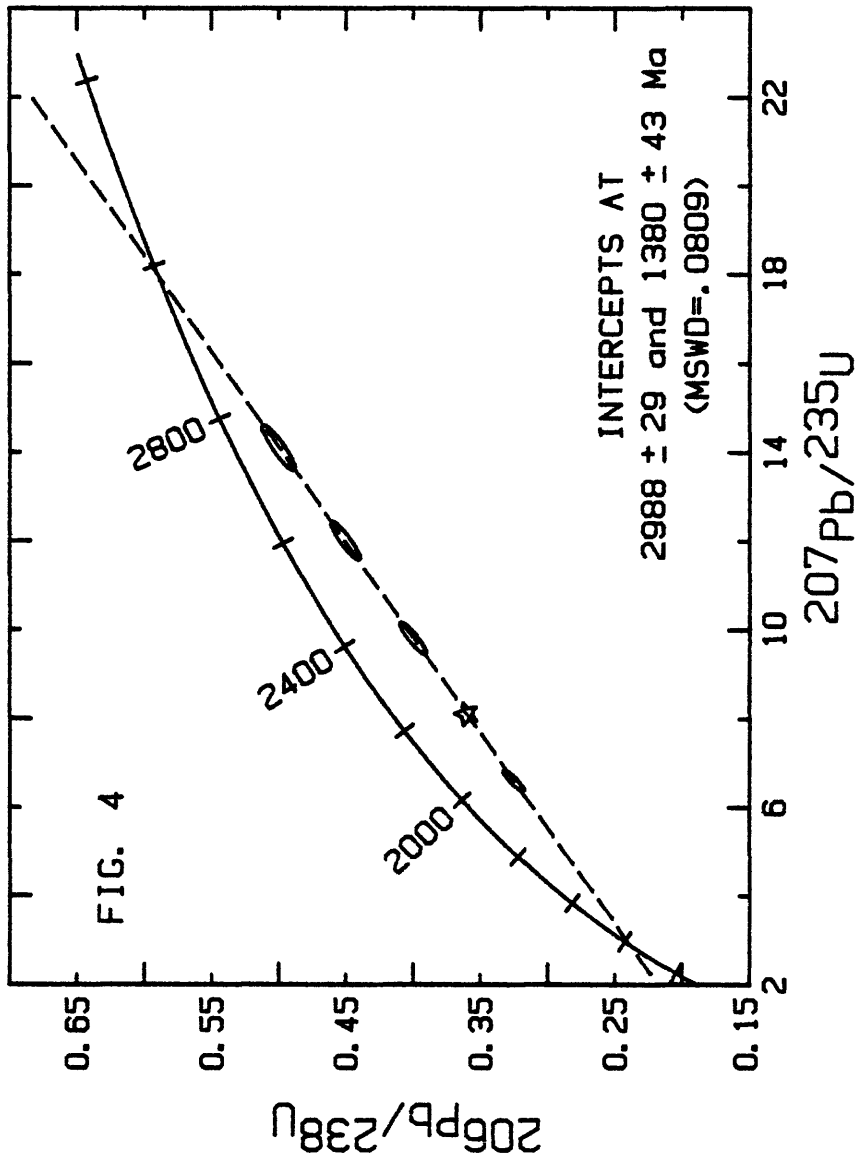


FIG. 2
46
42
38
34
30
26
0
400
800
1200
1600
$^{206}Pb/^{204}Pb$
$^{238}U/^{204}Pb$
ISOCHRON AGE: 69.4 Ma
INTERCEPT = 30.02 ± .07
PLOTTING SYMBOLS


INTERCEPTS AT
2988 ± 29 and 1380 ± 43 Ma
(MSWD=.0809)
2800
2400
2000
FIG. 4
0.65
0.55
0.45
0.35
0.25
0.15
2
6
10
14
18
22
$^{206}Pb/^{238}U$
$^{207}Pb/^{235}U$

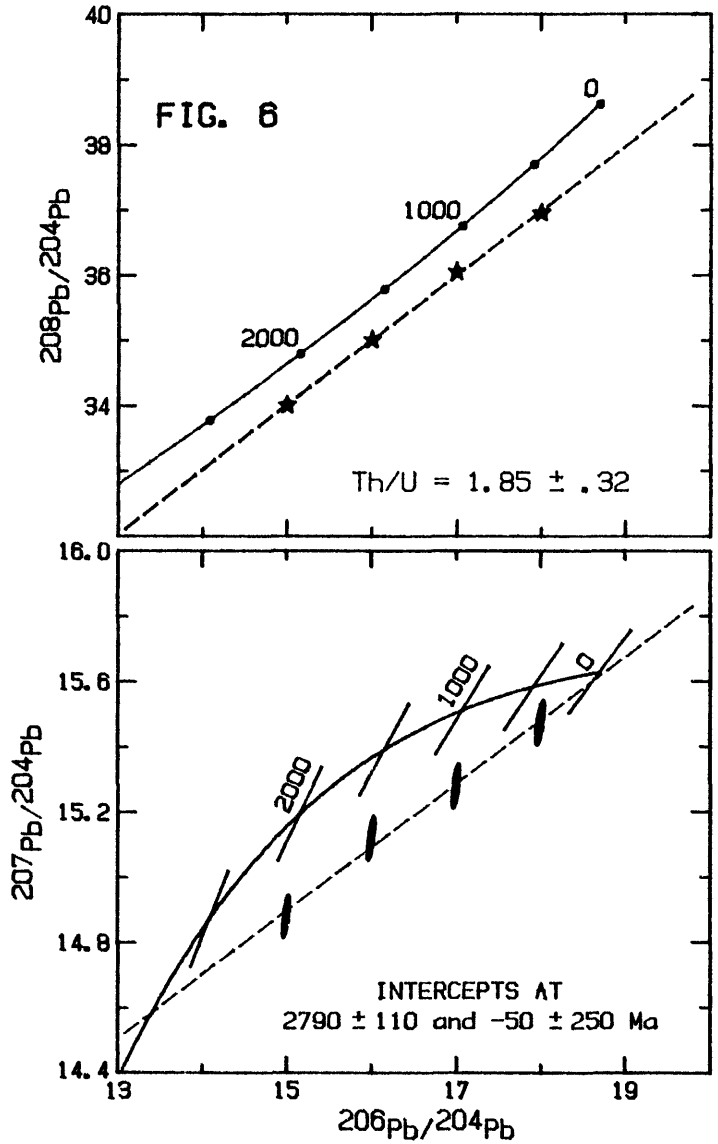


FIG. 6
40
38
36
34
208Pb/204Pb
0
1000
2000
Th/U = 1.85 ± .32
16.0
15.6
15.2
14.8
14.4
207Pb/204Pb
2000
1000
0
INTERCEPTS AT
2790 ± 110 and -50 ± 250 Ma
13
15
17
19
206Pb/204Pb

APPENDIX 1: PROGRAM LISTING OF ISOPLOT

```
10 ! ************************* ISOPLOT ****************************************
20 !
30 !  Program for plotting and regression of X-Y data. NOV 08, 1983.
40 !
50 !  Requires advanced programming ROM and 105 Kbytes RAM.
60 !   Data may be entered from either of  (1) Keyboard, (2) Visicalc data-
70 !   file, or (3) nX45 Numeric data-file, and plotted either on CRT or
80 !   plotter. Errors can be entered as either % or absolute, 1- or 2-sigma.
90 !   Brackets indicate optional input.
100 !  Takes 100 points for Yorkfit, or 300 points for polynomial regression.
110 !
120 ON ERROR GOTO 140
130 IF PLOT_STARTED THEN PLOTDATA
140 OFF ERROR @ OPTION BASE 1
150 !
160  DIM A(100),X_WT(100),Y_WT(100),R(100),Z(100),A$[72],X(300),Y(300),Ex(300),Ey(300),Name$[50],R0(300),Slope(5),Inter(5),L(5),Erri
nt(5),Errsl(5),O(5),R$[50]
170 DIM RR$[50],PLOTDEV(2),Pr(2),Nx$[32],Ny$[32],KEY$[1,N$(300)[18],S(300,5),XX(45),O$(5)[18],XERR(300),YERR(300),AXIS$[65],P$[80],
CRUD$[50],Nx_$[32],Ny_$[32]
180 DIM Nx0$[10],Ny0$[10],NxR$[32],NyR$[32],EL$[7],COLN(5),T1(100),T2(100),U(100),V(100),UV(100),U2(100),V2(100),Z2(100),Re$[50],ALL
KEY$[256]
190 DIM ZERO$[8],XTIK$(8)[12],YTIK$(8)[12],XTIK(8),YTIK(8),CACHE(100,5),SYMBOL(100,5),LETTER$(25)[80],LETTER(25,5),LINE(12,2),XWT(10
0),YWT(100)
200 INTEGER XDEC(8),YDEC(8),ISOTOPE(4)
210 !
220 !      Rb-87,   Sm-147,  U-238,      U-235,     Th-232  DECAY CONSTANTS
230 DATA 1.42E-11,6.54E-12,1.55125E-10,9.8485E-10,4.9475E-11
240 MAT READ L
250 !
260 ALLKEY$=RPT$ (" ",256)
270 FOR I=1 TO 256
280  IF I<128 OR I>172 THEN ALLKEY$[I,I]=CHR$ (I)
290 NEXT I
300 CRUD$=HGL$ (" #### IMPROPER RESPONSE OR NUMBER OF VALUES #### ")
310 ER$="%err" @ EA$="2-SIGMA" @ EB$=" %-ERRORS" @ PERCENT=1 @ SIG=2
320 CNT$=HGL$ (" CONT ") @ FILE$="" @ ZERO$="00000000"
330 PLOTDEV(1)=1 @ PLOTDEV(2)=705 @ CPEN=1 @ PEN 1 @ SLD,HP,DFILE,ALL=0 @ PRINTER IS 1 @ EL$=HGL$ ("ENDLINE") @ SPEED$="VS" @ PS=.35
 @ PLR,Pr(2),Pr(1),PR=1
340 POLYSIZE=2.5 @ DEG @ PAGESIZE 24 @ MAXNUM=300 @ MAXYOR=100 @ PLOT_STARTED,ALL_CONVERTED=0
350 !
360 REDRAW: LZERO,STACK,ODDSIZE,SIZETYPE,C_CT,NLINES,NPHRASES,PLOTTER_DUMP,SOLID_LINE=0 @ DEG @ PAGESIZE 24 @ CLEAR @ PEN 0 @ GOTO B
EGIN
370 !
371 CLEARKEY: ! CLEAR ALL SPECIAL-FUNCTION-KEY LABELS
372          FOR i=1 TO 14 @ ON KEY# i,"" GOSUB RETRN  @ NEXT i
373          RETURN
374 !
380 RETRIEVE_STR: ! Retrieve up to 5 substrings (separated by ',') from string
390      LINPUT R$ @ O$(1),O$(2),O$(3),O$(4),O$(5)="" @ Ninputs=0 @ RR$=R$
```

```
400      ON ERROR GOTO 440
410      Wh=POS (R$,",") @ SP=Wh*(Wh#0)+(LEN (R$)+1)*(Wh=0)
420      O$(1+Ninputs)=R$[1,SP-1] @ Ninputs=1+Ninputs
430      IF Wh THEN R$=R$[1+Wh] @ GOTO 410
440      OFF ERROR @ RETURN
450 !
460 RETRIEVE: ! Retrieve up to 5 numeric values (Separated by ',') from string
470      LINPUT R$
480 RTR: O(1),O(2),O(3),O(4),O(5),Ninputs=0 @ RR$=R$ ! Alternate entry line
490      ON ERROR GOTO 520
500      O(1+Ninputs)=VAL (R$) @ Ninputs=1+Ninputs
510      Wh=POS (R$,",") @ IF Wh THEN R$=R$[1+Wh] @ GOTO 500
520      OFF ERROR @ RETURN
530 !
540 BAD_INPUT: GOSUB CLUNK  @ DISP @ DISP CRUD$ @ DISP @ RETURN
550 !
560 RETRN: RETURN ! Dummy return
570 KRETRN: OFF KYBD @ RETURN
580 !
590 DEF FNA(X) = INT (LGT (ABS (X+(X=0))))
600 !
610 DEF FNTICK(X) !                        Returns tick-interval
620   A=10^FNA(X)/8
630   IF ABS (X/A)>12 THEN A=2*A @ GOTO 630
640   B=ABS (A)/10^FNA(A)
650   IF B*INT (B) THEN A=INT (B)*10^FNA(A)
660   FNTICK=A @ FN END
670 !
680 DEF FNXTICK(P) ! Draw X-axis & ticks
690     NTIX=Xspred/XTIK @ ETIX=NTIX/2=INT (NTIX/2) ! # Ticks even?
700     FOR X=1 TO NTIX
710      ETIK=X/2=INT (X/2) ! Even tick?
720      IF (P=1)*ETIK OR (P=-1)*((ETIK EXOR ETIX)=0) THEN M=XTICKSIZE ELSE M=XTICKSIZE/2
730      IDRAW P*XTIK,0
740      IDRAW 0,P*M @ IDRAW 0,-(P*M)
750     NEXT X
760     FNXTICK=0 @ FN END
770 !
780 DEF FNYTICK(P) ! Draw Y-axis & ticks
790     NTIX=Yspred/YTIK @ ETIX=NTIX/2=INT (NTIX/2) ! # Ticks even?
800     FOR Y=1 TO NTIX
810      ETIK=Y/2=INT (Y/2) ! Even tick?
820      IF (P=1)*ETIK OR (P=-1)*((ETIK EXOR ETIX)=0) THEN M=YTICKSIZE ELSE M=YTICKSIZE/2
830      IDRAW 0,P*YTIK
840      IDRAW -(P*M),0 @ IDRAW P*M,0
850     NEXT Y
860     FNYTICK=0 @ FN END
870 !
880 DEF FNDr(N,D) !                        Round value N to D sig. figures
890     IF N THEN Mu=10^(D-INT (LGT (ABS (N)))-1) ELSE Mu=1
900     FNDr=IP (N*Mu)/Mu @ FN END
910 !
920 DEF FNDECIMAL(X) !                     Returns #digits beyond decimal pt.
930   A=0 @ N=ABS (X)
940   N=10*(ABS (N)-INT (ABS (N)))
950   IF N THEN A=A+1 @ GOTO 940 ELSE FNDECIMAL=A
960   FN END
970 !
980 CLUNK: BEEP 115,40 @ BEEP 195,40 @ RETURN ! Clunky sound
990 !
```

```
1000 DEF FNCHAR(X) = 1+(ABS (X)>1)*(FNA(X)+(X*INT (X)))+(X<0)+FNDECIMAL(X)+(ABS (X)<1)*LZERO ! *Characters required for the number x
.
1010 !
1020 DEF FNR(A$) = POS (A$,"^")-POS (A$,"\")
1030 !
1040 REMOVE: L,A=LEN (A$) ! Remove superscript, subscript characters from string
1050     FOR J=1 TO LEN (A$)
1060     IF (A>1)*(A$[J,J]="^" OR A$[J,J]="\") THEN L=L-1 @ A$[J]=A$[J+1]
1070     NEXT J
1080     RETURN
1090 !
1100 BEGIN: ALPHA 1 @ CLEAR @ ALPHA 21,27 @ DISP "(LOWER OPTIONS ARE DEFAULT)" @ OFF CURSOR
1110              ON KEY# 1,"PRINT-DAT" GOSUB HARDCOPY
1120              ON KEY# 2," CRT-DAT" GOSUB DISPLAY
1130              ON KEY# 3," HARDPLOT" GOSUB HARDPLOT
1140              ON KEY# 4," CRT PLOT" GOSUB CRTPLOT
1150              ON KEY# 7,"  START" GOTO 60
1160   IF HP THEN ON KEY# 5," PEN 1" GOSUB PEN_1
1170 IF HP=0 THEN ON KEY# 5,"" GOSUB RETRN
1180   IF HP THEN ON KEY# 6," PEN 2" GOSUB PEN_2
1190 IF HP=0 THEN ON KEY# 6,"" GOSUB RETRN
1200   IF HP THEN ON KEY# 8,"PEN SPEED" GOSUB PEN_SPEED
1210 IF HP=0 THEN ON KEY# 8,"" GOSUB RETRN
1220              ON KEY# 9,"LDG ZEROES" GOSUB LZERO
1230   IF HP THEN ON KEY# 10,"  STACK" GOSUB STACK
1240 IF HP=0 THEN ON KEY# 10,"" GOSUB RETRN
1250              ON KEY# 11,"  HELP" GOTO HELP1
1260              ON KEY# 12,"  #NORM#" GOSUB NORMSIZE
1270              ON KEY# 13,"  SLIDE" GOSUB SLIDE
1280              ON KEY# 14," DATAFILE" GOTO GET_DATAFILE
1290 !
1300              ON KYBD KY,ALLKEY$ GOSUB keys
1310 !
1320   KEY LABEL
1330   GOTO 1330
1340 !
1350 keys: GOSUB CLUNK  @ RETURN
1360 !
1370 60: OFF KEY# @ CLEAR @ DISP "USE ^ TO INDICATE SUPERSCRIPT, \ FOR SUBSCRIPT."
1380 DISP @ DISP "X-AXIS NAME,  Y-AXIS NAME";@ INPUT Nx_$,Ny_$
1390 DISP @ DISP "X and Y LIMITS  ([Xmin,] Xmax, [Ymin,] Ymax)";@ GOSUB RETRIEVE
1400 IF Ninputs#2 AND Ninputs#4 THEN GOSUB BAD_INPUT  @ GOTO 1390
1410 IF Ninputs=2 THEN XMIN,YMIN=0 @ XMAX=O(1) @ YMAX=O(2) @ GOTO 1430
1420 XMIN=O(1) @ YMIN=O(3) @ XMAX=O(2) @ YMAX=O(4)
1430 IF XMAX<= XMIN OR YMAX<= YMIN THEN GOSUB BAD_INPUT  @ GOTO 1390
1440 IF PLR=1 THEN 1560 ELSE CLEAR @ DISP "PLOT-SIZE (3-10)" @ DISP @ DISP "[press "&EL$&" for nonstandard dimensions/locations]";@
DISP @ GOSUB RETRIEVE
1450 IF Ninputs#((O(1)<3)+(O(1)>10)) THEN GOSUB CLUNK  @ GOTO 1440
1460 IF Ninputs=0 THEN ODDSIZE=1 ELSE PLOTSIZE=O(1) @ GOTO 1560
1470 ALPHA 1 @ CLEAR @ OFF CURSOR @ ALPHA 8,7 @ AWRIT "PRESS "&HGL$ (" 1 ")&" TO DEFINE THE PLOT-LIMITS FOR THE ENTIRE FIGURE-AREA,"

1480  ALPHA 10,13 @ AWRIT HGL$ (" 2 ")&" TO DEFINE LIMITS FOR THE PLOT-BOX ONLY"
1490 ON KYBD KY,"0123456789" GOTO 1510
1500 GOTO 1500
1510 OFF KYBD @ IF KY#49 AND KY#50 THEN GOSUB CLUNK  @ GOTO 1490 ELSE SIZETYPE=KY-48 @ ALPHA 1 @ CLEAR @ OFF CURSOR
1520  ALPHA 10,10 @ DISP "ENTER PLOT-LIMITS IN mm: X-MIN, X-MAX, Y-MIN, Y-MAX" @ DISP TAB (14);"(max. limits are 0-257 [X]  and 0-19
1 [Y])";@ OFF CURSOR
1530  DISP @ INPUT LIMX1,LIMX2,LIMY1,LIMY2@ IF LIMX1#LIMY1<0 OR LIMX2>257 OR LIMY2>191 OR LIMX2-LIMX1<30 OR LIMY2-LIMY1<30 THEN GOSU
B BAD_INPUT  @ GOTO 1520
1540  PLOTSIZE=10*((LIMX2-LIMX1)/257+(LIMY2-LIMY1)/191)/2
```

```
1550 !
1560 ALPHA 1,1 @ CLEAR @ OFF CURSOR @ ALPHA 12,27 @ AWRIT "A FEW SECONDS, PLEASE..." @ Nx$=Nx_$ @ Ny$=Ny_$
1570 ON ERROR GOTO PLOTTER_ERROR  @ PLOTTER IS PLR @ OFF ERROR @ GOTO 1610
1580 !
1590 PLOTTER_ERROR: GOSUB CLUNK  @ CLEAR @ DISP TAB (24);HGL$ (" TURN THE PLOTTER ON, DUMMY! ") @ WAIT 3000 @ CLEAR @ GOTO 1570
1600 !
1610 IF PLOTTER_DUMP THEN 1990
1620 IF XMAX<= XMIN OR YMAX<= YMIN THEN GOSUB BAD_INPUT  @ GOTO 1390
1630 XTIK=FNTICK(XMAX-XMIN) @ YTIK=FNTICK(YMAX-YMIN)
1640 !
1650 ! Force lower-bound X & Y values to have a minimum # of sign. figures.
1660 FOR I=1 TO 7
1670  X=FNDr(ABS (XMIN),I) @ IF ABS (XMIN)-X<XTIK THEN 1690
1680 NEXT I
1690 IF XMIN>= 0 OR ABS (XMIN)=X THEN XMIN=SGN (XMIN)*X ELSE XMIN=-X-XTIK
1700 FOR I=1 TO 7
1710  Y=FNDr(ABS (YMIN),I) @ IF ABS (YMIN)-Y<= YTIK THEN 1730
1720 NEXT I
1730 IF YMIN>= 0 OR ABS (YMIN)=Y THEN YMIN=SGN (YMIN)*Y ELSE YMIN=-Y-YTIK
1740 !
1750 FOR X=XMIN TO XMAX STEP XTIK !        Force XMAX to lie on a tick
1760   IF X+XTIK>= XMAX THEN XMAX=X+XTIK @ GOTO 1780
1770 NEXT X
1780 FOR Y=YMIN TO YMAX STEP YTIK !        Force YMAX to lie on a tick
1790   IF Y+YTIK>= YMAX THEN YMAX=Y+YTIK @ GOTO 1810
1800 NEXT Y
1810 Xspred=XMAX-XMIN @ Yspred=YMAX-YMIN @ XDEC_MAX,YDEC_MAX,YCHAR,XCHAR=0
1820 !
1830 ! Find max. # characters in X & Y ticks & ticks w. max # decimal places
1840 FOR I=1 TO Xspred/XTIK/2+1
1850  XTIK(I)=XMIN+2*XTIK*(I-1)
1860  A=FNCHAR(XTIK(I)) @ IF A>XCHAR THEN XCHAR=A ! max # of characters
1870  XDEC(I)=FNDECIMAL(XTIK(I)) @ IF XDEC(I)>XDEC_MAX THEN XDEC_MAX=XDEC(I)
1880 NEXT I
1890 !
1900 FOR I=1 TO Yspred/YTIK/2+1
1910  YTIK(I)=YMIN+2*YTIK*(I-1)
1920  A=FNCHAR(YTIK(I)) @ IF A>YCHAR THEN YCHAR=A ! max # of characters
1930  YDEC(I)=FNDECIMAL(YTIK(I)) @ IF YDEC(I)>YDEC_MAX THEN YDEC_MAX=YDEC(I)
1940 NEXT I
1950 !
1960 CRT=PLR=1 @ PP=1+CRT @ HP=PLR*1 @ IF CRT THEN GCLEAR
1970 IF CRT THEN LIMX1,LIMY1=0 @ LIMX2=125 @ LIMY2=75 @ GOTO 2020
1980 IF ODDSIZE THEN 2010
1990  LIMX1=257*(1-.1*PLOTSIZE)/2 @ LIMX2=257-LIMX1 ! For centered plots
2000  LIMY1=191*(1-.1*PLOTSIZE)/2 @ LIMY2=191-LIMY1
2010 IF SIZETYPE<2 THEN LIMIT LIMX1,LIMX2,LIMY1,LIMY2
2020 AR=2/3 @ THT=5 @ LHT=6 @ IF CRT THEN 2070 ! Aspect ratio, tick label char. ht.
2030 IF SLD=0 THEN M=.7*(1+2.7*(PLOTSIZE<6)/PLOTSIZE) ! For plotsizes<6, somewhat buffer label-character size.
2040 IF SLD THEN M=.95*(1+2.7*(PLOTSIZE<6)/PLOTSIZE)
2050 THT=M*THT @ LHT=1.2*M*LHT ! Tick-label and axis-label character heights
2060 !
2070 Pu=POS (Nx$,"^") @ Pd=POS (Nx$,"\") @ Qu=POS (Ny$,"^") @ Qd=POS (Ny$,"\")
2080 P=Pu OR Pd @ Q=Qu OR Qd @ A=ABS (XMAX-XMIN)/XTIK @ B=ABS (YMAX-YMIN)/YTIK
2090 IF Qu AND Qd AND POS (Ny$,"\")<POS (Ny$,"^") THEN Ss=1 ELSE Ss=0
2100 ! Ss=1 if the Y-axis label has subscripts
2110 IF POS (Nx$,"g") OR POS (Nx$,"q") OR POS (Nx$,"p") THEN Xdsc=1 ELSE Xdsc=0
2120 IF POS (Ny$,"g") OR POS (Ny$,"q") OR POS (Ny$,"p") THEN Ydsc=1 ELSE Ydsc=0
2130 ! Need more room for chars. with descenders
2140 YL=1+THT*.75+LHT*(.75+.25*P+Xdsc/4) !                  GU's below X-axis
```

```
2150 XL=1+THT*AR*(1-HP/3)*YCHAR+LHT*(1.05+.05*CRT+Ss/2+Q/4+Ydsc/6-.3) ! GU's to left of Y-axis
2160 ! XL=1+THT*AR*YCHAR*(1-HP/6)-.25+LHT*(1+Ss/2+Ydsc/6)*(1-HP/3) ! to L of Y
2170 YEND=THT/2*(B/2=INT (B/2)) !                    GU's above top of Y-axis
2180 XEND=(XCHAR*AR*THT/2+1)*(A/2=INT (A/2)) !       GU's to right of X-axis
2190 !
2200 IF SIZETYPE=2 THEN C=0 @ GOSUB PLOTBOX_SIZE
2210 IF RATIO >1 THEN Xgu=100*RATIO  @ Ygu=100 ELSE Xgu=100 @ Ygu=100/RATIO
2220 ! Xgu,Ygu are GU's of total-plot area along X & y-directions
2230 !
2240 XLOC=(Xgu-XEND-XL)/Xspred !                     GU'S per UU (X-direction)
2250 YLOC=(Ygu-YEND-YL)/Yspred !                     GU'S per UU (Y-direction)
2260 Xmm=XLOC*(LIMX2-LIMX1)/Xgu !                    Millimetres/UU (X-dir)
2270 Ymm=YLOC*(LIMY2-LIMY1)/Ygu !                    Millimetres/UU (Y-dir)
2280 !
2290 LOCATE XL,Xgu-XEND,YL,Ygu-YEND
2300 IF HP THEN PRINTER IS PLOTDEV(2) @ PRINT SPEED$ @ PRINTER IS Pr(2)
2310  SCALE XMIN,XMAX,YMIN,YMAX
2320  CSIZE THT @ LW=AR*LHT*(1-HP/3)
2330 ! Label every 4th tick, not every other tick if little room available
2340  XTmm=2*XTIK*Xmm ! mm between every other tick on X-axis
2350  XTLmm=THT*AR*XCHAR*Xmm/XLOC*(1-HP/3) ! mm taken up by largest X-tick label
2360  IF XTLmm/XTmm<.5 THEN MTIK=2 ELSE MTIK=4 ! Avoid crowding X-tick labels
2370 !
2380 FOR I=1 TO Xspred/XTIK/2+1 !                    Add decimal pt & zero to
2390  IF XTIK(I)=0 THEN XTIK$(I)="0" @ GOTO 2470
2400  XTIK$(I)=VAL$ (XTIK(I)) !                      match format of X-tick with
2410  IF XDEC_MAX=0 OR XDEC_MAX=XDEC(I) THEN 2440 ! most significant figures.
2420  IF XDEC(I)=0 THEN XTIK$(I)=XTIK$(I)&"."
2430  XTIK$(I)=XTIK$(I)&ZERO$[1,XDEC_MAX-XDEC(I)]
2440  IF LZERO=0 THEN 2470
2450  IF XTIK(I)>0 AND XTIK(I)<1 THEN XTIK$(I)="0"&XTIK$(I)
2460  IF XTIK(I)<0 AND XTIK(I)>-1 THEN XTIK$(I)="-0"&XTIK$(I)[2]
2470 NEXT I
2480 !
2490 FOR I=1 TO Yspred/YTIK/2+1 !                    Add decimal pt & zero to
2500  IF YTIK(I)=0 THEN YTIK$(I)="0" @ GOTO 2580
2510  YTIK$(I)=VAL$ (YTIK(I)) !                      match format of Y-tick with
2520  IF YDEC_MAX=0 OR YDEC_MAX=YDEC(I) THEN 2550 ! most significant figures.
2530  IF YDEC(I)=0 THEN YTIK$(I)=YTIK$(I)&"."
2540  YTIK$(I)=YTIK$(I)&ZERO$[1,YDEC_MAX-YDEC(I)]
2550  IF LZERO=0 THEN 2580
2560  IF YTIK(I)>0 AND YTIK(I)<1 THEN YTIK$(I)="0"&YTIK$(I)
2570  IF YTIK(I)<0 AND YTIK(I)>-1 THEN YTIK$(I)="-0"&YTIK$(I)[2]
2580 NEXT I
2590 !
2600 ! LAXES -XTIK,YTIK,XMIN,YMIN,MTIK,2,5
2610  XTICKSIZE=2.5/YLOC @ YTICKSIZE=XTICKSIZE*Ymm/Xmm
2620  PEN CPEN @ SETUU @ MOVE XMIN,YMIN
2630  A=FNXTICK(1)+FNYTICK(1)+FNXTICK(-1)+FNYTICK(-1) @ FRAME ! Draw axes&ticks
2640  IF STACK THEN 2690 ELSE LORG 6 @ A=THT/YLOC/3
2650  FOR I=1 TO Xspred/XTIK/2+1 STEP MTIK/2 ! Label X-ticks
2660    MOVE XTIK(I),YMIN-A @ LABEL XTIK$(I)
2670  NEXT I
2680 !
2690  LORG 8 @ B=THT*AR/2/XLOC
2700  FOR I=1 TO Yspred/YTIK/2+1 ! Label Y-ticks
2710    IF STACK AND I=1 THEN 2740
2720 !  Slightly offset 1st y-tick to avoid overlab w. 1st x-tick
2730    MOVE XMIN-B,YTIK(I)+(I=1)*THT/9/YLOC @ LABEL YTIK$(I)
2740  NEXT I
```

```
2750 !
2760 ! AXES XTIK,0,XMIN,YMAX,2,0,5 @ AXES 0,YTIK,XMAX,YMIN,0,2,5 @ FRAME
2770  SETGU @ CSIZE LHT @ LDIR 0
2780  IF LEN (Ny$)=0 THEN Ny$=" " ! if no Y-axis label input
2790 A$=Nx$ @ GOSUB REMOVE  @ Lx=L @ NxR$=A$[1,L] ! Strip super/subscripts
2800 A$=Ny$ @ GOSUB REMOVE  @ Ly=L @ NyR$=A$[1,L]
2810 Pu1=(Pd#0)*(Pu<Pd) @ Qu1=(Qd#0)*(Qu<Qd) !   ^ or \ first indicator
2820 Pu1=(Pu*Pd#0)*(Pu<Pd)+(Pd=0) @ Qu1=(Qu*Qd#0)*(Qu<Qd)+(Qd=0)
2830 IF STACK THEN 2920
2840 LORG 1 @ MOVE (Xgu+XL-XEND)/2-LW*Lx/2,Xdsc*LHT/4 @ IF P=0 THEN LABEL Nx$ @ GOTO 2920
2850 !
2860 ! Label sub/superscripts X-axis
2870 IMOVE 0,LHT/4*(Pu1=0) ! START 1/4 CHAR-SPACE UP IF SUBSCRIPT 1st
2880 FOR I=1 TO LEN (Nx$)
2890  A=FNR(Nx$[I,I]) @ IF A THEN IMOVE 0,A*LHT/4 @ GOTO 2910
2900  LABEL Nx$[I,I] @ IMOVE LW,LHT
2910 NEXT I
2920 LDIR 90 @ LORG 6 @ MOVE 0,(Ygu+YL)/2 @ IF Q=0 THEN LABEL Ny$ @ GOTO 3000
2930 ! Label sub/superscripts Y-axis
2940 IMOVE Qu1*LHT/4,-(Ly*LW/2)
2950  FOR I=1 TO LEN (Ny$)
2960  A=FNR(Ny$[I,I]) @ IF A THEN IMOVE -(A*LHT/4),0 @ GOTO 2980
2970  LABEL Ny$[I,I] @ IMOVE -LHT,LW
2980 NEXT I
2990 !
3000 Nx$=NxR$ @ Ny$=NyR$ ! Stripped of ^ and \ characters
3010 Nx0$=Nx$[1,MIN (10,LEN (Nx$))] @ Ny0$=Ny$[1,MIN (10,LEN (Ny$))]
3020 PEN PLR=1 @ SETUU @ CLIP XMIN,XMAX,YMIN,YMAX
3030 IF PLOTTER_DUMP THEN RETURN ELSE PLOT_STARTED=1 @ GOTO STARTPLOT
3040 !
3050 STACK: STACK=1 @ BEEP @ ALPHA 1 @ DISP HGL$ (" SUPRESSED X-AXIS & X-TICK LABELS ") @ OFF CURSOR @ RETURN
3060 HARDCOPY: ALPHA 1 @ DISP HGL$ (" PRINTER PRINTOUT ") @ Pr(2)=701 @ PR=2 @ PRINTER IS Pr(2) @ OFF CURSOR @ RETURN ! Use printer
as printing device
3070 DISPLAY: ALPHA 1 @ DISP HGL$ (" CRT PRINTOUT ONLY ") @ Pr(2)=1 @ PR=1 @ PRINTER IS 1 @ OFF CURSOR @ RETURN ! Use CRT as printin
g device
3080 HARDPLOT: ALPHA 1 @ DISP HGL$ (" PLOTTER GRAPHICS ") @ PLR=PLOTDEV(2) @ HP=1 @ OFF CURSOR @ GOTO BEGIN  ! Use HP-7470A as plott
er
3090 NORMSIZE: SLD=0 @ ALPHA 1 @ DISP HGL$ (" NORMAL CHAR.-SIZE & PLOT-SPEED ") @ OFF CURSOR @ RETURN ! Use "normal" size labels & s
ymbols
3100 SLIDE: ALPHA 1 @ SLD=1 @ DISP HGL$ (" SLIDE-SIZE CHARACTERS ") @ OFF CURSOR @ RETURN ! use relatively large labels & symbols, a
s for 35mm slide.
3110 CRTPLOT: ALPHA 1 @ DISP HGL$ (" CRT GRAPHICS ") @ PLOTTER IS 1 @ PLR=1 @ HP=0 @ OFF CURSOR @ GOTO BEGIN  ! Use CRT as plotter
3120 !
3130 LZERO: BEEP @ ALPHA 1 @ DISP HGL$ (" ADD LEADING ZERO TO TICK-LABELS ") @ OFF CURSOR @ LZERO=1 @ RETURN ! Include leading zero
in front of dec pt for tick-labels
3140 GET_DATAFILE: GOSUB DATAFILE  @ GOTO BEGIN
3150 G_DATAFILE: GOSUB DATAFILE  @ GOTO RE_DATA
3160 PEN_SPEED: ALPHA 1 @ AWRIT HGL$ (" PEN SPEED (1-10) ")&"       " @ ALPHA 3 @ INPUT PENSPEED ! Define pen-speed for HP7470A plo
tter
3170     IF PENSPEED<1 OR PENSPEED>10 THEN GOSUB BAD_INPUT  @ GOTO PEN_SPEED
3180     IF PENSPEED=10 THEN SPEED$="VS" ! fast + max. acceleration
3190     IF PENSPEED<10 THEN SPEED$="VS"&VAL$ (INT (38.1*PENSPEED/10))
3200     GOTO BEGIN
3210 !
3220 STARTPLOT: NN=0 @ LDIR 0 @ PRINTER IS Pr(2) ! Re-entry line from CONT 1
3230 PLOTDATA: OFF KEY# @ OFF KYBD @ OFF ERROR @ N,Isochron,Pbpb=0 @ II,K=1
3240 RE_DATA: ALPHA 1 @ CLEAR @ ALPHA 18-2*CRT,27 @ AWRIT "press "&HGL$ (" CTRL H ")&" for help" @ OFF KYBD
3250          IF CRT THEN ALPHA 18,27 @ AWRIT "press "&HGL$ (" CTRL D ")&" for hard-plotter dump"
3260 ALPHA 21,25 @ AWRIT "PRESS ANY KEY FOR KEYBOARD-SYMBOL" @ OFF CURSOR @ PEN 0 @ SETUU @ MOVE XMIN,YMAX @ Kb=0
3270 REDIM X(MAXNUM),Y(MAXNUM)
```

```
3280                ON KEY# 1," OPEN Ebox" GOTO OEbox
3290                ON KEY# 2," OPEN Eell" GOTO OEell
3300                ON KEY# 3," POLYGON" GOTO Poly
3310                ON KEY# 4," ERRCROSS" GOTO ERRCross
3320     IF CRT THEN ON KEY# 5,"PRINT PLOT" GOTO PRINT_PLOT
3330      IF HP THEN ON KEY# 5,"" GOSUB RETRN
3340    IF NN>1 THEN ON KEY# 6,"  YORKFIT" GOTO Yorkfit
3350    IF NN<2 THEN ON KEY# 6,"" GOSUB RETRN
3360                ON KEY# 7,"  OPTIONS " GOTO OPTIONS
3370                ON KEY# 8,"SOLID Ebox" GOTO SEbox
3380                ON KEY# 9,"SOLID Eell" GOTO SEell
3390                ON KEY# 10," LETTER" GOTO LETTER
3400                ON KEY# 11,"  PEN 1" GOSUB PEN_1
3410      IF HP THEN ON KEY# 12,"  PEN 2" GOSUB PEN_2
3420     IF CRT THEN ON KEY# 12,"ERASE PEN" GOSUB ERASE_PEN
3430                ON KEY# 13," DATAFILE" GOTO G_DATAFILE
3440                ON KEY# 14,"  REDRAW" GOTO REDRAW
3450                ON KYBD KY,ALLKEY$ GOTO KYBRD
3460 PLABEL: KEY LABEL
3470        GOTO 3470
3480 !
3490 ! Select pen#1 (PEN_1) or pen#2 (PEN_2) of HP-7470A plotter
3500 PEN_1: CPEN=1 @ ALPHA 1 @ DISP HGL$ (" PEN 1 ") @ OFF CURSOR @ RETURN
3510 PEN_2: CPEN=2 @ ALPHA 1 @ DISP HGL$ (" PEN 2 ") @ OFF CURSOR @ RETURN
3520 !
3530 ERASE_PEN: CPEN=-1 @ ALPHA 1 @ DISP HGL$ (" ERASING PEN ") @ RETURN
3540 OEbox: SYMBOL=1 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" OPEN ERROR-BOX SYMBOL ") @ GOTO INPUT_DATA
3550 SEbox: SYMBOL=-1 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" SOLID ERROR-BOX SYMBOL ") @ GOTO INPUT_DATA
3560 OEell: SYMBOL=2 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" OPEN ERROR-ELLIPSE SYMBOL ") @ GOTO INPUT_DATA
3570 SEell: SYMBOL=-2 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" SOLID ERROR-ELLIPSE SYMBOL ") @ GOTO INPUT_DATA
3580 ERRCross: SYMBOL=4 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" ERROR-CROSS SYMBOL ") @ GOTO INPUT_DATA
3590 !
3600 Poly: SYMBOL=3 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" POLYGON SYMBOL ")
3610   DISP @ DISP "# SIDES (- for star), [OPEN/SOLID (0/1)  [,SIZE (1-5)  [,ROTATION (0-360)]]]";@ GOSUB RETRIEVE
3620   IF Ninputs<1 THEN GOSUB BAD_INPUT  @ GOTO 3610
3630   POLYSIDES=O(1)
3640   IF Ninputs>1 AND O(2)=1 THEN SOLID=1 ELSE SOLID=0
3650   IF Ninputs>2 THEN POLYSIZE=O(3) ELSE POLYSIZE=2.5
3660   IF Ninputs>3 THEN POLYROT=O(4) ELSE POLYROT=90+(POLYSIDES/2=INT (POLYSIDES/2))*(180/POLYSIDES-90)
3670   GOTO INPUT_DATA
3680 !
3690 KYBRD: OFF KYBD ! Keyboard-key depressed
3700        IF KY=40 OR KY=8 THEN HELP2 ! CTRL-H key
3710        IF KY=36 OR KY=4 THEN DUMP_TO_PLOTTER ! CTRL-D key
3720        IF KY<32 OR KY>122 THEN GOSUB CLUNK  @ GOTO PLABEL
3730        IF KY=19 THEN SOLID_LINE=1 @ ALPHA 1 @ DISP HGL$ (" USE SOLID LINE FOR YORKFIT LINES ") @ BEEP @ GOTO PLABEL  ! CTRL S k
ey
3740  CLEAR @ SYMBOL=5 @ OFF KYBD @ ALPHA 1 @ CLEAR @ KEY$=UPC$ (CHR$ (KY)) @ DISP HGL$ (" PLOTTING-SYMBOL IS ")&HGL$ (KEY$&" ") @ G
OTO INPUT_DATA
3750 !
3760 !
3770  INPUT_DATA: OFF CURSOR @ OFF KEY# @ OFF KYBD @ ALPHA 3 @ DISP EA$&EB$
3780  IF DFILE THEN DISP "INPUT DATAFILE SET #s AS 1st SET, LAST SET-ASTERISK (e.g. 2,12* or 2,12;15,18*)" @ DISP
3790  GOSUB INPUT_POINTS
3800  IF II>MAXNUM THEN Edit ELSE GOTO 3790
3810 !
3820  INPUT_POINTS: X(II),Y(II),Ex(II),Ey(II),R0(II)=0
3830  DISP @ DISP "#"&VAL$ (II)&":  ";@ DISP Nx0$&", ["&ER$&",]  ";Ny0$;" [,"&ER$&"] [,"&"err-corr.]  ("&EL$&" WHEN DONE)";@ LINPUT
Re$
3840 ! If datafile input, use commas to separate data-sets in a continuous series, semicolons to separate individual sets or differe
```

```
nt series.
3850 ! 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.
3860 Sc=POS (Re$,";") @ IF Sc THEN R$=Re$[1,Sc-1] ELSE R$=Re$ ! Semicolon?
3870 GOSUB RTR  @ IF Ninputs=0 THEN Edit
3880 IF NOT POS (Re$,"#") THEN DF=0 @ GOSUB KEYBOARD_INPUT  @ RETURN
3890 IF DFILE THEN DF=1 ELSE GOSUB CLUNK  @ DISP @ DISP "### NO DATAFILE IN MEMORY ###" @ GOTO 3790
3900 IF Ninputs=1 THEN S1,S2=D(1) ELSE S1=D(1) @ S2=D(2) ! 1st, last datafile set#s in a series
3910 DISP "SET#";TAB (27);Nx0$;TAB (38);ER$;TAB (51);Ny0$;TAB (62);ER$;TAB (73);"RHO" @ DISP
3920 !
3930  FOR ID=S1 TO S2 ! Transfer datafile sets to plotting array@display
3940    X(II)=S(ID,1) @ Ex(II)=S(ID,2) @ Y(II)=S(ID,3) @ Ey(II)=S(ID,4) @ R0(II)=S(ID,5) @ IF X(II)#Y(II)=0 THEN 3970
3950    DISP II;TAB (6);N$(ID);TAB (26);X(II);TAB (37);Ex(II);TAB (50);Y(II);TAB (61);Ey(II);TAB (69);R0(II) @ GOSUB 4050
3960    IF II=MAXNUM THEN Edit
3970  NEXT ID
3980  DISP @ IF Sc THEN Re$=Re$[1+Sc] @ GOTO 3860 ELSE RETURN ! Respond to ;
3990 !
4000 KEYBOARD_INPUT: Kb=1 ! Get typed-in data
4010    IF Ninputs<2 OR Ninputs=3 OR Ninputs>5 THEN GOSUB BAD_INPUT  @ GOTO 3830
4020    IF Ninputs=2 THEN X(II)=D(1) @ Y(II)=D(2) @ GOTO 4050
4030    IF Ninputs>3 THEN X(II)=D(1) @ Ex(II)=D(2) @ Y(II)=D(3) @ Ey(II)=D(4)
4040    IF Ninputs=5 THEN R0(II)=D(5)
4050  IF X(II)#Y(II)=0 THEN GOSUB CLUNK  @ DISP @ DISP HGL$ (" #### SET# "&VAL$ (II)&"  -NONZERO INPUT REQUIRED #### ") @ DISP @ DIS
P @ RETURN
4060 IF PERCENT THEN Ex=Ex(II) @ Ey=Ey(II) ELSE Ex=Ex(II)/X(II) @ Ey=Ey(II)/Y(II)
4070  IF ABS (R0(II))<= 1 AND Ex^2+Ey^2>= 2#R0(II)#Ex#Ey THEN II=II+1 @ N=N+1 @ RETURN
4080  GOSUB CLUNK  @ DISP @ DISP HGL$ (" #### ERROR-CORREL. OF "&VAL$ (R0(II))&" IS NOT POSSIBLE #### ") @ RETURN
4090 !
4100 Edit: IF Kb=0 THEN 4120 ELSE CLEAR @ DISP "SET#";TAB (8);Nx0$;TAB (20);ER$;TAB (40);Ny0$;TAB (52);ER$;TAB (62);"Rho" @ DISP
4110       FOR I=K TO N @ DISP I;TAB (8);X(I);TAB (20);Ex(I);TAB (40);Y(I);TAB (52);Ey(I);TAB (60);R0(I) @ NEXT I @ DISP
4120       IF Kb=0 AND DF AND Sc THEN PLOTT
4130       IF ABS (SYMBOL)=3 OR SYMBOL=5 THEN 4230 ELSE Ninputs,A=0
4140       FOR I=K TO N @ A=A+(Ex(I)#Ey(I)=0) @ NEXT I
4150       IF A=0 THEN 4230
4160       DISP "X-"&ER$&", Y-"&ER$&"  [, Err.-Corr.]  FOR SETS WITH UNASSIGNED OR ZERO ERRORS";
4170       GOSUB RETRIEVE  @ IF Ninputs<2 THEN GOSUB BAD_INPUT  @ GOTO 4160
4180       FOR I=K TO N
4190        IF Ninputs=3 AND R0(I)=0 THEN R0(I)=D(3)
4200        IF Ex(I)=0 THEN Ex(I)=D(1)
4210        IF Ey(I)=0 THEN Ey(I)=D(2)
4220       NEXT I @ GOTO Edit
4230       IF Kb=0 THEN PLOTT ! Don't ask for corrections if only datafile data
4240       DISP @ DISP "SET TO BE CORRECTED?  ("&EL$&" IF OK)";
4250       GOSUB RETRIEVE  @ I=D(1) @ IF I=0 THEN PLOTT ELSE II=I @ GOSUB INPUT_POINTS  @ N=N-1 @ GOTO Edit
4260 !
4270 ERROR_BOX: !                          Draw error-rectangle about data-point
4280   A=.36/Xmm/PP @ B=.36/Ymm/PP !  UU's per .36mm OR .18mm
4290   FOR J=0 TO 1000#(SYMBOL<0)
4300    K=J#A-ABS (XERR(I)) @ L=J#B-ABS (YERR(I)) @ IF K>= A OR L>= B THEN 4340
4310    CLIP X(I)+K,X(I)-K,Y(I)+L,Y(I)-L @ FRAME
4320    MOVE X(I),Y(I) @ IF SYMBOL>0 THEN 4330
4330   NEXT J
4340   CLIP XMIN,XMAX,YMIN,YMAX @ RETURN
4350 !
4360 ERROR_CROSS: !                          Draw error-cross through data-point
4370   IPLOT 0,YERR(I) @ IPLOT 0,-(2#YERR(I))
4380   IMOVE -XERR(I),YERR(I) @ IPLOT 2#XERR(I),0,-1 @ RETURN
4390 !
4400 ERROR_ELLIPSE: !                          Draw error-ellipse about data-point
4410   IF XERR(I)#YERR(I) THEN XE=XERR(I) ELSE XE=XERR(I)#1.0001
4420   IF R0(I)#1 THEN R=R0(I) ELSE R=.9999999
```

```
4430   ANGLE=.5*ATN (2*R*XE*YERR(I)/(XE^2-YERR(I)^2))
4440 ! ANGLE is an angle that the axis of the ellipse makes with the X-axis
4450   C1=2*(1-R^2)*1.224^2 @ C2=1/COS (2*ANGLE)
4460   Vx=XE^2 @ Vy=YERR(I)^2
4470   A=SQR (C1/((1+C2)/Vx+(1-C2)/Vy)) ! Length of major axis of ellipse
4480   B=SQR (C1/((1-C2)/Vx+(1+C2)/Vy)) ! Length of minor axis of ellipse
4490   St=A/INT (A/(Xspred/800)+1) !  Step-size along X-axis
4500   St=A/INT (A/St+.0001) !        force step to divide evenly
4510   SIN_ANGLE=SIN (ANGLE) @ COS_ANGLE=COS (ANGLE)
4520    FOR K=1 TO -1 STEP -2
4530     FOR X=K*A TO -(K*A)-K*St STEP -(K*St)
4540      IF (K=1)*(X<-(A*1.001)) OR (K=-1)*(X>A*1.001) THEN 4600
4550      Z=1-(X/A)^2 @ IF Z>= 0 THEN Y=K*B*SQR (Z) ELSE Y=0
4560      RPLOT X*COS_ANGLE-Y*SIN_ANGLE,X*SIN_ANGLE+Y*COS_ANGLE
4570      IF SYMBOL<0 THEN K=-K @ RPLOT X*COS_ANGLE+Y*SIN_ANGLE,X*SIN_ANGLE-Y*COS_ANGLE
4580     NEXT X
4590     IF SYMBOL<0 THEN RETURN
4600    NEXT K
4610    RETURN
4620 !
4630 POLYGON: SETGU ! Draw open or filled polygon or star symbol
4640          IF PLOTTER_DUMP THEN POLYSIDES=SYMBOL(JJ,2) @ SOLID=SYMBOL(JJ,3) @ POLYSIZE=SYMBOL(JJ,4) @ POLYROT=SYMBOL(JJ,5)
4650          FOR A=0 TO (HP AND SLD=0 AND POLYSIDES>0 AND POLYSIZE>3)+50*SOLID
4660           B=PS*(POLYSIZE-A/2)*PP @ S=360/ABS (POLYSIDES) @ E=.5-(ABS (POLYSIDES)=3)/5
4670           IF B<0 THEN SETUU @ RETURN
4680           FOR J=POLYROT TO POLYROT+360 STEP S
4690            RPLOT B*COS (J),B*SIN (J)
4700            IF POLYSIDES<0 THEN C=J+S/2 @ RPLOT E*B*COS (C),E*B*SIN (C)
4710           NEXT J
4720          NEXT A
4730          SETUU @ RETURN
4740 !
4750 KEYBOARD_SYMBOL: IF PLOTTER_DUMP THEN KEY$=CHR$ (SYMBOL(JJ,2))
4760                  IF KEY$="+" THEN B=1.5 @ C=AR*CRT+HP/2 ELSE B=0 @ C=1
4770                  IF KEY$="*" THEN C=CRT+.6*HP
4780                  LORG 5 @ CSIZE 3.5-HP+1.5*SLD+B,C,0 @ LABEL KEY$ @ RETURN
4790 !
4800 PLOTT: PEN CPEN @ C_ROW=CURSROW  !                Draw data-point symbols
4810        FOR I=K TO N
4820         IF HP THEN ALPHA C_ROW+1 @ DISP I;TAB (10);X(I);TAB (20);Y(I) @ OFF CURSOR @ GOTO 4870 ELSE C_CT=1+C_CT
4830         CACHE(C_CT,1)=X(I) @ CACHE(C_CT,2)=Ex(I) @ CACHE(C_CT,3)=Y(I) @ CACHE(C_CT,4)=Ey(I) @ CACHE(C_CT,5)=R0(I) @ SYMBOL(C_CT
,1)=SYMBOL
4840         IF SYMBOL=3 THEN SYMBOL(C_CT,2)=POLYSIDES @ SYMBOL(C_CT,3)=SOLID @ SYMBOL(C_CT,4)=POLYSIZE @ SYMBOL(C_CT,5)=POLYROT
4850         IF SYMBOL<3 AND SYMBOL<5 THEN SYMBOL(C_CT,1)=SYMBOL
4860         IF SYMBOL=5 THEN SYMBOL(C_CT,2)=KY
4870         IF PERCENT THEN XERR(I)=X(I)*Ex(I)/100 @ YERR(I)=Y(I)*Ey(I)/100
4880         IF PERCENT=0 THEN XERR(I)=Ex(I) @ YERR(I)=Ey(I)
4890         IF X(I)<XMIN OR Y(I)<YMIN OR X(I)>XMAX OR Y(I)>YMAX THEN 4920
4900         MOVE X(I),Y(I)
4910         ON ABS (SYMBOL) GOSUB ERROR_BOX ,ERROR_ELLIPSE ,POLYGON ,ERROR_CROSS ,KEYBOARD_SYMBOL
4920        NEXT I
4930        IF PLOTTER_DUMP=0 THEN MOVE XMIN,YMAX
4940        IF HP=1 AND PLOTTER_DUMP=0 THEN PEN 0
4950        IF PLOTTER_DUMP THEN RETURN
4960 !
4970 ADD: CLEAR @ IF CRT THEN ALPHA @ DISP "PRESS "&HGL$ (" A/G ")&" TO VIEW PLOT"
4980      NN=N @ DISP @ DISP "PRESS "&HGL$ (" RETAIN ")&" TO INCLUDE THESE POINTS WITH OTHERS FOR A YORKFIT." @ DISP
4990      DISP "PRESS "&HGL$ (" CONTINUE")&" TO CONTINUE." @ DISP
5000 IF N>1 THEN DISP "PRESS "&HGL$ (" YORKFIT ")&" FOR YORKFIT. "
5010 !
```

```
5020      GOSUB CLEARKEY
5050      ON KEY# 4,"  HELP" GOTO HELP4
5060      IF NN>1 THEN ON KEY# 5," YORKFIT" GOTO Yorkfit
5070      IF NN<2 THEN ON KEY# 5,"" GOSUB RETRN
5080      ON KEY# 6,"  RETAIN" GOTO RETAIN
5090      ON KEY# 7," CONTINUE" GOTO PLOTDATA
5170      OFF CURSOR @ KEY LABEL
5180      ON KYBD KY,ALLKEY$ GOSUB RETRN
5190      GOTO 5190
5200 !
5210 RETAIN: OFF KEY# @ OFF KYBD @ II,K=N+1 @ GOTO RE_DATA  ! Add next points to existing sequence
5220 OPTIONS: CLEAR @ GOSUB CLEARKEY
5230      IF CRT THEN ON KEY# 1,"PRINT PLOT" GOTO PRINT_PLOT
5240       IF HP THEN ON KEY# 1,"" GOSUB RETRN
5250                  ON KEY# 2,"ABS ERRORS" GOSUB ABS_ERRS
5260                  ON KEY# 3," % ERRORS" GOSUB PERCENT_ERRS
5270                  ON KEY# 4,"1-SIG ERRS" GOSUB ONE_SIG
5280                  ON KEY# 5,"2-SIG ERRS" GOSUB TWO_SIG
5290                  ON KEY# 7," ESCAPE" GOTO RE_DATA
5300                  ON KEY# 6,"STORE PLOT" GOSUB STORE_PLOT
5340                  ON KEY# 11,"  HELP" GOTO HELP3
5370                  ON KEY# 14," POLYFIT" GOTO REGRESS
5380                  ON KYBD KY,ALLKEY$ GOSUB RETRN
5390                  KEY LABEL
5400                  GOTO 5400
5410 !
5420 ABS_ERRS: PERCENT=0 @ EB$=" ABS.-ERRORS" @ DISP EB$ @ BEEP @ ER$="err" @ GOTO KRETRN
5430 PERCENT_ERRS: PERCENT=1 @ EB$=" %-ERRORS" @ DISP EB$ @ BEEP @ ER$="%err" @ GOTO KRETRN
5440 ONE_SIG: SIG=1 @ BEEP @ EA$="1-SIGMA" @ DISP EA$&" ERRORS" @ GOTO KRETRN
5450 TWO_SIG: SIG=2 @ BEEP @ EA$="2-SIGMA" @ DISP EA$&" ERRORS" @ GOTO KRETRN
5460 STORE_PLOT: DISP "FILE-NAME FOR THIS PLOT";@ INPUT FILE$@ GSTORE FILE$ @ DISP @ DISP "GRAPHICS-FILE '"&FILE$&"' STORED." @ DISP
 @ GOTO KRETRN
5470 !
5480 !
5490 ! ##########################################################################
5500 Yorkfit: ! LINE-FITTING ALGORITHM MODIFIED FROM D. YORK,  EPSL V. 5, P. 320-324, 1969
5510 ! ##########################################################################
5520 !
5530 OFF KYBD @ OFF KEY# @ IF N>MAXYOR THEN GOSUB CLUNK  @ DISP "(";MAXYOR;"POINTS FOR YORKFIT" @ WAIT 2000 @ GOTO RE_DATA
5540   OFF KEY# @ N=NN @ XERR,YERR=0 @ CLEAR @ DISP "YORKFIT:" @ DISP
5550 ! CHECK FOR UNASSIGNED ERRORS & ERROR-CORRELATIONS
5560   A,B,RHO=0 @ FOR I=1 TO N @ A=A+(Ex(I)*Ey(I)=0) @ B=B+(RO(I)=0) @ NEXT I
5570   IF A=0 AND B=0 THEN 5720
5580   IF A=0 OR B>0 THEN 5610
5590   CLEAR @ DISP "%X-ERRS. %Y-ERRS FOR SAMPLES WITH UNASSIGNED OR ZERO ERRORS";@ GOSUB RETRIEVE  @ IF Ninputs=0 THEN GOSUB BAD_IN
PUT  @ GOTO 5590
5600   XERR=O(1) @ YERR=O(2)*(O(2)#0)+O(1)*(O(2)=0) @ GOTO 5660
5610   IF A=0 OR B=0 THEN 5640
5620   DISP "%X-ERRS, %Y-ERRS [,ERR-CORRS] FOR SAMPLES WITH UNASSIGNED OR ZERO VALUES";@ GOSUB RETRIEVE  @ IF Ninputs=0 THEN GOSUB B
AD_INPUT  @ GOTO 5620
5630   XERR=O(1) @ YERR=O(2)*(O(2)#0)+O(1)*(O(2)=0) @ RHO=O(3) @ GOTO 5660
5640   DISP "ERR-CORRS FOR SAMPLES WITH UNASSIGNED OR ZERO VALUES ("&EL$&" IF ZERO)";
5650   GOSUB RETRIEVE  @ RHO=O(1) @ IF ABS (RHO)>1 THEN GOSUB BAD_INPUT  @ GOTO 5640
5660   FOR I=1 TO N
5670    IF Ex(I)=0 THEN Ex(I)=XERR @ XERR(I)=XERR*(PERCENT*X(I)/100+(PERCENT=0))
5680    IF Ey(I)=0 THEN Ey(I)=YERR @ YERR(I)=YERR*(PERCENT*Y(I)/100+(PERCENT=0))
5690    IF RO(I)=0 THEN RO(I)=RHO
5700   NEXT I
5710 !
5720  !  Determine if isochron data by looking for isotope #s in X & Y names
```

```
5730 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
5740 RESTORE 5730
5750 FOR I=1 TO 7
5760  MAT READ ISOTOPE
5770  A=POS (Ny$,VAL$ (ISOTOPE(1)))$POS (Ny$,VAL$ (ISOTOPE(2)))$POS (Nx$,VAL$ (ISOTOPE(3)))$POS (Nx$,VAL$ (ISOTOPE(4)))
5780  IF A AND I<= 5 THEN Isochron=I ! Type of std isochron (Rb-Sr,Sm/Nd...)
5790  IF A AND I>= 6 THEN Pbpb=I-5 !   Normal or inverse Pb-Pb isochron
5800 NEXT I
5810 !
5820  IF PR=2 THEN DISP "NAME FOR THIS YORKFIT";@ INPUT Name$[1,50]
5830  PRINTER IS Pr(2) @ CLEAR
5840  PRINT RPT$ ("*",30)&"YORKFIT"&RPT$ ("*",30)
5850  IF PR=2 THEN PRINT Name$ ELSE PRINT
5860  PRINT @ PRINT TAB (25);EA$&EB$
5870  PRINT "SET#";TAB (9);Nx0$;TAB (23);ER$;TAB (34);Ny0$;TAB (47);ER$;TAB (61);"Err-Correl." @ PRINT
5880 REDIM X(N),Y(N),Z(N),T1(N),T2(N),U(N),V(N),U2(N),V2(N),UV(N),Z2(N)
5890  Ok=0 @ Slope=.1 !    Ok Remains zero until a model-1 fit is obtained
5900  M=1 @ ITERMAX=15 !   Model-number, Max# of iterations
5910  M3_COUNT=0 !         MODEL-3 ITERATION COUNTER
5920  M5=0 !               Don't use Model-5.
5930  Slope0=Slope
5940  Iteration=0 !        Iteration-counter
5950  Slope=Slope0
5960  Iteration=1+Iteration
5970  IF Iteration>ITERMAX AND M#5 THEN Fail
5980  IF Iteration>ITERMAX AND M=5 THEN PRINT "UNCONVERGED MODEL-5 FIT" @ GOTO 6440
5990  IF Iteration>1 AND M=1 THEN 6260
6000  FOR I=1 TO N
6010   IF M=1 THEN PRINT I;TAB (8);X(I),TAB (22);Ex(I);TAB (33);Y(I);TAB (46);Ey(I);TAB (60);R0(I) @ GOTO 6200
6020 !
6030   ! Model-2 wting: Equal wts,  uncorrell. error (Implies knowing nothing about the cause of scatter).
6040   IF M=2 THEN X_WT(I)=1 @ Y_WT(I)=1/Slope^2 @ R(I)=0 @ GOTO 6230
6050   IF M#3 THEN 6110
6060 !
6070 ! Model-3 Weighting - anal. errors+init.-Y variation  (Implies scatter is due to analytical error plus a normally-distributed v
ariation i initial-Y)
6080   T=(YERR(I)/SIG)^2
6090   R(I)=R0(I)*SQR (T/(T+Sig_Y_Init^2)) @ Y_WT(I)=1/(T+Sig_Y_Init^2) @ GOTO 6230
6100 !
6110 ! MODEL-5 WEIGHTING - USE OBSERVED X- & Y-RESIDUALS TO ESTIMATE X- AND Y-ERRORS OF EACH POINT & ASSUME ZERO ERROR-CORRELATION
6120   Xres=-(Slope*Z(I)/X_WT(I)*(Inter+Slope*X(I)-Y(I)))
6130   Yres=Z(I)/Y_WT(I)*(Inter+Slope*X(I)-Y(I))
6140 ! IF RESIDUAL IS WITHIN .3-SIGMA(ANAL.), USE ANAL. ERROR
6150   A=1/Xres^2 @ IF A<11*XWT(I) THEN X_WT(I)=A ELSE X_WT(I)=XWT(I)
6160   A=1/Yres^2 @ IF A<11*YWT(I) THEN Y_WT(I)=A ELSE Y_WT(I)=YWT(I)
6170   R(I)=0 @ GOTO 6230
6180   !
6190 ! Model-1 weighting - analytical errs. only (York's original)
6200   Y_WT(I),YWT(I)=(SIG/YERR(I))^2
6210   X_WT(I),XWT(I)=(SIG/XERR(I))^2 @ R(I)=R0(I)
6220 !
6230   A(I)=SQR (X_WT(I)*Y_WT(I))
6240  NEXT I
6250  IF M=1 AND Iteration=1 THEN C_ROW=CURSROW +1 @ ALPHA C_ROW @ AWRIT RPT$ (" ",80)
6260  FOR I=1 TO N
6270   Z(I)=X_WT(I)*Y_WT(I)/(Slope^2*Y_WT(I)+X_WT(I)-2*Slope*R(I)*A(I))
6280  NEXT I
6290  C1=SUM (Z) @ MAT T1=Z.X@ MAT T2=Z.Y
6300  Xbar=SUM (T1)/C1 @ Ybar=SUM (T2)/C1
6310  C2,D,E=0
```

```
6320  MAT U=( -Xbar)+X@ MAT V=( -Ybar)+Y@ MAT U2=U.U@ MAT V2=V.V
6330  MAT Z2=Z.Z@ MAT UV=U.V
6340  FOR I=1 TO N
6350   C2=C2+(U2(I)/Y_WT(I)-V2(I)/X_WT(I))*Z2(I)
6360   D=D+(UV(I)/X_WT(I)-R(I)*U2(I)/A(I))*Z2(I)
6370   E=E+(UV(I)/Y_WT(I)-R(I)*V2(I)/A(I))*Z2(I)
6380  NEXT I
6390 Slope0=C2^2+4*D*E
6400 IF Slope0<0 THEN Fail
6410 Slope0=(SQR (Slope0)-C2)/(2*D)
6420 ALPHA C_ROW @ AWRIT "MODEL "&VAL$ (M)&":    Iteration "&VAL$ (Iteration)&"     Slope= "&VAL$ (Slope0)
6430 IF ABS ((Slope0-Slope)/Slope0)>.0001 THEN 5950
6440 Slope(M),Slope=Slope0 @ Inter(M),Inter=Ybar-Slope*Xbar
6450 IF M=3 THEN M3_COUNT=1+M3_COUNT .
6460 IF M=3 AND M3_COUNT>30 THEN Fail
6470 !
6480 ! Error algorithm of Titterington & Halliday, Chem Geol v 26, p 183
6490 D,E,SUMS=0
6500 FOR I=1 TO N
6510  Y=Y(I)-Slope*X(I)-Inter @ SUMS=SUMS+Z(I)*Y^2
6520  X=X(I)-Z(I)*Y*(R(I)*A(I)-Slope*Y_WT(I))/(Y_WT(I)*X_WT(I))
6530  D=D+X*Z(I) @ E=E+X^2*Z(I)
6540 NEXT I
6550 !
6560 V=N-2 ! Degrees of freedom
6570 MSWD=(V<> 0)*SUMS/(V+(V=0)) !      Mean square of weighted deviates
6580 IF M<3 OR ABS (MSWD-1)<.01 THEN 6610
6590  Sig_Y_Init=Sig_Y_Init*SQR (MSWD) ! Estimated variation in initial ratio required to account for scatter
6600  ALPHA C_ROW+1 @ AWRIT "Try "&VAL$ (M3_COUNT)&":  MSWD= "&VAL$ (FNDr(MSWD,3))&" for init-Y var. of "&VAL$ (FNDr(2*Sig_Y_Init,3)
) @ GOTO 5940
6610  St_t=(V<2)*12.7+(V>1)*1.96*V/SQR (V^2-2.43*V+1.696) ! Student's-t approximation (from Andy Turek)
6620 ErrSl_APr=C1/(E*C1-D^2) @ IF ErrSl_APr>0 THEN ErrSl_APr=SQR (ErrSl_APr) ELSE ErrSl_APr=0
6630 ErrInt_APr=ErrSl_APr*SQR (E/C1)
6640 ErrInt_IncSc=ErrInt_APr*SQR (MSWD) @ IF M=1 THEN Sig_Y_Init=ErrInt_IncSc
6650 ErrSl_IncSc=ErrSl_APr*SQR (MSWD)
6660 DISP
6670 IF M>1 THEN YORKPRINT
6680 !
6690 IF SUMS>80 THEN Prob=0 @ GOTO YORKPRINT
6700 D,A=1 ! Calculate chi-square probability
6710  IF INT (V/2)=V/2 THEN 6780
6720  FOR I=2 TO 2000 STEP 2 ! V odd
6730   D=D*SUMS/(V+I) @ A=A+D @ IF D<.000001 THEN 6750
6740  NEXT I
6750  D=SQR (PI )
6760  FOR I=.5 TO V/2 @ D=D*I @ NEXT I
6770  Prob=1-EXP (-(SUMS/2))*(SUMS/2)^(V/2)*A/D @ GOTO YORKPRINT
6780  FOR I=1 TO V/2-1 @ D=D*SUMS/(2*I) @ A=A+D @ NEXT I ! V even
6790  Prob=A*EXP (-(SUMS/2))
6800 !
6810 YORKPRINT: FOR Pr=1 TO PR
6820  PRINTER IS Pr(Pr) @ IF Pr=1 THEN CLEAR
6830  ON M GOTO 6840,6880,6860,6830,6900
6840  PRINT RPT$ ("-",80) @ PRINT "**MODEL 1 SOLUTION -- ASSUMES ALL SCATTER DUE TO ANALYTICAL ERROR**" @ Ok=1
6850  IF Prob>.15 THEN Errsl(1)=1.96*ErrSl_APr @ Errint(1)=1.96*ErrInt_APr @ GOTO 6920 ELSE GOTO 6910
6860  PRINT "*********MODEL 3 SOLUTION -- ASSUMES SCATTER IS DUE TO ANALYTICAL ERROR" @ PRINT "******PLUS NORMALLY-DISTRIBUTED ERROR
 IN INITIAL ";Ny$;"******"
6870  PRINT @ PRINT "CALCULATED VARIATION IN INITIAL ";Ny$;"=";FNDr(2*Sig_Y_Init,3);"  (2-SIGMA)" @ GOTO 6910
6880 PRINT @ PRINT "**********MODEL 2  SOLUTION -- EQUALLY WEIGHTED POINTS WITH UNCORRELATED"
6890   PRINT "*******************ERRORS; ASSUMES NORMAL DISTRIBUTION OF RESIDUALS*****" @ GOTO 6910
```

```
6900 PRINT @ PRINT "**MODEL 5  SOLUTION -- POINTS WEIGHTED BY OBSERVED SCATTER FROM BEST-FIT LINE**"
6910  Errsl(M)=St_t*ErrSl_IncSc @ Errint(M)=St_t*ErrInt_IncSc
6920  PRINT @ PRINT TAB (33);"SLOPE";TAB (57);"INTERCEPT" @ PRINT TAB (33);"-----";TAB (57);"---------" @ PRINT
6930  PRINT "BEST-FIT:";TAB (31);FNDr(Slope,6);TAB (56);FNDr(Inter,6) @ PRINT RPT$ ("-",80)
6940   IF M=1 THEN PRINT "ERROR (1-SIGMA, A PRIORI) ";TAB (31);FNDr(ErrSl_APr,4);TAB (56);FNDr(ErrInt_APr,4) @ PRINT
6950   PRINT "ERROR (1-SIGMA, FROM SCATTER) ";TAB (31);FNDr(ErrSl_IncSc,4);TAB (56);FNDr(ErrInt_IncSc,4) @ PRINT
6960   PRINT "ERROR (95% CONFIDENCE LIMITS)";TAB (31);FNDr(Errsl(M),4);TAB (56);FNDr(Errint(M),4) @ PRINT
6970   PRINT "COORDINATES OF CENTROID: X=";FNDr(Ibar,6);"Y=";FNDr(Ybar,6)
6980 IF M>1 THEN 7010
6990   PRINT @ PRINT "ANALYTICAL ERRORS ALONE WILL CAUSE THE OBSERVED AMOUNT OF SCATTER"
7000   PRINT "OR MORE ";VAL$ (FNDr(100*Prob,2));"% OF THE TIME, AS INDICATED BY THE M.S.W.D. OF";FNDr(MSWD,3)
7010   PRINT RPT$ ("-",80) @ PRINT
7020 NEXT Pr
7030 C_ROW=CURSROW
7040 IF M5=0 AND M=1 AND Prob(= .2 THEN M=2+(Isochron>0) @ Sig_Y_Init=ErrInt_IncSc @ AWRIT RPT$ (" ",80) @ AWRIT "MODEL "&VAL$ (M)&"
 YORKFIT..." @ GOTO 5940
7050 CHOOSE_MODEL: IF M>1 THEN DISP @ DISP "PRESS KEY TO CHOOSE MODEL-1 OR MODEL-"&VAL$ (M)&" SOLUTION." @ DISP @ DISP @ GOSUB CLEAR
KEY  ELSE 7190
7060   ON KEY# 6," MODEL-1" GOTO 7160
7070   ON KEY# 7," MODEL-"&VAL$ (M) GOTO 7170
7110   ON KEY# 4,"  HELP" GOTO HELP5
7130   ON KYBD KY,ALLKEY$ GOSUB RETRN
7140   KEY LABEL
7150   GOTO 7150
7160   OFF KEY# @ M=1 @ GOTO 7190
7170   OFF KEY# @ GOTO 7190
7180 !
7190 Slope=Slope(M) @ Inter=Inter(M) @ Errsl=Errsl(M) @ Errint=Errint(M) @ OFF KYBD
7200 ON 1+Pbpb GOTO 7210,7270,7260
7210 IF Isochron=0 THEN 7410
7220 Age=.000001*LOG (1+Slope)/L(Isochron) ! standard-isochron age
7230 Age_err=.000001*Errsl/(L(Isochron)*(1+Slope))
7240 GOTO PRINT_AGE
7250 !
7260 Slope=Inter @ Errsl=Errint
7270 Sl=Slope @ DISP "CALCULATING 207/206 AGE & ERRORS" @ GOSUB Pb_Pb_AGE
7280 Age=.000001*A76 @ Sl=Slope+Errsl @ GOSUB Pb_Pb_AGE
7290 A1=.000001*A76 @ Sl=Slope-Errsl @ GOSUB Pb_Pb_AGE
7300 A2=.000001*A76
7310 Age_err=(A1<> 0)*(A2<> 0)*ABS ((A1-A2)/2)
7320 !
7330 PRINT_AGE: FOR Pr=1 TO PR
7340             PRINTER IS Pr(Pr)
7350             IF Age THEN PRINT "MODEL";M;" ISOCHRON AGE =";FNDr(Age,5);
7360             IF Age_err THEN PRINT " +/-";FNDr(Age_err,3);
7370             IF Age THEN PRINT " Ma" @ PRINT @ PRINT RPT$ ("*",80)
7380            NEXT Pr
7390 C=CURSROW  @ CLEAR @ ALPHA C-4*(C>3) @ ALPHA C+1 @ OFF CURSOR
7400 !
7410 GOSUB DRAWLINE  @ BEEP @ GOTO KEYS
7420 !
7430 Fail: PRINT @ PRINT "DATA CANNOT BE FIT TO A STRAIGHT LINE USING MODEL";M @ PRINT @ PRINT
7440   IF M=1 THEN M=2+(Isochron>0) @ GOTO 5910
7450   IF Ok=0 THEN GOSUB CLUNK  @ GOTO KEYS  ELSE M=1 @ GOTO 7190
7460 !
7470 Pb_Pb_AGE: A76=0 @ IF Sl(.0156 OR Sl>1.9 THEN DISP @ DISP "**** CANNOT CALCULATE AN AGE FOR 207/206=";Sl;"****" @ DISP @ DISP @
 RETURN
7480   S=4500000000*(SGN (-(1/2)+(Sl>L(4)/(L(3)*137.88)))+(Sl>.7))
7490   F=L(3)*EXP (S*(L(3)-L(4)))/L(4)
7500   A76=LOG (1+(EXP (S*L(3))-1-F*(EXP (S*L(4))-1))/(1/(137.88*Sl)-F))/L(4)
```

```
7510   IF ABS (A76-S)<1000 THEN RETURN ELSE S=A76 @ GOTO 7490
7520 !
7530 PRINT_PLOT: OFF KEY# ! Dump CRT-plot to printer
7540       ALPHA 1 @ CLEAR @ ALPHA 21,36 @ DISP "WHICH PRINTER-MODEL?" @ DISP @ DISP @ GOSUB CLEARKEY
7550       ON KEY# 4," HP-82905" GOTO 7720
7560       ON KEY# 5,"  HP-2673" GOTO 7730
7600       ON KEY# 7,"  ESCAPE" GOTO 7740
7690       ON KYBD KY,ALLKEY$ GOSUB RETRN
7700       KEY LABEL
7710       GOTO 7710
7720    OFF KEY# @ PRINTER IS 701 @ DUMP GRAPHICS @ GOTO 7740
7730    OFF KEY# @ PRINTER IS 701 @ DUMP GRAPHICS 0,0,0,1 @ GOTO 7740
7740    PRINTER IS Pr(2) @ GOTO RE_DATA
7750 !
7760 DRAWLINE: ! Draw best-fit line, using dashed line-type
7770           NLINES=1+NLINES @ IF CRT THEN LINE(NLINES,1)=Slope @ LINE(NLINES,2)=Inter
7780           IF SOLID_LINE=0 THEN LINE TYPE 4+NLINES
7790           CLIP XMIN,XMIN+.97*Xspred,YMIN,YMIN+.97*Yspred
7800           MOVE XMIN,XMIN*Slope+Inter @ PEN CPEN
7810           DRAW XMAX,XMAX*Slope+Inter
7820           MOVE XMIN,YMIN @ LINE TYPE 1 @ PEN CRT
7830           CLIP XMIN,XMAX,YMIN,YMAX @ RETURN
7840 !
7850 KEYS: IF CRT THEN OFF CURSOR @ DISP "PRESS "&HGL$ (" A/G ")&" TO VIEW PLOT " @ DISP @ DISP @ DISP @ OFF CURSOR @ GOSUB CLEARKEY

7860               ON KEY# 1," RESTART" GOTO BEGIN
7870               ON KEY# 2,"ADD POINTS" GOTO RETAIN
7880   IF CRT THEN ON KEY# 3,"PRINT PLOT" GOTO PRINT_PLOT
7900               ON KEY# 4,"  HELP" GOTO HELP6
7930               ON KEY# 7,"CONTINUE" GOTO PLOTDATA
8010               ON KYBD KY,ALLKEY$ GOSUB RETRN
8020       KEY LABEL
8030       GOTO 8030
8040 !
8050 LETTER: OFF KEY# @ OFF KYBD @ CLEAR ! Draft a phrase on the plot
8060         UNCLIP @ SETUU @ IF PLOTTER_DUMP THEN 8100
8070         DISP @ DISP "(PRESS "&EL$&" TO ESCAPE)" @ DISP
8080         DISP "CHAR.-HEIGHT (tick labels were";FNDr(THT*Ymm/YLOC/2,2);")[, SLANT (0-30) [, ROTATION]]";
8090    GOSUB RETRIEVE  @ IF Ninputs THEN H=0(1) @ SL=0(2) @ ROT=0(3) ELSE GOTO RE_DATA
8100         PEN CPEN @ CS=2*H*YLOC/Ymm @ W=AR*CS*(1-HP/3.1) @ CSIZE CS,AR,SL @ LDIR ROT
8110         IF HP THEN 8150
8120 DISP @ DISP "ENTER COORDINATES (X, Y)  OF LOWER-LEFT CORNER OF PHRASE";
8130         GOSUB RETRIEVE  @ IF Ninputs<2 THEN GOSUB BAD_INPUT  @ GOTO 8120
8140         X=0(1) @ Y=0(2) @ PEN CPEN @ GOTO 8170
8150         IF PLOTTER_DUMP THEN 8170 ELSE DISP @ DISP "POSITION PEN WITH PLOTTER-CONTROLS, THEN PRESS "&CNT$ @ PAUSE
8160         CURSOR X,Y
8170         MOVE X,Y @ LORG 1
8180         IF PLOTTER_DUMP=0 THEN DISP @ DISP "PHRASE TO BE DRAFTED";@ LINPUT P$
8190         L=LEN (P$) @ SETGU @ IF HP THEN 8230
8200         NPHRASES=1+NPHRASES @ LETTER$(NPHRASES)=P$ @ LETTER(NPHRASES,1)=X @ LETTER(NPHRASES,2)=Y
8210         LETTER(NPHRASES,3)=H @ LETTER(NPHRASES,4)=SL @ LETTER(NPHRASES,5)=ROT
8220         IF CRT AND CPEN=-1 THEN NPHRASES=NPHRASES-2
8230         IF POS (P$,"\")=0 AND POS (P$,"^")=0 AND POS (P$,"+/-")=0 THEN LABEL P$ @ GOTO 8310
8240         IF ROT#0 THEN GOSUB CLUNK  @ DISP @ DISP "CAN'T USE \, ^, OR +/- SYMBOLS WITH NONZERO ANGLES." @ DISP @ GOTO 8070
8250         FOR I=1 TO L
8260          IF I>L-2 THEN 8280
8270          IF P$[I,I+2]="+/-" THEN GOSUB PM  @ I=I+2 @ GOTO 8300
8280          A=FNR(P$[I,I]) @ IF A THEN IMOVE 0,A*CS/4 @ GOTO 8300
8290          LABEL P$[I,I] @ IMOVE W,CS
8300         NEXT I
```

```
8310        PEN CRT @ SETUU @ LDIR 0 @ IF PLOTTER_DUMP THEN RETURN ELSE 8070
8320 !
8330 PM: IMOVE 0,.13*CS @ LABEL "+" @ IMOVE 0,.62*CS @ LABEL "-" @ IMOVE W,1.25*CS @ RETURN ! Label +/- symbol
8340 !
8350 DATAFILE: CLEAR @ DISP "NAME OF DATA-FILE TO BE RETRIEVED?"
8360  DISP @ DISP "(to get a disk catalog, enter "&H6L$ (" C ")&" for drive 0, "&H6L$ (" c ")&" for drive 1)"
8370  IF ALL THEN DISP @ DISP "(press "&EL$&" for more data from ";FILE$
8380  DISP @ GOSUB RETRIEVE_STR
8390  IF Ninputs=0 THEN GOTO FINISH
8400  IF O$(1)="C" THEN CAT ":D700" @ GOTO 8420
8410  IF O$(1)="c" THEN CAT ":D701" ELSE 8440
8420  DISP @ DISP "PRESS "&CNT$&" TO CONTINUE." @ PAUSE
8430  GOTO DATAFILE
8440  FILE$=O$(1) @ ON ERROR GOTO WRONGFILE
8450  ASSIGN# 1 TO  FILE$ @ ON ERROR GOTO YORVERT
8460  READ# 1,1 ; N$(1),XX()@ OFF ERROR
8470 !
8480 ! ############### RETRIEVE nX45 NUMERIC DATA-FILE ROWS FROM TAPE ##########
8490 !
8500  DISP @ DISP "COLUMN #s FOR X, [%Xerr,]  Y  [,%Yerr] [, Err-Correl]";@ GOSUB RETRIEVE  @ MAT COLN=ZER
8510  IF Ninputs=2 THEN COLN(1)=O(1) @ COLN(3)=O(2)
8520  IF Ninputs>3 THEN COLN(1)=O(1) @ COLN(2)=O(2) @ COLN(3)=O(3) @ COLN(4)=O(4) @ COLN(5)=O(5)
8530  IF Ninputs<2 OR Ninputs>5 OR Ninputs=3 THEN GOSUB BAD_INPUT  @ GOTO 8500
8540  CLEAR @ PRINT "SET#";TAB (8);"NAME";TAB (28);"X";TAB (37);"%Err";TAB (49);"Y";TAB (58);"%Err";TAB (70);"RHO" @ PRINT
8550 GOSUB NAMEBLANK
8560  ON ERROR GOTO DATAFILE_RETURN
8570  FOR I=1 TO MAXNUM
8580   READ# 1,I ; N$(I),XX()
8590   FOR J=1 TO 5
8600    IF COLN(J) THEN S(I,J)=XX(COLN(J)) ELSE S(I,J)=0
8610   NEXT J
8620   PRINT I;TAB (6);N$(I);TAB (26);S(I,1);TAB (37);S(I,2);TAB (47);S(I,3);TAB (58);S(I,4);TAB (68);S(I,5)
8630  NEXT I
8640 DATAFILE_RETURN: OFF ERROR @ DFILE=1 @ PRINT @ DISP @ DISP "PRESS "&CNT$&" TO CONTINUE" @ PAUSE
8650 RETURN
8660 !
8670 YORVERT: ! Convert VISICALC files to 100X5 arrays for plotting/Yorkfit
8680 !
8690 OFF ERROR
8700 DIM VC$[24],VC(300,50),I$[20],J$[20],PARAM$(5)[5]
8710 DATA "X","X-ERR","Y","Y-ERR","RHO"
8720 RESTORE 8710 @ READ PARAM$(1),PARAM$(2),PARAM$(3),PARAM$(4),PARAM$(5)
8730 CLEAR @ B_COL=0 @ DISP "INPUT   BEGINNING, ENDING   ROW-NUMBERS CONTAINING SAMPLE-DATA";@ GOSUB RETRIEVE
8740 IF Ninputs#2 THEN GOSUB BAD_INPUT  @ GOTO 8730 ELSE RMIN=O(1) @ RMAX=O(2)
8750 ALL=0 @ DISP @ DISP "(enter "&H6L$ (" ALL ")&" if ALL columns are to be converted)"
8760 DISP "(press "&EL$&" for list of column names)" @ GOSUB COLINPUT  @ GOTO 8880
8770 !
8780 COLINPUT: DISP @ DISP "INPUT COLUMNS (e.g. C,D,AK...) FOR: X, [%Xerr,]  Y [,%Yerr]  [,Rho(X-Y)]:"
8790  GOSUB RETRIEVE_STR  @ IF POS (UPC$ (O$(1)),"ALL") THEN ALL=1 @ NCOLS=45 @ CNUM,B_COL=0 @ RETURN ELSE B_COL=0
8800  IF Ninputs=0 THEN GOSUB COLUMN_NAMES  @ GOTO COLINPUT
8810  IF Ninputs<2 THEN GOSUB BAD_INPUT  @ GOTO COLINPUT  ELSE NCOLS=Ninputs
8820  FOR I=1 TO Ninputs
8830   IF O$(I)="A" THEN GOSUB BAD_INPUT  @ GOTO COLINPUT
8840   IF O$(I)="B" THEN B_COL=1
8850  NEXT I
8860  CNUM=2+2*(O$(2)#"")+(O$(5)#"") @ RETURN
8870 !
8880 CLEAR @ DISP "ZEROING ARRAY..." @ MAT S=ZER @ IF ALL THEN MAT VC=ZER
8890 CLEAR @ DISP "CONVERTING VISICALC FILE...." @ NUMVALS=0 @ DISP @ DISP
8900 GOSUB NAMEBLANK  @ ASSIGN# 1 TO  FILE$ @ C_ROW=CURSROW  @ OFF CURSOR @ GOTO 8980
```

```
8910  VC_STRINGS: READ# 1 ; VC$@ OFF ERROR @ ALPHA C_ROW,1 @ AWRIT VC$&"  "
8920    IF VC$[1,1]#")" THEN NEXT_I
8930    A=1+POS (VC$,":") !          String position where data starts
8940    E=NUM (VC$[A,A]) @ IF E=34 OR E=39 THEN A=A+1 ! 1st data-char.
8950    B=NUM (VC$[3,3]) @ D=NUM (VC$[2,2]) @ C=3+(B>64 AND B<91) ! C is 1st row-# character
8960    VCROW=VAL (VC$[C]) @ RETURN ! VISICALC row-number
8970 !
8980 FOR I=1 TO 3150 ! Read VISICALC strings
8990  ON ERROR GOTO FINISH
9000  GOSUB VC_STRINGS  @ IF VCROW<RMIN THEN FINISH
9010  IF VCROW>RMAX THEN GOTO NEXT_I  ELSE ROWN=VCROW-RMIN+1 ! Sample#
9020  IF C=4 AND B>86 THEN NEXT_I
9030  IF C=3 THEN COLN=D-66
9040  IF C=4 THEN COLN=24+26#(D-65)+B-64
9050  IF COLN=-1 THEN L=LEN (VC$) @ N$(ROWN)[1,9]=VC$[A,L-1] @ ALPHA C_ROW,L+4 @ AWRIT N$(ROWN)&"   " @ GOTO NEXT_I
9060  IF B_COL+COLN=0 THEN N$(ROWN)[10,18]=VC$[A,LEN (VC$)-1] @ GOTO NEXT_I
9070  IF ALL=0 THEN 9130
9080  ON ERROR GOTO 9120
9090    A=VAL (VC$[A]) @ OFF ERROR @ VC(VCROW,COLN)=A
9100    ALPHA C_ROW,38 @ OFF CURSOR
9110    DISP VCROW;COLN;A @ OFF CURSOR
9120  OFF ERROR @ GOTO NEXT_I
9130  OFF ERROR @ V$=VC$[2,C-1]
9140  FOR PARNUM=1 TO NCOLS
9150   R,F=0 @ GOSUB EXTRACT  @ IF F THEN 9170
9160   IF R THEN NEXT_I
9170  NEXT PARNUM
9180  GOTO NEXT_I
9190 !
9200  EXTRACT: ! Extract numeric values from VISICALC String
9210    IF V$#O$(PARNUM) THEN F=1 @ RETURN @ !    If not correct column
9220    IF PARNUM=2 AND NCOLS=2 THEN COLN=3 ELSE COLN=PARNUM
9230    ON ERROR GOTO 9270
9240    A=VAL (VC$[A]) @ OFF ERROR @ S(ROWN,COLN)=A @ R=1
9250    ALPHA C_ROW,38 @ OFF CURSOR
9260    DISP ROWN;PARAM$(COLN);" =";A
9270    OFF CURSOR @ OFF ERROR @ RETURN
9280 !
9290 NEXT_I: NEXT I
9300 !
9310 FINISH: OFF ERROR @ IF ALL THEN MAT S=ZER @ GOSUB COLSELECT
9320  CLEAR @ PRINT "SET#";TAB (31);"X";TAB (38);"%err";TAB (52);"Y";TAB (61);"%err";TAB (69);"RHO" @ PRINT
9330 FOR I=1 TO RMAX-RMIN+1
9340  PRINT I;TAB (5);N$(I);TAB (28);S(I,1);TAB (37);FNDr(S(I,2),3);TAB (49);S(I,3);TAB (59);FNDr(S(I,4),3);TAB (67);S(I,5)
9350 NEXT I
9360 GOTO DATAFILE_RETURN
9370 !
9380 NAMEBLANK: FOR I=1 TO MAXNUM @ N$(I)="" @ NEXT I @ RETURN
9390 !
9400 WRONGFILE: GOSUB CLUNK  @ DISP @ DISP "### FILE '";FILE$;"' NOT RECOGNIZED ON SPECIFIED DISC;   TRY AGAIN ###" @ DISP @ WAIT 10
00 @ GOTO DATAFILE
9410 !
9420  COLSELECT: ! Select columns from VISICALC array already in memory
9430  GOSUB COLINPUT  @ DISP @ DISP "TRANSFERRING ELEMENTS..."
9440  MAT COLN=ZER
9450  FOR I=1 TO NCOLS
9460   ON ERROR GOTO 9500
9470   IF LEN (O$(I))=1 THEN COLN(I)=NUM (O$(I)[1])-66
9480   IF LEN (O$(I))>1 THEN COLN(I)=26#(NUM (O$(I)[1,1])-64)+NUM (O$(I)[2,2])-66
9490   IF COLN(I)<1 OR COLN(I)>47 THEN GOSUB BAD_INPUT  @ GOTO COLSELECT
```

```
9500   OFF ERROR
9510  NEXT I
9520  FOR I=0 TO RMAX-RMIN
9530   FOR J=1 TO NCOLS
9540    IF NCOLS=2 AND J=2 THEN COLN=3 ELSE COLN=J
9550    S(I+1,COLN)=VC(RMIN+I,COLN(J))
9560   NEXT J
9570  NEXT I
9580 RETURN
9590 !
9600 PLOTBOX_SIZE: A,B=0 ! Calculate limits for plot defined by box-size
9610  RATIO_=(LIMX2-LIMX1)/(LIMY2-LIMY1)
9620  IF RATIO_>1 THEN Bgu=100 @ A=1 @ GOSUB AGU  @ GOSUB BGU
9630  IF RATIO_<= 1 THEN Agu=100 @ B=1 @ GOSUB BGU  @ GOSUB AGU
9640  ON ERROR GOTO 9690
9650  LIMIT A1,A2,B1,B2
9660  IF (C=0)*(RATIO >1 EXOR RATIO_>1) THEN C=1 @ GOTO PLOTBOX_SIZE
9670  OFF ERROR @ LIMX1=A1 @ LIMX2=A2 @ LIMY1=B1 @ LIMY2=B2 @ RETURN
9680 !
9690 OFF ERROR @ GOSUB CLUNK  @ DISP @ DISP "**** INSUFFICIENT ROOM FOR LABELS WITH SPECIFIED PLOTBOX-LIMITS ****" @ DISP @ DISP @ G
OTO 1520
9700 !
9710 AGU:
9720  MGy=(LIMY2-LIMY1)/(Bgu-YL-YEND) ! millimeters/GU in Y-direction
9730  YL_MM=YL*MGy @ YEND_MM=YEND*MGy
9740  B1=LIMY1-YL_MM @ B2=LIMY2+YEND_MM
9750  IF A THEN Agu=XL+XEND+100*(LIMX2-LIMX1)/(B2-B1)
9760 RETURN
9770 !
9780 BGU:
9790  MGx=(LIMX2-LIMX1)/(Agu-XL-XEND) ! millimeters/GU in X-direction
9800  XL_MM=XL*MGx @ XEND_MM=XEND*MGx
9810  A1=LIMX1-XL_MM @ A2=LIMX2+XEND_MM
9820  IF B THEN Bgu=YL+YEND+100*(LIMY2-LIMY1)/(A2-A1)
9830 RETURN
9840 !
9850 COLUMN_NAMES:
9860 ! Find and list column-names of VISICALC files.
9870 ! Assumes that column-naames are underline with repeating "=", so that
9880 ! the VC-string is "/-=", and that rows 3 and 4 contain the column-names.
9890 ! Also assumes a 9-character column-width for sample-names.
9900 DIM CN$(61,2)[9]
9910 !
9920 ALPHA 1 @ CLEAR @ C_ROW=CURSROW  @ B$="         "
9930 DISP "Locating column-name rows.  Please wait..."
9940 FOR I=1 TO 61 @ CN$(I,1),CN$(I,2)=B$ @ NEXT I
9950 ASSIGN# 1 TO  FILE$ @ ON ERROR GOTO 9990
9960 FOR I=1 TO 3150
9970  READ# 1 ; VC$@ IF POS (VC$,"/-=") THEN 10000
9980 NEXT I
9990 OFF ERROR @ DISP @ DISP HGL$ (" NO '/-=' VISICALC-CELLS FOUND - CAN'T FIND NAMES ") @ DISP @ DISP @ RETURN
10000 ALPHA C_ROW @ CLEAR
10010 FOR I=1 TO 3150
10020   ON ERROR GOTO 10110 @ GOSUB VC_STRINGS
10030   ON 1+(VCROW>4)+2*(VCROW<3) GOTO 10040,10090,10110
10040   IF C=3 THEN COLN=D-66
10050   IF C=4 THEN COLN=24+26*(D-65)+B-64
10060   IF COLN<1 THEN NEXT I
10070   IF VCROW=3 OR VCROW=4 THEN P=POS (VC$,CHR$ (13)) @ CN$(COLN,VCROW-2)=TRIM$ (VC$[A,P-1])
10080 !
```

```
10090 NEXT_I: NEXT I
10100 !
10110 CLEAR @ PRINT "COLUMN";TAB (10);"COLUMN-NAME" @ PRINT
10120 FOR I=1 TO 61
10130  IF I<25 THEN I$=CHR$ (I+66)
10140  IF I>24 AND I<51 THEN I$="A"&CHR$ (I+40)
10150  IF I>50 THEN I$="B"&CHR$ (I+14)
10160  IF LEN (CN$(I,1))=0 AND LEN (CN$(I,2))=0 THEN 10190
10170  IF CN$(I,1)=B$ AND CN$(I,2)=B$ THEN 10190
10180  PRINT "  ";I$;TAB (10);CN$(I,1);TAB (20);CN$(I,2)
10190 NEXT I
10200 PRINT @ PRINT
10210 RETURN
10220 !
10230 REGRESS: CLEAR @ OFF KYBD @ GOSUB POLYFIT  ! Do polynomial regression & plot curve
10240 SETUU @ PEN CPEN @ Y=0 @ X=XMIN%(XMIN$0)+Xspred/100%(XMIN=0)
10250 FOR I=X TO XMAX STEP (XMAX-XMIN)/100
10260  Y=0 @ IF I=0 THEN 10290
10270  FOR J=1 TO P+1 @ Y=Y+Coef(J)%I^(J-1) @ NEXT J
10280  IF I$X THEN DRAW I,Y ELSE MOVE I,Y
10290 NEXT I
10300 PEN 0 @ DISP "PRESS "&CNT$&" TO CONTINUE" @ PAUSE
10310 GOTO RE_DATA
10320 !
10330 POLYFIT: OFF KEY$
10340 ! Up to 5th order polynomial regression on up to 300 points.
10350 ! Unweighted for analytical errors.
10360 ! A-Array is the X Power-Matrix,
10370 ! Coef-Array Contains the P+1 Coefficients of the Pth-Order Polynomial.
10380 !
10390  DIM XA(6,300),XB(6,6),XD(6,300),P(300,6),Coef(6)
10400 DISP "WHAT ORDER POLYNOMIAL (1-5)";@ INPUT P
10410 IF P<1 OR P>5 OR NN<P+1 THEN GOSUB BAD_INPUT  @ GOTO 10400
10420 DISP @ DISP "Please Wait..."
10430  REDIM XA(P+1,NN),XD(P+1,NN),P(NN,P+1),Y(NN)
10440 !
10450 ! Create Power-matrix from X-values
10460  FOR I=1 TO NN
10470   FOR J=1 TO P+1
10480    P(I,J)=X(I)^(J-1)
10490   NEXT J
10500  NEXT I
10510 !
10520 ! Solve matrix equations for best-fit polynomial coefficients
10530   MAT XA=TRN (P)@ MAT XB=XA$P
10540   MAT XD=INV (XB)$XA
10550   MAT Coef=XD$Y
10560   CLEAR @ PRINT @ PRINT "ORDER";P;"POLYNOMIAL COEFFICIENTS:  " @ MAT PRINT Coef @ PRINT
10570 RETURN
10580 !
10590 DUMP_TO_PLOTTER: ! Transfer CRT-plot to HP-7470A plotter
10600   CPEN=1 @ SPEED$="VS" @ PLOTSIZE=8 @ PLOTTER_DUMP=1 @ HP=1 @ CRT=0 @ PLR=705 @ GOSUB 1560 @ CLEAR
10610   FOR JJ=1 TO C_CT
10620     K,N=1 @ X(K)=CACHE(JJ,1) @ Y(K)=CACHE(JJ,3) @ Ex(K)=CACHE(JJ,2) @ Ey(K)=CACHE(JJ,4) @ RO(K)=CACHE(JJ,5)
10630     SYMBOL=SYMBOL(JJ,1) @ GOSUB PLOTT
10640   NEXT JJ
10650   FOR I=1 TO NLINES
10660    Slope=LINE(I,1) @ Inter=LINE(I,2) @ GOSUB DRAWLINE
10670   NEXT I
10680 !
```

```
10690   FOR JJ=1 TO NPHRASES
10700    P$=LETTER$(JJ) @ X=LETTER(JJ,1) @ Y=LETTER(JJ,2) @ H=LETTER(JJ,3) @ SL=LETTER(JJ,4) @ ROT=LETTER(JJ,5) @ GOSUB LETTER
10710   NEXT JJ
10720 !
10730   PLOTTER_DUMP=0 @ GOTO RE_DATA
10740 !
10750 !
10760 HELP: OFF KYBD @ CLEAR ! Display instructions for special-function keys.
10770 DIM DF1$[2],DF2$[10],DF3$[60],DF4$[60],DF5$[60]
10780  PRINTER IS 1 @ PRINT TAB (16);HGL$ (" FUNCTIONS OF KEYS DEFINED DURING LAST DISPLAY: ") @ PRINT @ PRINT
10790  RETURN
10800 !
10810 HELP2: GOSUB HELP  @ RESTORE 10820 @ NUM_KEYS=17 @ GOSUB PRINT_HELP  @ GOTO RE_DATA
10820 DATA "1",OPEN Ebox,Choose an open (unfilled) 2-sigma error box as the,plotting-symbol.,"","8",SOLID Ebox
10830 DATA Choose a solid (filled-in) 2-sigma error box as the,plotting symbol.,"","2","OPEN Eell",Choose an open (unfilled) 2-sigma
 error-ellipse as the
10840 DATA plotting symbol.,"","9",SOLID Eell,Choose a solid (filled-in) 2-sigma error-ellipse as the,plotting-symbol.,""
10850 DATA "3",POLYGON,Choose an open or solid polygon (either regular or,star-shaped) of arbitrary size & orientation as the,plotti
ng-symbol.
10860 DATA "10",LETTER,Draft a phrase anywhere on the plot.,"",""
10870 DATA "","  +$0X ...",Use any keyboard-key character as the plotting-symbol.,"",""
10880 DATA "4",ERRCROSS,Choose a 2-sigma error-cross as the plotting-symbol.,"",""
10890 DATA "5",PRINT PLOT,Dump CRT-plot to printer (CRT-plot only).,"",""
10900 DATA "--",CTRL D,Dump CRT-plot to HP-7470A plotter.,"",""
10910 DATA "6",YORKFIT,Calculate a York-style best-fit line to the last-,"plotted set of data, starting with set# 1.  Also"
10920 DATA "calculates isochron age, if appropriate."
10930 DATA "7",OPTIONS,"Change errors to 1 or 2 sigma, absolute or percent;",Store CRT plot on disk; do polynomial regression to dat
a.,""
10940 DATA "11",PEN 1,Use pen# 1 (plotter) or solid-line (CRT).,"",""
10950 DATA "12",PEN 2,Use pen# 2 (plotter only).,"",""
10960 DATA "12",ERASE PEN,Use 'erasing pen' to un-draw symbols or letters,(CRT only).,""
10970 DATA "13",DATAFILE,Bring data from either a VISICALC or numeric,data-file into memory.,""
10980 DATA "14",REDRAW,Start a new plot.,"",""
10990 !
11000 PRINT_HELP: OFF KEY#
11010 FOR I=1 TO NUM_KEYS
11020  READ DF1$,DF2$,DF3$,DF4$,DF5$
11030  PRINT "KEY# "&DF1$;TAB (9);"- "&DF2$&":";TAB (23);DF3$
11040  IF DF4$#"" THEN PRINT TAB (23);DF4$
11050  IF DF5$#"" THEN PRINT TAB (23);DF5$
11060  PRINT
11070 NEXT I
11080 PRINT @ PRINT "Use "&HGL$ (" ROLL ")&" key to scroll display, press "&CNT$&" to return to previous display."
11090 PAUSE @ RETURN
11100 !
11110 HELP1: GOSUB HELP  @ RESTORE 11130 @ NUM_KEYS=13 @ GOSUB PRINT_HELP  @ GOTO BEGIN
11120 !
11130 DATA "1",PRINT-DAT,Use printer as the printout device.,"","","2",CRT-DAT,Use the CRT as the printout device.,"",""
11140 DATA "3",HARDPLOT,Use the plotter (HP-7470A) as the plotting device.,"","","4",CRT PLOT,Use the CRT as the plotting device.,""
,""
11150 DATA "5",PEN 1,Use pen# 1 (plotter only) as the plotting pen.,"","","6",PEN 2,Use pen# 2 (plotter only) as the plotting pen.,"
","",""
11160 DATA "8",PEN SPEED,Specify the speed of the plotter pen (default value is,the fastest speed)  (plotter only).,""
11170 DATA "12",$NORM$,"Use 'normal'-size ticks, labels,& symbols.","","",""
11180 DATA "13",SLIDE,"Use 'slide-size' ticks, labels, & symbols.  These",are somewhat larger relative to the plot-size than,$NORM$
size.
11190 DATA "14",DATAFILE,Bring data from either a VISICALC or numeric data-,file into memory.,""
11200 DATA "7",START,"Answer the resulting queries about axis-labels, intervals,",& plot-size.  The plot framework will then be draw
n.  You
11210 DATA don't need to press any other keys for just a CRT plot.
```

```
11220 DATA "10",STACK,Suppress X-axis label and X-axis tick-labels for more,convenient vertical stacking of plots sharing the same,X
-axis.
11230 DATA "9",LOG ZEROES,Include a leading zero in tick-labels for ticks with,absolute values less than zero.,""
11240 !
11250 HELP3: GOSUB HELP  @ RESTORE 11270 @ NUM_KEYS=8 @ GOSUB PRINT_HELP  @ GOTO OPTIONS
11260 !
11270 DATA "1",PRINT PLOT,Dump the CRT plot to an HP-82905B or 2673A printer.,"",""
11280 DATA "2",STORE PLOT,Store the plot as a graphics file on a disk (CRT,plots only).,""
11290 DATA "4","1-SIG ERRS",Input data-point errors at the 1-sigma level instead,of 2-sigma.,""
11300 DATA "5","2-SIG ERRS",Input data-point errors at the 2-sigma level (default),instead of 1-sigma.,""
11310 DATA "2",ABS ERRORS,Input data-point errors as absolute values rather,than as percent values.,""
11320 DATA "3",% ERRORS,Input data-point errors as percent values (default),rather than as absolute values.,""
11330 DATA "14",POLYFIT,Calculate a best-fit polynomial curve (up to 5th order),"to the last-input set of data, and plot the curve."
,""
11340 DATA "7",ESCAPE,Return to the previous key-function display.,"",""
11350 !
11360 HELP4: GOSUB HELP  @ RESTORE 11380 @ NUM_KEYS=3 @ GOSUB PRINT_HELP  @ GOTO ADD
11370 !
11380 DATA "5",YORKFIT,Calculate a York-type best-fit line to the last set of,"data, & calculate an isochron age if appropriate.","",

11390 DATA "6",RETAIN,"Add any subsequent-input points to the previous ones,",so that points with different plot-symbols can be
11400 DATA pooled for a later Yorkfit or Polyfit.
11410 DATA "7",CONTINUE,Return to the input-data part of the program.,"",""
11420 !
11430 HELP5: GOSUB HELP
11440 PRINT "MODEL 1:  Use the Model 1 solution for drawing the best-fit line and for any" @ PRINT TAB (12);"isochron calculation.
 Model 1 assumes that the";
11450 PRINT " only cause of scatter" @ PRINT TAB (12);"is from the assigned analytical errors.  The points are weighted";
11460 PRINT TAB (12);"according to those errors only.  If the probability of fit is less";TAB (12);"than 15%, the assigned errors ar
e expanded until they can";
11470 PRINT " account" @ PRINT TAB (12);"for the scatter, and a Student's-t multiplier of the errors is used." @ PRINT
11480 PRINT "MODEL 2:  Use the Model 2 solution for drawing the best-fit line and for any"
11490 PRINT TAB (12);"isochron calculation.  Model 2 assumes that nothing is known about"
11500 PRINT TAB (12);"the reason for the scatter of the points, and so weights all points"
11510 PRINT TAB (12);"equally, with zero error-correlations." @ PRINT
11520 PRINT "MODEL 3:  Use the Model 3 solution for drawing the best-fit line and for any"
11530 PRINT TAB (12);"isochron calculation.  Model 3 assumes that the cause of the scatter"
11540 PRINT TAB (12);"from the best-fit line is a combination of analytical error plus a"
11550 PRINT TAB (12);"normally-distributed variation in the initial ratios.  A Student's-t"
11560 PRINT TAB (12);"is applied to the errors." @ PRINT @ PRINT
11570 !
11580  PRINT "Press "&CNT$&" to return to previous display." @ PAUSE @ GOTO CHOOSE_MODEL
11590 !
11600 HELP6: GOSUB HELP  @ RESTORE 11620 @ NUM_KEYS=4 @ GOSUB PRINT_HELP  @ GOTO KEYS
11610 !
11620 DATA "1",RESTART,"Re-draw the plot, retaining any datafile data in memory.","",""
11630 DATA "2",ADD POINTS,Retain the just-Yorkfit points so that they can be added,to the next-input points for another Yorkfit.,""
11640 DATA "3",PRINT PLOT,Dump the CRT plot to an HP-82905B or HP-2673A printer.,"",""
11650 DATA "7",CONTINUE,Return to plot-symbol-select display.,"",""
```

APPENDIX 2: PROGRAM LISTING OF CPLOT

```
10 ! ############################## CPLOT ########################################
20 !  Program for Concordia plotting and Yorkfit of Pb/U isotope data.
30 !  Revision of NOV 01, 1983.  Requires Adv. Progr. ROM., about 100 kbytes.
40 !
50 !   Data may be entered from either of  (1) Keyboard, (2) Visicalc data-
60 !   file, or (3) nX45 Numeric data-file, and plotted either on CRT or
70 !   plotter.
80 !   Brackets indicate optional input.
90 !
100 ON ERROR GOTO 120
110 IF PLOT_STARTED THEN PLOTDATA ! SKIP PLOT SET-UP IF STARTING FROM "CONT 1"
120 OFF ERROR
130 !
140  OPTION BASE 1
150 DIM A(100),G(100),Q(100),R(100),Z(100),A$[72],X(100),Y(100),Ex(100),Ey(100),Name$[50],R0(100),Slope(4),Inter(4),Errint(4),Errsl(
4),Xbar(4),Ybar(4),CT$[1]
160 DIM R$[50],RR$[50],PLOTDEV(2),Pr(2),KEY$[1],N$(100)[18],XX(45),CRUD$[50]
170 DIM ALLKEY$[256],O(5),O$(5)[3],XERR(100),YERR(100),PHRASE$[80],S(100,5),Cint(8),E76(100),ZERO$[8],XTIK$(8)[12],YTIK$(8)[12],XTIK
(8),YTIK(8),Re$[50]
180 DIM CACHE(100,5),LETTER$(25)[80],LINE(12,2)
190 SHORT Ex0(100),Ey0(100),SYMBOL(100,5),LETTER(25,5)
200 INTEGER XDEC(8),YDEC(8)
210 !
220 ! Initialize constants
230 L8=.000155125 @ L5=.00098485 @ PS=.35 ! DECAY CONSTS., POLY-SIZE CONST.
240 CRUD$=HGL$ (" #### IMPROPER RESPONSE OR NUMBER OF VALUES #### ") @ SLD,HP=0
250 EL$=" "&HGL$ ("ENDLINE")&" " @ SPEED$="VS" @ FILE$="" @ ZERO$="00000000"
260 ALLKEY$=RPT$ (" ",256)
270 FOR I=1 TO 256
280  IF I<128 OR I>172 THEN ALLKEY$[I,I]=CHR$ (I)
290 NEXT I
300 PLOTDEV(1)=1 @ PLOTDEV(2)=705 @ CPEN=1 @ PEN 1 @ DFILE=0 @ ERRTYPE=-1 @ PLR,Pr(2),Pr(1),PR=1 @ PRINTER IS 1 @ CA=0 @ CT$="0" @ P
AGESIZE 24 @ DEG
310 REDRAW: PLOTTER_DUMP,C_CT,NPHRASES,LZERO,YCHAR,XCHAR,ODDSIZE,SIZETYPE,NCHORDS,SOLIDCHORD=0 @ GOTO BEGIN
320 !
321 CLEARKEY: ! Clear all special-function-key labels
322          FOR i=1 TO 14 @ ON KEY# i,"" GOSUB RETRN  @ NEXT i
323          RETURN
324 !
330 RETRIEVE_STR: ! Retrieve up to 5 substrings (separated by ',') from string
340      ON ERROR GOTO 390
350      LINPUT R$ @ O$(1),O$(2),O$(3),O$(4),O$(5)="" @ Ninputs=0 @ RR$=R$
360      Wh=POS (R$,",") @ SP=Wh*(Wh#0)+(LEN (R$)+1)*(Wh=0)
370      O$(1+Ninputs)=R$[1,SP-1] @ Ninputs=1+Ninputs
380      IF Wh THEN R$=R$[1+Wh] @ GOTO 360
390      OFF ERROR @ RETURN
400 !
410 RETRIEVE: ! Retrieve up to 5 numeric values separated by ',' from string
```

```
420     LINPUT R$
430 RTR: O(1),O(2),O(3),O(4),O(5),Ninputs=0 @ RR$=R$ ! Alternate entry line
440     ON ERROR GOTO 470
450     O(1+Ninputs)=VAL (R$) @ Ninputs=1+Ninputs
460     Wh=POS (R$,",") @ IF Wh THEN R$=R$[1+Wh] @ GOTO 450
470     OFF ERROR @ RETURN
480 !
490 BAD_INPUT: GOSUB CLUNK  @ DISP @ DISP CRUD$ @ DISP @ RETURN
500 !
510 DEF FNA(T) = EXP (L5*T)-1 !  207/235
520 DEF FNB(T) = EXP (L8*T)-1 !  206/238
530 DEF FNC(T) = L8*EXP ((L8-L5)*T)/L5 !  Slope of concordia curve for time T
540 DEF FND(X) = LOG (1+X)/L5 !           206/238 age
550 DEF FNE(Y) = LOG (1+Y)/L8 !           207/235 age
560 DEF FNF(X) = INT (LGT (ABS (X+(X=0))))
570 !
580 DEF FNTICK(X) !                       Returns tick-interval
590   A=10^FNF(X)/8
600   IF ABS (X/A)>12 THEN A=2*A @ GOTO 600
610   B=ABS (A)/10^FNF(A)
620   IF B*INT (B) THEN A=INT (B)*10^FNF(A)
630   FNTICK=A @ FN END
640 !
650 DEF FNXTICK(P) !                      Draw X-axis & ticks
660   NTIX=Xspred/XTIK @ ETIX=NTIX/2=INT (NTIX/2) ! # ticks even?
670   FOR X=1 TO NTIX
680     ETIK=X/2=INT (X/2) ! Even tick?
690     IF (P=1)*ETIK OR (P=-1)*((ETIK EXOR ETIX)=0) THEN M=XTICKSIZE ELSE M=XTICKSIZE/2
700     IDRAW P*XTIK,0 @ IDRAW 0,P*M @ IDRAW 0,-(P*M)
710   NEXT X
720   FNXTICK=0 @ FN END
730 !
740 DEF FNYTICK(P) !                      Draw Y-axis & ticks
750   NTIX=Yspred/YTIK @ ETIX=NTIX/2=INT (NTIX/2) ! # ticks even?
760   FOR Y=1 TO NTIX
770     ETIK=Y/2=INT (Y/2) ! Even tick?
780     IF (P=1)*ETIK OR (P=-1)*((ETIK EXOR ETIX)=0) THEN M=YTICKSIZE ELSE M=YTICKSIZE/2
790     IDRAW 0,P*YTIK @ IDRAW -(P*M),0 @ IDRAW P*M,0
800   NEXT Y
810   FNYTICK=0 @ FN END
820 !
830 DEF FNDr(N,D) !                       Round value N to D significant figs.
840     IF N THEN Mu=10^(D-INT (LGT (ABS (N)))-1) ELSE Mu=1
850     FNDr=IP (N*Mu)/Mu
860     FN END
870 !
880 DEF FNDECIMAL(X) !                    Returns #digits beyond decimal pt
890   A=0 @ N=X
900   N=10*(ABS (N)-INT (ABS (N)))
910   IF N THEN A=A+1 @ GOTO 900 ELSE FNDECIMAL=A
920   FN END
930 !
940 CLUNK: BEEP 115,40 @ BEEP 195,40 @ RETURN ! ERROR-SOUND
950 !
960 DEF FNCHAR(X) = 1+(ABS (X)>1)*(FNF(X)+(X*INT (X)))+(X<0)+FNDECIMAL(X)+(X<1)*LZERO ! #CHARACTERS REQUIRED FOR X.
970 !
980 CA=0 @ CT$="0" @ ALPHA 1 @ DISP HGL$ (" HORIZONTAL CONCORDIA TICK-LABELS ") @ RETURN
990 CA=1 @ CT$="-" @ ALPHA 1 @ DISP HGL$ (" OBLIQUE CONCORDIA TICK-LABELS ") @ RETURN
1000 !
1010 BEGIN: ALPHA 1 @ CLEAR @ ALPHA 21,27 @ AWRIT "(LOWER OPTIONS ARE DEFAULT)"
```

```
1020 ALPHA 19,19 @ AWRIT "Press "&HGL$ (" L ")&" to obtain leading zeroes for ticks" @ OFF CURSOR @ GOSUB CLEARKEY
1030              ON KEY# 1,"0  CTICKS" GOSUB 980
1040              ON KEY# 2,"CRT-PRINT" GOSUB DISPLAY
1050              ON KEY# 3," CRT PLOT" GOSUB CRTPLOT
1060 IF PLR#1 THEN ON KEY# 4,"  PEN 1" GOSUB PEN_1
1080              ON KEY# 6,"  $NORM$" GOSUB NORMSIZE
1090              ON KEY# 7,"  START" GOTO 60
1100              ON KEY# 8,"\  CTICKS" GOSUB 990
1110              ON KEY# 9," PRINTER" GOSUB HARDCOPY
1120              ON KEY# 10," HARDPLOT" GOSUB HARDPLOT
1130 IF PLR#1 THEN ON KEY# 11,"  PEN 2" GOSUB PEN_2
1150              ON KEY# 5," DATAFILE" GOSUB DATAFILE
1160              ON KEY# 13,"  SLIDE" GOSUB SLIDE
1170              ON KEY# 14,"  HELP" GOTO HELP1
1180 IF PLR=1 THEN ON KEY# 12,"" GOSUB RETRN
1190 IF PLR#1 THEN ON KEY# 12," PENSPEED" GOTO PEN_SPEED
1200              ON KYBD K,ALLKEY$ GOTO keys
1210      KEY LABEL
1220      GOTO 1220
1230 !
1240 keys: IF K=76 OR K=108 THEN OFF KYBD @ GOTO LZERO  ELSE GOSUB CLUNK  @ GOTO 1220
1250 !
1260 GO: CLEAR @ OFF KEY# !  Enter plot-limits & calculate graphics parameters
1270 OFF KYBD @ DISP @ DISP "[MIN. AGE,]  MAX. AGE (Ma)    (PRESS"&EL$&"TO DEFINE BY 6/8-7/5 LIMITS)";
1280  GOSUB RETRIEVE  @ IF O(2)<O(1) AND O(2) THEN GOSUB BAD_INPUT  @ GOTO 1270
1290  IF Ninputs=0 THEN T=0 @ GOTO 1320 ELSE AMN=O(1) @ AMX=O(2) @ T=1
1300  IF Ninputs=1 THEN T=1 @ AMN=0 @ AMX=O(1) @ GOTO 1470 ELSE T=1 @ GOTO 1470
1310 !
1320  DISP "207/235, 206/238 LIMITS";@ GOSUB RETRIEVE  @ IF Ninputs#2 AND Ninputs#4 THEN GOSUB BAD_INPUT  @ GOTO 1320
1330  IF Ninputs=2 THEN XMIN,YMIN=0 @ XMAX=O(1) @ YMAX=O(2) @ GOTO 1360
1340  XMIN=O(1) @ XMAX=O(2) @ YMIN=O(3) @ YMAX=O(4)
1350  IF XMIN>= XMAX OR YMIN>= YMAX THEN GOSUB BAD_INPUT  @ GOTO 1320
1360  XTIK=FNTICK(XMAX-XMIN) @ YTIK=FNTICK(YMAX-YMIN)
1370 !
1380 ! Force lower-bound X & Y values to have a minimum # of sig. figures.
1390 FOR I=1 TO 8
1400  X=FNDr(XMIN,I) @ IF XMIN-X<XTIK THEN XMIN=X @ GOTO 1420
1410 NEXT I
1420 FOR I=1 TO 8
1430  Y=FNDr(YMIN,I) @ IF YMIN-Y<YTIK THEN YMIN=Y @ GOTO 1450
1440 NEXT I
1450  AMX=FND(XMAX-(XMAX-XMIN)/100) @ IF FNB(AMX)>YMAX THEN AMX=FNE(YMAX)
1460  AMN=FND(XMIN)
1470 IF PLR=1 THEN 1600 ELSE CLEAR @ DISP "PLOT-SIZE (3-10)" @ DISP @ DISP "[press "&EL$&" for nonstandard dimensions/locations]";@
DISP @ GOSUB RETRIEVE
1480 IF Ninputs$((O(1)<3)+(O(1)>10)) THEN GOSUB BAD_INPUT  @ GOTO 1470
1490 IF Ninputs=0 THEN ODDSIZE=1 ELSE PLOTSIZE=O(1) @ GOTO 1600
1500 !
1510 CLEAR @ DISP @ DISP "PRESS "&HGL$ (" 1 ")&" TO DEFINE PLOT-LIMITS FOR THE ENTIRE FIGURE-AREA,"
1520 DISP @ DISP "      "&HGL$ (" 2 ")&" TO DEFINE FOR JUST THE PLOT-BOX." @ DISP
1530 ON KYBD KY,"0123456789" GOTO 1550
1540 GOTO 1540
1550 OFF KYBD @ IF KY#49 AND KY#50 THEN GOSUB CLUNK  @ GOTO 1530 ELSE SIZETYPE=KY-48
1560  DISP @ DISP "ENTER PLOT-LIMITS IN mm:  X-MIN, X-MAX, Y-MIN, Y-MAX" @ DISP "(max. limits are 0-257 [X] and 0-191 [Y])";@ DISP
1570  INPUT LIMX1,LIMX2,LIMY1,LIMY2@ IF LIMX1<0 OR LIMX2>257 OR LIMY1<0 OR LIMY2>191 OR LIMX2-LIMX1<30 OR LIMY2-LIMY1<30 THEN GOSUB
BAD_INPUT  @ GOTO 1560
1580  PLOTSIZE=10$((LIMX2-LIMX1)/257+(LIMY2-LIMY1)/191)/2
1590 !
1600  ALPHA 1,1 @ CLEAR @ OFF CURSOR @ ALPHA 12,27 @ AWRIT "A FEW SECONDS, PLEASE..."
1610 ON ERROR GOTO PLOTTER_NOT_ON  @ PLOTTER IS PLR @ OFF ERROR @ GOTO 1660
```

```
1620 !
1630 PLOTTER_NOT_ON: GOSUB CLUNK  @ CLEAR @ DISP TAB (24);HGL$ (" PLOTTER NOT READY! ") @ WAIT 3000 @ CLEAR @ GOTO 1610
1640 !
1650 IF PLOTTER_DUMP THEN 2040
1660 IF T=0 THEN 1770
1670 ! CALCULATE PLOT-BOUNDARIES -------------------------
1680 !
1690  XMAX=FNA(AMX) @ YMAX=FNB(AMX)
1700  XTIK=FNTICK(XMAX-FNA(AMN)) @ YTIK=FNTICK(YMAX-FNB(AMN))
1710  FOR XMIN=0 TO XMAX STEP XTIK !          CALC. MIN. X FOR AGE-DEFINED PLOT
1720   IF FND(XMIN+XTIK)>AMN THEN 1740
1730  NEXT XMIN
1740  FOR YMIN=0 TO YMAX STEP YTIK !          CALC. MIN. Y FOR AGE-DEFINED PLOT
1750   IF FNE(YMIN+2*YTIK)>AMN THEN 1780
1760  NEXT YMIN
1770  XTIK=FNTICK(XMAX-XMIN) @ YTIK=FNTICK(YMAX-YMIN)
1780  CTIK=FNTICK((AMX-AMN)/(1+(AMX-AMN>3000)))
1790 FOR X=XMIN TO XMAX STEP XTIK !           CALC. MAX. X  "   "    "      "
1800  IF XMAX-X<= XTIK THEN XMAX=X+XTIK @ GOTO 1820
1810 NEXT X
1820 FOR Y=YMIN TO YMAX STEP YTIK !           CALC. MAX. Y  "   "    "      "
1830  IF YMAX-Y<= YTIK THEN YMAX=Y+YTIK @ GOTO 1850
1840 NEXT Y
1850 XTIK=FNTICK(XMAX-XMIN) @ YTIK=FNTICK(YMAX-YMIN)
1860  Xspred=XMAX-XMIN @ Yspred=YMAX-YMIN @ XDEC_MAX,YDEC_MAX,XCHAR,YCHAR=0 @ HP=PLR#1
1870 !
1880 ! Find max. # characters in X & Y ticks & ticks w. max. # of decimal pl.
1890 FOR I=1 TO Xspred/XTIK/2+1
1900   XTIK(I)=XMIN+2*XTIK*(I-1)
1910   A=FNCHAR(XTIK(I)) @ IF A>XCHAR THEN XCHAR=A ! Max. # of characters
1920   XDEC(I)=FNDECIMAL(XTIK(I)) @ IF XDEC(I)>XDEC_MAX THEN XDEC_MAX=XDEC(I)
1930 NEXT I
1940 !
1950 FOR I=1 TO Yspred/YTIK/2+1
1960   YTIK(I)=YMIN+2*YTIK*(I-1)
1970   A=FNCHAR(YTIK(I)) @ IF A>YCHAR THEN YCHAR=A ! Max. # of characters
1980   YDEC(I)=FNDECIMAL(YTIK(I)) @ IF YDEC(I)>YDEC_MAX THEN YDEC_MAX=YDEC(I)
1990 NEXT I
2000 !
2010 HP=PLR#1 @ CRT=PLR=1 @ PP=1+CRT @ IF PLR=1 THEN GCLEAR
2020 IF PLR=1 THEN LIMX1,LIMY1=0 @ LIMX2=125 @ LIMY2=75 @ GOTO 2070
2030 IF ODDSIZE THEN 2060
2040  LIMX1=257*(1-.1*PLOTSIZE)/2 @ LIMX2=257-LIMX1 ! Centered plot
2050  LIMY1=191*(1-.1*PLOTSIZE)/2 @ LIMY2=191-LIMY1
2060  IF SIZETYPE<2 THEN LIMIT LIMX1,LIMX2,LIMY1,LIMY2
2070 AR=2/3 @ THT=5 @ LHT=6 !       ASPECT RATIO, TICK-LABEL CHAR. HT
2080 IF HP THEN M=.7*(1+2.7*(PLOTSIZE<6 OR SLD)/PLOTSIZE) @ THT=M*THT @ LHT=1.2*M*LHT ! For plotsize<6, character-height is somewhat
 buffered
2090 IF HP THEN PRINTER IS PLOTDEV(2) @ PRINT SPEED$ @ PRINTER IS Pr(2)
2100 !
2110 A=(XMAX-XMIN)/XTIK @ B=(YMAX-YMIN)/YTIK
2120 YL=1+THT*.75+LHT !                          GU's below X-axis
2130 XL=THT*AR*(1-HP/3)*YCHAR+LHT*1.2 !          GU's to left of Y-axis
2140 ! XL=1+THT*AR*(YCHAR-HP)*(1-HP/3)-.25+LHT*1.5 ! GU's to left of Y-axis
2150 YEND=THT/2*(B/2=INT (B/2)) !                GU's above top of Y-axis
2160 XEND=(1+XCHAR*AR*THT/2)*(A/2=INT (A/2)) !   GU's to right of X-axis
2170 !
2180 IF SIZETYPE=2 THEN C=0 @ GOSUB PLOTBOX_SIZE
2190 IF RATIO >1 THEN Xgu=100*RATIO  @ Ygu=100 ELSE Xgu=100 @ Ygu=100/RATIO
2200 ! Xgu, Ygu are GU's of total-plot area along X&Y directions
```

```
2210 !
2220 LOCATE XL,Xgu-XEND,YL,Ygu-YEND
2230 XLOC=(Xgu-XEND-XL)/Xspred !          GU's/UU in x-direction
2240 YLOC=(Ygu-YEND-YL)/Yspred !                *   *  *  * *    *
2250 Xmm=XLOC*(LIMX2-LIMX1)/Xgu !         Millimeters/UU in X-direction
2260 Ymm=YLOC*(LIMY2-LIMY1)/Ygu !         Millimeters/UU in Y-direction
2270 SCALE XMIN,XMAX,YMIN,YMAX
2280 CSIZE THT @ LDIR 0 @ PEN CPEN @ LW=AR*LHT*(1-HP/3)
2290 IF PLOTTER_DUMP THEN 2490
2300 !
2310 FOR I=1 TO Xspred/XTIK/2+1 !                  Add decimal pt & zero to
2320  XTIK$(I)=VAL$ (XTIK(I)) !                   match format of X-tick with
2330  IF XDEC_MAX=0 OR XDEC_MAX=XDEC(I) THEN 2360 ! most significant figures.
2340  IF XDEC(I)=0 THEN XTIK$(I)=XTIK$(I)&"."
2350  XTIK$(I)=XTIK$(I)&ZERO$[1,XDEC_MAX-XDEC(I)]
2360  IF LZERO AND XTIK(I)>0 AND XTIK(I)<1 THEN XTIK$(I)="0"&XTIK$(I) ! Add leading zero
2370  IF XTIK(I)=0 THEN XTIK$(I)="0"
2380 NEXT I
2390 !
2400 FOR I=1 TO Yspred/YTIK/2+1 !                  Add decimal pt & zero to
2410  YTIK$(I)=VAL$ (YTIK(I)) !                   match format of Y-tick with
2420  IF YDEC_MAX=0 OR YDEC_MAX=YDEC(I) THEN 2450 ! most significant figures.
2430  IF YDEC(I)=0 THEN YTIK$(I)=YTIK$(I)&"."
2440  YTIK$(I)=YTIK$(I)&ZERO$[1,YDEC_MAX-YDEC(I)]
2450  IF LZERO AND YTIK(I)>0 AND YTIK(I)<1 THEN YTIK$(I)="0"&YTIK$(I) ! Add leading zero
2460  IF YTIK(I)=0 THEN YTIK$(I)="0"
2470 NEXT I
2480 !
2490 XTICKSIZE=2.5/YLOC @ YTICKSIZE=XTICKSIZE*Ymm/Xmm
2500 PEN CPEN @ SETUU @ MOVE XMIN,YMIN
2510 A=FNXTICK(1)+FNYTICK(1)+FNXTICK(-1)+FNYTICK(-1) @ FRAME ! Draw axes$ticks
2520 LORG 6 @ A=THT/YLOC/3
2530 FOR I=1 TO Xspred/XTIK/2+1 ! Label X-axis ticks
2540   MOVE XTIK(I),YMIN-A @ LABEL XTIK$(I)
2550 NEXT I
2560 !
2570 LORG 8 @ B=THT*AR/2/XLOC
2580 FOR I=1 TO Yspred/YTIK/2+1 ! Label Y-axis ticks
2590   MOVE XMIN-B,YTIK(I)+(I=1)*THT/8/YLOC @ LABEL YTIK$(I)
2600 ! lift 1st Y-tick to avoid interference with 1st X-tick
2610 NEXT I
2620 !
2630 SETGU @ CSIZE LHT
2640 LORG 1 @ MOVE (Xgu+XL-XEND)/2-LW*4.5,LHT/4 @ LABEL "207"
2650 IMOVE 3*LW,LHT*3/4 @ LABEL "Pb/" @ IMOVE 3*LW,LHT*5/4 @ LABEL "235"
2660 IMOVE 3*LW,LHT*3/4 @ LABEL "U"
2670 LDIR 90 @ LORG 6 @ MOVE 0,(Ygu+YL)/2-LW*4.5 @ LABEL "206"
2680 IMOVE -(LHT*3/4),3*LW @ LABEL "Pb/" @ IMOVE -(LHT*5/4),3*LW @ LABEL "238"
2690 IMOVE -(LHT*3/4),2*LW @ LABEL "U" @ SETUU @ LDIR 0
2700 !
2710 PLOT_CONCORDIA: K=8*CTIK @ C=0 @ CHT=THT @ CTHT=THT-2*(CA=0) @ D=0
2720      CLIP XMIN,XMAX,YMIN,YMAX
2730      FOR AMN=C TO AMX STEP K
2740       IF FNA(AMN)>= XMIN AND FNB(AMN)>= YMIN THEN 2760
2750      NEXT AMN
2760      IF K*CTIK THEN C=AMN-K @ K=CTIK @ GOTO 2730
2770      FOR T=AMN-CTIK TO AMX STEP CTIK/8
2780       X=FNA(T) @ Y=FNB(T) @ IF (X>XMIN)*(Y>YMIN) THEN DRAW X,Y ELSE MOVE X,Y
2790        IF T/CTIK*INT (T/CTIK) THEN 3000
2800        IF T<AMN THEN 2990
```

```
2810 !
2820 ! ------ Concordia ticks & tick-labels -----
2830 !
2840       IF T=0 THEN 2990
2850       IF CA=0 THEN 2880
2860       RISE=FNC(T)*YLOC @ RUN_=-XLOC ! OF TICK-ANGLE
2870       SETGU @ LDIR RISE,RUN_ ! RT-ANGLE TO CONCORDIA CURVE
2880       SETUU @ LORG 5 @ CSIZE CTHT,1,0 @ LABEL CT$ @ IF D=0 THEN 2980
2890 ! Check if tick-labels lie within plot-box
2900        L=CHT*(1.5+INT (LGT (T))) @ IF CA THEN 2930
2910        Xt=XL+XLOC*(X-XMIN)-L*AR-2 ! location of label-start in X-GU's
2920        Yt=YL+YLOC*(Y-YMIN)+CHT+2 @ GOTO 2960 ! loc. of label-top, Y-GU's
2930        Tangle=90+ATN (RISE/RUN_) ! tick-angle, real degrees
2940        Xt=XL+XLOC*(X-XMIN)-COS (Tangle)*L*AR-1
2950        Yt=YL+YLOC*(Y-YMIN)+SIN (Tangle)*L+1
2960        IF Xt<XL OR Yt>Ygu-YEND THEN 2980
2970       MOVE X,Y @ LORG 7+CA @ CSIZE CHT,AR,0 @ LABEL VAL$ (T)&" "
2980       MOVE X,Y
2990       D=D=0
3000      NEXT T
3010 IF FILE$="" THEN ERRTYPE=0
3020 !
3030 PLOT_STARTED=1 @ GOTO STARTPLOT
3040 RETRN: RETURN !    Dummy return for key-labels
3050 !
3060 HARDCOPY: ALPHA 1 @ DISP HGL$ (" PRINTER PRINTOUT ") @ Pr(2)=701 @ PR=2 @ PRINTER IS Pr(2) @ RETURN
3070 NORMSIZE: ALPHA 1 @ DISP HGL$ (" NORMAL-SIZE PLOT-SYMBOLS & LINES ") @ SLD=0 @ RETURN
3080 LZERO: BEEP @ ALPHA 1 @ DISP HGL$ (" ADD LEADING ZEROES TO TICK-LABELS ") @ OFF CURSOR @ LZERO=1 @ GOTO 1200 ! Include ldg zero
 before <1 tick dec pt
3090 PEN_SPEED: ALPHA 1 @ CLEAR @ AWRIT HGL$ (" INPUT PEN-SPEED (1-10) ") @ ALPHA 3 @ INPUT PENSPEED
3100      IF PENSPEED<1 OR PENSPEED>10 THEN GOSUB CLUNK  @ GOTO PEN_SPEED
3110      IF PENSPEED=10 THEN SPEED$="VS"
3120      IF PENSPEED<10 THEN SPEED$="VS"&VAL$ (INT (38.1*PENSPEED/10))
3130      GOTO BEGIN
3140 !
3150 SLIDE: ALPHA 1 @ DISP HGL$ (" SLIDE-SIZE SYMBOLS & LINES ") @ SLD=1 @ RETURN
3160 DISPLAY: ALPHA 1 @ DISP HGL$ (" CRT PRINTOUT ONLY ") @ Pr(2)=1 @ PR=1 @ PRINTER IS 1 @ RETURN
3170 !
3180 HARDPLOT: ALPHA 1 @ DISP HGL$ (" PLOTTER GRAPHICS ") @ PLR=PLOTDEV(2) @ WAIT 500 @ GOTO BEGIN
3190 CRTPLOT: ALPHA 1 @ DISP HGL$ (" CRT GRAPHICS ") @ PLR=1 @ WAIT 500 @ GOTO BEGIN
3200 !
3210 STARTPLOT: NN=0 @ LDIR 0 @ PDIR 0 @ PRINTER IS Pr(2)
3220 PLOTDATA: OFF KEY# @ OFF ERROR @ N,Iso,Pbpb=0 @ II,K=1
3230 RE_DATA: OFF KYBD @ OFF KEY# @ PEN 0 @ SETUU @ MOVE XMIN,YMAX @ Kb=0 @ IF PLOTTER_DUMP THEN RETURN
3240         ALPHA 1 @ CLEAR @ ALPHA 21,23 @ AWRIT "(PRESS ANY KEY FOR KEYBOARD-SYMBOL)" @ OFF CURSOR
3250 !
3260             ON KEY# 13," ?ERRORS?" GOTO 3490
3270             ON KEY# 2,"OPEN Ell" GOTO OEll
3280             ON KEY# 4," POLYGON" GOTO Poly
3290             ON KEY# 8,"CONT DIFF" GOTO CONT_DIF
3300  IF CRT THEN ON KEY# 11,"PRINT PLOT" GOTO PRINT_PLOT
3310   IF HP THEN ON KEY# 11,"" GOSUB RETRN
3320 IF NN>1 THEN ON KEY# 7,"  YORKFIT" GOTO Yorkfit
3330 IF NN<2 THEN ON KEY# 7,"" GOSUB RETRN
3340  IF CRT THEN ON KEY# 12,"ERASE PEN" GOSUB ERASE_PEN
3350   IF HP THEN ON KEY# 12,"" GOSUB RETRN
3360             ON KEY# 5,"  HELP" GOTO HELP2
3370             ON KEY# 9,"SOLIDEll" GOTO SEll
3380             ON KEY# 1," LETTER" GOTO LETTER
3390             ON KEY# 10,"  PEN 1" GOSUB PEN_1
```

```
3400   IF HP THEN ON KEY# 3,"  PEN 2" GOSUB PEN_2
3410  IF CRT THEN ON KEY# 3,"HPLOT DUMP" GOTO DUMP_TO_PLOTTER
3420              ON KEY# 14," DATAFILE" GOSUB DATAFILE
3430              ON KEY# 6,"  REDRAW" GOTO REDRAW
3440              ON KYBD KY,ALLKEY$ GOTO KYBRD
3450 !
3460 PLABEL: KEY LABEL
3470         GOTO 3470
3480 !
3490 GOSUB ERRORS  @ GOTO RE_DATA
3500 ERRORS: ALPHA 1 @ CLEAR @ DISP @ DISP "PRESS "&HGL$ (" 1 ")&" FOR STANDARD ERROR-FORMAT (207/235 & 206/238 errors and err.-corr
el.)," @ DISP
3510  DISP "PRESS "&HGL$ (" 2 ")&" FOR OPTIONAL FORMAT (206/238 and 207/206 errors)." @ DISP
3520  ON KYBD KY,"0123456789" GOTO 3540
3530  GOTO 3530
3540  IF KY<49 OR KY>50 THEN GOSUB CLUNK  @ GOTO 3520 ELSE ERRTYPE=KY-49 @ BEEP 200,20 @ RETURN
3550 PEN_1: CPEN=1 @ ALPHA 1 @ DISP HGL$ (" PEN 1 ") @ RETURN
3560 PEN_2: CPEN=2 @ ALPHA 1 @ DISP HGL$ (" PEN 2 ") @ RETURN
3570 ERASE_PEN: CPEN=-1 @ ALPHA 1 @ DISP HGL$ (" ERASE-PEN ") @ RETURN
3580 !
3590 OEell: SYMBOL=2 @ ALPHA 1 @ DISP HGL$ (" OPEN ERROR-ELLIPSE SYMBOL ") @ GOTO INPUT_DATA
3600 SEell: SYMBOL=-2 @ ALPHA 1 @ DISP HGL$ (" SOLID ERROR-ELLIPSE SYMBOL ") @ GOTO INPUT_DATA
3610 !
3620 Poly: SYMBOL=3 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" POLYGON SYMBOL ")
3630       DISP @ DISP "# SIDES (- for star) [,OPEN/SOLID (0/1) [,SIZE (1-5) [,ROTATION (0-360)]]]";@ GOSUB RETRIEVE
3640   IF Ninputs<1 THEN GOSUB BAD_INPUT  @ GOTO 3630
3650   POLYSIDES=O(1)
3660   IF Ninputs>1 AND O(2)=1 THEN SOLID=1 ELSE SOLID=0
3670   IF Ninputs>2 THEN POLYSIZE=O(3) ELSE POLYSIZE=2.5
3680   IF Ninputs>3 THEN POLYROT=O(4) ELSE POLYROT=90+(POLYSIDES/2=INT (POLYSIDES/2))*(180/POLYSIDES-90)
3690   GOTO INPUT_DATA
3700 !
3710 KYBRD: IF KY=19 THEN SOLIDCHORD=1 @ ALPHA 1 @ DISP HGL$ (" USE SOLID LINE FOR CHORDS ") @ BEEP @ GOTO PLABEL
3720      CLEAR @ SYMBOL=5 @ KEY$=UPC$ (CHR$ (KY)) @ OFF KYBD @ ALPHA 1 @ DISP HGL$ (" PLOTTING-SYMBOL IS ")&HGL$ (KEY$&" ") @ GOTO
INPUT_DATA
3730 !
3740  INPUT_DATA: CLEAR @ OFF KEY# @ IF DFILE THEN DISP HGL$ (" INPUT DATAFILE SET #s AS 1st SET, LAST SET-ASTERISK ");" (e.g. 2,12*
)" @ DISP
3750  GOSUB INPUT_POINTS
3760  IF II>100 THEN Edit ELSE GOTO 3750
3770 !
3780  INPUT_POINTS: OFF KYBD @ X(II),Y(II),Ex(II),Ey(II),R(II)=0
3790  IF ERRTYPE=0 THEN DISP @ DISP "#"&VAL$ (II)&":  ";@ DISP "207/235, [%err,]  206/238, [%err] [,err-corr.]";
3800  IF ERRTYPE THEN DISP @ DISP "#"&VAL$ (II)&":  ";@ DISP "207/235, 206/238, [%6/8-err] [,%7/6-err]";
3810  DISP " (*&EL$&"WHEN DONE)";@ LINPUT Re$
3820  Sc=POS (Re$,";") @ IF Sc THEN R$=Re$[1,Sc-1] ELSE R$=Re$ ! Semicolon?
3830  GOSUB RTR  @ IF Ninputs=0 THEN Edit
3840  IF NOT POS (Re$,"*") THEN DF=0 @ GOSUB KEYBOARD_INPUT  @ RETURN
3850 IF DFILE THEN DF=1 ELSE GOSUB CLUNK  @ DISP @ DISP "*** NO DATAFILE IN MEMORY ***" @ GOTO 3750
3860 IF Ninputs=1 THEN S1,S2=O(1) ELSE S1=O(1) @ S2=O(2) ! 1st,last sets in series
3870 DISP "SET#";TAB (29);"207/235";TAB (38);"%Err";TAB (50);"206/238";TAB (60);"%Err";TAB (69);"RHO" @ DISP
3880 FOR ID=S1 TO S2
3890  IF ERRTYPE=0 THEN X(II)=S(ID,1) @ Ex(II)=S(ID,2) @ Y(II)=S(ID,3) @ Ey(II)=S(ID,4) @ R(II)=S(ID,5) @ GOTO 3930
3900  X(II)=S(ID,1) @ Y(II)=S(ID,2)
3910  IF S(ID,3)=0 OR S(ID,4)=0 THEN Ex(II),Ey(II),R(II)=0 @ GOTO 3930
3920  Ey(II)=S(ID,3) @ Ex(II)=FNDr(SQR (Ey(II)^2+S(ID,4)^2),3) @ R(II)=FNDr((Ex(II)^2+Ey(II)^2-S(ID,4)^2)/(2*Ex(II)*Ey(II)),4)
3930  IF X(II)*Y(II)=0 THEN 3960
3940  DISP II;TAB (5);N$(ID);TAB (28);X(II);TAB (37);Ex(II);TAB (49);Y(II);TAB (59);Ey(II);TAB (67);R(II) @ GOSUB 4080
3950  IF II=100 THEN Edit
3960 NEXT ID
```

```
3970 DISP @ IF Sc THEN Re$=Re$[1,Sc] @ GOTO 3820 ELSE RETURN
3980 !
3990 KEYBOARD_INPUT: Kb=1 @ IF ERRTYPE=0 THEN 4040
4000    IF Ninputs#2 AND Ninputs#4 THEN GOSUB BAD_INPUT  @ GOTO 3790
4010    X(II)=O(1) @ Y(II)=O(2) @ IF Ninputs=2 THEN 4080
4020    Ey(II)=O(3) @ Ex(II)=FNDr(SQR (O(3)^2+O(4)^2),3)
4030    R(II)=FNDr((Ex(II)^2+Ey(II)^2-O(4)^2)/(2*Ex(II)*Ey(II)),4) @ GOTO 4080
4040    IF Ninputs<2 OR Ninputs=3 OR Ninputs>5 THEN GOSUB BAD_INPUT  @ GOTO 3790
4050    IF Ninputs=2 THEN X(II)=O(1) @ Y(II)=O(2) @ GOTO 4080
4060    IF Ninputs>3 THEN X(II)=O(1) @ Ex(II)=O(2) @ Y(II)=O(3) @ Ey(II)=O(4)
4070    IF Ninputs=5 THEN R(II)=O(5)
4080  IF X(II)*Y(II)=0 THEN GOSUB CLUNK  @ DISP @ DISP HGL$ (" **** SET# "&VAL$ (II)&"  -NONZERO INPUT REQUIRED **** ") @ DISP @ DIS
P @ RETURN
4090  IF ERRTYPE OR (Ex(II)^2+Ey(II)^2)= 2*R(II)*Ex(II)*Ey(II))*(R(II)#1) THEN 4120
4100  GOSUB CLUNK  @ DISP @ IF R(II)#1 THEN DISP HGL$ ("**** ERR.-CORR. OF "&VAL$ (R(II))&" IS TOO HIGH **** ") @ RETURN
4110  DISP HGL$ (" AN ERROR-CORRELATION OF 1 IMPLIES A 207/206 ERROR OF ZERO- NOT POSSIBLE. ") @ RETURN
4120  II=II+1 @ N=N+1 @ RETURN
4130 !
4140 Edit: IF Kb=0 THEN 4160 ELSE CLEAR @ DISP "SET#";TAB (8);"207/235";TAB (20);"%Err";TAB (40);"206/238";TAB (52);"%Err";TAB (62);
"Rho" @ DISP
4150       FOR I=K TO N @ DISP I;TAB (8);X(I);TAB (20);Ex(I);TAB (40);Y(I);TAB (52);Ey(I);TAB (60);R(I) @ NEXT I @ DISP
4160       IF Kb=0 AND DF AND Sc THEN PLOTT
4170       IF ABS (SYMBOL)=3 OR SYMBOL=5 THEN 4280 ELSE Ninputs,A=0
4180       FOR I=K TO N @ A=A+(Ex(I)*Ey(I)=0) @ NEXT I
4190       IF A=0 THEN 4280
4200       IF ERRTYPE THEN DISP "%Errors:  206/238, 207/206";
4210       IF ERRTYPE=0 THEN DISP "%207/235-Err, %206/238-Err  [, Err.-Corr.]";
4220       DISP " FOR SETS WITH ZERO ERRORS";@ GOSUB RETRIEVE  @ IF Ninputs<2 THEN GOSUB BAD_INPUT  @ GOTO 4200
4230       FOR I=K TO N @ IF Ex(I) AND Ey(I) THEN 4260
4240        IF ERRTYPE THEN Ey(I)=O(1) @ Ex(I)=FNDr(SQR (O(1)^2+O(2)^2),3) @ R(I)=FNDr((Ex(I)^2+Ey(I)^2-O(2)^2)/(2*Ex(I)*Ey(I)),4) @
 GOTO 4260
4250        Ex(I)=O(1) @ Ey(I)=O(2) @ IF Ninputs=3 THEN R(I)=O(3)
4260       NEXT I @ GOTO Edit
4270       IF Kb=0 THEN PLOTT
4280       DISP @ DISP "SET TO BE CORRECTED?  ("&HGL$ ("ENDLINE")&" IF OK) ";
4290  GOSUB RETRIEVE  @ I=O(1) @ IF I=0 THEN PLOTT ELSE II=I @ GOSUB 3780 @ N=N-1 @ GOTO Edit
4300 !
4310 ERROR_ELLIPSE: !                        Draw error-ellipse about data-point
4320   IF XERR(I)*YERR(I) THEN XE=XERR(I) ELSE XE=XERR(I)*1.0001
4330   ANGLE=.5*ATN (2*R(I)*XE*YERR(I)/(XE^2-YERR(I)^2))
4340 ! ANGLE is the angle that an axis of the ellipse makes with the X-axis.
4350   C1=2*(1-R(I)^2)*1.224^2 @ C2=1/COS (2*ANGLE)
4360   Vx=XE^2 @ Vy=YERR(I)^2
4370   A=SQR (C1/((1+C2)/Vx+(1-C2)/Vy)) ! Length of major axis of ellipse
4380   B=SQR (C1/((1-C2)/Vx+(1+C2)/Vy)) ! Length of minor axis of ellipse
4390   St=A/INT (A/(Xspred/800)+1) !      Step-size, in x-direction
4400   St=A/INT (A/St+.0001)
4410   SIN_ANGLE=SIN (ANGLE) @ COS_ANGLE=COS (ANGLE)
4420    FOR K=1 TO -1 STEP -2
4430     FOR X=K*A TO -(K*(A+St)) STEP -(K*St)
4440     IF (K<0)*(X<-(A*1.00001)) OR (K>0)*(X>A*1.00001) THEN 4500
4450      Z=1-(X/A)^2 @ IF Z>= 0 THEN Y=K*B*SQR (Z) ELSE 4480
4460      RPLOT X*COS_ANGLE-Y*SIN_ANGLE,X*SIN_ANGLE+Y*COS_ANGLE
4470      IF SYMBOL<0 THEN K=-K @ RPLOT X*COS_ANGLE+Y*SIN_ANGLE,X*SIN_ANGLE-Y*COS_ANGLE
4480     NEXT X
4490     IF SYMBOL<0 THEN 4510
4500    NEXT K
4510    RETURN
4520 !
4530 POLYGON: SETGU ! Draw open or filled polygon or star-symbol
```

```
4540          IF PLOTTER_DUMP THEN POLYSIDES=SYMBOL(JJ,2) @ SOLID=SYMBOL(JJ,3) @ POLYSIZE=SYMBOL(JJ,4) @ POLYROT=SYMBOL(JJ,5)
4550          FOR A=0 TO (HP AND SLD=0 AND POLYSIDES>0 AND POLYSIZE>3.5)+50*SOLID
4560           B=PS*(POLYSIZE-A/2)*PP @ S=360/ABS (POLYSIDES) @ E=.5-(ABS (POLYSIDES)=3)/5
4570           IF B<0 THEN SETUU @ RETURN
4580           FOR J=POLYROT TO POLYROT+360 STEP S
4590            RPLOT B*COS (J),B*SIN (J)
4600            IF POLYSIDES<0 THEN C=J+S/2 @ RPLOT E*B*COS (C),E*B*SIN (C)
4610           NEXT J
4620          NEXT A
4630          SETUU @ RETURN
4640 !
4650 KEYBOARD_SYMBOL: IF PLOTTER_DUMP THEN KEY$=CHR$ (SYMBOL(JJ,2))
4660                  IF KEY$="+" THEN B=1.5 @ C=AR*CRT+.5*HP ELSE B=0 @ C=1
4670                  IF KEY$="*" THEN C=CRT+.6*HP
4680            LORG 5 @ CSIZE 3.5-HP+1.5*SLD+B,C,0 @ LABEL KEY$ @ RETURN
4690 !
4700 PLOTT: PEN CPEN !                                  Plot data-point symbols
4710        FOR I=K TO N
4720         IF HP THEN ALPHA C_ROW+1 @ DISP I;TAB (10);X(I);TAB (20);Y(I) @ OFF CURSOR @ GOTO 4760 ELSE C_CT=1+C_CT
4730         CACHE(C_CT,1)=X(I) @ CACHE(C_CT,3)=Y(I) @ CACHE(C_CT,2)=Ex(I) @ CACHE(C_CT,4)=Ey(I) @ CACHE(C_CT,5)=R(I) @ SYMBOL(C_CT,
1)=SYMBOL
4740         IF SYMBOL=3 THEN SYMBOL(C_CT,2)=POLYSIDES @ SYMBOL(C_CT,3)=SOLID @ SYMBOL(C_CT,4)=POLYSIZE @ SYMBOL(C_CT,5)=POLYROT
4750         IF SYMBOL=5 THEN SYMBOL(C_CT,2)=KY
4760         XERR(I)=X(I)*Ex(I)/100 @ YERR(I)=Y(I)*Ey(I)/100
4770         IF X(I)<XMIN OR Y(I)<YMIN OR X(I)>XMAX OR Y(I)>YMAX THEN 4800
4780         MOVE X(I),Y(I)
4790         ON ABS (SYMBOL) GOSUB RETRN ,ERROR_ELLIPSE ,POLYGON ,RETRN ,KEYBOARD_SYMBOL
4800        NEXT I
4810        IF PLOTTER_DUMP THEN RETURN ELSE MOVE XMIN,YMIN @ PEN 0
4820        IF CRT THEN ALPHA @ CLEAR @ DISP "PRESS "&HGL$ (" A/G ")&" TO VIEW PLOT." @ DISP
4830 !
4840 ADD: NN=N @ DISP @ DISP "PRESS "&HGL$ (" RETAIN ")&" TO INCLUDE THESE POINTS WITH OTHERS FOR A YORKFIT." @ DISP @ GOSUB CLEARKE
Y
4850      DISP "PRESS "&HGL$ (" CONTINUE")&" TO CONTINUE." @ DISP @ DISP
4890              ON KEY# 4,"  HELP" GOTO HELP3
4900 IF NN>1 THEN ON KEY# 5,"  YORKFIT" GOTO Yorkfit
4920              ON KEY# 6,"  RETAIN" GOTO RETAIN
4930              ON KEY# 7," CONTINUE" GOTO PLOTDATA
5010              ON KYBD KY,ALLKEY$ GOSUB RETRN
5020       OFF CURSOR @ KEY LABEL
5030       GOTO 5030
5040 RETAIN: OFF KYBD @ OFF KEY# @ II,K=N+1 @ GOTO RE_DATA  ! Add next points to exisitng sequence
5050 !
5060 !
5070 ! ##########################################################################
5080 Yorkfit: ! LINE-FITTING ALGORITHM MODIFIED FROM D. YORK,  EPSL V. 5, P. 320
5090 ! ##########################################################################
5100 !
5110  OFF KYBD @ OFF KEY# @ CLEAR @ DISP "YORKFIT:" @ DISP
5120  XERR,YERR,RHO,NO_ERRS,NO_RHOS=0 ! Check for zero errs & err-correls
5130  FOR I=1 TO NN @ NO_ERRS=NO_ERRS+(Ex(I)*Ey(I)=0) @ NO_RHOS=NO_RHOS+(R(I)=0) @ NEXT I
5140  IF ERRTYPE=0 THEN 5180
5150  IF NO_ERRS=0 THEN 5390
5160  CLEAR @ DISP "% Errors: 206/238,  207/206  FOR SAMPLES WITH NO ASSIGNED ERRORS";@ INPUT YERR,Z
5170  XERR=FNDr(SQR (YERR^2+Z^2),3) @ RHO=FNDr((XERR^2+YERR^2-Z^2)/(2*XERR*YERR),4) @ GOTO 5330
5180   IF NO_ERRS AND NO_RHOS THEN 5230
5190   IF NO_ERRS=0 AND NO_RHOS THEN 5280
5200   IF NO_ERRS AND NO_RHOS=0 THEN 5210 ELSE 5390
5210   CLEAR @ DISP "%Errors: 207/235, 206/238  FOR SAMPLES WITH NO ASSIGNED ERRORS";@ GOSUB RETRIEVE  @ IF Ninputs=0 THEN GOSUB BAD
_INPUT  @ GOTO 5210
```

```
5220   XERR=D(1) @ YERR=D(2) @ GOTO 5330
5230   CLEAR @ DISP "%207/235 Errs, %206/238 Errs, Err-Corrs FOR SAMPLES WITH NO ASSIGNED VALUES"
5240   GOSUB RETRIEVE  @ IF Ninputs=0 THEN GOSUB BAD_INPUT  @ GOTO 5230
5250   XERR=D(1) @ YERR=D(2) @ RHO=D(3)
5260   IF RHO=0 THEN GOSUB 5310 @ GOTO 5230
5270   IF ABS (RHO)>1 THEN GOSUB 5320 @ GOTO 5230 ELSE 5330
5280   DISP HGL$ (" ERR-CORRS FOR SAMPLES WITH ZERO ASSIGNED VALUES ");@ GOSUB RETRIEVE  @ RHO=D(1)
5290   IF RHO=0 THEN GOSUB 5310 @ GOTO 5280
5300   IF ABS (RHO)>1 THEN GOSUB 5320 @ GOTO 5280 ELSE 5330
5310   GOSUB CLUNK  @ DISP "CONCORDIA ERRORS ARE "&HGL$ (" ALWAYS ")&" CORRELATED (typical values are .95-.995)." @ DISP @ DISP @ WA
IT 2000 @ RETURN
5320   GOSUB CLUNK  @ DISP "ERROR-CORRELATIONS MUST BE WITHIN +1 AND -1" @ DISP @ DISP @ WAIT 2000 @ RETURN
5330   FOR I=1 TO NN ! Assign errors to sets with zero assigned values
5340    IF Ex(I)=0 THEN Ex(I)=XERR
5350    IF Ey(I)=0 THEN Ey(I)=YERR
5360    IF R(I)=0 THEN R(I)=RHO
5370   NEXT I
5380 !
5390  IF PR=2 THEN DISP "NAME FOR THIS YORKFIT";@ INPUT Name$[1,50]
5400  PRINTER IS Pr(2) @ CLEAR @ PRINT @ PRINT RPT$ ("%",32)&"YORKFIT"&RPT$ ("%",33)
5410 !
5420 ! MODEL-1 is Yorks original 'a priori' fit, assumes all scatter due to analytical error and weights points accordingly.  Studen
t's-t is 1.96.
5430 !
5440 ! MODEL-2 weights all points equally (with zero error-correlation), uses Student's-t for N-2 degrees of freedom.  Equivalent to
 knowing nothing about the
5450 ! actual cause of scatter, and invoked only if the MODEL-1 fit has excess
5460 !  scatter.
5470 !
5480 ! MODEL-4 weights points according to their degree of discordance along the best-fit line.  The errors of each point are calcul
ated from the sum of the
5490 ! MODEL-4 weights points according to their degree of discordance along the best-fit line.  The errors of each point are calcul
ated from the sum of the
5500 ! variances due to analytical error and to an error-ellipse for equal X and Y errors and an error-correlation of -1 (equivalent
 to increasing 207/206
5510 ! errors).  The size of the added error-ellipse goes from zero for comletely concordant points to infinite for completely disco
rdant points.
5520 ! Student's-t for MODEL-4 is for N-2 degrees of freedom.
5530 !
5540 !
5550  IF PR=2 THEN PRINT Name$ ELSE PRINT
5560  PRINT @ PRINT TAB (24);"ERRORS ARE PERCENT 2-SIGMA"
5570  PRINT "SET#";TAB (8);"207/235";TAB (18);"%err";TAB (32);"206/238";TAB (42);"%err";TAB (54);"CORREL.";TAB (65);"%7/6err" @ PRIN
T
5580  Ok=0 @ N=NN @ Slope=.1 @ TOLER=.05 @ p=.2 @ M=1
5590 ! Ok set to 1 if one fit completed; toler is MSWD tolerance for model-4 fit;      p is starting error-additive for model 4; M i
s model#.
5600  Slope0=Slope @ I1=0 ! I1 is Yorkfit interation counter
5610  IF M>1 THEN DISP "MODEL";M;" YORKFIT..."
5620  Slope=Slope0
5630  I1=1+I1
5640  IF I1>10 THEN Fail ! Give up if not converged within 10 iterations
5650  IF I1>1 AND M#2 THEN 5890
5660   IF M#4 THEN 5690
5670   ALPHA C_ROW @ FOR Z=1 TO N+4 @ DISP @ NEXT Z @ OFF CURSOR @ ALPHA C_ROW
5680  DISP TAB (5);"%DISCORDANT";TAB (21);"%XERR";TAB (33);"%YERR";TAB (44);"%7/6ERR";TAB (58);"RHO"
5690  FOR I=1 TO N
5700   ON M GOTO 5710,5740,5740,5770
5710   E76(I)=SQR (Ex(I)^2+Ey(I)^2-2*R(I)*Ex(I)*Ey(I)) ! % 207/206 ERR
5720   Ex0(I)=Ex(I) @ Ey0(I)=Ey(I) @ R0(I)=R(I)
```

```
5730   PRINT I;TAB (7);X(I);TAB (17);Ex(I);TAB (31);Y(I);TAB (41);Ey(I);TAB (53);R(I);TAB (65);FNDr(E76(I),3) @ GOTO 5850
5740   G(I)=1 @ Q(I)=1/Slope^2 @ R(I)=0 @ GOTO 5870 ! Model-2 weighting
5750 !
5760 !  Model-4 weighting: Fd is fractional discordance
5770    IF OLD_APEX THEN Fd=SQR ((X_UINT-X(I))^2+(Y_UINT-Y(I))^2)/CHORDLENGTH
5780    IF YOUNG_APEX THEN Fd=SQR ((X_LINT-X(I))^2+(Y_LINT-Y(I))^2)/CHORDLENGTH
5790    Fm=Fd/(1-Fd) ! Multiplier for added errors according to discordance
5800    ! add variances proportional to discordance
5810    Ex(I)=SQR (Ex0(I)^2+(p*Fm)^2) @ Ey(I)=SQR (Ey0(I)^2+(p*Fm)^2)
5820    V76=E76(I)^2+(p*Fm*2)^2 ! New 207/206 variance
5830    R(I)=(Ex(I)^2+Ey(I)^2-V76)/(2*Ex(I)*Ey(I)) ! New rho
5840    DISP I;TAB (8);FNDr(100*Fd,3);TAB (20);FNDr(Ex(I),3);TAB (32);FNDr(Ey(I),3);TAB (44);FNDr(SQR (V76),3);TAB (56);FNDr(R(I),4)

5850   Q(I)=(1/(Y(I)*Ey(I)/200))^2
5860   G(I)=(1/(X(I)*Ex(I)/200))^2
5870   A(I)=SQR (G(I)*Q(I))
5880  NEXT I
5890  IF M=1 AND I1=1 THEN C_ROW=CURSROW  @ ALPHA C_ROW @ AWRIT RPT$ (" ",80)
5900  S1,B5,D,E,Xbar,Ybar=0
5910  IF M=4 AND I1=1 THEN DISP
5920  FOR I=1 TO N
5930   Z(I)=G(I)*Q(I)/(Slope^2*Q(I)+G(I)-2*Slope*R(I)*A(I))
5940   S1=S1+Z(I) @ Xbar=Xbar+Z(I)*X(I) @ Ybar=Ybar+Z(I)*Y(I)
5950  NEXT I
5960  Xbar=Xbar/S1 @ Ybar=Ybar/S1
5970  FOR I=1 TO N
5980   U=X(I)-Xbar @ V=Y(I)-Ybar
5990   B5=B5+(U^2/Q(I)-V^2/G(I))*Z(I)^2
6000   D=D+(U*V/G(I)-R(I)*U^2/A(I))*Z(I)^2
6010  E=E+V*(U/Q(I)-R(I)*V/A(I))*Z(I)^2
6020 NEXT I
6030 Slope0=B5^2+4*D*E
6040 IF Slope0<0 THEN Fail
6050 Slope0=(SQR (Slope0)-B5)/(2*D)
6060 ALPHA C_ROW @ AWRIT "MODEL "&VAL$ (M)&":    Iteration "&VAL$ (I1)&"    Slope="&VAL$ (Slope0)
6070 IF ABS ((Slope0-Slope)/Slope0)>.0001 THEN 5620
6080 Slope(M),Slope=Slope0 @ Inter(M),Inter=Ybar-Slope*Xbar
6090 D,E,SUMS=0
6100 !
6110 ! ERROR ALGORITHM OF TITTERINGTON & HALLIDAY, CHEM GEOL V 26, P 183
6120 FOR I=1 TO N
6130  Y=Y(I)-Slope*X(I)-Inter @ SUMS=SUMS+Z(I)*Y^2
6140  X=X(I)-Z(I)*Y*(R(I)*A(I)-Slope*Q(I))/(Q(I)*G(I))
6150  D=D+X*Z(I) @ E=E+X^2*Z(I)
6160 NEXT I
6170 !
6180 V=N-2 ! DEG. FREEDOM
6190 MSWD=(V<> 0)*SUMS/(V+(V=0)) ! Mean square of weighted deviates
6200 IF M=4 AND ABS (MSWD-1)>TOLER THEN p=p*SQR (MSWD) @ DISP "MSWD: ";MSWD @ GOTO INTERCEPTS
6210 St_t=(V<2)*12.7+(V>1)*1.96*V/SQR (V^2-2.43*V+1.696) ! Student's-t approximation   (from Andy Turek)
6220 B3=SQR (S1/(E*S1-D^2))
6230 F1=B3*SQR (E/S1)
6240 S1=F1*SQR (MSWD)
6250 B5=B3*SQR (MSWD)
6260 IF M>1 THEN 6400
6270 !
6280 IF SUMS<80 THEN 6290 ELSE Prob=0 @ GOTO 6400
6290 D,A=1 ! Calculate chi-square probability
6300  IF INT (V/2)=V/2 THEN 6370
6310  FOR I=2 TO 2000 STEP 2 ! V ODD
```

```
6320   D=D*SUMS/(V+I) @ A=A+D @ IF D<.000001 THEN 6340
6330  NEXT I
6340  D=SQR (PI )
6350  FOR I=.5 TO V/2 @ D=D*I @ NEXT I
6360  Prob=1-EXP (-(SUMS/2))*(SUMS/2)^(V/2)*A/D @ GOTO 6400
6370  FOR I=1 TO V/2-1 @ D=D*SUMS/(2*I) @ A=A+D @ NEXT I ! V EVEN
6380  Prob=A*EXP (-(SUMS/2))
6390 !
6400 FOR Pr=1 TO PR
6410  PRINTER IS Pr(Pr) @ DISP @ IF Pr=1 THEN CLEAR
6420  ON M GOTO 6430,6450,6450,6470
6430  PRINT RPT$ ("-",80) @ PRINT "****MODEL 1 SOLUTION -- ASSUMES ALL SCATTER DUE TO ANALYTICAL ERROR****" @ Ok=1
6440  IF Prob>.15 THEN Errsl(1)=1.96*B3 @ Errint(1)=1.96*F1 @ GOTO 6510 ELSE 6500
6450  PRINT @ PRINT "**********MODEL 2  SOLUTION -- EQUALLY WEIGHTED POINTS WITH UNCORRELATED"
6460  PRINT RPT$ ("*",19)&"ERRORS; ASSUMES NORMAL DISTRIBUTION OF RESIDUALS*****" @ GOTO 6500
6470  PRINT @ PRINT "********* MODEL 4  SOLUTION -- POINTS WEIGHTED ACCORDING TO DEGREE ******" @ PRINT RPT$ ("*",30);" OF CONCORDAN
CE TOWARDS ";
6480  IF OLD_APEX THEN PRINT "UPPER INTERCEPT ****"
6490  IF YOUNG_APEX THEN PRINT "LOWER INTERCEPT ****"
6500  Errsl(M)=St_t*B5 @ Errint(M)=St_t*S1
6510  PRINT @ PRINT TAB (33);"SLOPE";TAB (57);"INTERCEPT" @ PRINT TAB (33);"-----";TAB (57);"---------" @ PRINT
6520  PRINT "BEST-FIT:";TAB (31);FNDr(Slope,6);TAB (56);FNDr(Inter,6) @ PRINT RPT$ ("-",80)
6530   IF M=1 THEN PRINT "ERROR (1-SIGMA, A PRIORI) ";TAB (31);FNDr(B3,4);TAB (56);FNDr(F1,4) @ PRINT @ MSWD_1=MSWD
6540   PRINT "ERROR (1-SIGMA, FROM SCATTER) ";TAB (31);FNDr(B5,4);TAB (56);FNDr(S1,4) @ PRINT
6550   PRINT "ERROR (95% CONFIDENCE LIMITS)";TAB (31);FNDr(Errsl(M),4);TAB (56);FNDr(Errint(M),4) @ PRINT
6560   PRINT "COORDINATES OF CENTROID: X=";FNDr(Xbar,6);"Y=";FNDr(Ybar,6)
6570   Xbar(M)=Xbar @ Ybar(M)=Ybar @ IF M>1 THEN 6600
6580   PRINT @ PRINT "ANALYTICAL ERRORS ALONE WILL CAUSE THE OBSERVED AMOUNT OF SCATTER"
6590   PRINT "OR MORE "&VAL$ (FNDr(100*Prob,2))&"% OF THE TIME, AS INDICATED BY THE M.S.W.D. OF";FNDr(MSWD,3)
6600   PRINT RPT$ ("-",80) @ PRINT
6610   NEXT Pr
6620 FOR I=1 TO NN @ Ex(I)=Ex0(I) @ Ey(I)=Ey0(I) @ R(I)=R0(I) @ NEXT I
6630 C_ROW=CURSROW  @ IF M>1 THEN 6660
6640 IF Prob>.2 THEN INTERCEPTS ELSE 6890
6650 !
6660 CHOOSE_MODEL: DISP "PRESS KEY FOR MODEL-SOLUTION DESIRED." @ DISP USING "3/" @ GOSUB CLEARKEY
6710   ON KEY# 5,"  HELP" GOTO HELP4
6720   ON KEY# 6," MODEL 1" GOTO 6870
6730   ON KEY# 7," MODEL "&VAL$ (M) GOTO 6870
6810   ON KYBD KY,ALLKEY$ GOSUB RETRN
6820 !
6830  KEY LABEL
6840  GOTO 6840
6850 !
6860  OFF KEY# @ M=1 @ GOTO INTERCEPTS
6870  OFF KEY# @ GOTO INTERCEPTS
6880 !
6890 DISP "PRESS k5 FOR MODEL 2    (equal wts for all points)" @ DISP TAB (7);"k6 FOR MODEL 4-UI (7/6 errors increase towards lower
intercept)"
6900 DISP TAB (7);"k7 FOR MODEL 4-LI (7/6 errors increase towards upper intercept)" @ DISP USING "3/" @ GOSUB CLEARKEY
6940  ON KEY# 4,"  HELP" GOTO HELP4
6950  ON KEY# 5," MODEL 2" GOTO 7090
6960  ON KEY# 6,"MODEL 4-UI" GOTO 7100
6970  ON KEY# 7,"MODEL 4-LI" GOTO 7110
7050  ON KYBD KY,ALLKEY$ GOSUB RETRN
7060  KEY LABEL
7070 !
7080  GOTO 7080
7090  OFF KEY# @ OFF KYBD @ M=2 @ GOTO 5600
7100  OLD_APEX=1 @ YOUNG_APEX=0 @ GOTO 7120
```

```
7110  OLD_APEX=0 @ YOUNG_APEX=1
7120  OFF KEY# @ OFF KYBD @ M=4 @ Slope=Slope(1) @ Inter=Inter(1) @ GOTO CALC_INTERCEPTS
7130  INTERCEPTS: Slope=Slope(M) @ Inter=Inter(M) @ OFF KYBD
7140 !
7150 CALC_INTERCEPTS: R=CURSROW  @ DISP "CALCULATING CHORD-INTERCEPTS....." @ OFF CURSOR
7160    FOR J=0 TO 1
7170     T1=-500+5500*J ! 1st Trial-T is -500 Ma (lower-int) or 5500 Ma (upper)
7180     C=FNC(T1) @ X=(Inter+C*FNA(T1)-FNB(T1))/(C-Slope)
7190     IF X<-1 THEN NO_INTER ELSE T=FND(X) @ ALPHA R,42+J*10 @ AWRIT VAL$ (INT (T))&"   "
7200     IF ABS (T-T1)<.01 THEN Cint(J+1)=T ELSE T1=T @ GOTO 7180
7210    NEXT J
7220    IF M<4 OR ABS (MSWD-1)<TOLER THEN 7260
7230    X_UINT=FNA(Cint(2)) @ Y_UINT=FNB(Cint(2)) @ X_LINT=FNA(Cint(1)) @ Y_LINT=FNB(Cint(1)) ! x & y values or concordia-intercepts
 of chord.
7240    CHORDLENGTH=SQR ((X_UINT-X_LINT)^2+(Y_UINT-Y_LINT)^2) @ GOTO 5600
7250 !
7260 ! Calculate concordia-intercept errors ------------------
7270 R=CURSROW +1 @ ALPHA R,1 @ DISP "CALCULATING ERROR-ENVELOPE INTERCEPTS..."
7280 Errsl=Errsl(M) @ Errint=Errint(M)
7290 Xbar=Xbar(M) @ Ybar=Ybar(M)
7300   FOR J=3 TO 6
7310    T1=7000*(J>4)-1000 @ A=1-2*(J/2#INT (J/2)) @ I=INF
7320    GOSUB ERRCALC  @ Cint(J)=T
7330   NEXT J @ GOTO PRINT_SOLUTION
7340 ERRCALC: ! Calculate concordia-intercepts of error-envelope about line
7350  S=FNC(T1) @ B=FNB(T1)-S*FNA(T1)-Inter
7360  D=2*(B*(S-Slope)+Xbar*Errsl^2) @ E=(S-Slope)^2-Errsl^2
7370  G=D^2-4*E*(B^2-Errint^2) @ IF G<0 THEN NO_ENV_INTER
7380  X=(A*SQR (G)-D)/(2*E) @ IF X<-1 THEN NO_ENV_INTER
7390  T=FND(X) @ ALPHA R,42+(J-3)*7 @ AWRIT VAL$ (INT (T))&"   " @ IF ABS (T-T1)>I THEN NO_ENV_INTER
7400  IF ABS (T-T1)<.01 THEN RETURN ELSE I=ABS (T-T1) @ T1=T @ GOTO ERRCALC
7410 !
7420 PRINT_SOLUTION: ALPHA C_ROW @ Cint(7)=FNDr(ABS (Cint(4)-Cint(3))/2*(Cint(3)*Cint(4)#0),2)
7430    Cint(8)=FNDr(ABS (Cint(6)-Cint(5))/2*(Cint(5)*Cint(6)#0),2)
7440   PRINT @ PRINT TAB (12);"<<<<<<< MODEL";M;" CONCORDIA-INTERCEPT SOLUTION >>>>>>>" @ PRINT @ PRINT
7450 A,B,D1,D2=1
7460   IF Cint(7) THEN A=10^INT (LGT (ABS (Cint(7))))
7470   IF Cint(8) THEN B=10^INT (LGT (ABS (Cint(8))))
7480   IF Cint(1) THEN D1=1-INT (LGT (ABS (A/Cint(1))))
7490   IF Cint(2) THEN D2=1-INT (LGT (ABS (B/Cint(2))))
7500   PRINT "LOWER INTERCEPT:  ";@ IF Cint(1) THEN PRINT FNDr(Cint(1),D1);
7510   IF Cint(1) AND Cint(7) THEN PRINT " +/-";Cint(7);"Ma";ELSE PRINT "Ma";
7520   IF Cint(3)=0 AND Cint(4)=0 THEN PRINT ELSE PRINT TAB (52);"( ";
7530   IF Cint(3) THEN PRINT ")";FNDr(Cint(3),D1);
7540   IF Cint(4) THEN PRINT "(";FNDr(Cint(4),D1);")" ELSE PRINT ")"
7550   PRINT "UPPER INTERCEPT:  ";@ IF Cint(2) THEN PRINT FNDr(Cint(2),D2);
7560   IF Cint(2) AND Cint(8) THEN PRINT " +/-";Cint(8);"Ma";ELSE PRINT "Ma";
7570   IF Cint(5)=0 AND Cint(6)=0 THEN PRINT ELSE PRINT TAB (52);"( ";
7580   IF Cint(5) THEN PRINT ")";FNDr(Cint(5),D2);
7590   IF Cint(6) THEN PRINT "(";FNDr(Cint(6),D2);")" ELSE PRINT ")"
7600   PRINT @ PRINT "(UNCERTAINTIES ARE 95%-CONFIDENCE LIMITS)" @ PRINT RPT$ ("-",80)
7610 NCHORDS=1+NCHORDS @ GOSUB DRAWLINE  @ BEEP @ IF CRT THEN LABEL_INTERCEPTS
7620 !
7630 DISP @ DISP "PRESS "&HGL$ (" LABEL INT. ")&" FOR PLOT-LABEL OF INTERCEPTS."
7640 DISP @ DISP "PRESS "&HGL$ (" CONTINUE ")&" TO DECLINE." @ DISP USING "3/" @ GOSUB CLEARKEY
7700       ON KEY# 6,"LABEL INT." GOTO LABEL_INTERCEPTS
7710       ON KEY# 7," CONTINUE" GOTO 7960
7790       ON KYBD KY,ALLKEY$ GOSUB RETRN
7800       KEY LABEL
7810       GOTO 7810
```

```
7820 !
7830 LABEL_INTERCEPTS: OFF KEY# @ OFF KYBD
7840  CS=THT @ W=CS*AR*(1-.25*HP) @ MOVE XMAX,YMIN @ SETGU
7850  IF Cint(2) THEN I1$=VAL$ (FNDr(Cint(2),D2)) ELSE I1$="***"
7860  IF Cint(1) THEN I3$=VAL$ (FNDr(Cint(1),D1)) ELSE I3$="***"
7870  IF Cint(8) THEN I2$=VAL$ (Cint(8)) ELSE I2$="***"
7880  IF Cint(7) THEN I4$=VAL$ (Cint(7)) ELSE I4$="***"
7890  L1=LEN (I1$) @ L2=LEN (I2$) @ L3=LEN (I3$) @ L4=LEN (I4$)
7900  NCAR=14+L1+L2+L3+L4 @ CSIZE CS,AR @ LORG 1
7910  IMOVE -((NCAR+14+0*HP)/2*W),3.5*CS*NCHORDS-CS/2 @ LABEL "INTERCEPTS AT"
7920  IMOVE -((NCAR-15)/2*W),0 @ LABEL I1$ @ IMOVE (.5+0*HP+L1)*W,CS @ GOSUB PM  @ IMOVE (1+0*HP)*W/2,0 @ LABEL I2$&" and "&I3$
7930  IMOVE (L2+L3+5.5+0*HP)*W,CS @ GOSUB PM  @ IMOVE (.5+0*HP)*W,0 @ LABEL I4$&" Ma"
7940  SETUU @ MOVE XMAX,YMIN @ SETGU @ IMOVE -((NCAR+12)/2*W),3.5*CS*NCHORDS-2.5*CS @ LABEL "(MSWD="&VAL$ (FNDr(MSWD_1,3))&")"
7950   !
7960   OFF KYBD @ OFF KEY# @ DEG @ SETUU @ MOVE XMIN,YMAX @ PEN PLR=1 @ ALPHA
7970 !
7980  GOTO ADD
7990 !
8000 Fail: PRINT @ PRINT "DATA CANNOT BE FIT TO A STRAIGHT LINE USING MODEL";M @ PRINT @ PRINT @ GOSUB CLUNK
8010  IF Ok=0 THEN 7980
8020  IF M=4 THEN M=2 @ GOTO 5600
8030  IF M=2 THEN M=1 @ GOTO 7130
8040 !
8050 NO_INTER: PRINT "***** CHORD DOES NOT HAVE TWO CONCORDIA INTERCEPTS *****" @ PRINT @ Cint(1+J)=0 @ GOTO 7210
8060 NO_ENV_INTER: PRINT @ PRINT "***** ARM";J-2;"OF ERROR-ENVELOPE DOES NOT INTERSECT CONCORDIA *****" @ PRINT @ T=0 @ RETURN
8070 END
8080 !
8090 PRINT_PLOT: ! Dump CRT-plot to printer
8100          CLEAR @ DISP "PRESS "&HGL$ (" 8 ")&" FOR 82905 PRINTER, "&HGL$ (" 2 ")&" FOR 2673 PRINTER."
8110          ON KYBD KY,"1234567890" GOTO 8130
8120          GOTO 8120
8130          OFF KYBD @ PRINTER IS 701,80 @ K=KY-48
8140          IF K=8 THEN DUMP GRAPHICS
8150          IF K=2 THEN DUMP GRAPHICS 0,0,0,1
8160          IF K#8 AND K#2 THEN GOSUB CLUNK  @ GOTO PRINT_PLOT
8170          PRINTER IS Pr(2) @ GOTO RE_DATA
8180 !
8190 DRAWLINE: IF CRT THEN LINE(NCHORDS,1)=Slope @ LINE(NCHORDS,2)=Inter
8200         CLIP XMIN+Xspred/100,XMAX-Xspred/35,YMIN+Yspred/100,YMAX-Yspred/35
8210         IF SOLIDCHORD=0 THEN LINE TYPE NCHORDS+4
8220         MOVE XMIN,XMIN*Slope+Inter @ PEN CPEN
8230         DRAW XMAX,XMAX*Slope+Inter
8240         MOVE XMIN,YMIN @ LINE TYPE 1
8250         CLIP XMIN,XMAX,YMIN,YMAX @ RETURN
8260 !
8270 LETTER: CLEAR @ SETUU @ UNCLIP ! Draft a phrase on the plot
8280         IF PLOTTER_DUMP THEN 8330
8290         DISP @ DISP "PRESS "&EL$&" TO ESCAPE" @ DISP
8300         DISP "CHAR.-HEIGHT IN mm (tick labels are";FNDr(THT*Ymm/YLOC/2,2);")  [, SLANT (0-30)  [, ROTATION]]";
8310         GOSUB RETRIEVE  @ IF Ninputs=0 THEN RE_DATA
8320         H=0(1) @ SL=0(2) @ ROT=0(3)
8330         PEN CPEN @ CS=2*H*YLOC/Ymm @ W=AR*CS*(1-HP/3.1) @ CSIZE CS,AR,SL @ LDIR ROT
8340         IF HP THEN 8380
8350  DISP @ DISP "ENTER COORDINATES (X, Y) OF LOWER-LEFT CORNER OF PHRASE"
8360         GOSUB RETRIEVE  @ IF Ninputs(2 THEN GOSUB BAD_INPUT  @ GOTO 8350
8370         X=0(1) @ Y=0(2) @ PEN CPEN @ GOTO 8400
8380         IF PLOTTER_DUMP THEN 8400 ELSE DISP @ DISP "POSITION PEN WITH PLOTTER-CONTROLS, THEN PRESS "&HGL$ (" CONT ") @ PEN CPEN
 @ PAUSE
8390         CURSOR X,Y
8400         MOVE X,Y @ LORG 1
```

```
8410        IF PLOTTER_DUMP=0 THEN DISP @ DISP "PHRASE TO BE DRAFTED";@ LINPUT PHRASE$
8420        L=LEN (PHRASE$) @ SETGU @ IF HP THEN 8450
8430        NPHRASES=1+NPHRASES @ LETTER$(NPHRASES)=PHRASE$ @ LETTER(NPHRASES,1)=X @ LETTER(NPHRASES,2)=Y @ LETTER(NPHRASES,3)=H @
LETTER(NPHRASES,4)=SL
8440        LETTER(NPHRASES,5)=ROT @ IF CPEN=-1 THEN NPHRASES=NPHRASES-2
8450        IF POS (PHRASE$,"\")=0 AND POS (PHRASE$,"^")=0 AND POS (PHRASE$,"+/-")=0 THEN LABEL PHRASE$ @ GOTO 8530
8460        IF O(3)#0 THEN GOSUB CLUNK  @ DISP @ DISP "SORRY, YOU CAN'T USE \, ^, OR +/- SYMBOLS WITH NONZERO ANGLES." @ DISP @ GOT
O LETTER
8470        FOR I=1 TO L
8480         IF I>L-2 THEN 8500
8490         IF PHRASE$[I,I+2]="+/-" THEN GOSUB PM  @ I=I+2 @ GOTO 8520
8500         A=(PHRASE$[I,I]="^")-(PHRASE$[I,I]="\")
8510         IF A=0 THEN LABEL PHRASE$[I,I] @ IMOVE W,CS ELSE IMOVE 0,A*CS/4
8520        NEXT I
8530        LDIR 0 @ PEN PLOTTER_DUMP @ SETUU @ IF PLOTTER_DUMP THEN RETURN ELSE 8290
8540 PM: IMOVE 0,.13*CS @ LABEL "+" @ IMOVE 0,.62*CS @ LABEL "-" @ IMOVE W,1.25*CS @ RETURN ! +/- symbol
8550 !
8560 DATAFILE: CLEAR @ DISP "NAME OF DATA-FILE TO BE RETRIEVED?"
8570   DISP @ DISP "(to get a disk catalog, enter "&HGL$ (" C ")&" for drive 0, "&HGL$ (" c ")&" for drive 1)" @ DISP @ INPUT FILE$
8580   IF FILE$="C" THEN CAT ":D700" @ GOTO 8600
8590   IF FILE$="c" THEN CAT ":D701" ELSE GOTO 8620
8600   DISP @ DISP "PRESS "&HGL$ (" CONT ")&" TO CONTINUE." @ PAUSE
8610   GOTO DATAFILE
8620   IF ERRTYPE<0 THEN GOSUB ERRORS
8630   ON ERROR GOTO WRONGFILE
8640   INTEGER CLN(5)
8650   ASSIGN# 1 TO  FILE$ @ ON ERROR GOTO YORVERT
8660   READ# 1,1 ; N$(1),XX()@ OFF ERROR
8670 !
8680 ! ############### RETRIEVE nX45 NUMERIC DATA-FILE ROWS FROM TAPE ##########
8690   DISP @ DISP "COLUMN #s FOR 207/235,";
8700   IF ERRTYPE THEN DISP " 206/238  [,%6/8-Err] [, %7/6-Err]";
8710   IF ERRTYPE=0 THEN DISP " [%Err,]  206/238  [,%Err] [, Err-Correl]";
8720   GOSUB RETRIEVE  @ MAT CLN=ZER @ IF ERRTYPE=0 THEN 8770
8730   IF Ninputs=2 THEN CLN(1)=O(1) @ CLN(2)=O(2) @ GOTO 8800
8740   IF Ninputs=4 THEN MAT CLN=O@ GOTO 8800
8750   IF Ninputs#2 AND Ninputs#4 THEN GOSUB BAD_INPUT  @ GOTO 8700
8760 !
8770   IF Ninputs=2 THEN CLN(1)=O(1) @ CLN(3)=O(2)
8780   IF Ninputs>3 THEN MAT CLN=O
8790   IF Ninputs<2 OR Ninputs>5 OR Ninputs=3 THEN GOSUB BAD_INPUT  @ GOTO 8690
8800   CLEAR @ PRINT "SET#";TAB (8);"NAME";TAB (27);"207/235";
8810   IF ERRTYPE THEN PRINT TAB (41);"206/238";TAB (55);"%6/8-Err";TAB (69);"%7/6-Err"
8820   IF ERRTYPE=0 THEN PRINT TAB (38);"%Err";TAB (48);"206/238";TAB (59);"%Err";TAB (70);"RHO"
8830   PRINT @ GOSUB NAMEBLANK  @ MAT S=ZER
8840   ON ERROR GOTO 8940
8850   FOR I=1 TO 100
8860    READ# 1,I ; N$(I),XX()
8870    FOR J=1 TO 5-ERRTYPE
8880     IF CLN(J) THEN S(I,J)=XX(CLN(J))
8890    NEXT J
8900    PRINT I;TAB (6);N$(I);TAB (26);
8910    IF ERRTYPE THEN PRINT S(I,1);TAB (41);S(I,2);TAB (55);S(I,3);TAB (69);S(I,4)
8920    IF ERRTYPE=0 THEN PRINT S(I,1);TAB (37);S(I,2);TAB (47);S(I,3);TAB (58);S(I,4);TAB (68);S(I,5)
8930   NEXT I
8940    GOTO 9640
8950 !
8960 YORVERT: ! CONVERT VISICALC FILES TO 100X5 ARRAYS FOR PLOTTING/YORKFIT
8970 !
8980 OFF ERROR
```

```
8990 DIM VC$[24],I$[20],J$[20],PA$(5)[7]
9000 IF ERRTYPE=0 THEN PA$(1)="207/235" @ PA$(2)="%7/5Err" @ PA$(3)="206/238" @ PA$(4)="%6/8Err" @ PA$(5)="Rho"
9010 IF ERRTYPE THEN PA$(1)="207/235" @ PA$(2)="206/238" @ PA$(3)="%6/8Err" @ PA$(4)="%7/6Err"
9020 CLEAR @ DISP "INPUT   BEGINNING, ENDING   ROW-NUMBERS CONTAINING SAMPLE-DATA";@ GOSUB RETRIEVE
9030 IF Ninputs#2 THEN GOSUB BAD_INPUT  @ GOTO 9020 ELSE RMIN=0(1) @ RMAX=0(2)
9040  COLINPUT: DISP @ DISP "(press "&EL$&" for column-name listing)"
9050  DISP @ DISP "INPUT COLUMNS FOR: 207/235, ";
9060  IF ERRTYPE=0 THEN DISP "[%Err,]  206/238 [,%Err]  [, Rho(X-Y)]";@ GOTO 9080
9070  IF ERRTYPE THEN DISP "206/238   [,%6/8-Err]  [, %7/6-Err]";
9080  GOSUB RETRIEVE_STR  @ IF Ninputs=1 THEN GOSUB BAD_INPUT  @ GOTO COLINPUT  ELSE NCOLS=Ninputs
9090  IF NCOLS=0 THEN GOSUB COLUMN_NAMES  @ GOTO 9050
9100 FOR I=1 TO 5-ERRTYPE
9110   IF NUM (O$(I))>47 AND NUM (O$(I))<59 THEN GOSUB CLUNK  @ DISP @ DISP "### COLUMNS ARE DESIGNATED BY LETTERS (e.g. M, B, AK...
) ###" @ DISP @ GOTO 9040
9120 NEXT I
9130 CNUM=2+2#(O$(2)#"")+(O$(5)#"")
9140 FOR I=1 TO RMAX-RMIN+1 !             ZERO ALL ARRAY-ELEMENTS
9150 FOR J=1 TO 5 @ S(I,J)=0 @ NEXT J
9160 NEXT I
9170 ALPHA 1 @ CLEAR @ DISP "CONVERTING VISICALC FILE...." @ NUMVALS=0
9180 GOSUB NAMEBLANK  @ ASSIGN# 1 TO  FILE$ @ MAT S=ZER @ DISP @ C_ROW=CURSROW  @ GOTO 9270
9190 !
9200 VC_STRINGS: READ# 1 ; VC$@ OFF ERROR @ ALPHA C_ROW,1 @ AWRIT VC$&"   "
9210   IF VC$[1,1]#">" THEN NEXT_I
9220   A=1+POS (VC$,":") !                STRING POSITION WHERE DATA STARTS
9230   E=NUM (VC$[A,A]) @ IF E=34 OR E=39 THEN A=A+1 ! 1st DATA-CHAR.
9240   B=NUM (VC$[3,3]) @ D=NUM (VC$[2,2]) @ C=3+(B>64 AND B<91) ! C is 1st row-# char.
9250   VCROW=VAL (VC$[C]) @ RETURN ! VISICALC ROW-#
9260 !
9270 FOR I=1 TO 3150
9280  ON ERROR GOTO FINISH
9290  GOSUB VC_STRINGS  @ IF VCROW<RMIN THEN FINISH
9300  IF VCROW>RMAX THEN GOTO NEXT_I  ELSE ROWN=VCROW-RMIN+1 ! SAMPLE#
9310  IF C=3 THEN COLN=D-66
9320  IF C=4 THEN COLN=24+26#(D-65)+B-64
9330  IF COLN=-1 THEN L=LEN (VC$) @ N$(ROWN)[1,9]=VC$[A,L-1] @ ALPHA C_ROW,L+4 @ AWRIT N$(ROWN)&"   " @ GOTO NEXT_I
9340  IF COLN=0 THEN N$(ROWN)[10,18]=VC$[A,LEN (VC$)-1] @ GOTO NEXT_I
9350  V$=VC$[2,C-1]
9360  FOR PARNUM=1 TO NCOLS
9370   F,R=0 @ GOSUB EXTRACT  @ IF F THEN 9390
9380   IF R THEN NEXT_I
9390  NEXT PARNUM
9400  GOTO NEXT_I
9410 !
9420 NAMEBLANK: FOR I=1 TO 100 @ N$(I)="" @ NEXT I @ RETURN
9430 !
9440  EXTRACT: ! EXTRACT NUMERIC VALUES FROM VISICALC STRING
9450   IF V$#O$(PARNUM) THEN F=1 @ RETURN !     IF NOT CORRECT COLUMN
9460   IF ERRTYPE THEN COLN=PARNUM @ GOTO 9480
9470   IF PARNUM=2 AND NCOLS=2 THEN COLN=3 ELSE COLN=PARNUM
9480   ON ERROR GOTO 9520
9490   A=VAL (VC$[A]) @ OFF ERROR @ S(ROWN,COLN)=A @ R=1
9500   ALPHA C_ROW,41 @ OFF CURSOR
9510   DISP "SAMPLE#";ROWN;"   ";PA$(COLN);" =";A
9520   OFF CURSOR @ OFF ERROR @ RETURN
9530 !
9540 NEXT_I: NEXT I
9550 FINISH: OFF ERROR @ CLEAR @ PRINT " SET#";TAB (29);"207/235";
9560  IF ERRTYPE=0 THEN PRINT TAB (38);"%Err";TAB (50);"206/238";TAB (60);"%Err";TAB (70);"RHO" @ PRINT @ GOTO 9580
9570  IF ERRTYPE THEN PRINT TAB (42);"206/238";TAB (55);"%6/8-Err";TAB (68);"%7/6-Err" @ PRINT
```

```
9580 ON ERROR GOTO 9630
9590 FOR I=1 TO RMAX-RMIN+1
9600  PRINT I;TAB (5);N$(I);TAB (28);S(I,1);
9610  IF ERRTYPE=0 THEN PRINT TAB (37);FNDr(S(I,2),3);TAB (49);S(I,3);TAB (59);FNDr(S(I,4),3);TAB (67);S(I,5) @ GOTO 9630
9620  IF ERRTYPE THEN PRINT TAB (41);S(I,2);TAB (54);S(I,3);TAB (67);FNDr(S(I,4),3)
9630 NEXT I
9640 OFF ERROR @ DFILE=1 @ DISP @ DISP "PRESS "&H6L$ (" CONT ")&" TO CONTINUE" @ PAUSE
9650 ALPHA 1 @ CLEAR @ KEY LABEL @ RETURN
9660 WRONGFILE: GOSUB CLUNK  @ DISP @ DISP "### FILE '";FILE$;"' NOT RECOGNIZED ON SPECIFIED DISC;   TRY AGAIN ###" @ DISP @ WAIT 10
00 @ GOTO DATAFILE
9670 CONT_DIF: ! SOLUTION FOR COUPLED Pb-Rn DIFFUSION FROM HOMOGENEOUS U-BEARING REGION INTO U-Pb-Ra FREE REGION.
9680  DISP "AGE (Ma), Rn/Pb DIFFUSION RATIO/1E10";@ INPUT T,R@ R=R*10000000000
9690  L0=66200000 @ Z=137.88 ! LAMBDA Rn, 238/235@PEN CPEN
9700  FOR I=1 TO 100
9710   L1=.0000001*(COS (.9*I)+1)*EXP (I*.15) @ R0=R*L1
9720   K4=L8/(L0+R0-L8) @ K1=L0*K4/(L1-L8) @ K2=-(L0*K4/(L1-L0-R0)) @ K3=-K1-K2
9730   X=L5/(L1-L5)*(1-EXP ((L5-L1)*T)) @ A=(L8-L0-R0)*T
9740   IF A>-227 THEN C=K2*EXP (A) ELSE C=0
9750   Y=K1+C+K3*EXP ((L8-L1)*T)
9760   IF I>1 THEN DRAW X,Y ELSE MOVE X,Y
9770  NEXT I
9780  MOVE XMIN,YMAX @ BEEP @ CLEAR @ GOTO PLABEL
9790 !
9800 PLOTBOX_SIZE: A,B=0 ! CALCULATE LIMITS FOR PLOT DEFINED BY BOX-SIZE
9810  RATIO_=(LIMX2-LIMX1)/(LIMY2-LIMY1)
9820  IF RATIO_>1 THEN Bgu=100 @ A=1 @ GOSUB AGU  @ GOSUB BGU
9830  IF RATIO_<= 1 THEN Agu=100 @ B=1 @ GOSUB BGU  @ GOSUB AGU
9840  ON ERROR GOTO 9890
9850  LIMIT A1,A2,B1,B2
9860  IF (C=0)*(RATIO >1 EXOR RATIO_>1) THEN C=1 @ GOTO PLOTBOX_SIZE
9870  OFF ERROR @ LIMX1=A1 @ LIMX2=A2 @ LIMY1=B1 @ LIMY2=B2 @ RETURN
9880 !
9890 OFF ERROR @ GOSUB CLUNK  @ DISP @ DISP "#### INSUFFICIENT ROOM FOR LABELS WITH SPECIFIED PLOTBOX-LIMITS ####" @ DISP @ DISP @ G
OTO 1560
9900 !
9910 AGU:
9920  MGy=(LIMY2-LIMY1)/(Bgu-YL-YEND) ! millimeters/GU in Y-direction
9930  YL_MM=YL*MGy @ YEND_MM=YEND*MGy
9940  B1=LIMY1-YL_MM @ B2=LIMY2+YEND_MM
9950  IF A THEN Agu=XL+XEND+100*(LIMX2-LIMX1)/(B2-B1)
9960 RETURN
9970 !
9980 BGU:
9990  MGx=(LIMX2-LIMX1)/(Agu-XL-XEND) ! millimeters/GU in X-direction
10000  XL_MM=XL*MGx @ XEND_MM=XEND*MGx
10010  A1=LIMX1-XL_MM @ A2=LIMX2+XEND_MM
10020  IF B THEN Bgu=YL+YEND+100*(LIMY2-LIMY1)/(A2-A1)
10030 RETURN
10040 COLUMN_NAMES:
10050 ! SUBROUTINE TO FIND AND LIST COLUMN-NAMES OF VISICALC FILES.
10060 ! ASSUMES THAT COLUMN-NAMES ARE UNDERLINED WITH REPEATING "=", SO THAT
10070 ! THE VC-STRING IS "/-=", AND THAT ROWS 3 AND 4 CONTAIN THE COLUMN-NAMES
10080 ! ALSO ASSUMES A 9-CHARACTER COLUMN-WIDTH.
10090 DIM CN$(61,2)[9]
10100 !
10110 ALPHA 1 @ CLEAR @ C_ROW=CURSROW  @ B$="         "
10120 DISP "Locating column-name rows.  Please wait..."
10130 FOR I=1 TO 61 @ CN$(I,1),CN$(I,2)=B$ @ NEXT I
10140 ASSIGN# 1 TO  FILE$ @ ON ERROR GOTO 10180
10150 FOR I=1 TO 3150
```

```
10160  READ# 1 ; VC$@ IF POS (VC$,"/-=") THEN 10190
10170 NEXT I
10180 OFF ERROR @ DISP @ DISP HGL$ (" NO '/-=' VISICALC-CELLS FOUND - CAN'T FIND NAMES ") @ DISP @ DISP @ RETURN
10190 ALPHA C_ROW @ CLEAR
10200 FOR I=1 TO 3150
10210   ON ERROR GOTO 10300 @ GOSUB VC_STRINGS
10220   ON 1+(VCROW>4)+2*(VCROW<3) GOTO 10230,10280,10300
10230   IF C=3 THEN COLN=D-66
10240   IF C=4 THEN COLN=24+26*(D-65)+B-64
10250   IF COLN<1 THEN NEXT_I
10260   IF VCROW=3 OR VCROW=4 THEN P=POS (VC$,CHR$ (13)) @ CN$(COLN,VCROW-2)=TRIM$ (VC$[A,P-1])
10270 !
10280 NEXT_I: NEXT I
10290 !
10300 CLEAR @ PRINT "COLUMN";TAB (10);"COLUMN-NAME" @ PRINT
10310 FOR I=1 TO 61
10320  IF I<25 THEN I$=CHR$ (I+66)
10330  IF I>24 AND I<51 THEN I$="A"&CHR$ (I+40)
10340  IF I>50 THEN I$="B"&CHR$ (I+14)
10350  IF LEN (CN$(I,1))=0 AND LEN (CN$(I,2))=0 THEN 10380
10360  IF CN$(I,1)=B$ AND CN$(I,2)=B$ THEN 10380
10370  PRINT "  ";I$;TAB (10);CN$(I,1);TAB (20);CN$(I,2)
10380 NEXT I
10390 PRINT @ PRINT @ RETURN
10400 !
10410 DUMP_TO_PLOTTER: OFF KEY# @ OFF KYBD ! Transfer CRT-plot to HP-7470A plotter
10420 CLEAR @ DISP "Press "&HGL$ (" ENDLINE ")&" to confirm dump of CRT-plot to plotter, " @ DISP "enter any number to decline."
10430 GOSUB RETRIEVE  @ IF Ninputs>0 THEN RE_DATA
10440  CPEN=1 @ SPEED$="VS" @ PLOTSIZE=8 @ PLOTTER_DUMP,HP=1 @ CRT=0 @ PLR=705 @ GOSUB 1600 @ CLEAR @ C_ROW=CURSROW +1
10450  FOR JJ=1 TO C_CT
10460   K,N=1 @ X(K)=CACHE(JJ,1) @ Y(K)=CACHE(JJ,3) @ Ex(K)=CACHE(JJ,2) @ Ey(K)=CACHE(JJ,4) @ R(K)=CACHE(JJ,5) @ SYMBOL=SYMBOL(JJ,1)

10470   GOSUB PLOTT
10480  NEXT JJ
10490 !
10500  FOR I=1 TO NCHORDS
10510   Slope=LINE(I,1) @ Inter=LINE(I,2) @ GOSUB DRAWLINE
10520  NEXT I
10530 !
10540 FOR JJ=1 TO NPHRASES
10550  PHRASE$=LETTER$(JJ) @ X=LETTER(JJ,1) @ Y=LETTER(JJ,2) @ H=LETTER(JJ,3) @ SL=LETTER(JJ,4) @ ROT=LETTER(JJ,5) @ GOSUB LETTER
10560 NEXT JJ
10570 PLOTTER_DUMP=0 @ GOTO RE_DATA
10580 !
10590 HELP: OFF KYBD @ OFF KEY# @ CLEAR ! Display instructions for special-function keys.
10600 DIM DF1$[2],DF2$[10],DF3$[60],DF4$[60],DF5$[60]
10610  PRINTER IS 1 @ PRINT TAB (16);HGL$ (" FUNCTIONS OF KEYS DEFINED DURING LAST DISPLAY: ") @ PRINT @ PRINT
10620  RETURN
10630 !
10640 PRINT_HELP: OFF KEY#
10650 FOR I=1 TO NUM_KEYS
10660  READ DF1$,DF2$,DF3$,DF4$,DF5$
10670  PRINT "KEY# "&DF1$;TAB (9);"- "&DF2$&":";TAB (23);DF3$
10680  IF DF4$#"" THEN PRINT TAB (23);DF4$
10690  IF DF5$#"" THEN PRINT TAB (23);DF5$
10700  PRINT
10710 NEXT I
10720 PRINT @ PRINT "Use "&HGL$ (" ROLL ")&" key to scroll display, press "&HGL$ (" CONT ")&" to return to previous display."
10730 PAUSE @ RETURN
10740 !
```

```
10750 HELP2: GOSUB HELP  @ RESTORE 10760 @ NUM_KEYS=16 @ GOSUB PRINT_HELP  @ GOTO RE_DATA
10760 DATA "1",LETTER,Draft a phrase anywhere on the plot.,"","
10770 DATA "2","OPEN Eell",Choose an open (unfilled) 2-sigma error-ellipse as the,plotting symbol.,""
10780 DATA "9",SOLID Eell,Choose a solid (filled-in) 2-sigma error-ellipse as the,plotting-symbol.,""
10790 DATA "4",POLYGON,Choose an open or solid polygon (either regular or,star-shaped) of arbitrary size & orientation as the,plotti
ng-symbol.
10800 DATA "7",YORKFIT,Calculate a York-style best-fit line to the last-,"plotted set of data, starting with set# 1.  Also"
10810 DATA "calculates isochron age, if appropriate."
10820 DATA "10",PEN 1,Use pen# 1 (plotter) or solid-line (CRT).,"","
10830 DATA "3",PEN 2,Use pen# 2 (plotter only).,"","
10840 DATA "12",ERASE PEN,Use 'erasing pen' to un-draw symbols or letters,(CRT only).,""
10850 DATA "13",?ERRORS?,"Choose error-format: default is 206/238 %errors, 207/235","%errors, err.-correl.  Optional is 206/238 %err
ors,","207/206 %errors."
10860 DATA "8",CONT.DIFF.,Calculate and draw a continuous-diffusion curve for,combined Pb and intermediate-daughter leakage.,""
10870 DATA "11",PRINT PLOT,Dump the CRT plot to an HP-82905B or 2673 printer.,"","
10880 DATA "3",HPLOT DUMP,"Dump the CRT plot to an HP-7470a plotter,using a","plot-size of 8, pen# 1, and fastest pen-speed.","
10890 DATA "6",REDRAW,Start another plot from scratch.,"","
10900 DATA "14",DATAFILE,Bring data from either a numeric or VISICALC data-file,into memory.,""
10910 DATA "","+‡OX ...",Use any keyboard-key symbol as the plotting-symbol.,"","
10920 DATA "",CTL S,Use a solid line instead of a dashed line for concordia,chords.,""
10930 !
10940 HELP1: GOSUB HELP  @ RESTORE 10960 @ NUM_KEYS=14 @ GOSUB PRINT_HELP  @ GOTO BEGIN
10950 !
10960 DATA "1",0 CTICKS,Use small circles as concordia ticks and label the,concordia ticks horizontally.,""
10970 DATA "2",CRT-PRINT,"Use the CRT as the printing device, rather than a",hard-copy printer.,""
10980 DATA "3",CRT PLOT,"Use the CRT as the plotting device, rather than a",hard-copy printer.,""
10990 DATA "4",PEN 1,"Use pen# 1 of the HP-7470A plotter, or the white-line pen","(CRT) as the plotting-pen.","
11000 DATA "5",DATAFILE,Load data from a numeric or VISICALC data-file into,memory.,""
11010 DATA "6",‡NORM‡,Use 'normal'-size labels and symbols.,"","
11020 DATA "7",START,Enter axis-names & axis-limits for plot and start,plotting.,""
11030 DATA "8",\ CTICKS,"Use short line, normal to the concordia curve, as the",concordia ticks and label the ticks normal to the cu
rve.,""
11040 DATA "9",PRINTER,Use a hard-copy printer as the printing device rather,than the CRT.,""
11050 DATA "10",HARDPLOT,Use an HP-7470A plotter as the plotting device rather,than the CRT.,""
11060 DATA "11",PEN 2,Use pen$ 2 of the HP-7470A plotter.,"","
11070 DATA "12",PEN SPEED,Specify the speed of the plotter pen (default value is,the fastest speed)  (plotter only).,""
11080 DATA "13",SLIDE,Use larger-than-normal labels and symbols: useful for 35mm,transparencies or other reduced reproductions.,""
11090 DATA L,"",Include a leading zero before the decimal point of,each tick-label less than zero (e.g. 0.41 instead,of .41).
11100 DATA "5",PEN 1,Use pen# 1 (plotter only) as the plotting pen.,"","","6",PEN 2,Use pen# 2 (plotter only) as the plotting pen.,"
","
11110 DATA "8",PEN SPEED,Specify the speed of the plotter pen (default value is,the fastest speed)  (plotter only).,""
11120 DATA "12",‡NORM‡,"Use 'normal'-size ticks, labels,& symbols.","","
11130 DATA "13",SLIDE,"Use 'slide-size' ticks, labels, & symbols.  These",are somewhat larger relative to the plot-size than,‡NORM‡
size.
11140 DATA "14",DATAFILE,Bring data from either a VISICALC or numeric data-,file into memory.,""
11150 DATA "7",START,"Answer the resulting queries about axis-labels, intervals,",& plot-size.  The plot framework will then be draw
n.  You
11160 DATA don't need to press any other keys for just a CRT plot.
11170 DATA "10",STACK,Suppress X-axis label and X-axis tick-labels for more,convenient vertical stacking of plots sharing the same,X
-axis.
11180 DATA "9",LDG ZEROES,Include a leading zero in tick-labels for ticks with,absolute values less than zero.,""
11190 !
11200 HELP3: GOSUB HELP  @ RESTORE 11220 @ NUM_KEYS=3 @ GOSUB PRINT_HELP  @ GOTO ADD
11210 !
11220 DATA "5",YORKFIT,Calculate a York-type best-fit line to the last set of,"data, & calculate an isochron age if appropriate.","

11230 DATA "6",RETAIN,"Add any subsequent-input points to the previous ones,",so that points with different plot-symbols can be
11240 DATA pooled for a later Yorkfit or Polyfit.
11250 DATA "7",CONTINUE,Return to the input-data part of the program.,"","
11260 !
```

```
11270 HELP4: GOSUB HELP  @ GOSUB 11280 @ DISP @ DISP @ GOTO CHOOSE_MODEL
11280 PRINT "MODEL 1:   Use the Model 1 solution for drawing the best-fit line and for any" @ PRINT TAB (12);"concordia-intercepts.
 Model 1 assumes that the";
11290 PRINT " only cause of scatter" @ PRINT TAB (12);"is from the assigned analytical errors.  The points are weighted";
11300 PRINT TAB (12);"according to those errors only.  If the probability of fit is less";TAB (12);"than 15%, the assigned errors ar
e expanded until they can";
11310 PRINT " account" @ PRINT TAB (12);"for the scatter, and a Student's-t multiplier of the errors is used." @ PRINT
11320 PRINT "MODEL 2:   Use the Model 2 solution for drawing the best-fit line and for any"
11330 PRINT TAB (12);"concordia-intercepts.  Model 2 assumes that nothing is known about"
11340 PRINT TAB (12);"the reason for the scatter of the points, and so weights all points"
11350 PRINT TAB (12);"equally, with zero error-correlations.  A Student's-t multiplier is" @ PRINT TAB (12);"always used." @ PRINT @
 PRINT
11360 PRINT "MODEL 4:   Use the Model 4 solution for drawing the best-fit line and for"
11370 PRINT TAB (12);"calculating concordia-intercepts.  Model 4-UI assumes that there is a"
11380 PRINT TAB (12);"second disturbance-event, so that the more discordant the points, the"
11390 PRINT TAB (12);"more they scatter from a line.  To compensate for this trend, the"
11400 PRINT TAB (12);"207/206 errors of each point are expanded according to their"
11410 PRINT TAB (12);"discordance, so that the most-concordant points are more heavily" @ PRINT TAB (12);"weighted.  Model 4-LI is s
imilar, except that the ";
11420 PRINT "207/206" @ PRINT TAB (12);"errors are expanded towards the upper intercept, so that the points" @ PRINT TAB (12);
11430 PRINT "lying closest to the lower intercept are more heavily weighted." @ PRINT TAB (12);"Model 4-LI is especially useful if a
n inherited component";
11440 PRINT "is present." @ PRINT @ PRINT
11450 !
11460  PRINT "Press "&HGL$ (" CONT ")&" to return to previous display." @ PAUSE @ RETURN
11470 !
```

APPENDIX 3: PROGRAM LISTING OF PBPLOT

```
10 ! ############################ PBPLOT ########################################
20 !  Program for plotting and Yorkfit of Pb isotope data, with single-stage
30 !   growth-curves.
40 !  Revision of NOV 01, 1983. Requires Adv. Progr. ROM & ABT 100 Kbytes.
50 !
60 !   Data may be entered from either of  (1) Keyboard, (2) Visicalc data-
70 !   file, or (3) nX45 Numeric data-file, and plotted either on CRT or
80 !   plotter.
90 !   Brackets indicate optional input.
100 !
110 ON ERROR GOTO 130
120 IF PLOT_STARTED THEN PLOTDATA ! Skip plot set-up if starting from "CONT 1"
130 OFF ERROR
140  OPTION BASE 1
150  DIM A(100),G(100),Q(100),R(100),Z(100),A$[72],X(100),Y(100),Ex(100),Ey(100),Name$[50],R0(100),Slope(2),Inter(2),Errint(2),Errsl
(2),Xbar(2),Ybar(2),CT$[1]
160 DIM R$[50],RR$[50],PLOTDEV(2),Pr(2),KEY$[1],N$(100)[18],XX(45),CRUD$[50]
170 DIM O(8),O$(8)[18],XERR(100),YERR(100),P$[80],S(100,5),Cint(8),Re$[50]
180 DIM E76(100),Yparam(2,3),YI$[7],ALLKEY$[256],U(100),V(100),UV(100),U2(100),V2(100),Z2(100),T1(100),T2(100),XTIK$(8)[12],YTIK$(8)
[12],XTIK(8),YTIK(8)
190 DIM ZERO$[8],LETTER$(25)[80],CACHE(100,5)
200 SHORT Ex0(100),Ey0(100),SYMBOL(100,5),LETTER(25,5)
210 INTEGER XDEC(8),YDEC(8)
220 GOSUB DATARESTORE  @ GOTO 270
230 DATARESTORE: ! Initialize constants
240 DATA 137.88,.26455,.00098485,.000049475,.000155125,11.152,12.998,31.23,9.74,3700
250 RESTORE 240 @ READ Yparam(1,1),Yparam(2,1),Yparam(1,2),Yparam(2,2),L8,ALPHA0,Yparam(1,3),Yparam(2,3),Mu0,T0
260 RETURN
270 CRUD$=HGL$ (" #### IMPROPER RESPONSE OR NUMBER OF VALUES #### ") @ SLD,HP=0
280 EL$=" "&HGL$ ("ENDLINE")&" " @ SPEED$="VS" @ PLOT_STARTED,PLOTTER_DUMP=0 @ ZERO$="00000000"
290 !
300 ALLKEY$=RPT$ (" ",256)
310 FOR I=1 TO 156
320  IF I<128 OR I>172 THEN ALLKEY$[I,I]=CHR$ (I)
330 NEXT I
340 PLOTDEV(1)=1 @ PLOTDEV(2)=705 @ CPEN=1 @ PEN 1 @ CA,DFILE=0 @ PS=.35 @ PLR,Pr(2),Pr(1),PR=1 @ CT$="0" @ PAGESIZE 24 @ PRINTER IS
 1
350 REDRAW: ODDSIZE,SIZETYPE,Ytype,NPHRASES,NLINES,C_CT,SOLID_LINE=0 @ CLEAR @ PEN 0
360 XDEC,YDEC,YCHAR,XCHAR=0 @ GOTO BEGIN
370 !
371 CLEARKEY: ! Clear all special-function-key labels
372          FOR i=1 TO 14 @ ON KEY# i,"" GOSUB RETRN  @ NEXT i
373          RETURN
374 !
380 RETRIEVE_STR: ! Retrieve up to 8 substrings (separated by ',') from string
390      ON ERROR GOTO 440
400      LINPUT R$ @ O$(1),O$(2),O$(3),O$(4),O$(5),O$(6),O$(7),O$(8)="" @ Ninputs=0 @ RR$=R$
410      Wh=POS (R$,",") @ SP=Wh*(Wh#0)+(LEN (R$)+1)*(Wh=0)
```

```
420     O$(1+Ninputs)=R$[1,SP-1] @ Ninputs=1+Ninputs
430     IF Wh THEN R$=R$[1+Wh] @ GOTO 410
440     OFF ERROR @ RETURN
450 !
460 RETRIEVE: ! Retreive up to 8 numeric values (separated by ',') from string
470     LINPUT R$
480 RTR: Ninputs=0 @ RR$=R$ @ MAT O=ZER
490     ON ERROR GOTO 520
500     O(1+Ninputs)=VAL (R$) @ Ninputs=1+Ninputs
510     Wh=POS (R$,",") @ IF Wh THEN R$=R$[1+Wh] @ GOTO 500
520     OFF ERROR @ RETURN
530 !
540 BAD_INPUT: GOSUB CLUNK  @ DISP @ DISP CRUD$ @ DISP @ RETURN
550 !
560 DEF FNA(T) = ALPHA0+Mu0*(E0-EXP (L8*T)) ! 206/204 from T
570 DEF FNB(T) = rparam(Ytype,3)+Mu0*(E1-EXP (Yparam(Ytype,2)*T))/Yparam(Ytype,1) ! 207/204 or 208/204 from T
580 DEF FNC(T) = rparam(Ytype,2)*EXP ((Yparam(Ytype,2)-L8)*T)/(L8*Yparam(Ytype,1)) ! Growth-curve slope
590 DEF FND(X) = LOG (E0-(X-ALPHA0)/Mu0)/L8 !   T from 206/204
600 DEF FNE(Y) = LOG (E1-(Y-Yparam(Ytype,3))/(Mu0/Yparam(Ytype,1)))/Yparam(Ytype,2) ! T from 207/204 or 208/204
610 DEF FNF(X) = INT (L6T (ABS (X+(X=0))))
620 !
630 DEF FNTICK(X) !                         Returns tick-interval
640   A=10^FNF(X)/8
650   IF ABS (X/A)>12 THEN A=2*A @ GOTO 650
660   B=ABS (A)/10^FNF(A)
670   IF B*INT (B) THEN A=INT (B)*10^FNF(A)
680   FNTICK=A @ FN END
690 !
700 DEF FNXTICK(P) ! Draw X-axis & ticks
710   NTIX=Xspred/XTIK @ ETIX=NTIX/2=INT (NTIX/2) ! Even # of ticks?
720   FOR X=1 TO NTIX
730     ETIK=X/2=INT (X/2) ! Even tick?
740     IF (P=1)*ETIK OR (P=-1)*((ETIK EXOR ETIX)=0) THEN M=XTICKSIZE ELSE M=XTICKSIZE/2
750     IDRAW P*XTIK,0 @ IDRAW 0,P*M @ IDRAW 0,-(P*M)
760   NEXT X
770   FNXTICK=0 @ FN END
780 !
790 DEF FNYTICK(P) ! Draw Y-axis & ticks
800   NTIX=Yspred/YTIK @ ETIX=NTIX/2=INT (NTIX/2) ! Even # of ticks?
810   FOR Y=1 TO NTIX
820     ETIK=Y/2=INT (Y/2) ! Even tick?
830     IF (P=1)*ETIK OR (P=-1)*((ETIK EXOR ETIX)=0) THEN M=YTICKSIZE ELSE M=YTICKSIZE/2
840     IDRAW 0,P*YTIK @ IDRAW -(P*M),0 @ IDRAW P*M,0
850   NEXT Y
860   FNYTICK=0 @ FN END
870 !
880 DEF FNDr(N,D) !                  Round value N to D significant figures
890     IF N THEN Mu=10^(D-INT (L6T (ABS (N)))-1) ELSE Mu=1
900     FNDr=IP (N*Mu)/Mu
910     FN END
920 !
930 DEF FNDECIMAL(X) !               Returns #digits beyond decimal point
940   A=0 @ N=X
950   N=10*(ABS (N)-INT (ABS (N)))
960   IF N THEN A=A+1 @ GOTO 950 ELSE FNDECIMAL=A
970   FN END
980 !
990 DEF FNPb_Pb_AGE(R) ! Calculate 207/206 age
1000  DEFAULT ON @ COUNT,A76=0
1010  S=SGN (1.5-Ytype)*10000*SGN (-.5+(R>Yparam(Ytype,2)/(L8*Yparam(Ytype,1))))
```

```
1020  F=L8*EXP (S*(L8-Yparam(Ytype,2)))/Yparam(Ytype,2)
1030  A=1+(EXP (S*L8)-1-F*(EXP (S*Yparam(Ytype,2))-1))/(1/(Yparam(Ytype,1)*R)-F)
1040  IF A<= 0 THEN 1070 ELSE A76=LOG (A)/Yparam(Ytype,2) @ COUNT=1+COUNT
1050  IF ABS (A76-S)<.01 THEN 1080
1060  IF COUNT<30 THEN S=A76 @ GOTO 1020
1070  DISP @ DISP HGL$ (" CANNOT CALCULATE AN AGE FOR A "&YI$&"-206/204 SLOPE OF "&VAL$ (R)&" ")
1080  FNPb_Pb_AGE=A76 @ DEFAULT OFF
1090  FN END
1100 !
1110 CLUNK: BEEP 115,40 @ BEEP 195,40 @ RETURN ! Error-sound
1120 !
1130 DEF FNCHAR(X) = 1+(ABS (X)>1)*(FNF(X)+(X*INT (X)))+(X<0)+FNDECIMAL(X) ! #Characters required for X.
1140 !
1150 CA=0 @ CT$="0" @ ALPHA 1 @ DISP HGL$ (" HORIZONTAL GROWTH-CURVE TICK-LABELS ") @ RETURN
1160 CA=1 @ CT$="-" @ ALPHA 1 @ DISP HGL$ (" OBLIQUE GROWTH-CURVE TICK-LABELS ") @ RETURN
1170 !
1180 DFILE_BEGIN: GOSUB DATAFILE  @ GOTO BEGIN
1190 !
1200 !
1210 BEGIN: GOSUB DATARESTORE  @ DEG @ NOCURVE,STACK,NCHORDS=0 @ GOSUB CLEARKEY
1220        ALPHA 1 @ CLEAR @ ALPHA 21,28 @ AWRIT "Lower Options are Default"
1230        ALPHA 15,28 @ AWRIT "press "&HGL$ (" N ")&" for no growth-curve"
1240        ALPHA 17,28 @ AWRIT "press "&HGL$ (" T ")&" for no growth-curve ticks"
1250        ALPHA 19,28 @ AWRIT "press "&HGL$ (" H ")&" for help"
1260                  ON KYBD KY,ALLKEY$ GOTO CANCEL
1270                  ON KEY# 1,"0  GTICKS" GOSUB 1150
1280                  ON KEY# 2,"CRT-PRINT" GOSUB DISPLAY
1290                  ON KEY# 3," CRT PLOT" GOSUB CRTPLOT
1300    IF PLR#1 THEN ON KEY# 4,"  PEN 1" GOSUB PEN_1
1320  IF Ytype=0 THEN ON KEY# 5," STACK" GOSUB STACK
1330                  ON KEY# 6,"  *NORM*" GOSUB NORMSIZE
1340    IF Ytype THEN ON KEY# 7,"  START" GOTO 60
1350  IF Ytype=0 THEN ON KEY# 7," 207/204" GOTO 1490
1360                  ON KEY# 8,"\  GTICKS" GOSUB 1160
1370                  ON KEY# 9," PRINTER" GOSUB HARDCOPY
1380                  ON KEY# 10," HARDPLOT" GOSUB HARDPLOT
1390    IF PLR#1 THEN ON KEY# 11,"  PEN 2" GOSUB PEN_2
1410    IF Ytype THEN ON KEY# 5," DATAFILE" GOSUB DFILE_BEGIN
1420                  ON KEY# 13,"  SLIDE" GOSUB SLIDE
1440 IF Ytype=0 THEN ON KEY# 14," 208/204" GOTO 1500
1450   IF PLR#1 THEN ON KEY# 12," PENSPEED" GOSUB PEN_SPEED
1470        KEY LABEL
1480        GOTO 1480
1490  YI$="207/204" @ Ytype=1 @ ALPHA 1 @ DISP HGL$ ("  207/204 - 206/204 PLOT ") @ OFF CURSOR @ GOTO 1270
1500  YI$="208/204" @ Ytype=2 @ ALPHA 1 @ DISP HGL$ ("  208/204 - 206/204 PLOT ") @ OFF CURSOR @ GOTO 1270
1510 !
1520 CANCEL: OFF KYBD
1530         IF KY=84 OR KY=116 THEN CA=-1 @ CT$="" @ ALPHA 1 @ DISP HGL$ (" SUPPRESS GROWTH-CURVE TICKS ") @ GOTO 1260
1540         IF KY=78 OR KY=110 THEN NOCURVE=1 @ ALPHA 1 @ DISP HGL$ (" SUPPRESS GROWTH-CURVE  ") @ GOTO 1260
1550         IF KY=72 OR KY=104 THEN GOTO HELP1
1560         GOSUB CLUNK  @ GOTO 1260
1570 !
1580 GO: ALPHA 1,1 @ CLEAR @ OFF CURSOR @ ALPHA 10,13 @ OFF KEY# @ OFF KYBD
1590   AWRIT "PRESS "&HGL$ (" ENDLINE ")&" FOR STACEY-KRAMERS GROWTH-CURVE, OR"
1600   ALPHA 12,22 @ AWRIT HGL$ (" * ")&"    FOR ARBITRARY GROWTH-CURVE PARAMETERS."
1610   ON KYBD KY,CHR$ (154)&"*" GOTO 1630
1620   GOTO 1620
1630   CLEAR @ OFF KYBD @ A=(KY=154)+2*(KY=42)
1640 !
1650   IF A=2 THEN DISP @ DISP "ENTER INITIAL 206/204, INITIAL "&YI$&", STARTING-AGE (Ma), AND SOURCE 238/204" @ DISP @ INPUT ALPHA0
```

```
,Yparam(Ytype,3),T0,Mu0
1660   IF A=2 AND Ytype=2 THEN DISP @ DISP "SOURCE 238/232";@ INPUT Yparam(1,2)
1670   E0=EXP (L8*T0) @ E1=EXP (Yparam(Ytype,2)*T0)
1680  DISP @ DISP "206/204  and "&YI$&" PLOT-LIMITS (min., max., min., max.)" @ DISP @ GOSUB RETRIEVE  @ IF Ninputs#4 THEN GOSUB BAD
_INPUT  @ GOTO 1680
1690  XMIN=O(1) @ XMAX=O(2) @ YMIN=O(3) @ YMAX=O(4)
1700  IF XMIN>= XMAX OR YMIN>= YMAX THEN GOSUB BAD_INPUT  @ GOTO 1680
1710  XTIK=FNTICK(XMAX-XMIN) @ YTIK=FNTICK(YMAX-YMIN)
1720 !
1730 ! Force lower-bound X & Y values to have a minimum # of sig. figures
1740 FOR I=1 TO 8
1750  X=FNDr(XMIN,I) @ IF XMIN-X<XTIK THEN XMIN=X @ GOTO 1770
1760 NEXT I
1770 FOR I=1 TO 8
1780  Y=FNDr(YMIN,I) @ IF YMIN-Y<YTIK THEN YMIN=Y @ GOTO 1800
1790 NEXT I
1800  AMN=0
1810  AMX=FND(XMIN*(XMIN>= ALPHA0)+ALPHA0*(XMIN<ALPHA0))
1820 IF PLR=1 THEN 1960 ELSE CLEAR @ DISP "PLOT-SIZE (3-10)" @ DISP @ DISP "[press "&EL$&" for nonstandard dimensions/location]";@ D
ISP @ GOSUB RETRIEVE
1830 IF Ninputs*(O(1)<3 OR O(1)>10) THEN GOSUB CLUNK  @ GOTO 1820
1840 IF Ninputs=0 THEN ODDSIZE=1 ELSE PLOTSIZE=O(1) @ GOTO 1960
1850 !
1860 CLEAR @ DISP @ DISP "PRESS "&HGL$ (" 1 ")&" TO DEFINE LIMITS FOR THE ENTIRE FIGURE-AREA,"
1870 DISP @ DISP "      "&HGL$ (" 2 ")&" TO DEFINE LIMITS FOR JUST THE PLOT-BOX ITSELF." @ DISP
1880 ON KYBD KY,"0123456789" GOTO 1900
1890 GOTO 1890
1900 OFF KYBD @ IF KY#49 AND KY#50 THEN GOSUB CLUNK  @ GOTO 1880 ELSE SIZETYPE=KY-48
1910 PLOTLIMITS: DISP @ DISP "ENTER PLOT-LIMITS IN mm: X-MIN, X-MAX, Y-MIN, Y-MAX" @ DISP @ DISP "(max. limits are 0-257 [X]  and 0-
191 [Y])";
1920 INPUT LIMX1,LIMX2,LIMY1,LIMY2
1930 IF LIMX1<0 OR LIMX2>257 OR LIMY1<0 OR LIMY2>191 OR LIMX2-LIMX1<30 OR LIMY2-LIMY1<30 THEN GOSUB BAD_INPUT  @ GOTO 1920
1940 PLOTSIZE=10*((LIMX2-LIMX1)/257+(LIMY2-LIMY1)/191)/2
1950 !
1960  ALPHA 1,1 @ CLEAR @ OFF CURSOR @ ALPHA 12,27 @ AWRIT "A FEW SECONDS, PLEASE..."
1970 !
1980 ! Calculate plot-boundaries -------------------------
1990 !
2000 CTIK=FNTICK((AMX-AMN)*2) @ IF CA=0 AND (AMX-AMN)/CTIK<6 THEN CTIK=CTIK/2
2010 XTIK=FNTICK(XMAX-XMIN) @ YTIK=FNTICK(YMAX-YMIN)
2020 !
2030 ! Force lower-bound X & Y-values to have a minimum# of significant figs.
2040 FOR I=1 TO 7
2050  X=FNDr(XMIN,I) @ IF XMIN-X<XTIK THEN 2070
2060 NEXT I
2070 FOR I=1 TO 7
2080  Y=FNDr(YMIN,I) @ IF YMIN-Y<YTIK THEN 2100
2090 NEXT I
2100 FOR X=XMIN TO XMAX STEP XTIK ! Force X-MAX to lie on a tick
2110  IF XMAX-X<= XTIK THEN XMAX=X+XTIK @ GOTO 2130
2120 NEXT X
2130 FOR Y=YMIN TO YMAX STEP YTIK ! Force Y-MAX to lie on a tick
2140  IF YMAX-Y<= YTIK THEN YMAX=Y+YTIK @ GOTO 2160
2150 NEXT Y
2160 XDEC_MAX,YDEC_MAX,XCHAR,YCHAR=0
2170  Xspred=XMAX-XMIN @ Yspred=YMAX-YMIN @ HP=PLR#1
2180 !
2190 ! Find max# characters in X & Y ticks w. max# decimal places
2200 FOR I=1 TO Xspred/XTIK/2+1
2210  XTIK(I)=XMIN+2*XTIK*(I-1)
```

```
2220  A=FNCHAR(XTIK(I)) @ IF A>XCHAR THEN XCHAR=A
2230  XDEC(I)=FNDECIMAL(XTIK(I)) @ IF XDEC(I)>XDEC_MAX THEN XDEC_MAX=XDEC(I)
2240 NEXT I
2250 FOR I=1 TO Yspred/YTIK/2+1
2260  YTIK(I)=YMIN+2*YTIK*(I-1)
2270  A=FNCHAR(YTIK(I)) @ IF A>YCHAR THEN YCHAR=A
2280  YDEC(I)=FNDECIMAL(YTIK(I)) @ IF YDEC(I)>YDEC_MAX THEN YDEC_MAX=YDEC(I)
2290 NEXT I
2300 !
2310 ON ERROR GOTO PLOTTER_NOT_ON  @ PLOTTER IS PLR @ OFF ERROR @ GOTO 2350
2320 !
2330 PLOTTER_NOT_ON: GOSUB CLUNK  @ CLEAR @ DISP TAB (24);HGL$ (" PLOTTER NOT READY ") @ WAIT 3000 @ CLEAR @ GOTO 2310
2340 !
2350 IF PLOTTER_DUMP THEN 2390
2360 HP=PLR$1 @ CRT=PLR=1 @ PP=1+CRT @ IF PLR=1 THEN GCLEAR
2370 IF CRT THEN LIMX1,LIMY1=0 @ LIMX2=125 @ LIMY2=75 @ GOTO 2420
2380 IF ODDSIZE THEN 2410
2390 LIMX1=257*(1-.1*PLOTSIZE)/2 @ LIMX2=257-LIMX1 ! For centered plots
2400 LIMY1=191*(1-.1*PLOTSIZE)/2 @ LIMY2=191-LIMY1 ! For centered plots
2410 IF SIZETYPE<2 THEN LIMIT LIMX1,LIMX2,LIMY1,LIMY2
2420 AR=2/3 @ THT=5 @ LHT=6 !       Aspect ratio, tick-label character height
2430 IF HP*(STACK=0 OR PLOT_STARTED=0) THEN M=.7*(1+2.7*(PLOTSIZE<4 OR SLD)/PLOTSIZE) @ THT=M*THT @ LHT=1.2*M*LHT
2440 IF HP THEN PRINTER IS PLOTDEV(2) @ PRINT SPEED$ @ PRINTER IS Pr(2)
2450 !
2460 A=(XMAX-XMIN)/XTIK @ B=(YMAX-YMIN)/YTIK
2470 YL=1+THT*.75+LHT !                          GU's below X-axis
2480 ! XL=1+THT*AR*(YCHAR-HP)*(1-HP/3)-.25+LHT*1.5 ! GU's to left of Y-axis
2490 ! XL=THT*AR*(YCHAR6)*YCHAR+LHT*2/3
2500 XL=THT*AR*(YCHAR+.5)*(1-HP/3)+LHT*1.2
2510 YEND=THT/2*(B/2=INT (B/2)) !                 GU's above top of Y-axis
2520 XEND=(1+XCHAR*AR*THT/2)*(A/2=INT (A/2)) !    GU's to right of X-axis
2530 !
2540 IF SIZETYPE=2 THEN C=0 @ GOSUB PLOTBOX_SIZE
2550 IF RATIO >1 THEN Xgu=100*RATIO  @ Ygu=100 ELSE Xgu=100 @ Ygu=100/RATIO
2560 ! Xgu & Ygu are GU's of total plot-area along X & Y directions
2570 !
2580 Xu=Xgu-XL-XEND !                             GU's plot-area width
2590 Yu=Ygu-YL-YEND !                             GU's  "    "   height
2600 !
2610 LOCATE XL,Xgu-XEND,YL,Ygu-YEND
2620 XLOC=(Xgu-XEND-XL)/Xspred !                 GU's per UU in X-direction
2630 YLOC=(Ygu-YEND-YL)/Yspred !                 "    "  "  "  Y    "
2640 Xmm=XLOC*(LIMX2-LIMX1)/Xgu !                mm per UU in X-direction
2650 Ymm=YLOC*(LIMY2-LIMY1)/Ygu !                "  "  "  "  Y    "
2660 LOCATE XL,Xgu-XEND,YL,Ygu-YEND
2670 SCALE XMIN,XMAX,YMIN,YMAX
2680 CSIZE THT @ LW=AR*LHT*(1-HP/3)
2690 !
2700 FOR I=1 TO Xspred/XTIK/2+1
2710  XTIK$(I)=VAL$ (XTIK(I))
2720  IF XDEC_MAX=0 OR XDEC_MAX=XDEC(I) THEN 2750
2730  IF XDEC(I)=0 THEN XTIK$(I)=XTIK$(I)&"."
2740  XTIK$(I)=XTIK$(I)&ZERO$[1,XDEC_MAX-XDEC(I)]
2750 NEXT I
2760 !
2770 FOR I=1 TO Yspred/YTIK/2+1
2780  YTIK$(I)=VAL$ (YTIK(I))
2790  IF YDEC_MAX=0 OR YDEC_MAX=YDEC(I) THEN 2820
2800  IF YDEC(I)=0 THEN YTIK$(I)=YTIK$(I)&"."
2810  YTIK$(I)=YTIK$(I)&ZERO$[1,YDEC_MAX-YDEC(I)]
```

```
2820 NEXT I
2830 !
2840 XTICKSIZE=2.5/YLOC @ YTICKSIZE=XTICKSIZE*Yaa/Xaa
2850 PEN CPEN @ SETUU @ MOVE XMIN,YMIN
2860 A=FNXTICK(1)+FNYTICK(1)+FNXTICK(-1)+FNYTICK(-1) @ FRAME
2870 IF STACK THEN 2920 ELSE LORG 6 @ A=THT/YLOC/3
2880 FOR I=1 TO Xspred/XTIK/2+1
2890  MOVE XTIK(I),YMIN-A @ LABEL XTIK$(I)
2900 NEXT I
2910 !
2920 LORG 8 @ B=THT*AR/2/XLOC
2930 FOR I=1 TO Yspred/YTIK/2+1
2940  IF STACK AND I=1 THEN 2960
2950  IF I>1 THEN MOVE XMIN-B,YTIK(I) @ LABEL YTIK$(I) ELSE MOVE XMIN-B,YTIK(I)+THT/10/YLOC @ LABEL YTIK$(I)
2960 NEXT I
2970 !
2980 SETGU @ CSIZE LHT @ IF STACK THEN 3020
2990 LORG 1 @ MOVE (Xgu+XL-XEND)/2-LW*4.5,LHT/4 @ LABEL "206"
3000 IMOVE 3*LW,LHT*3/4 @ LABEL "Pb/" @ IMOVE 3*LW,LHT*5/4 @ LABEL "204"
3010 IMOVE 3*LW,LHT*3/4 @ LABEL "Pb"
3020 LDIR 90 @ LORG 6 @ MOVE 0,(Ygu+YL)/2-LW*4.5 @ LABEL YI$[1,3]
3030 IMOVE -(LHT*3/4),3*LW @ LABEL "Pb/" @ IMOVE -(LHT*5/4),3*LW @ LABEL "204"
3040 IMOVE -(LHT*3/4),2.5*LW @ LABEL "Pb" @ SETUU @ LDIR 0
3050 !
3060 PLOT_GROWTH_CURVE: IF NOCURVE THEN 3350
3070     K=8*CTIK @ C=0 @ CHT=THT @ D=0
3080                 IF CA>0 THEN CTHT=7*THT ELSE CTHT=THT/2.5
3090     C=10^FNF(T0)
3100     FOR AMX=C*(1+INT (T0/C)) TO AMN STEP -CTIK
3110      IF FNA(AMX)>XMIN AND FNB(AMX)>YMIN THEN 3130
3120     NEXT AMX
3130     T1=AMX+2*CTIK
3140      CLIP XMIN,XMAX,YMIN,YMAX
3150      FOR T=T1 TO AMN STEP -(CTIK/8)
3160       X=FNA(T) @ Y=FNB(T) @ IF (X>XMIN)*(Y>YMIN) AND T<= T0 THEN DRAW X,Y ELSE MOVE X,Y
3170        IF T/CTIK*INT (T/CTIK) OR CA=-1 THEN 3330
3180 !
3190 ! ------ Growth-curve ticks & tick-labels -----
3200 !
3210        IF X>XMAX OR Y>YMAX OR X<XMIN OR Y<YMIN OR T>T0 THEN 3330
3220        IF CA=0 THEN 3250
3230         RISE=(Y-Yparam(Ytype,3))/(X-ALPHA0)*YLOC @ RUN_=XLOC ! Defines slope of isochrons
3240         SETGU @ LDIR RUN_,RISE ! Isochron slope
3250       SETUU @ LORG 5 @ CSIZE CTHT,1,0 @ LABEL CT$ @ IF D=0 THEN 3310
3260 IF CA THEN MOVE X-Xspred/50,Y+Yspred/200 @ LORG 1 @ CSIZE CHT,AR,0 @ LABEL VAL$ (T)&" " @ GOTO 3310
3270       L=CHT*(1.5+INT (LGT (T+(T=0))))
3280       Xt=XL+XLOC*(X-XMIN)-L*AR-2 ! location of ticklabel-start, in GU's
3290       Yt=YL+YLOC*(Y-YMIN)+CHT+2 ! location of ticklabel-start, in GU's
3300 IF Xt>XL AND Yt<Ygu-YEND THEN MOVE X+Xspred/60,Y+Yspred/60 @ LORG 7 @ CSIZE CHT,AR,0 @ LABEL VAL$ (T)&" "
3310        MOVE X,Y
3320        D=D=0
3330       NEXT T
3340 !
3350 MOVE XMIN,YMAX @ PEN 0
3360 IF PLOTTER_DUMP THEN RETURN ELSE PLOT_STARTED=1 @ GOTO STARTPLOT
3370 DFILE_STARTED: GOSUB DATAFILE  @ GOTO RE_DATA
3380 RETRN: RETURN !    Dummy return for key-labels
3390 HARDCOPY: ALPHA 1 @ DISP HGL$ (" PRINTER PRINTOUT ") @ Pr(2)=701 @ PR=2 @ PRINTER IS Pr(2) @ RETURN
3400 NORMSIZE: ALPHA 1 @ DISP HGL$ (" NORMAL-SIZE PLOT-SYMBOLS & LINES ") @ SLD=0 @ RETURN
3410 PEN_SPEED: ALPHA 1 @ CLEAR @ AWRIT HGL$ (" INPUT PEN-SPEED (1-10) ") @ ALPHA 3 @ INPUT PENSPEED
```

```
3420      IF PENSPEED<1 OR PENSPEED>10 THEN GOSUB CLUNK  @ GOTO PEN_SPEED
3430      IF PENSPEED=10 THEN SPEED$="VS"
3440      IF PENSPEED<10 THEN SPEED$="VS"&VAL$ (INT (38.1*PENSPEED/10))
3450      GOTO BEGIN
3460 !
3470 STACK: ALPHA 1 @ DISP HGL$ (" SUPPRESS X-AXIS LABEL & TICK-LABELS ") @ STACK=1 @ RETURN
3480 SLIDE: ALPHA 1 @ DISP HGL$ (" SLIDE-SIZE SYMBOLS & LINES ") @ SLD=1 @ RETURN
3490 DISPLAY: ALPHA 1 @ DISP HGL$ (" CRT PRINTOUT ONLY ") @ Pr(2)=1 @ PR=1 @ PRINTER IS 1 @ RETURN
3500 !
3510 HARDPLOT: ALPHA 1 @ DISP HGL$ (" PLOTTER GRAPHICS ") @ PLR=PLOTDEV(2) @ WAIT 500 @ GOTO 1270
3520 CRTPLOT: ALPHA 1 @ DISP HGL$ (" CRT GRAPHICS ") @ PLR=1 @ WAIT 500 @ GOTO BEGIN
3530 !
3540 STARTPLOT: NN=0 @ LDIR 0 @ PDIR 0 @ PRINTER IS Pr(2) @ DEG
3550 PLOTDATA: OFF ERROR @ N,Iso,Pbpb=0 @ II,K=1
3560 !
3570 RE_DATA: ALPHA 1 @ CLEAR @ ALPHA 21,23 @ AWRIT "(PRESS ANY KEY FOR KEYBOARD-SYMBOL)" @ OFF CURSOR @ PEN 0 @ SETUU @ MOVE XMIN,Y
MAX @ Kb=0 @ GOSUB CLEARKEY
3580 REDIM X(100),Y(100)
3600              ON KEY# 2,"OPEN Ell" GOTO OEll
3610              ON KEY# 4," POLYGON" GOTO Poly
3630   IF CRT THEN ON KEY# 11,"PRINT PLOT" GOTO PRINT_PLOT
3650  IF NN>1 THEN ON KEY# 7,"  YORKFIT" GOTO Yorkfit
3670   IF CRT THEN ON KEY# 12,"ERASE PEN" GOSUB ERASE_PEN
3690              ON KEY# 5," HELP" GOTO HELP2
3700              ON KEY# 9,"SOLIDEll" GOTO SEll
3710              ON KEY# 1," LETTER" GOTO LETTER
3720              ON KEY# 10,"  PEN 1" GOSUB PEN_1
3730    IF HP THEN ON KEY# 3,"  PEN 2" GOSUB PEN_2
3740   IF CRT THEN ON KEY# 3,"TO PLOTTER" GOTO DUMP_TO_PLOTTER
3750              ON KEY# 14," DATAFILE" GOTO DFILE_STARTED
3760              ON KEY# 6,"  REDRAW" GOTO REDRAW
3770              ON KYBD KY,ALLKEY$ GOTO KYBRD
3780 PLABEL: KEY LABEL
3790          GOTO 3790
3800 !
3810 PEN_1: CPEN=1 @ ALPHA 1 @ DISP HGL$ (" PEN 1 ") @ RETURN
3820 PEN_2: CPEN=2 @ ALPHA 1 @ DISP HGL$ (" PEN 2 ") @ RETURN
3830 ERASE_PEN: CPEN=-1 @ ALPHA 1 @ DISP HGL$ (" ERASE-PEN ") @ RETURN
3840 OEll: SYMBOL=2 @ ALPHA 1 @ DISP HGL$ (" OPEN ERROR-ELLIPSE SYMBOL ") @ GOTO INPUT_DATA
3850 SEll: SYMBOL=-2 @ ALPHA 1 @ DISP HGL$ (" SOLID ERROR-ELLIPSE SYMBOL ") @ GOTO INPUT_DATA
3860 !
3870 Poly: SYMBOL=3 @ ALPHA 1 @ CLEAR @ DISP HGL$ (" POLYGON SYMBOL ")
3880   DISP @ DISP "# SIDES (- for star) [,OPEN/SOLID (0/1) [,SIZE (1-5) [,ROTATION (0-360)]]]";@ GOSUB RETRIEVE
3890   IF Ninputs<1 THEN GOSUB BAD_INPUT  @ GOTO 3880
3900   POLYSIDES=D(1)
3910   IF Ninputs>1 AND D(2)=1 THEN SOLID=1 ELSE SOLID=0
3920   IF Ninputs>2 THEN POLYSIZE=D(3) ELSE POLYSIZE=2.5
3930   IF Ninputs>3 THEN POLYROT=D(4) ELSE POLYROT=90+(POLYSIDES/2=INT (POLYSIDES/2))*(180/POLYSIDES-90)
3940   GOTO INPUT_DATA
3950 !
3960 KYBRD: IF KY=19 THEN SOLID_LINE=1 @ BEEP @ ALPHA 1 @ DISP HGL$ (" USE SOLID LINE FOR YORKFIT LINES ") @ GOTO PLABEL  ! CTRL S k
ey
3970       CLEAR @ SYMBOL=5 @ KEY$=UPC$ (CHR$ (KY)) @ OFF KYBD @ ALPHA 1 @ DISP HGL$ (" PLOTTING-SYMBOL IS ")&HGL$ (KEY$&" ") @ GOT
O INPUT_DATA
3980 !
3990 !
4000  INPUT_DATA: OFF KEY# @ OFF KYBD @ IF DFILE THEN DISP HGL$ (" INPUT DATAFILE SET #s AS 1st SET, LAST SET-ASTERISK ");" (e.g. 2,
12*)" @ DISP
4010  GOSUB INPUT_POINTS
4020  IF II>100 THEN Edit ELSE GOTO 4010
```

```
4030 !
4040  INPUT_POINTS: X(II),Y(II),Ex(II),Ey(II),R(II)=0
4050  DISP @ DISP "#"&VAL$ (II)&":  ";@ DISP "206/204, [%err,]  "&YI$&", [%err] [,err-corr.]  ("&EL$&"WHEN DONE)";@ LINPUT Re$
4060  ! If datafile input, use commas to separate data sets in a continuous series, semicolons to separate individual sets or differ
ent series.
4070 ! 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.
4080 Sc=POS (Re$,";") @ IF Sc THEN R$=Re$[1,Sc-1] ELSE R$=Re$ ! semicolon?
4090 GOSUB RTR  @ IF Ninputs=0 THEN Edit
4100 IF NOT POS (Re$,"#") THEN DF=0 @ GOSUB KEYBOARD_INPUT  @ RETURN
4110 IF DFILE THEN DF=1 ELSE GOSUB CLUNK  @ DISP @ DISP "### NO DATAFILE IN MEMORY ###" @ GOTO 4010
4120 IF DFILE THEN DF=1 ELSE GOSUB CLUNK  @ DISP @ DISP "### NO DATAFILE IN MEMORY ###" @ GOTO 4010
4130 IF Ninputs=1 THEN S1,S2=O(1) ELSE S1=O(1) @ S2=O(2) ! 1st, last datafile set#s in a series
4140 DISP "SET#";TAB (29);"206/204";TAB (38);"%Err";TAB (50);YI$;TAB (60);"%Err";TAB (69);"RHO" @ DISP
4150 FOR ID=S1 TO S2 ! transfer datafile sets to plotting array & display
4160  X(II)=S(ID,1) @ Ex(II)=S(ID,2) @ Y(II)=S(ID,3) @ Ey(II)=S(ID,4) @ R(II)=S(ID,5) @ IF X(II)#Y(II)=0 THEN 4190
4170  DISP II;TAB (5);N$(ID);TAB (28);X(II);TAB (37);Ex(II);TAB (49);Y(II);TAB (59);Ey(II);TAB (67);R(II) @ GOSUB 4270
4180  IF II=100 THEN Edit
4190 NEXT ID
4200 DISP @ IF Sc THEN Re$=Re$[1+Sc] @ GOTO 4080 ELSE RETURN ! Respond to ";"
4210 !
4220 KEYBOARD_INPUT: Kb=1 ! Transfer typed-in data to plotting array
4230   IF Ninputs<2 OR Ninputs=3 OR Ninputs>5 THEN GOSUB BAD_INPUT  @ GOTO 4050
4240   IF Ninputs=2 THEN X(II)=O(1) @ Y(II)=O(2) @ GOTO 4270
4250   IF Ninputs>3 THEN X(II)=O(1) @ Ex(II)=O(2) @ Y(II)=O(3) @ Ey(II)=O(4)
4260   IF Ninputs=5 THEN R(II)=O(5)
4270   IF X(II)#Y(II) THEN 4290
4280   IF DF THEN RETURN ELSE GOSUB CLUNK  @ DISP @ DISP HGL$ (" #### SET# "&VAL$ (II)&"  -NONZERO INPUT REQUIRED #### ") @ DISP @ D
ISP @ RETURN
4290    IF Ex(II)^2+Ey(II)^2<2#R(II)#Ex(II)#Ey(II) THEN GOSUB CLUNK  @ DISP @ DISP HGL$ ("#### ERR.-CORR. OF "&VAL$ (R(II))&" IS TOO
 HIGH #### ") @ RETURN
4300  II=II+1 @ N=N+1 @ RETURN
4310 !
4320 Edit: IF Kb=0 THEN 4350 ELSE CLEAR @ DISP "SET#";TAB (8);"206/204";TAB (20);"%Err";TAB (40);YI$;TAB (52);"%Err";TAB (62);"Rho"
@ DISP
4330       FOR I=K TO N @ DISP I;TAB (8);X(I);TAB (20);Ex(I);TAB (40);Y(I);TAB (52);Ey(I);TAB (60);R(I) @ NEXT I @ DISP
4340       IF Kb=0 AND DF THEN PLOTT
4350       IF ABS (SYMBOL)=3 OR SYMBOL=5 THEN 4420 ELSE Ninputs,A=0
4360       FOR I=K TO N @ A=A+(Ex(I)#Ey(I)=0) @ NEXT I
4370       IF A=0 THEN 4420
4380       DISP "% 6/4-Err, % "&YI$[3,3]&"/4-ERR, Err.-Corrs  FOR SETS WITH ZERO ERRORS";@ GOSUB RETRIEVE  @ IF Ninputs<2 THEN GOSUB
 BAD_INPUT  @ GOTO 4380
4390       FOR I=K TO N @ IF Ex(I) AND Ey(I) THEN 4410
4400       Ex(I)=O(1) @ Ey(I)=O(2) @ IF Ninputs=3 THEN R(I)=O(3)
4410       NEXT I @ GOTO Edit
4420       IF Kb=0 THEN PLOTT ! Don't ask for corrections if no keyboard-data
4430       DISP @ DISP "SET TO BE CORRECTED?  ("&HGL$ ("ENDLINE")&" IF OK) ";
4440  GOSUB RETRIEVE  @ I=O(1) @ IF I=0 THEN PLOTT ELSE II=I @ GOSUB 4040 @ N=N-1 @ GOTO Edit
4450 !
4460 ERROR_ELLIPSE: !    Draw error-ellipse about data-point
4470   IF XERR(I)#YERR(I) THEN XE=XERR(I) ELSE XE=XERR(I)#1.0001
4480   ANGLE=.5#ATN (2#R(I)#XE#YERR(I)/(XE^2-YERR(I)^2))
4490 ! ANGLE is the angle that an axis of the ellipse makes with the X-axis.
4500   C1=2#(1-R(I)^2)#1.224^2 @ C2=1/COS (2#ANGLE)
4510   Vx=XE^2 @ Vy=YERR(I)^2
4520   A=SQR (C1/((1+C2)/Vx+(1-C2)/Vy)) ! Length of major axis of ellipse
4530   B=SQR (C1/((1-C2)/Vx+(1+C2)/Vy)) ! Length of minor axis of ellipse
4540   St=A/INT (A/(Xspred/800)+1) !  Step-size, in x-direction
4550   St=A/INT (A/St+.0001)
4560   SIN_ANGLE=SIN (ANGLE) @ COS_ANGLE=COS (ANGLE)
4570    FOR K=1 TO -1 STEP -2
```

```
4580     FOR X=K*A TO -(K*(A+St)) STEP -(K*St)
4590      IF (K)0)*(X(-(A*1.001)) OR (K(0)*(X)A*1.001) THEN 4650
4600      Z=1-(X/A)^2 @ IF Z)= 0 THEN Y=K*B*SQR (Z) ELSE GOTO 4630
4610      RPLOT X*COS_ANGLE-Y*SIN_ANGLE,X*SIN_ANGLE+Y*COS_ANGLE
4620      IF SYMBOL(0 THEN K=-K @ RPLOT X*COS_ANGLE+Y*SIN_ANGLE,X*SIN_ANGLE-Y*COS_ANGLE ! Fill in ellipse if solid
4630     NEXT X
4640     IF SYMBOL(0 THEN 4660
4650    NEXT K
4660    RETURN
4670 !
4680 POLYGON: SETGU ! Draw open or filled polygon or star-symbol
4690       IF PLOTTER_DUMP THEN POLYSIDES=SYMBOL(JJ,2) @ SOLID=SYMBOL(JJ,3) @ POLYSIZE=SYMBOL(JJ,4) @ POLYROT=SYMBOL(JJ,5)
4700       FOR A=0 TO (HP AND SLD=0 AND POLYSIDES)0 AND POLYSIZE)3)+50*SOLID
4710        B=PS*(POLYSIZE-A/2)*PP @ S=360/ABS (POLYSIDES) @ E=.5-(ABS (POLYSIDES)=3)/5
4720           IF B(0 THEN SETUU @ RETURN
4730           FOR J=POLYROT TO POLYROT+360 STEP S
4740            RPLOT B*COS (J),B*SIN (J)
4750            IF POLYSIDES(0 THEN C=J+S/2 @ RPLOT E*B*COS (C),E*B*SIN (C)
4760           NEXT J
4770          NEXT A
4780          SETUU @ RETURN
4790 !
4800 KEYBOARD_SYMBOL: IF PLOTTER_DUMP THEN KEY$=CHR$ (SYMBOL(JJ,2))
4810                  IF KEY$="+" THEN B=1.5 @ C=AR*CRT+HP/2 ELSE B=0 @ C=1
4820                  IF KEY$="*" THEN C=CRT+.6*HP
4830                  LORG 5 @ CSIZE 3.5-HP+1.5*SLD+B,C,0 @ LABEL KEY$ @ RETURN
4840 !
4850 PLOTT: PEN CPEN @ C_ROW=CURSROW  !                Draw data-point symbols
4860        FOR I=K TO N
4870         IF HP THEN ALPHA C_ROW+1 @ DISP I;TAB (10);X(I);TAB (20);Y(I) @ OFF CURSOR @ GOTO 4920 ELSE C_CT=1+C_CT
4880         CACHE(C_CT,1)=X(I) @ CACHE(C_CT,2)=Ex(I) @ CACHE(C_CT,3)=Y(I) @ CACHE(C_CT,4)=Ey(I) @ CACHE(C_CT,5)=R(I) @ SYMBOL(C_CT,
1)=SYMBOL
4890         IF SYMBOL=3 THEN SYMBOL(C_CT,2)=POLYSIDES @ SYMBOL(C_CT,3)=SOLID @ SYMBOL(C_CT,4)=POLYSIZE @ SYMBOL(C_CT,5)=POLYROT
4900         IF SYMBOL*3 AND SYMBOL*5 THEN SYMBOL(C_CT,1)=SYMBOL
4910         IF SYMBOL=5 THEN SYMBOL(C_CT,2)=KY
4920         XERR(I)=X(I)*Ex(I)/100 @ YERR(I)=Y(I)*Ey(I)/100
4930         IF X(I)(XMIN OR Y(I)(YMIN OR X(I)>XMAX OR Y(I)>YMAX THEN 4960
4940         MOVE X(I),Y(I)
4950         ON ABS (SYMBOL) GOSUB RETRN ,ERROR_ELLIPSE ,POLYGON ,RETRN ,KEYBOARD_SYMBOL
4960        NEXT I
4970        IF PLOTTER_DUMP=0 THEN MOVE XMIN,YMAX
4980        IF HP=1 AND PLOTTER_DUMP=0 THEN PEN 0
4990        IF PLOTTER_DUMP THEN RETURN
5000 !
5010 ADD: NN=N @ DISP @ DISP "PRESS "&HGL$ (" RETAIN ")&" TO INCLUDE THESE POINTS WITH OTHERS FOR A YORKFIT." @ DISP @ GOSUB CLEARKE
Y
5020      DISP "PRESS "&HGL$ (" CONTINUE")&" TO CONTINUE." @ DISP @ DISP
5060       ON KEY# 4," HELP" GOTO HELP3
5070 IF NN>1 THEN ON KEY# 5," YORKFIT" GOTO Yorkfit
5090       ON KEY# 6," RETAIN" GOTO RETAIN
5100       ON KEY# 7," CONTINUE" GOTO PLOTDATA
5180       OFF CURSOR @ KEY LABEL
5190       GOTO 5190
5200 RETAIN: II,K=N+1 @ GOTO RE_DATA  ! Add next points to existing sequence
5210 !
5220 !
5230 ! ##########################################################################
5240 Yorkfit: ! LINE-FITTING ALGORITHM MODIFIED FROM D. YORK,  EPSL V. 5, P. 320
5250 ! ##########################################################################
5260 !
```

```
5270  CLEAR @ DISP "YORKFIT:" @ DISP
5280  XERR,YERR=0 ! Check for unassigned errors & error-correlations
5290   A,B,RHO=0 @ FOR I=1 TO NN @ A=A+(Ex(I)#Ey(I)=0) @ B=B+(R(I)=0) @ NEXT I
5300   IF A AND B THEN 5350
5310   IF A=0 AND B THEN 5390
5320   IF A AND B=0 THEN 5330 ELSE 5500
5330   DISP "%X-ERRS, %Y-ERRS FOR SAMPLES WITH NO ASSIGNED ERRORS";@ GOSUB RETRIEVE  @ IF Ninputs=0 THEN GOSUB BAD_INPUT  @ GOTO 533
0
5340   XERR=O(1) @ YERR=O(2)#(O(2)#0)+O(1)#(O(2)=0) @ GOTO 5440
5350   DISP "%X-Errs, %Y-Errs, Err-Corrs FOR SAMPLES WITH ZERO ASSIGNED VALUES";@ GOSUB RETRIEVE  @ IF Ninputs=0 THEN GOSUB BAD_INPU
T  @ GOTO 5350
5360   XERR=O(1) @ YERR=O(2)#(O(2)#0)+O(1)#(O(2)=0) @ RHO=O(3)
5370   IF RHO=0 THEN GOSUB 5420 @ GOTO 5350
5380   IF ABS (RHO)>1 THEN GOSUB 5430 @ GOTO 5350 ELSE GOTO 5440
5390   DISP HGL$ (" ERR-CORRS FOR SAMPLES WITH ZERO ASSIGNED VALUES ");@ GOSUB RETRIEVE  @ RHO=O(1)
5400   IF RHO=0 THEN GOSUB 5420 @ GOTO 5390
5410   IF ABS (RHO)>1 THEN GOSUB 5430 @ GOTO 5390 ELSE GOTO 5440
5420   GOSUB CLUNK  @ DISP "Pb-Pb ERRORS ARE "&HGL$ (" ALWAYS ")&" CORRELATED (typical values are .90-.995)." @ DISP @ DISP @ RETURN

5430   GOSUB CLUNK  @ DISP "ERROR-CORRELATIONS MUST BE WITHIN +1 AND -1" @ DISP @ DISP @ RETURN
5440   FOR I=1 TO NN ! Assign errors to sets with zero assigned values
5450    IF Ex(I)=0 THEN Ex(I)=XERR
5460    IF Ey(I)=0 THEN Ey(I)=YERR
5470    IF R(I)=0 THEN R(I)=RHO
5480   NEXT I
5490 !
5500  IF PR=2 THEN DISP "NAME FOR THIS YORKFIT";@ INPUT Name$[1,50]
5510  PRINTER IS Pr(2) @ CLEAR @ PRINT @ PRINT RPT$ ("#",32)&"YORKFIT"&RPT$ ("#",33)
5520 !
5530 ! MODEL-1 is Yorks original 'a priori' fit, assumes all scatter due to analytical error and weights points accordingly.  Studen
t's-t is 1.96.
5540 !
5550 ! MODEL-2 weights all points equally (with zero error-correlation), uses Student's-t for N-2 degrees of freedom.  Equivalent to
 knowing nothing about the
5560 ! actual cause of scatter, and invoked only if the MODEL-1 fit has excess
5570 !  scatter.
5580 !
5590 !
5600  IF PR=2 THEN PRINT Name$ ELSE PRINT
5610  PRINT @ PRINT TAB (24);"ERRORS ARE PERCENT 2-SIGMA"
5620  PRINT "SET#";TAB (8);"206/204";TAB (18);"%err";TAB (32);Y1$;TAB (42);"%err";TAB (54);"CORREL.";TAB (65);"%"&Y1$[3,3]&"/6err" @
 PRINT
5630  Ok,M=0 @ N=NN @ Slope=.1
5640  REDIM X(N),Y(N),Z(N),U(N),V(N),UV(N),U2(N),V2(N),Z2(N),T1(N),T2(N)
5650 ! Ok set to 1 if one fit completed;  M is model#.
5660  M=M+1 @ Slope0=Slope @ I1=0 ! I1 is iteration-counter
5670  IF M>1 THEN DISP "MODEL";M;" YORKFIT..."
5680  Slope=Slope0
5690  I1=1+I1
5700  IF I1>10 THEN Fail ! Give up if unconverged in 10 iterations
5710  IF I1>1 AND M#2 THEN 5830
5720  FOR I=1 TO N
5730   ON M GOTO 5740,5770
5740   E76(I)=SQR (Ex(I)^2+Ey(I)^2-2#R(I)#Ex(I)#Ey(I)) ! % 207/206 ERR
5750   Ex0(I)=Ex(I) @ Ey0(I)=Ey(I) @ R0(I)=R(I)
5760   PRINT I;TAB (7);X(I);TAB (17);Ex(I);TAB (31);Y(I);TAB (41);Ey(I);TAB (53);R(I);TAB (65);FNDr(E76(I),3) @ GOTO 5790
5770   G(I)=1 @ Q(I)=1/Slope^2 @ R(I)=0 @ GOTO 5810 ! Model-2 weighting
5780 !
5790   Q(I)=(1/(Y(I)#Ey(I)/200))^2
5800   G(I)=(1/(X(I)#Ex(I)/200))^2
```

```
5810   A(I)=SQR (G(I)*Q(I))
5820  NEXT I
5830  B5,D,E=0 @ IF M=1 AND I1=1 THEN DISP @ DISP
5840  FOR I=1 TO N
5850   Z(I)=G(I)*Q(I)/(Slope^2*Q(I)+G(I)-2*Slope*R(I)*A(I))
5860  ! S1=S1+Z(I) @ Xbar=Xbar+Z(I)*X(I) @ Ybar=Ybar+Z(I)*Y(I)
5870  NEXT I
5880 ! Xbar=Xbar/S1 @ Ybar=Ybar/S1
5890  S1=SUM (Z) @ MAT T1=Z.X@ MAT T2=Z.Y
5900  Xbar=SUM (T1)/S1 @ Ybar=SUM (T2)/S1 @ MAT U=( -Xbar)+X@ MAT V=( -Ybar)+Y
5910  MAT U2=U.U@ MAT V2=V.V@ MAT UV=U.V@ MAT Z2=Z.Z
5920  FOR I=1 TO N
5930  ! U=X(I)-Xbar @ V=Y(I)-Ybar
5940   B5=B5+(U2(I)/Q(I)-V2(I)/G(I))*Z2(I)
5950   D=D+(UV(I)/G(I)-R(I)*U2(I)/A(I))*Z2(I)
5960   E=E+(UV(I)/Q(I)-R(I)*V2(I)/A(I))*Z2(I)
5970 NEXT I
5980 Slope0=B5^2+4*D*E
5990 IF Slope0<0 THEN Fail
6000 Slope0=(SQR (Slope0)-B5)/(2*D) @ ALPHA CURSROW -1 @ DISP Slope0
6010 IF ABS ((Slope0-Slope)/Slope0)>.0001 THEN 5680
6020 Slope(M),Slope=Slope0 @ Inter(M),Inter=Ybar-Slope*Xbar
6030 D,E,SUMS=0
6040 !
6050 ! Error-algorithm of Titterington & Halliday, Chem. Geol. v. 26, p. 183
6060 FOR I=1 TO N
6070  Y=Y(I)-Slope*X(I)-Inter @ SUMS=SUMS+Z(I)*Y^2
6080  X=X(I)-Z(I)*Y*(R(I)*A(I)-Slope*Q(I))/(Q(I)*G(I))
6090  D=D+X*Z(I) @ E=E+X^2*Z(I)
6100 NEXT I
6110 !
6120 V=N-2 ! Degrees of freedom
6130 MSWD=(V<> 0)*SUMS/(V+(V=0)) ! Mean square of weighted deviates
6140 St_t=(V(2)*12.7+(V>1)*1.96*V/SQR (V^2-2.43*V+1.696) ! Students-t approximation from Andy Turek
6150 B3=SQR (S1/(E*S1-D^2))
6160 F1=B3*SQR (E/S1)
6170 S1=F1*SQR (MSWD)
6180 B5=B3*SQR (MSWD)
6190 IF M>1 THEN 6330
6200 !
6210 IF SUMS<80 THEN 6220 ELSE Prob=0 @ GOTO 6330
6220 D,A=1 ! Calculate chi-square probability
6230  IF INT (V/2)=V/2 THEN 6300
6240  FOR I=2 TO 2000 STEP 2 ! V odd
6250   D=D*SUMS/(V+I) @ A=A+D @ IF D<.000001 THEN 6270
6260  NEXT I
6270  D=SQR (PI )
6280  FOR I=.5 TO V/2 @ D=D*I @ NEXT I
6290  Prob=1-EXP (-(SUMS/2))*(SUMS/2)^(V/2)*A/D @ GOTO 6330
6300  FOR I=1 TO V/2-1 @ D=D*SUMS/(2*I) @ A=A+D @ NEXT I ! V even
6310  Prob=A*EXP (-(SUMS/2))
6320 !
6330 FOR Pr=1 TO PR
6340  PRINTER IS Pr(Pr) @ DISP @ IF Pr=1 AND M>1 THEN ALPHA C_ROW
6350  ON M GOTO 6360,6380
6360  PRINT RPT$ ("-",80) @ PRINT "****MODEL 1 SOLUTION -- ASSUMES ALL SCATTER DUE TO ANALYTICAL ERROR****" @ Ok=1
6370  IF Prob>.15 THEN Errsl(1)=1.96*B3 @ Errint(1)=1.96*F1 @ GOTO 6410 ELSE 6400
6380  PRINT @ PRINT "**********MODEL 2  SOLUTION -- EQUALLY WEIGHTED POINTS WITH UNCORRELATED"
6390  PRINT RPT$ ("*",19)&"ERRORS; ASSUMES NORMAL DISTRIBUTION OF RESIDUALS*****" @ GOTO 6400
6400  Errsl(M)=St_t*B5 @ Errint(M)=St_t*S1
```

```
6410  PRINT @ PRINT TAB (33);"SLOPE";TAB (57);"INTERCEPT" @ PRINT TAB (33);"-----";TAB (57);"---------" @ PRINT
6420  PRINT "BEST-FIT:";TAB (31);FNDr(Slope,6);TAB (56);FNDr(Inter,6) @ PRINT RPT$ ("-",80)
6430   IF M=1 THEN PRINT "ERROR (1-SIGMA, A PRIORI) ";TAB (31);FNDr(B3,4);TAB (56);FNDr(F1,4) @ PRINT
6440   PRINT "ERROR (1-SIGMA, FROM SCATTER) ";TAB (31);FNDr(B5,4);TAB (56);FNDr(S1,4) @ PRINT
6450   PRINT "ERROR (95% CONFIDENCE LIMITS)";TAB (31);FNDr(Errs1(M),4);TAB (56);FNDr(Errint(M),4) @ PRINT
6460   PRINT "COORDINATES OF CENTROID: X=";FNDr(Xbar,6);"Y=";FNDr(Ybar,6)
6470   Xbar(M)=Xbar @ Ybar(M)=Ybar @ IF M>1 THEN 6500
6480   PRINT @ PRINT "ANALYTICAL ERRORS ALONE WILL CAUSE THE OBSERVED AMOUNT OF SCATTER"
6490   PRINT "OR MORE "&VAL$ (FNDr(100*Prob,2))&"% OF THE TIME, AS INDICATED BY THE M.S.W.D. OF";FNDr(MSWD,3)
6500   PRINT RPT$ ("-",80) @ PRINT
6510   NEXT Pr
6520 FOR I=1 TO NN @ Ex(I)=Ex0(I) @ Ey(I)=Ey0(I) @ R(I)=R0(I) @ NEXT I
6530 C_ROW=CURSROW  @ IF M>1 THEN 6560
6540 IF Prob>.2 THEN 6690 ELSE 5660
6550 !
6560 CHOOSE_MODEL: DISP "PRESS KEY FOR MODEL-SOLUTION DESIRED." @ DISP USING "3/" @ GOSUB CLEARKEY
6570   ON KEY# 6," MODEL 1" GOTO 6670
6580   ON KEY# 7," MODEL 2" GOTO 6680
6620   ON KEY# 4,"  HELP" GOTO HELP4
6640  KEY LABEL
6650  GOTO 6650
6660 !
6670  OFF KEY# @ M=1 @ GOTO 6690
6680  OFF KEY# @ M=2 @ GOTO 6690
6690 Slope=Slope(M) @ Inter=Inter(M) @ GOSUB DRAWLINE
6700 Errs1=Errs1(M) @ Errint=Errint(M) @ Xbar=Xbar(M) @ Ybar=Ybar(M)
6710 ALPHA C_ROW
6720 !
6730 Age=FNPb_Pb_AGE(Slope)
6740 Lower_Age=FNPb_Pb_AGE(Slope-Errs1) @ Upper_Age=FNPb_Pb_AGE(Slope+Errs1)
6750 Age_Err=(Lower_Age#0)*(Upper_Age#0)*ABS (Upper_Age-Lower_Age)/2
6760 FOR Pr=1 TO PR
6770  PRINTER IS Pr(Pr)
6780  IF Age THEN PRINT @ PRINT "### AGE FROM SLOPE ALONE IS";FNDr(Age,4);
6790  IF Age_Err THEN PRINT "+/-";FNDr(Age_Err,3);
6800  IF Age THEN PRINT " Ma ###" @ PRINT @ PRINT
6810 NEXT Pr
6820 !
6830 CALC_INTERCEPTS: DISP "CALCULATING GROWTH-CURVE-INTERCEPTS....."
6840    R=CURSROW -1 @ OFF CURSOR
6850    FOR J=0 TO 1
6860     T1=-1000+9000*J
6870     C=FNC(T1) @ X=(Inter+C*FNA(T1)-FNB(T1))/(C-Slope)
6880     IF X<-1 OR (X-ALPHA0)/Mu0>= E0 THEN NO_INTER ELSE T=FND(X) @ ALPHA R,42 @ DISP INT (T)
6890     IF ABS (T-T1)<.01 THEN Cint(J+1)=T ELSE T1=T @ GOTO 6870
6900    NEXT J
6910 !
6920 ! Calculate errors of growth-curve intercepts ------------------
6930   DISP "CALCULATING ERROR-ENVELOPE INTERCEPTS..." @ R=CURSROW -1
6940   FOR J=3 TO 6
6950    T1=9000*(J>4)-1000 @ A=1-2*(J/2#INT (J/2)) @ I=INF  @ COUNT=0
6960    GOSUB ERRCALC  @ Cint(J)=T
6970   NEXT J @ GOTO PRINT_SOLUTION
6980 ERRCALC: ! Calculate growth-curve intercepts of error-envelope about line
6990  COUNT=1+COUNT @ IF COUNT>20 THEN NO_ENV_INTER
7000  S=FNC(T1) @ B=FNB(T1)-S*FNA(T1)-Inter
7010  D=2*(B*(S-Slope)+Xbar*Errs1^2) @ E=(S-Slope)^2-Errs1^2
7020  G=D^2-4*E*(B^2-Errint^2) @ IF G<0 THEN NO_ENV_INTER
7030  X=(A*SQR (G)-D)/(2*E) @ IF X<-1 OR (X-ALPHA0)/Mu0>= E0 THEN NO_ENV_INTER
7040  T=FND(X) @ ALPHA R,42 @ DISP INT (T) @ IF ABS (T-T1)>I THEN NO_ENV_INTER
```

```
7050  IF ABS (T-T1)<.01 THEN RETURN ELSE I=ABS (T-T1) @ T1=T @ GOTO ERRCALC
7060 !
7070 PRINT_SOLUTION: ALPHA CURSROW -2 @ Cint(7)=FNDr(ABS (Cint(4)-Cint(3))/2*(Cint(3)*Cint(4)#0),2)
7080    Cint(8)=FNDr(ABS (Cint(6)-Cint(5))/2*(Cint(5)*Cint(6)#0),2)
7090   PRINT @ PRINT TAB (10);"<<<<<<< MODEL";M;" GROWTH-CURVE-INTERCEPT SOLUTION >>>>>>>>" @ PRINT @ PRINT
7100 A,B,D1,D2=1
7110   IF Cint(7) THEN A=10^INT (LGT (ABS (Cint(7))))
7120   IF Cint(8) THEN B=10^INT (LGT (ABS (Cint(8))))
7130   IF Cint(1) THEN D1=1-INT (LGT (ABS (A/Cint(1))))
7140   IF Cint(2) THEN D2=1-INT (LGT (ABS (B/Cint(2))))
7150   PRINT "LOWER INTERCEPT:  ";@ IF Cint(1) THEN PRINT FNDr(Cint(1),D1);
7160   IF Cint(1) AND Cint(7) THEN PRINT " +/-";Cint(7);"Ma";
7170   IF Cint(1)#0 AND Cint(7)=0 THEN PRINT "Ma";
7180   IF Cint(3)=0 AND Cint(4)=0 THEN PRINT ELSE PRINT TAB (52);"( ";
7190   IF Cint(3) THEN PRINT " >"&VAL$ (FNDr(Cint(3),D1));
7200   IF Cint(4) THEN PRINT " <"&VAL$ (FNDr(Cint(4),D1))&")" ELSE PRINT ")"
7210   PRINT "UPPER INTERCEPT:  ";@ IF Cint(2) THEN PRINT FNDr(Cint(2),D2);
7220   IF Cint(2) AND Cint(8) THEN PRINT " +/-";Cint(8);"Ma";
7230   IF Cint(2)#0 AND Cint(8)=0 THEN PRINT "Ma";
7240   IF Cint(5)=0 AND Cint(6)=0 THEN PRINT ELSE PRINT TAB (52);"( ";
7250   IF Cint(5) THEN PRINT " <"&VAL$ (FNDr(Cint(5),D2));
7260   IF Cint(6) THEN PRINT " >"&VAL$ (FNDr(Cint(6),D2))&")" ELSE PRINT ")"
7270   PRINT @ PRINT "(UNCERTAINTIES ARE 95%-CONFIDENCE LIMITS)" @ PRINT RPT$ ("-",80)
7280 NCHORDS=1+NCHORDS @ BEEP
7290 DISP @ DISP "PRESS "&HGL$ (" LABEL INT. ")&" FOR PLOT-LABEL OF INTERCEPTS."
7300 DISP @ DISP "PRESS "&HGL$ (" CONTINUE ")&" TO DECLINE." @ DISP USING "3/" @ GOSUB CLEARKEY
7360       ON KEY# 6,"LABEL INT." GOTO LABEL_INTERCEPTS
7370       ON KEY# 7," CONTINUE" GOTO 7600
7450       KEY LABEL
7460       GOTO 7460
7470 !
7480 LABEL_INTERCEPTS:
7490  H=THT @ W=H*AR*(1-.25*HP) @ MOVE XMAX,YMIN @ SETGU
7500  IF Cint(2) THEN I1$=VAL$ (FNDr(Cint(2),D2)) ELSE I1$="***"
7510  IF Cint(1) THEN I3$=VAL$ (FNDr(Cint(1),D1)) ELSE I3$="***"
7520  IF Cint(8) THEN I2$=VAL$ (Cint(8)) ELSE I2$="***"
7530  IF Cint(7) THEN I4$=VAL$ (Cint(7)) ELSE I4$="***"
7540  L1=LEN (I1$) @ L2=LEN (I2$) @ L3=LEN (I3$) @ L4=LEN (I4$)
7550  NCAR=14+L1+L2+L3+L4 @ CSIZE H,AR @ LORG 1
7560  IMOVE -((NCAR+17+0*HP)/2*W),2.5*H*NCHORDS @ LABEL "INTERCEPTS AT"
7570  IMOVE -((NCAR-15)/2*W),0 @ LABEL I1$ @ IMOVE (.5+0*HP+L1)*W,H @ GOSUB PM  @ IMOVE (1+0*HP)*W/2,0 @ LABEL I2$&" and "&I3$
7580  IMOVE (L2+L3+5.5+0*HP)*W,H @ GOSUB PM  @ IMOVE (.5+0*HP)*W,0 @ LABEL I4$&" Ma"
7590 !
7600   OFF KEY# @ DEG @ SETUU @ MOVE XMIN,YMAX @ PEN PLR=1 @ ALPHA
7610 !
7620  GOTO ADD
7630 !
7640 Fail: PRINT @ PRINT "DATA CANNOT BE FIT TO A STRAIGHT LINE USING MODEL";M @ PRINT @ PRINT @ GOSUB CLUNK
7650  IF Ok=0 THEN 7620
7660  IF M=2 THEN M=1 @ GOTO 6690
7670 !
7680 NO_INTER: PRINT "***** CHORD DOES NOT HAVE TWO GROWTH-CURVE INTERCEPTS *****" @ PRINT @ Cint(1+J)=0 @ GOTO 6900
7690 NO_ENV_INTER: PRINT @ PRINT "***** ARM";J-2;"OF ERROR-ENVELOPE DOES NOT INTERSECT GROWTH-CURVE *****" @ PRINT @ T=0 @ RETURN
7700 !
7710 PRINT_PLOT: ! Dump CRT-plot to printer
7720         CLEAR @ DISP "PRESS "&HGL$ (" 8 ")&" FOR 82905 PRINTER, "&HGL$ (" 2 ")&" FOR 2673 PRINTER."
7730         ON KYBD KY,"1234567890" GOTO 7750
7740         GOTO 7740
7750         OFF KYBD @ PRINTER IS 701,80 @ K=KY-48
7760         IF K=8 THEN DUMP GRAPHICS
```

```
7770          IF K=2 THEN DUMP GRAPHICS 0,0,0,1
7780          IF K#8 AND K#2 THEN GOSUB CLUNK  @ GOTO PRINT_PLOT
7790          PRINTER IS Pr(2) @ GOTO RE_DATA
7800 !
7810 DRAWLINE: ! Draw a dashed line of the Yorkfit slope&intercept
7820         CLIP XMIN+Xspred/100,XMAX-Xspred/33,YMIN+Yspred/100,YMAX-Yspred/33
7830         NLINES=1+NLINES @ IF CRT THEN LINE(NLINES,1)=Slope @ LINE(NLINES,2)=Inter
7840         IF SOLID_LINE=0 THEN LINE TYPE NLINES+4
7850         MOVE XMIN,XMIN*Slope+Inter @ PEN CPEN
7860         DRAW XMAX,XMAX*Slope+Inter
7870         MOVE XMIN,YMIN @ LINE TYPE 1
7880         CLIP XMIN,XMAX,YMIN,YMAX @ RETURN
7890 !
7900 LETTER: OFF KEY# @ OFF KYBD @ CLEAR ! Draft a phrase on the plot
7910         UNCLIP @ SETUU @ IF PLOTTER_DUMP THEN 7950
7920         DISP @ DISP "(PRESS "&EL$&" TO ESCAPE)" @ DISP
7930         DISP "CHAR.-HEIGHT IN mm (tick labels were";FNDr(THT*Ymm/YLOC/2,2);")[, SLANT (0-30) [, ROTATION]]";
7940         GOSUB RETRIEVE  @ IF Ninputs THEN H=O(1) @ SL=O(2) @ ROT=O(3) ELSE GOTO RE_DATA
7950         PEN CPEN @ CS=2*H*YLOC/Ymm @ W=AR*CS*(1-HP/3.1) @ CSIZE CS,AR,SL @ LDIR ROT
7960         IF HP THEN 8000
7970 DISP @ DISP "ENTER COORDINATES (X, Y)  OF LOWER-LEFT CORNER OF PHRASE";
7980         GOSUB RETRIEVE  @ IF Ninputs<2 THEN GOSUB BAD_INPUT  @ GOTO 7860
7990         X=O(1) @ Y=O(2) @ PEN CPEN @ GOTO 8020
8000         IF PLOTTER_DUMP THEN 8020 ELSE DISP @ DISP "POSITION PEN WITH PLOTTER-CONTROLS, THEN PRESS "&CNT$ @ PAUSE
8010         CURSOR X,Y
8020         MOVE X,Y @ LORG 1
8030         IF PLOTTER_DUMP=0 THEN DISP @ DISP "PHRASE TO BE DRAFTED";@ LINPUT P$
8040         L=LEN (P$) @ SETGU @ IF HP THEN 8080
8050         NPHRASES=1+NPHRASES @ LETTER$(NPHRASES)=P$ @ LETTER(NPHRASES,1)=X @ LETTER(NPHRASES,2)=Y
8060         LETTER(NPHRASES,3)=H @ LETTER(NPHRASES,4)=SL @ LETTER(NPHRASES,5)=ROT
8070         IF CRT AND CPEN=-1 THEN NPHRASES=NPHRASES-2
8080         IF POS (P$,"\")=0 AND POS (P$,"^")=0 AND POS (P$,"+/-")=0 THEN LABEL P$ @ GOTO 8150
8090         IF ROT#0 THEN GOSUB CLUNK  @ DISP @ DISP "CAN'T USE \, ^, OR +/- SYMBOLS WITH NONZERO ANGLES." @ DISP @ GOTO 7790
8100         FOR I=1 TO L
8110          IF I>L-2 THEN 8130
8120          IF P$[I,I+2]="+/-" THEN GOSUB PM  @ I=I+2 @ GOTO 8140
8130          A=(P$[I,I]="^")-(P$[I,I]="\") @ IF A THEN IMOVE 0,A*CS/4 ELSE LABEL P$[I,I] @ IMOVE W,CS
8140         NEXT I
8150         PEN CRT @ SETUU @ LDIR 0 @ IF PLOTTER_DUMP THEN RETURN ELSE 7920
8160 PM: IMOVE 0,.13*CS @ LABEL "+" @ IMOVE 0,.62*CS @ LABEL "-" @ IMOVE W,1.25*CS @ RETURN ! Label a plus-or-minus symbol
8170 !
8180 DATAFILE: CLEAR @ DISP "NAME OF DATA-FILE TO BE RETRIEVED?" @ DISP
8190  DISP "(to get a disk catalog, enter "&HGL$ (" C ")&" for drive 0, "&HGL$ (" c ")&" for drive 1)"
8200  IF DFILE THEN DISP "(press ";EL$;" to re-use data from file ";FILE$&")"
8210  DISP @ GOSUB RETRIEVE_STR  @ IF DFILE AND Ninputs=0 THEN FINISH ELSE FILE$=D$(1)
8220  IF FILE$="C" THEN CAT ":D701" @ GOTO 8240
8230  IF FILE$="c" THEN CAT ":D701" ELSE 8260
8240  DISP @ DISP "PRESS "&HGL$ (" CONT ")&" TO CONTINUE." @ PAUSE
8250  GOTO DATAFILE
8260 DIM CLM(8),VC(100,8)
8270   ON ERROR GOTO WRONGFILE
8280   ASSIGN# 1 TO  FILE$ @ ON ERROR GOTO YORVERT
8290   READ# 1,1 ; N$(1),XX()@ OFF ERROR
8300 !
8310 ! ############### RETRIEVE nX45 NUMERIC DATA-FILE ROWS FROM TAPE ##########
8320 !
8330   CLEAR @ DISP "INPUT COLUMN #s FOR:" @ DISP @ GOSUB COLINPUT_DISPLAY  @ GOSUB RETRIEVE  @ MAT CLM=ZER
8340   NCOLS=Ninputs @ IF NCOLS=1 OR NCOLS=4 OR NCOLS=6 OR NCOLS=7 THEN GOSUB BAD_INPUT  @ GOTO 8330
8350   IF NCOLS=2 THEN CLM(1)=O(1) @ CLM(3)=O(2)
8360   IF NCOLS=3 THEN CLM(1)=O(1) @ CLM(3)=O(2) @ CLM(6)=O(3)
```

```
8370  IF NCOLS>3 THEN MAT CLM=0
8380  CLEAR @ GOSUB NAMEBLANK
8390  MAT VC=ZER
8400  ON ERROR GOTO 8470
8410  FOR I=1 TO 100
8420   READ# 1,I ; N$(I),XX()@ DISP N$(I)
8430   FOR J=1 TO 8
8440    IF CLM(J) THEN VC(I,J)=XX(CLM(J))
8450   NEXT J
8460  NEXT I
8470 RMIN=1 @ RMAX=I-1 @ GOTO FINISH
8480 !
8490 PLOTBOX_SIZE: A,B=0 ! Calculate limits for plot defined by box-size
8500  RATIO_=(LIMX2-LIMX1)/(LIMY2-LIMY1)
8510  IF RATIO_>1 THEN Bgu=100 @ A=1 @ GOSUB AGU  @ GOSUB BGU
8520  IF RATIO_<= 1 THEN Agu=100 @ B=1 @ GOSUB BGU  @ GOSUB AGU
8530  ON ERROR GOTO 8580
8540  LIMIT A1,A2,B1,B2
8550  IF (C=0)*(RATIO >1 EXOR RATIO_>1) THEN C=1 @ GOTO PLOTBOX_SIZE
8560  OFF ERROR @ LIMX1=A1 @ LIMX2=A2 @ LIMY1=B1 @ LIMY2=B2 @ RETURN
8570 !
8580  OFF ERROR @ GOSUB CLUNK  @ DISP @ DISP "**** INSUFFICIENT ROOM FOR LABELS WITH SPECIFIED PLOTBOX-LIMITS ****" @ DISP @ DISP @
GOTO PLOTLIMITS
8590 !
8600 AGU:
8610  MGy=(LIMY2-LIMY1)/(Bgu-YL-YEND) ! millimeters/GU in Y-direction
8620  YL_MM=YL*MGy @ YEND_MM=YEND*MGy
8630  B1=LIMY1-YL_MM @ B2=LIMY2+YEND_MM
8640  IF A THEN Agu=XL+XEND+100*(LIMX2-LIMX1)/(B2-B1)
8650 RETURN
8660 !
8670 BGU:
8680  MGx=(LIMX2-LIMX1)/(Agu-XL-XEND) ! millimeters/GU in X-direction
8690  XL_MM=XL*MGx @ XEND_MM=XEND*MGx
8700  A1=LIMX1-XL_MM @ A2=LIMX2+XEND_MM
8710  IF B THEN Bgu=YL+YEND+100*(LIMY2-LIMY1)/(A2-A1)
8720 RETURN
8730 !
8740 YORVERT: ! Convert VISICALC files to 100X5 arrays for plotting/Yorkfit
8750 !
8760 OFF ERROR
8770 DIM VC$[24],I$[20],J$[20],PARAM$(8)[6]
8780 DATA "6/4","6/4err","7/4","7/4err","RHO-A","8/4","8/4err","RHO-B"
8790 RESTORE 8780 @ READ PARAM$(1),PARAM$(2),PARAM$(3),PARAM$(4),PARAM$(5),PARAM$(6),PARAM$(7),PARAM$(8)
8800 CLEAR @ DISP "INPUT   BEGINNING, ENDING   ROW-NUMBERS CONTAINING SAMPLE-DATA";@ GOSUB RETRIEVE
8810 IF Ninputs#2 THEN GOSUB BAD_INPUT  @ GOTO 8800 ELSE RMIN=O(1) @ RMAX=O(2)
8820  DISP @ DISP "(press "&EL$&" for listing of column names)"
8830  DISP @ DISP "INPUT COLUMNS (e.g. C,D,AK...) FOR:" @ DISP
8840  COL_LIST=0 @ GOSUB COLINPUT_DISPLAY  @ GOSUB COLINPUT  @ IF COL_LIST THEN 8830 ELSE 8970
8850  COLINPUT_DISPLAY: DISP "    6/4, %err,  7/4, %err,  Rho(6/4-7/4),  8/4, %err,  Rho(6/4-8/4)"
8860  DISP @ DISP "or: 6/4, %err,  7/4, %err,  Rho(6/4-7/4)" @ DISP @ DISP "or: 6/4, 7/4, 8/4" @ DISP @ DISP "OR: 6/4,  7/4" @ DISP
8870  RETURN
8880 !
8890 COLINPUT: GOSUB RETRIEVE_STR
8900 IF Ninputs=0 THEN GOSUB COLUMN_NAMES  @ COL_LIST=1 @ RETURN
8910  IF Ninputs=1 OR Ninputs=4 OR Ninputs=6 OR Ninputs=7 THEN GOSUB BAD_INPUT  @ GOTO COLINPUT  ELSE NCOLS=Ninputs
8920  FOR I=1 TO NCOLS
8930   IF O$(I)="A" OR O$(I)="B" THEN GOSUB BAD_INPUT  @ GOTO COLINPUT
8940  NEXT I
8950  RETURN
```

```
8960 !
8970 IF NCOLS=2 THEN CLM(1)=1 @ CLM(2)=3
8980 IF NCOLS=3 THEN CLM(1)=1 @ CLM(2)=3 @ CLM(3)=6
8990 IF NCOLS=5 THEN CLM(1)=1 @ CLM(2)=2 @ CLM(3)=3 @ CLM(4)=4 @ CLM(5)=5
9000 IF NCOLS=8 THEN CLM(1)=1 @ CLM(2)=2 @ CLM(3)=3 @ CLM(4)=4 @ CLM(5)=5 @ CLM(6)=6 @ CLM(7)=7 @ CLM(8)=8
9010 !
9020 CLEAR @ MAT VC=ZER
9030 DISP @ DISP "CONVERTING VISICALC FILE...." @ NUMVALS=0 @ DISP @ DISP
9040 GOSUB NAMEBLANK  @ C_ROW=CURSROW  @ OFF CURSOR @ ASSIGN# 1 TO  FILE$ @ GOTO 9130
9050 !
9060 VC_STRINGS: READ# 1 ; VC$@ OFF ERROR @ ALPHA C_ROW,1 @ AWRIT VC$&"    "
9070   IF VC$[1,1]#">" THEN NEXT_I
9080   A=1+POS (VC$,":") !             String position where data starts
9090   E=NUM (VC$[A,A]) @ IF E=34 OR E=39 THEN A=A+1 ! 1st data-character
9100   B=NUM (VC$[3,3]) @ D=NUM (VC$[2,2]) @ C=3+(B)64 AND B(91)
9110   VCROW=VAL (VC$[C]) @ RETURN ! VISICALC row#
9120 !
9130 FOR I=1 TO 3150 ! Read VISICALC strings from datafile
9140  ON ERROR GOTO FINISH
9150  GOSUB VC_STRINGS  @ IF VCROW(RMIN THEN FINISH
9160  IF VCROW)RMAX THEN GOTO NEXT_I  ELSE ROWN=VCROW-RMIN+1 ! Sample#
9170  IF C=3 THEN COLN=D-66
9180  IF C=4 THEN COLN=24+26*(D-65)+B-64
9190  IF COLN=0 THEN L=LEN (VC$) @ N$(ROWN)[1,9]=VC$[A,L-1] @ ALPHA C_ROW,L+4 @ AWRIT N$(ROWN)&"   " @ GOTO NEXT_I
9200  IF COLN=-1 THEN N$(ROWN)[10,18]=VC$[A,LEN (VC$)-1] @ GOTO NEXT_I
9210  V$=VC$[2,C-1]
9220  FOR PARNUM=1 TO NCOLS
9230   F,R=0 @ GOSUB EXTRACT  @ IF F THEN 9250
9240   IF R THEN NEXT_I
9250  NEXT PARNUM
9260  GOTO NEXT_I
9270 !
9280  EXTRACT: ! Extract numeric values from VISICALC string
9290   IF V$#0$(PARNUM) THEN F=1 @ RETURN ! If not correct column#
9300    ON ERROR GOTO 9330
9310    VC(VCROW,CLM(PARNUM)),E=VAL (VC$[A]) @ OFF ERROR
9320    ALPHA C_ROW,41 @ DISP ROWN;"    ";PARAM$(CLM(PARNUM));" =";E
9330    R=1 @ OFF CURSOR @ OFF ERROR @ RETURN
9340 !
9350 NEXT_I: NEXT I
9360 !
9370 FINISH: OFF ERROR @ MAT S=ZER @ GOSUB COLSELECT
9380  CLEAR @ PRINT "SET#";TAB (29);"206/204";TAB (39);"%err";TAB (49);YI$;TAB (59);"%err";TAB (69);"RHO" @ PRINT
9390 ON ERROR GOTO 9420
9400 FOR I=1 TO RMAX-RMIN+1
9410  PRINT I;TAB (5);N$(I);TAB (29);S(I,1);TAB (39);FNDr(S(I,2),3);TAB (49);S(I,3);TAB (59);FNDr(S(I,4),3);TAB (69);S(I,5)
9420 NEXT I
9430 OFF ERROR @ BEEP @ DFILE=1
9440 IF PR=2 THEN RETURN ELSE DISP @ DISP "PRESS "&HGL$ (" CONT ")&" TO START PLOT" @ PAUSE @ RETURN
9450 !
9460 NAMEBLANK: FOR I=1 TO 100 @ N$(I)="" @ NEXT I @ RETURN
9470 !
9480 WRONGFILE: GOSUB CLUNK  @ DISP @ DISP "### FILE '";FILE$;"' NOT RECOGNIZED ON SPECIFIED DISC;   TRY AGAIN ###" @ DISP @ WAIT 10
00 @ GOTO DATAFILE
9490 !
9500  COLSELECT: ! Select columns from VISICALC array already in memory
9510  DISP @ DISP "Wait.." @ DISP .
9520  FOR I=0 TO RMAX-RMIN ! Transfer elements from complete Pb-isotope array to array for plotting (e.g. 6/4-7/4-8/4 to 6/4-7/4)
9530   FOR J=1 TO 5
9540    IF Ytype=1 THEN K=J ELSE K=J*(J(3)+(J+3)*(J)2)
```

```
9550    S(I+1,J)=VC(RMIN+I,K)
9560   NEXT J
9570  NEXT I
9580 RETURN
9590 !
9600 COLUMN_NAMES:
9610 ! Subroutine to find and list column-names of VISICALC files.
9620 ! Assumes that column-names are underlined with repeating "=", so that
9630 ! the VC-string is "/-=", and that rows 3 AND 4 contain the column-names.
9640 ! Also assumes a 9-character column-width for the VISICALC file.
9650 DIM CN$(61,2)[9]
9660 !
9670 ALPHA 1 @ CLEAR @ C_ROW=CURSROW  @ B$="         "
9680 DISP "Locating column-name rows.  Please wait..."
9690 FOR I=1 TO 61 @ CN$(I,1),CN$(I,2)=B$ @ NEXT I
9700 ASSIGN# 1 TO  FILE$ @ ON ERROR GOTO 9740
9710 FOR I=1 TO 3150
9720  READ# 1 ; VC$@ IF POS (VC$,"/-=") THEN 9750
9730 NEXT I
9740 OFF ERROR @ DISP @ DISP HGL$ (" NO '/-=' VISICALC-CELLS FOUND - CAN'T FIND NAMES ") @ DISP @ DISP @ RETURN
9750 ALPHA C_ROW @ CLEAR
9760 FOR I=1 TO 3150
9770   ON ERROR GOTO 9860 @ GOSUB VC_STRINGS
9780   ON 1+(VCROW>4)+2*(VCROW<3) GOTO 9790,9840,9860
9790   IF C=3 THEN COLN=D-66
9800   IF C=4 THEN COLN=24+26*(D-65)+B-64
9810   IF COLN<1 THEN NEXT_I
9820   IF VCROW=3 OR VCROW=4 THEN P=POS (VC$,CHR$ (13)) @ CN$(COLN,VCROW-2)=TRIM$ (VC$[A,P-1])
9830 !
9840 NEXT_I: NEXT I
9850 !
9860 OFF ERROR @ CLEAR @ PRINT "COLUMN";TAB (10);"COLUMN-NAME" @ PRINT
9870 FOR I=1 TO 61
9880  IF I<25 THEN I$=CHR$ (I+66)
9890  IF I>24 AND I<51 THEN I$="A"&CHR$ (I+40)
9900  IF I>50 THEN I$="B"&CHR$ (I+14)
9910  IF LEN (CN$(I,1))=0 AND LEN (CN$(I,2))=0 THEN 9940
9920  IF CN$(I,1)=B$ AND CN$(I,2)=B$ THEN 9940
9930  PRINT "  ";I$;TAB (10);CN$(I,1);TAB (20);CN$(I,2)
9940 NEXT I
9950 PRINT @ PRINT @ RETURN
9960 !
9970 DUMP_TO_PLOTTER: DISP "PRESS "&HGL$ (" ENDLINE ")&" TO BEGIN HARD-PLOTTER DUMP, ENTER ANY NUMBER TO DECLINE" @ GOSUB RETRIEVE
@ IF Ninputs THEN RE_DATA
9980  OFF KYBD @ CPEN=1 @ SPEED$="VS" @ PLOTSIZE=8 @ PLOTTER_DUMP=1 @ HP=1 @ CRT=0 @ PLR=705 @ GOSUB 1960 @ CLEAR
9990   FOR JJ=1 TO C_CT
10000     K,N=1 @ X(K)=CACHE(JJ,1) @ Y(K)=CACHE(JJ,3) @ Ex(K)=CACHE(JJ,2) @ Ey(K)=CACHE(JJ,4) @ R(K)=CACHE(JJ,5)
10010     SYMBOL=SYMBOL(JJ,1) @ GOSUB PLOTT
10020   NEXT JJ
10030   FOR I=1 TO NLINES
10040    Slope=LINE(I,1) @ Inter=LINE(I,2) @ GOSUB DRAWLINE
10050   NEXT I
10060 !
10070   FOR JJ=1 TO NPHRASES
10080    P$=LETTER$(JJ) @ X=LETTER(JJ,1) @ Y=LETTER(JJ,2) @ H=LETTER(JJ,3) @ SL=LETTER(JJ,4) @ ROT=LETTER(JJ,5) @ GOSUB LETTER
10090   NEXT JJ
10100 !
10110 PLOTTER_DUMP=0 @ GOTO RE_DATA
10120 !
10130 HELP: CLEAR ! Display instructions for special-function keys.
```

```
10140 DIM DF1$[2],DF2$[10],DF3$[60],DF4$[60],DF5$[60]
10150  PRINTER IS 1 @ PRINT TAB (16);HGL$ (" FUNCTIONS OF KEYS DEFINED DURING LAST DISPLAY: ") @ PRINT @ PRINT
10160  RETURN
10170 !
10180 PRINT_HELP: OFF KEY#
10190 FOR I=1 TO NUM_KEYS
10200  READ DF1$,DF2$,DF3$,DF4$,DF5$
10210  PRINT "KEY# "&DF1$;TAB (9);"- "&DF2$&":";TAB (23);DF3$
10220  IF DF4$#"" THEN PRINT TAB (23);DF4$
10230  IF DF5$#"" THEN PRINT TAB (23);DF5$
10240  PRINT
10250 NEXT I
10260 PRINT @ PRINT "Use "&HGL$ (" ROLL ")&" key to scroll display, press "&HGL$ (" CONT ")&" to return to previous display."
10270 PAUSE @ RETURN
10280 !
10290 HELP2: GOSUB HELP  @ RESTORE 10300 @ NUM_KEYS=13 @ GOSUB PRINT_HELP  @ GOTO RE_DATA
10300 DATA "2","OPEN Eell",Choose an open (unfilled) 2-sigma error-ellipse as the,plotting symbol.,""
10310 DATA "9",SOLID Eell,Choose a solid (filled-in) 2-sigma error-ellipse as the,plotting-symbol.,""
10320 DATA "4",POLYGON,Choose an open or solid polygon (either regular or,star-shaped) of arbitrary size & orientation as the,plotti
ng-symbol.
10330 DATA "7",YORKFIT,Calculate a York-style best-fit line to the last-,"plotted set of data, starting with set# 1.  Also"
10340 DATA "calculates isochron age, if appropriate."
10350 DATA "10",PEN 1,Use pen# 1 (plotter) or solid-line (CRT).,"",""
10360 DATA "3",PEN 2,Use pen# 2 (plotter only).,"",""
10370 DATA "12",ERASE PEN,Use 'erasing pen' to un-draw symbols or letters,(CRT only).,""
10380 DATA "1",LETTER,Draft a phrase anywhere on the plot.,"",""
10390 DATA "11",PRINT PLOT,Dump the CRT plot to an HP-82905B or 2673 printer.,"",""
10400 DATA "3",TO PLOTTER,Dump the CRT plot to an HP-7470A plotter.,"",""
10410 DATA "6",REDRAW,Start another plot from scratch.,"",""
10420 DATA "14",DATAFILE,Bring data from either a numeric or VISICALC data-file,into memory.,""
10430 DATA "","+#DX ...",Use any keyboard-key symbol as the plotting-symbol.,"",""
10440 !
10450 HELP1: GOSUB HELP  @ RESTORE 10470 @ NUM_KEYS=18 @ GOSUB PRINT_HELP  @ GOTO BEGIN
10460 !
10470 DATA "1",0 GTICKS,Use small circles as the growth-curve ticks and label,the growth-curve ticks horizontally.,""
10480 DATA "8",\ GTICKS,"For growth-curve ticks, use segments of the constant-mu",isochrons and label the ticks parallel to the tick
s.,""
10490 DATA "2",CRT-PRINT,"Use the CRT as the printing device, rather than a",hard-copy printer.,""
10500 DATA "3",CRT PLOT,"Use the CRT as the plotting device, rather than a",hard-copy printer.,""
10510 DATA "9",PRINTER,Use a hard-copy printer as the printing device rather,than the CRT.,""
10520 DATA "10",HARDPLOT,Use an HP-7470A plotter as the plotting device rather,than the CRT.,""
10530 DATA "4",PEN 1,"Use pen# 1 of the HP-7470A plotter, or the white-line pen","(CRT) as the plotting-pen.","
10540 DATA "11",PEN 2,Use pen# 2 of the HP-7470A plotter.,"",""
10550 DATA "12",PEN SPEED,Specify the speed of the plotter pen (default value is,the fastest speed)  (plotter only).,""
10560 DATA "5",DATAFILE,Load data from a numeric or VISICALC data-file into,memory.,""
10570 DATA "6",#NORM#,Use 'normal'-size labels and symbols.,"",""
10580 DATA "13",SLIDE,Use larger-than-normal labels and symbols: useful for 35mm,transparencies or other reduced reproductions.,""
10590 DATA "7",START,Enter axis-names & axis-limits for plot and start,"plotting.  You don't need to press any other keys for",just
a CRT plot.
10600 DATA "5",STACK,Suppress X-axis label and X-axis tick-labels for more,convenient vertical stacking of plots sharing the same,X-
axis.
10610 DATA "7","207/204",Use 207/204 as the Y-axis of the plot.,"",""
10620 DATA "14","208/204",Use 208/204 as the Y-axis of the plot.,"",""
10630 DATA N,"",Don't draw a lead-isotope growth-curve on the plot.,"",""
10640 DATA T,"",Don't draw ticks or tick-labels to the lead-isotope growth,curve.,""
10650 !
10660 HELP3: GOSUB HELP  @ RESTORE 10680 @ NUM_KEYS=3 @ GOSUB PRINT_HELP  @ GOTO ADD
10670 !
10680 DATA "5",YORKFIT,Calculate a York-type best-fit line to the last set of,"data, & calculate an isochron age if appropriate.","
```

```
10690 DATA "6",RETAIN,"Add any subsequent-input points to the previous ones,",so that points with different plot-symbols can be
10700 DATA pooled for a later Yorkfit or Polyfit.
10710 DATA "7",CONTINUE,Return to the input-data part of the program.,"",""
10720 !
10730 HELP4: GOSUB HELP  @ GOSUB 10740 @ DISP @ DISP @ GOTO CHOOSE_MODEL
10740 PRINT "MODEL 1:   Use the Model 1 solution for drawing the best-fit line and for any" @ PRINT TAB (12);"concordia-intercepts.
 Model 1 assumes that the";
10750 PRINT " only cause of scatter" @ PRINT TAB (12);"is from the assigned analytical errors.  The points are weighted";
10760 PRINT TAB (12);"according to those errors only.  If the probability of fit is less";TAB (12);"than 15%, the assigned errors ar
e expanded until they can";
10770 PRINT " account" @ PRINT TAB (12);"for the scatter, and a Student's-t multiplier of the errors is used." @ PRINT
10780 PRINT "MODEL 2:   Use the Model 2 solution for drawing the best-fit line and for any"
10790 PRINT TAB (12);"concordia-intercepts.  Model 2 assumes that nothing is known about"
10800 PRINT TAB (12);"the reason for the scatter of the points, and so weights all points"
10810 PRINT TAB (12);"equally, with zero error-correlations.  A Student's-t multiplier is" @ PRINT TAB (12);"always used." @ PRINT @
 PRINT
10820 RETURN
```