APPLICATIONS OF COMPUTERS TO BALANCED CROSS-SECTIONS

By

SARAH DAWN SALTZER

SUBMITTED TO THE DEPARTMENT OF EARTH, ATMOSPHERIC AND PLANETARY SCIENCES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY, 1986

© Massachusetts Institute of Technology

Signature of Author

[Signature]

Department of Earth, Atmospheric and Planetary Sciences, May 9, 1986

Certified by

Kip Hodges

Thesis Supervisor

Accepted by

Chairman, Department Committee on Graduate Students
Applications of Computers to Balanced Cross-Sections

By

Sarah Dawn Saltzer

Submitted to the Department of Earth, Atmospheric and Planetary Sciences in partial fulfillment of the degree of Master of Science at the Massachusetts Institute of Technology.

May 9, 1986

Abstract

Making the subsurface area of a geologic section balance is one technique aimed at helping the geologist determine the validity of a proposed deformed cross section. However, checking for bed length balance in concentric fold regimes, area balance in regions of similar folding and finally retrodeforming a section are all very time consuming steps.

Using a series of four computer programs this entire process can be simplified and the errors greatly reduced. The program SECTION uses topographic and structural orientation data to constrain the rough structural geometry along a line of section. Tests for balance can be made using the program BALANCE once the geologist has integrated his knowledge and interpretations with the computer generated cross section. The program BALANCE uses an iterative method for finally generating an area balanced cross section, a bed length balanced cross section, or both. This program also retrodeforms, further constraining the validity of the section. A last pair of programs, THREEDIM and PROJECTION assist three-dimensional balancing of a volume of deformed strata.

This software package yielded successful results in the Canadian Rockies Front Range. Four cross sections were interpreted independently from this area and were balanced and retrodeformed to test their validity. Using the data from these sections, two intermediate sections were created which passed all tests for balance.

Thesis Supervisor: Kip Hodges
Professor of Geology
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>4</td>
</tr>
<tr>
<td>Introduction</td>
<td>5</td>
</tr>
<tr>
<td>SECTION</td>
<td>8</td>
</tr>
<tr>
<td>BALANCE</td>
<td>22</td>
</tr>
<tr>
<td>THREEDIM</td>
<td>35</td>
</tr>
<tr>
<td>PROJECTION</td>
<td>39</td>
</tr>
<tr>
<td>Example from the Canadian Rockies</td>
<td>42</td>
</tr>
<tr>
<td>Conclusions</td>
<td>53</td>
</tr>
<tr>
<td>References</td>
<td>54</td>
</tr>
<tr>
<td>Appendix 1 SECTION and EXTRASUBS program listings</td>
<td>56</td>
</tr>
<tr>
<td>Appendix 2 BALANCE program listing</td>
<td>74</td>
</tr>
<tr>
<td>Appendix 3 THREEDIM program listing</td>
<td>90</td>
</tr>
<tr>
<td>Appendix 4 PROJECTION program listing</td>
<td>95</td>
</tr>
</tbody>
</table>
Acknowledgements

I would first like to thank Kip Hodges for being a wonderful thesis advisor, and having faith in my computer programs when I had none. Peter Tilke has never stopped helping and has had a lot of input into the structure of these computer programs. Most of all, I would like to thank Dave Lineman for all of his time and effort in helping me survive the writing of this thesis and for being a great friend.
Introduction

The geologic cross section is a fundamental tool which permits the structural geologist to depict her or his interpretation of subsurface geology. The permissibility of this interpretation can be examined by testing the cross section for balance. The concept of balanced cross sections was formally introduced by Dahlstrom (1969) and has since been expanded upon by a variety of workers including Elliot (1983) and Woodward, Boyer and Suppe (1985). Strictly speaking, a balanced cross section is one in which the two dimensional areas representing strata in the section are equal in area to these strata in the undeformed state. Therefore, restricting cross sections to conform to a balanced geometry eliminates possible interpretations which do not adequately describe a series of events leading to a body of deformed rock.

Cross section balancing, however, is a tedious task with many sources of error. In this paper I describe an interactive software package which permits use of a desk top computer to streamline the balancing process, reduce the errors, and increase the reproducibility of interpretive sections. The initial programs assist the user in the construction of preliminary sections from geologic data. The final programs permit the integration of several balanced sections from a single area into a three dimensional view of the structural style of the area.
The programs are written in Hewlett Packard BASIC(2.1) for the HP 200 series personal computers. With minor changes, however, they can be altered to run on any computer system that supports BASIC. All of these programs require some sort of digitizing tablet for data input. The digitizing and plotting commands assume a Hewlett Packard device which supports HP Graphics Language (HPGL) for this purpose. Therefore, lines in the program with these statements may have to be altered if different equipment is to be used. The program BALANCE requires a RAM memory of one megabyte.

Finally, I have taken advantage of the soft-key feature of the HP personal computer in writing of the program SECTION. On the upper left hand corner of the keyboard there are ten keys of which eight are defined within the context of the program and correspond to different program segments. The need for these keys can be eliminated with GOTO and IF....THEN statements if user defined function keys do not exist on the keyboard of another system.

These four computer programs assist in all phases of cross section development and interpretation. The first program SECTION, assembles topographic, fold, well and structural data on to a line of section and makes a preliminary guess at the geometries. The next program, BALANCE tests an interpretation for balance and then assists in attaining a balanced section. The Program THREEDIM takes balanced cross sections from the same area and places them in a three dimensional grid defining the entire volume of
deformed rock. The final program, PROJECTION, looks at structures off of the line of section and attempts to balance the structures in three dimensions.

Figure 1 Flowchart illustrating general sequence of programs.
Figure 2 Flowchart for program SECTION.
SECTION

The first step in drawing balanced cross sections is to assemble all of the pertinent geologic data from the map and to project it to the desired line of section. Then, from this skeletal section, the geologist can attempt to interpret the often complicated underlying structures. The program SECTION performs some of these more tedious tasks needed to create the base section, including generating topographic profiles, plotting geologic data, and drawing folds.

The program is broken into sub-programs which call various subroutines to perform some complex and often repeated functions. A few of these subroutines are stored in a separate program, EXTRASUBS, and are independently loaded and deleted as they are needed by a program segment. When a given program option is completed, the user returns to the primary command level of the program with the option of choosing another branch defined by a different soft-key.

Getting Started

The computer needs some basic information in order to do any calculations and manipulations. At the start of the program the user must input the scale of the map. The orientation of the line of section is extremely important and the two endpoints must be input (digitized on the plotter). The map should be oriented on the plotter with north pointed to the top of the plotter. All output that is generated will be at the same scale as the original map in metric units.
**TOPODIGIT**

This program segment digitizes the topography for the construction of a topographic profile. The user first inputs the elevation of each point which is to be digitized along the line of section. Given that most Canadian and USGS topo maps do not have elevations printed in metric, these numbers can be input in American Standard Units (feet) and will be converted internally.

The user next digitizes, from left to right, each data point along the line of section. The program rotates the line of section to be parallel with the base of the plotter and then automatically creates a binary data file containing these data on the current floppy disk. The user inputs a file name and the file that is generated has the word "Topo" concatenated to it. This file will be used by the next program segment, TOPODRAW.

**TOPODRAW**

There are two options available for drawing topographic profiles. The computer can simply "connect the dots" (the elevation data points) or instead can fit a cubic spline curve, Strang (1985). In either case, the computer concludes by drawing the vertical axis and labelling the high and low elevation in meters (Fig. 3).

The cubic spline fit uses the subroutine SPLINE which is automatically loaded from the batch of extra subroutines. A cubic spline interpolation is exact and the curve is constrained to pass through each point.
The first part of the cubic spline routine calculates a best fit slope at each point for the curve. For a set of x-coordinate points $X_0...X_N$ and corresponding elevations $Y_0...Y_N$, the best slopes $S_0...S_N$ are determined by the matrix equation:

$$A s = B$$

(1)

where,

$$s = S_0...S_N$$

and,

$$A = \begin{bmatrix} \frac{2}{H_1} & \frac{1}{H_1} \\ \frac{1}{H_1} & \frac{2}{H_2} + \frac{2}{H_1} & \frac{1}{H_2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{H_N} & \frac{2}{H_N} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{Y_1-Y_0}{H_1^2} \\ \frac{Y_3-Y_2}{H_3^2} + \frac{Y_2-Y_1}{H_2^2} \\ \vdots \\ \frac{Y_N-Y_{N-1}}{H_N^2} \end{bmatrix}$$
The variable $H$ is the spacing between the $x$-coordinate values where $H_1 = X_1 - X_0$, $H_2 = X_2 - X_1$, .... $H_N = X_N - X_{N-1}$. Once the slopes have been determined, a hermite fit, Strang (1985), is used to calculate the curve connecting the points with known slopes.

A hermite curve fit simply connects two points each with known elevation and slope. Unit distance between points is assumed and the resulting curve is scaled at a later time. The curve is calculated using the equation:

$$u(x) = A(x-1)^2(2x+1) + B(x-1)^2x + Cx^2(3-2x) + Dx^2(x-1) \quad (2)$$

where, $A = Y_0 \quad B = S_0 \quad C = Y_1 \quad D = S_1$,

and the resulting curve is fit between the first two points. This procedure is then repeated for points 2 through $N$. The result is a smooth curve passing through the center of each topographic elevation point.

FOLDDIGIT

The program segment FOLDDIGIT enables the user to produce vertical or axial down-plunge projections of a fold outcrop. The user marks a number of points along the contact of a folded layer at its intersection with the topography. Next the user enters the elevation of each of these data points, in metric or in feet, and then proceeds to digitize each point along the outcropping contact. All of this information is stored in a file with "Fold" concatenated to the file name, for use by either of the program segments FOLDDRAW or AXIAL.
Technically, one would not put an axial projection on to a cross section. However, the down plunge view can be interesting to the geologist in order to classify the types of folding within the region. An axial projection is a view of the fold across a plane which is perpendicular to the plunge vector in space.

The program segment AXIAL reads a data file created by FOLDDIGIT and projects each point individually along a user input plunge and trend onto a plane normal to the plunge vector. After viewing the projection, the user has the option of choosing a slightly different plunge and trend, if these values are not well constrained.
The user next connects the projected data points defining the down plunge projection via short straight line segments or by the use of a best-fit polynomial. The best fit polynomial curve is obtained with the use of the subroutines POLY and INTERVAL which are automatically loaded from EXTRASUBS. INTERVAL redefines the location of each point along the x-axis. The accuracy of each point location is slightly reduced as the x-value is rounded to two significant figures where the fractional value is a multiple of 0.25. The purpose of INTERVAL is to redefine each point so that the final polynomial, which is drawn at intervals of 0.25 plotter units (1/16 of an inch) will actually pass through each point exactly.

Once the points have been redefined, the subroutine POLY determines the best fit polynomial. POLY uses x and y coordinate values obtained from INTERVAL and renames them WW(*) and UU(*) respectively. The user inputs a suggested order for the polynomial which can be easily changed if the resulting curve is unsatisfactory. A polynomial of order n has the form:

\[ y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_n x^n \]  (3)

where x goes from the minimum value of UU(*) to the maximum value of UU(*) and y corresponds to the resulting WW(*) between these 2 endpoints. The constants \( a_0 \) through \( a_n \) are
best determined in a least-squares sense by the relation:

\[ A^T A C = A^T B \]  

(4)

where,

\[
A = \begin{bmatrix}
1 & x_0 & \cdots & x_0^{n-1} \\
1 & x_1 & \cdots & x_1^{n-1} \\
& \vdots & \ddots & \vdots \\
1 & x_{m-1} & \cdots & x_{m-1}^{n-1}
\end{bmatrix}, \quad C = \begin{bmatrix}
a_{n-1} \\
a_{n-2} \\
\vdots \\
a_0
\end{bmatrix}, \quad B = \begin{bmatrix}
Y_0 \\
Y_1 \\
\vdots \\
Y_{m-1}
\end{bmatrix}
\]

for a set of m distinct points. The final curve contains points spaced every 1/16 cm which are connected by straight line segments (Fig. 4).

**FOLDDRAW**

This program segment accesses the same file used by AXIAL to create a vertical view of the fold along a user input line of section. The points are projected along their plunge vector in space to the intersection of this vector with a vertical plane containing the cross section (Fig. 5). Finally, the data points can be connected by straight line segments or a polynomial best fit curve, as described in the last section.
GEODIGIT

This procedure assembles files containing the locations and the values of a number of geologic data points. The geologic information can be: 1) strike and dip measurements to be projected along strike to the line of section; 2) strike and dip measurements to be projected along the plunge and trend of the surrounding structure to the line of section; 3) or well locations. The information for each of these three data types is input in a slightly differing manner.

For strikes and dips projected along strike to the line of section, the user inputs both the strike in degrees from north and also the direction, east or west. Dip is input in the same manner and the location of the symbol on the map is digitized. These data are stored in a binary data file with "Geo" concatenated to the user input file name. These data are then projected by the program segment GEODRAW to the appropriate x-coordinate position along the line of section and are plotted on the resulting section showing now the apparent dip at the elevation of the topography at that same
x-coordinate position. If the computer also asks for the unit number. Each stratigraphic unit should be assigned a number so that contacts may be fit to the data.

If the rocks in the location of these same symbols are involved in large scale folding then as an alternative the projection can occur along the regional trend and plunge of the underlying structure. In this case, the strike, dip, and unit number are entered as before, and then in addition the user inputs the regional trend measured in degrees from north, the direction E or W, and the plunge and plunge direction measured in the same manner. The user next digitizes the location of the strike and dip symbol on the map and all of the above information is then stored sequentially in the same "Geo" file. When these points are eventually drawn onto the line of section by the procedure GEODRAW, they are projected along trend, down plunge, and plotted at their resulting elevation with their apparent dip.

For well locations, the user inputs only the general strike of the beds and then digitizes the location of the well on the map. The file containing all well information is created separately using the procedure WELLFILE1.
WELLFILE1

This program option is used only in conjunction with GEODIGIT and basically creates binary data files with "Well" concatenated to the file name, containing well information. For each unit or contact encountered in the well for which there is information the user enters the depth, the general dip and dip direction and the unit number in order of increasing depth. Wells are projected horizontally along strike to the line of section and are drawn onto the cross section so that the depths of units correspond to the same scale as the topographic elevations.

GEODRAW

In order to actually invoke this program segment the user must choose option 8 from the main command level, "Draw Cross Section". This procedure will itself call GEODRAW as it is needed.

GEODRAW goes a step beyond plotting geologic dip symbols on a cross section correctly. It can also draw hermitian curves to fit the data by invoking the subroutine HERMITE. A hermite cubic finds the best fitting curve to two points when for each point the y-coordinate (elevation) and the slope (in this case, apparent dip dy/dx) are known. This corresponds to:
\[ A = y(0) = \text{elevation of point 1} \quad (5) \]
\[ B = \frac{dy(0)}{dx} = \text{dip of point 1} \quad (6) \]
\[ C = y(1) = \text{elevation of point 2} \quad (7) \]
\[ D = \frac{dy(1)}{dx} = \text{dip of point 2}. \quad (8) \]

Using equation (2) the resulting curve, \( u(x) \), has the correct dip as it passes through each data symbol on the line of section. This same procedure is carried out continuously connecting data corresponding to the same unit number, and repeats itself for each successive unit number. The curve can intersect data projected along strike, along plunge, and well data (Fig. 6)

![Figure 6 Hermite fit to geologic data.](image)

In some cases, an oversupply of geologic data can create a problem for hermite curve fitting. A final option in GEODRAW allows the user to selectively remove excess and obviously out of place data from the file created by GEODIGIT or WELLFILL1 so that a curve can nicely fit the data.
**Draw Cross Section**

This procedure enables the user to combine two or more types of output onto one line of section. This reduces to superimposing topographic and geologic data together, or topographic and fold data together on to one line of section. With fold and topographic data this is quite straightforward. The fold is drawn to intersect the topography at the same x-coordinate position and elevation on the cross section as on the map. The merging of topographic and geologic data is much more involved.

Certain geologic data, for example symbols which are projected along strike, are plotted at their resulting x-coordinate position and at the elevation of the topography at that point. Therefore, the computer must already have the necessary topographic information to plot the point correctly. Once all of the symbols have been placed on the line of section by invoking the abilities of GEODRAW a hermite curve, as described earlier, may be fit to the data.
Figure 7 Flowchart for program BALANCE
BALANCE

The program BALANCE uses a succession of methods to determine whether geologic cross sections can be defined as balanced. The input for this program consists of digitized cross sections drawn by the geologist. After a preliminary test for balance, using an iterative approach, the user can simply dictate a change in the cross section to successively converge on a cross section that is either area or bed length balanced. Files of data are maintained and updated for reproduction purposes.

The first step in checking for balance is to digitize a cross section which has been drawn using accurate geologic data from a map. The computer can handle data for up to 4 layers and 8 faults. The data arrays are easily redimensioned if extra memory is available. The user inputs the scale of the section and the name for the output file containing the digitized cross section information. Initially, up to 20 points are digitized along each fault from left to right, starting at the left-most fault. The last point is digitized twice to inform the computer of the fault end. After all of the faults have been digitized, the x and y location coordinate values for each point are stored in a separate file from the layer contact information, with II concatenated to the file name.

Next, the layer contacts are digitized starting on the left side of the cross section. The user digitizes individual
points along a path defined by the contact boundary. When either a fault or end of a layer is encountered, the point is digitized twice. A "segment", which is defined as a contact bounded by 2 faults or a fault and the end of the cross section can have up to 25 points digitized along it. A separate binary data file with "Bal" concatenated to the file name is created.

If mistakes are made while digitizing, and are recognized early enough, they can be corrected. If a point along the middle of a segment is digitized twice and yet does not border along a fault, pressing the number (3) clears the second digitized location. In addition, because a file is created once the faults are all digitized correctly, if a serious error occurs subsequently in digitizing along the contacts and it is necessary to start over, then the faults do not have to be redigitized if the location of the cross section on the plotter has not changed. The user simply reactivates the program, inputs the same file name, and enters zero (0) for the number of faults.

Digitizing Techniques

This program assumes that all faults cut all bed contacts and all layers. In some cases, for example blind thrusts (Boyer and Elliot, 1983) this criterion is not satisfied. However, simple digitizing techniques can allow for more complicated sections to be tested for balance. In the case of blind faults, simply extending the fault with no displacement
into the upper layers can solve the problem.

Another problematic situation arises when sections contain horse structures, created when imbricate thrusts propagating in front of earlier thrusts, finally rejoin the earlier thrust. Some layer contacts may not exist within a horse bounded by 2 faults. To solve this problem, the user digitizes consecutively along the contact in question, digitizes twice at the first fault, digitizes one imaginary point 2 times within the horse structure and then continues digitizing immediately to the right of the second bounding fault. Finally, overturned beds create a problem for polynomial curve fitting. One solution is to draw an imaginary fault with no displacement along the axial plane of the fold. The result will be to divide the bed into two segments for separate curve fitting.

**Polynomial curves**

After all digitizing has been completed, polynomial functions can be fit to the individual segments which were defined earlier. The object of the polynomials is to obtain detailed shapes of beds so that the computer can have more accurate information for balancing calculations.

When a cross section containing only a simple fold is digitized and calculations are performed without a polynomial fit the bedlength is calculated by summing the straight line segments defining the fold between the digitized points. The errors associated with this procedure can be noticeable. In
perfectly cylindrical folds, for example, with a radius of curvature of 100 m and digitized points spaced every 100 m the error is 2.7%. Clearly, if spacing between digitized points is decreased, then the accuracy of calculations increases in both bed length and area balanced sections. This is the basis for the polynomial fit.

The program, BALANCE calls the same subroutine, POLY which was described in the program SECTION. Polynomials for each segment can be up to 8th order, yet must have an order less than or equal to the number of points in the segment. The user decides for which segments polynomial fits may aid, and may settle for the straight line fits in unfolded regions in order to save time. The digitized points within the segment are projected onto the CRT so that the user can choose an appropriate order for the resulting polynomial. If the fit is not adequate, the process can be repeated a number of times to satisfy the user. Polynomial fits are also calculated for the faults.

The polynomial curve which is generated is not actually a curve, but rather a collection of short connected line segments, each segment being 1/16 cm long in the x-direction. The result appears as a continuous curve. If folds have been digitized with care, for example hinge and trough points have been entered as well as more points in zones of increased complexity (inflexion zones), then polynomial curves can actually fit the original folds on the cross section exactly.
After polynomial curves have been fit to the desired sections, the entire cross section can be plotted on the CRT or on the plotter. It is generally a good practice to view the section as a check for digitizing errors.

**Tests for Area Balance**

In many foreland fold and thrust belts the deformation path is approximately plane strain. Therefore, one can orient the section parallel to the transport direction so that conservation of rock volume becomes conservation of area in cross section. The bed area in a deformed cross section must equal the area in the undeformed state for a section to be acceptably balanced.

The general method of the area balance procedure is to integrate to find the entire area below the highest contact down to the bottom of the cross section. This procedure is repeated to find the area under the next lower contact. By subtracting these 2 quantities, the area of the bed layer in cross section can be determined (Fig. 8).

Layer area = [Diagram of area calculation]

Figure 8 Integration procedure.
The integration procedure involves no calculus. Instead, trapezoidal approximations are made to calculate areas by the formula:

\[
\frac{(b_1+b_2) \cdot n}{2} = \text{Area} \quad (9)
\]

The variable \( n \), for an individual trapezoid, is defined by the horizontal distance between two neighboring digitized points. The variables \( b_1 \) and \( b_2 \) correspond to the \( y \)-coordinates of 2 adjacent digitized points (Fig. 9).

In an unfaulted, but complexly folded region, the sequence is very straightforward and the calculations are instantaneous. Complications arise with the addition of faults (Fig.10). If trapezoidal area \( j \) is subtracted from \( i \) the resulting area is not equal to the bed area to the left of the fault. In fact, this bed area is incorrect by an amount \( m \). The calculation \( i+m-j = \text{area} \) results in the correct area.

\[\text{Figure 9 Trapezoidal areas.}\]

\[\text{Figure 10 Determining fault areas.}\]
determination. Therefore, the variable m must be calculated in order to obtain the correct bed area value. m is simply a trapezoidal area with \( b_1 \) and \( b_2 \) defined by the value of the intersection of each of the contacts with the fault surface. A similar calculation must be made to the right side of the fault, yet m (which has a different value here) is subtracted. When faults are curved and are fit by polynomials the value m is simply the sum of many small areas.

Sub-areas for individual sections are next summed up across faults and discontinuities to determine the total area per bed within the cross section. Since we assume that the area of this bed remains constant before and after deformation, then, the original, undeformed length can be determined by relation (9) if the undeformed thickness of the bed is known. The value of the original thickness of the bed can be input by the user or digitized off of the plotter. The value for scale, input at the start of the program influences the technique to be used. If the true thickness of the bed is well constrained, the user should input the actual scale of the cross section at the start of the program and the undeformed length in relevant units and correct scale will be output. If the original bed thicknesses are not well constrained then the user can digitize the thickness of the layers directly from the cross section. Each layer thickness can be digitized a few times across the section to obtain an
average thickness. Using this method, the scale input at the start of the program should be 1. The resulting output will be in real units relative to the digitized cross section.

The geologist must judge whether a cross section is balanced by examining and comparing the computer generated pre-deformational bed lengths. The computer calculates the mean bed length and the standard deviation. The final error analysis compares the maximum bed length difference to the mean bed length. Ideally, all bedlengths should be exactly equal, or some plausible explanation should exist for a deviant bed. However, in a perfectly balanced section, slight digitizing errors can result in minor area errors. Woodward et al. (1985) discuss this phenomenon in detail. The computer allows for the user to change the thickness slightly to attain perfect balance. The computer calculates the standard deviation of the bed lengths and also a percent confidence. My experience shows that standard deviations in balanced sections with a scale of one should lie in the $10^{-5}$ range. The percent confidence uses the calculation:

$$\text{%confidence} = \frac{\text{maximum bed length} - \text{minimum bed length}}{\text{mean bed length}} \times 100$$

(10)

the values of %confidence should be greater than 97% to consider a cross section balanced.
Bed length Balance

Sections which exhibit area balance may be tested for bed length balance as well if the folding is concentric. In regions of similar folding bed length balance is inappropriate because the length of a layer before and after similar folding does not remain constant. Bed length balance requires significantly fewer calculations and has a lower probability of error.

This routine sums up the distances of the individual straight line segments connecting digitized points by segment for an entire curved contact offset by faults. If the curve has been fit by a polynomial then a large number of significantly shorter segments are summed with a more accurate result. Individual bed lengths should be equal. In addition, as a final check, the bedlength balance technique should yield a mean length with a value similar to the mean length determined in the area balance routine.

Tests for bedlength balance generate values for strain as well for each contact length. Strain ($\varepsilon$) is calculated independently with the formula:

$$\varepsilon = \frac{L - L_0}{L_0}$$

where $L$ is the length of the deformed bed in cross section, and $L_0$ is the unfolded and unfaulted length.
Palinspastic Reconstruction

Retrodeforming a cross section is a final technique used to balance cross sections. The palinspastic reconstruction is basically a view of the undisturbed stratigraphic wedge with the predeformational insipient fault geometries sketched in. Cross sections which are not balanced will contain fault surfaces which do not adequately describe a deformation path, showing vertical faults and faults showing an incorrect dip.

The retrodeforming procedure in the program BALANCE first draws out the flat stratigraphic wedge using the lengths and thicknesses calculated earlier. For viewing purposes, the mean length may be plotted for all beds. Any cross sections which are retrodeformed must already exhibit area balance or the retrodeformation is meaningless. Faults are drawn in assuming that the bed area between two faults remains constant before and after deformation. This is the only requirement affecting the resulting geometry of the reconstructed wedge.

Relative contact lengths between faults are not taken into account nor is the deformed fault shape or hanging wall and foot wall cutoff lengths. These "sub-areas" are calculated for each layer between each fault and the computer calculates each "sub-length" using relation (9). If rectangles containing the correct areas are plotted, the result could by stepping faults (Fig. 11a). Instead, to give the fault a curved surface, a polynomial function is fit which goes through a
point on each layer halfway through the thickness of each layer at the edge of the rectangle (Fig. 11b). The user should choose polynomials of order 3 or 4 depending on the number of strata in the wedge.

The resulting palinspastic reconstruction (Fig. 11c) must be further interpreted by the user. Fault surfaces must be extended to the top and bottom of the wedge, and in some cases the curvature must be reduced (Fig. 11d). In addition, the user must selectively rule out and redraw some surfaces as the computer may not have enough information to draw the correct surface.
Redigitizing Sections

Most cross sections, after a first check do not pass the requirements for area balance. As a result, subtle changes are generally needed to obtain balance. This computer program interacts with the user to obtain satisfactory changes in the cross section configuration.

If changes are desired, the entire cross section is drawn onto the plotter with the location of every point which was originally digitized by the user starred and numbered. These digitized points can be moved, one at a time to change the bed shape. In addition, points can be added. The user in this case enters the location numbers of the points to the left and right of the location for the new point, then digitizes the point. After a single change has occurred, the cross section may be redrawn and renumbered. This is especially useful if points have been added, as the numbering sequence changes slightly.

When cross section changes have been completed, binary data files are updated to reflect the changes and renewed tests for balance can occur. The procedure must be entirely reactivated starting with fitting polynomials to sections.
Figure 12 Flowchart for THREEDIM.
THREEDIM

The main object of the program THREEDIM is to set up a three-dimensional grid defining a volume of deformed rock. The geologic map forms the 2-D base of this grid and the vertical cross sections, in their proper positions along the map create the third dimension in space. In this way, the relationships between a number of cross sections can be interpreted and the implications for intermediate cross sections can be examined.

The grid itself does not contain complete cross sections. Instead, the user chooses key points on the cross section which can aid eventually in the construction of other intermediate sections. The next program, PROJECTION, takes these points and projects them along structural trend and plunge to desired cross section locations. These key markers should include hinge and trough locations, inflection points, and hanging wall and footwall cutoffs at fault boundaries.

Therefore, the user must initially compile a list of "projectable" points from all of the cross sections on the map. This list should include key points common to all cross sections in addition to key points on only one or more sections. This entire list must be entered, in the same order, each time THREEDIM is re-run for a cross section from this map. This process helps to avoid data confusion later on as THREEDIM is re-run for successive cross sections. The next step is to digitize the location of one section on
the map to be included in the grid. Because this program is rerun for each cross section, the location of the map on the plotter cannot change or future projections will be incorrect.

A binary data file containing x and y coordinate values of every point on the cross section is next read by the computer and the geologic cross section is projected on to the CRT display. This file must have previously been created by the program BALANCE. It is assumed that any cross sections placed in the grid have been tested successfully for balance.

There are two types of files which can be retrieved in this step. The simplest file contains only points which were digitized off of the original cross section. This file which has "Bal" concatenated to the file name was created and modified by BALANCE and is intended basically for use with that program, yet is sufficient for use here if folding is minimal. As an alternative, and a necessity in complexly folded regions, a file containing polynomial fits to the data may be used. This file with "Poly" appended to the name was created by BALANCE near the end of the program for specific use by THREEDIM.

With a given cross section in full view, the geologist can selectively choose those aspects of the geometry which are deemed critical to the three dimensional picture. These points are easily digitized on the screen using the knob on the keyboard. If inflection points are to be included, the
computer can automatically locate the correct spots. In fact, a binary file, with "Infl" concatenated to the file name, containing all inflection point data was created by the program BALANCE after polynomial curves were determined. The user simply enters the segment number containing an inflection point, and the actual location is marked on the screen.

The first point digitized, however, must correspond to the left endpoint of the line of section on the map. This is the "reference point" and its elevation must also be entered.

Each successive point to be digitized now has a corresponding letter assigned to it. Pressing the soft-key labeled "menu" provides a list of all points which can be digitized and their associated letter. When a point is digitized, the user enters this letter equivalent which is then used to label the point on the screen. It should be emphasized that every element on the list does not need to be digitized, as some elements may not exist on the cross section under review.

When digitizing is completed, a file with "3_D" concatenated to the user input file name, containing all grid data is created for use in the program PROJECTION. Successive cross sections from the same map should be digitized at this point.
Project these points defining location of structural features from cross section J to new line of section

Calculate weighted average for each structure from scatter of points

Plot weighted average points defining a new cross section

End

Figure 13 Flowchart for PROJECTION
The final program, PROJECTION helps to constrain the geometries and study the structural features in additional, randomly oriented cross sections. The points in data files created by THREEDIM are projected along their structural trend and plunge and are then plotted along the desired line of section. A best fit point is then determined for points projected from two or more cross sections. One goal, then, is to examine whether an intermediate cross section, oriented parallel to orogenic strike, which is derived from two or more balanced cross sections, is itself balanced. This algorithm is also useful for viewing structures off the line of section and at other orientations.

The first step is for the user to enter the scale of the map and cross sections and the names of the files created by THREEDIM which contain the digitized data for the individual cross sections. The section line defining the location of each of the cross sections is displayed on the CRT. At this point, the user chooses a new line of section and has the option of digitizing its endpoints on the plotter, if the map is still in the same spot, or CRT.

Individual data files are read in turn. After a file for a given cross section is read, the data points (which are locations of specific geologic structures) are projected along the structural trend and plunge to their respective positions along the desired line of section. This new
section, containing only data from the most recently projected cross section is then rotated to be parallel with the base of the plotter and finally drawn onto the CRT for viewing purposes. Axes, elevations, and tickmarks are also included on the plot. This same procedure is repeated for successive cross sections. In the end, a scatter of points defines the general location for each important structural point on the new line of section.

For each point which has been projected from a number of cross sections, a best fit point must be determined for the intermediate section. Therefore, the projected distance of each and every point from its cross section to the intermediate section is calculated. The average x and y coordinate value for the resulting point is not calculated because data points projected across larger distances have greater uncertainty. Instead, a weighted average is determined so that cross sections closer to the intermediate section have a greater influence on the resulting location of the structural data point, and therefore on the resulting geometry of the intermediate cross section. These weighted average locations are all plotted onto a cross section along with their corresponding label.

In the final step, the geologist is left without the aid of the computer and must interpret the finer details of the underlying structures using this plot of points in an effort to determine more conclusively the geometrical relationships in the intermediate cross section.
Example From the Canadian Front Ranges

The Front Ranges of the Canadian Cordillera are a heavily explored region due to their petroleum resources and thus provide a good base to test these programs. The Front Ranges are comprised of north-south trending linear mountains which are composed of west dipping thrust sheets. The McConnel thrust is the primary detachment and subsequent thrusts, Exshaw, Lac Des Arcs, Rundle and Sulfer Mountain converge and finally join the McConnel thrust at depth. The displacement (along strike) is balanced by transfer from one fault to another (Price and Mountjoy, 1970). Therefore, this region is also interesting to study for three dimensional geometries.

My goal was to balance the volume of rock in the Cranmore area of the Canadian Front Ranges (Fig. 14). This area has been mapped and interpreted by Price and Mountjoy (1970) and was reinterpreted by Price and Fermor (1984). Four cross sections were chosen A-A' thru D-D' oriented perpendicular to orogenic strike. The program SECTION was used to draw cubic splines to fit the topography. Cross sections were obtained from the geologic map and iteration. These four cross sections were tested for balance using BALANCE and all required adjustments to finally attain area balance, after which they were retrodeformed. The cross sections, along with their inferred palinspastic reconstructions are shown in figures 15a-d.
Figure 14 Map showing location of cross sections.
The technique used to balance these sections was very straight forward. Each cross section was split into two separate sections to fit on the plotter and were individually balanced using BALANCE. In the right half of cross section C-C' for example, 118 points were digitized along the contacts (Fig. 16). Because the beds are not flat lying on the right edge of the section, an imaginary fault was drawn in to bound the beds. Files "Balcan4b" and and "IIcan4b" were created to store the digitized data. Polynomial functions were determined for the beds only because faults in this section are generally oddly shaped and polynomial functions can not accurately describe their geometry. Bed areas were calculated and thicknesses were digitized off the plotter. Redigitizing some areas was necessary to attain perfect balance (Fig. 16). This altered cross section was tested for balance with much better results (Fig. 17). This series of steps was repeated for each cross section.
DATA FROM CROSS SECTION CAN4B

LISTING OF LAYER AREAS AND TOTAL CROSS SECTION AREAS
-------------------------------------------------------------
( 1 ) .003758504375 SQUARE METERS
( 2 ) .0037621921875 SQUARE METERS
( 3 ) .003407818125 SQUARE METERS
TOTAL AREA= .0109285146875 SQUARE METERS

LISTING OF LAYER LENGTHS AND THICKNESSES
---------------------------------------------
UNDEFORMED LENGTH OF LAYER 1 WAS .377 METERS
THICKNESS .00997 METERS
UNDEFORMED LENGTH OF LAYER 2 WAS .372 METERS
THICKNESS .0101 METERS
UNDEFORMED LENGTH OF LAYER 3 WAS .372 METERS
THICKNESS .00917 METERS

MEAN LENGTH IS .374 METERS
SAMPLE STANDARD DEVIATION= 8.39E-6 METERS
PERCENT CONFIDENCE=-99.3654653048%

Figure 17 Computer output from cross section 1-1'

Cross sections A-A' and B-B' were used next to attain the geometries along a new line of section, 1-1'. Using THREEDIM, data points from A-A' and B-B' were projected using PROJECTION to the new line of section. On each cross section, points were chosen from areas of structural complexity (Fig. 18a&b). These locations were digitized using THREEDIM and were stored in files "3_Dcana" and "3_Dcanb". Next, using PROJECTION the points were individually projected along trend and plunge to the location of cross section 1-1' (Fig. 14). A weighted average location was calculated then to determine the best location for each structural data point on the new line of section (Fig 18c). The underlying geology along cross section 1-1' was then interpreted using surface geology and the
weighted average fit to each data point. My geological interpretations were required to pass through any projected points. The final cross section was tested for balance and passed tests for area balance without any required changes.

Figure 18 a&b) points projected using THREEDIM. c) resulting cross section.
A second exercise attempted to reinterpret a geologic section of Price and Fermor, (1984). Their section, figure 19a and labeled 2-2' on figure 14 was tested for balance repeatedly. Minor changes and redigitizing did not help in attaining balance, and their interpretation was deemed unbalanceable. To see if another slightly different interpretation would work, I used cross sections C-C' and D-D' to infer the geometries along section 2-2'. The second geological interpretation (Fig. 19b) was obtained from the use of THREEDIM and PROJECTION. This section was tested for balance using BALANCE and was determined to be area balanced.

These two interpretations of sections 2-2' show most dissimilarities in the strata projected above the topography. In the interpretation of Price and Fermor there is an excess area in the uppermost lithologic unit. My interpretation of the strata above the topography eliminates some of this unit. In addition, the two horse structures show slightly differing geometries, depths, size, and ratios of different lithologic units. All of my units contain inflection points inferred from sections C-C' and D-D'. The strata in the section of Price and Fermor are generally not curved. Finally, the main detachment in the Price and Fermor section obtains depths of greater than 5000m. This results in an excess of the lowest stratigraphic unit to accommodate for the extra space. My interpretation shows the detachment only to depths of slightly less than 5000m.
Figure 19  Two interpretations of section 2-2.'
CONCLUSIONS

This software package was designed to aid the interpretation skills of the structural geologist. Continually checking a proposed section for balance imposes more realistic constraints on the geologist's interpretation. This automated approach to drawing, balancing, and retrodeforming allows immense time savings when only slight modifications in geometry are needed. Also, the programs' ability to project a theoretical section in between two previously balanced sections is a good start to interpreting new areas. Finally, the examples from the Canadian Cordillera show that these programs not only work for theoretical models, but also can be valuable tools in complex geologic situations.
References

Bally, Gordy, and Stewart, 1966, Structure, seismic data on orogenic evolution of the southern Canadian Rocky Mountains, Bull. of Canadian Petroleum Geology, 14, p. 337-381.


Ollerenshaw, N.C., 1978, Geology, Calgary, West of Fifth Meridian, Alberta and British Columbia: Geologic Survey of Canada map 1457A.


10 ! ***********************************************************
20 ! *** SECTION ***
30 ! ***********************************************************
40 DIM Hh(1:150),Xx(1:150),Ii(300),Jj(300)
50 COM /Comdata/ R,C,D,E,F,Direct,Scal,Xx(1:150),Hh(1:150),O(1500),Oo(1500),Aa(1:40),Bb(1:40),Yy(1:600),X2(25),App(25),Unit(25),Dip$(25)[1],Str(25)
60 COM /Geodata/ X1(25),Y1(25),Strike$(25)[1],Dp(25),Tr(25),Trend$(25)[1],Pl(25),Plunge$(25)[1],Elev(25),Atype(25)
70 PRINTER IS 1
80 GINIT
90 GRAPHICS ON
100 GRAPHICS INPUT IS 705, "HPGL"
110 PLOTTER IS 705, "HPGL"
120 PEN 0
130 PRINT
140 PRINT
150 PRINT "*****************************************************************
160 PRINT "* OPTION ONE * OPTION TWO *"
170 PRINT "* DIGITIZE TOPOGRAPHY (0) * DRAW TOPOGRAPHY (5) *"
180 PRINT "* DIGITIZE GEOLOGY (1) * DRAW FOLD (6) *"
190 PRINT "* DIGITIZE FOLD (2) * DRAW AXIAL PROF (7) *"
200 PRINT "* CREATE WELL FILE (3) * DRAW CROSS SECTION (8) *"
210 PRINT "*******************************************************************
220 ON KEY 0 LABEL "TOPODIGIT" GOTO 390
230 ON KEY 1 LABEL "GEODIGIT" GOTO 450
240 ON KEY 2 LABEL "FOLDDIGIT" GOTO 480
250 ON KEY 3 LABEL "WELLFILE" GOTO 620
260 ON KEY 4 LABEL " " GOTO 270
270 ON KEY 5 LABEL "TOPODRAW" GOTO 420
280 ON KEY 6 LABEL "FOLDDRAW" GOTO 510
290 ON KEY 7 LABEL "AXIAL" GOTO 590
300 ON KEY 8 LABEL "CROSSSECTION" GOTO 550
310 ON KEY 9 LABEL " " GOTO 270
320 GOTO Spin1
330 CALL To odigit(Topofile,New$)
340 GOTO 130
350 !
360 CALL Topodraw(Topofile,New$,Ntotpts,Topo,Mm,Ii(*),Jj(*),L)
370 GOTO 130
380 !
390 CALL Geodigit(Geofile,Nice$,C)
400 GOTO 130

GOTO 130

CALL Folddraw(Foldfile,Name$,Ndat,Fold)
GOTO 130

GOSUB 670

CALL Crs(Ntotpts,Ndat,Npts,Fold,Topo,Geo,Xxx(*),Hhh(*),G,Ii(*),Jj(*),L)
GOTO 130

CALL Axial(Scal,Foldfile,Name$,O(*),Oo(*))
GOTO 130

LOADSUB Wellfilel FROM "EXTRASUBS"
CALL Wellfilel
DELSUB Wellfilel
GOTO 130
STOP

DEG

!**********************************************************************

Topodigit:SUB Topodigit(Topofile,New$)
DIM X(40),X2(1:40),H(1:40),H2(40),Y(40)
PRINT "START DIGITIZING TOPOGRAPHY FROM LEFT TO RIGHT"
PRINT "WHEN DONE DIGITIZING PRESS 1 FOR THE ELEVATION"
M=1
LOOP
PRINT "ENTER ELEVATION OF POINT",M
INPUT H(M)
EXIT IF H(M)=1
M=M+1
END LOOP
M=M-1
Ndat=M
FOR M=1 TO Ndat
PRINT "DIGITIZE POINT",M,"ELEVATION IS";H(M)
DIGITIZE X(M),Y(M)
BEEP
NEXT M
R=ATN((ABS(Y(Ndat)-Y(1))/ABS(X(1)-X(Ndat))))
FOR M=1 TO Ndat
X2(M)=X(M)/COS(R)
NEXT M
INPUT "WHAT WOULD YOU LIKE TO CALL THE NEW FILE?",New$
CREATE BDAT "TOPO"&New$,1,700
ASSIGN @File TO "TOPO"&New$
SUBEND

!**************************************************************************

Folddigit:SUB Folddigit(Foldfile,Name$)

J=0

INPUT "WHAT WOULD YOU LIKE TO CALL THE NEW FILE?",Name$

CREATE BDAT "FOLD"&Name$,1,40000

ASSIGN @File TO "FOLD"&Name$

DIM X(150),Y(150),H(150),O(150),Oo(105),X1(150),Y1(150),H1(150)

J=1

PRINT "START DIGITIZING FOLD. WHEN DONE DIGITIZING, PRESS 1 FOR ELEVATION"

LOOP

PRINT "ENTER ELEVATION OF POINT",J

INPUT H(J)

EXIT IF H(J)=1

PRINT "POINT";J;"ELEVATION";H(J);"METERS"

J=J+1

END LOOP

J=J-1

Ndat=J

FOR J=1 TO Ndat

PRINT "DIGITIZE POINT";J;"ELEVATION IS";H(J)

DIGITIZE X(J),Y(J)

IF J=1 THEN GOTO 1480

IF X(J)=X(J-1) THEN

IF Y(J)=Y(J-1) THEN

PRINT "YOU JUST DIGITIZED THIS POINT TWICE!"

BEEP 651,1

PRINT "DIGITIZE THE CORRECT POINT THIS TIME"

GOTO 1360

ELSE

BEEP

END IF

ELSE

BEEP

END IF

PRINT "LOCATION OF POINT";J;"IS",X(J),Y(J)

NEXT J

PRINT "TOTAL NUMBER OF POINTS IS";Ndat

OUTPUT @File;X(*),Y(*),H(*)

Foldfile=1

SUBEND

!**************************************************************************

Folddraw:SUB Folddraw(Foldfile,Name$,Ndat,Fold)

COM /Comdata/ R,C,D,E,F,Direc,Scal,Xx(1:150),Hh(1:150),O(150),Oo(150)

Aa(1:40),Bb(1:40),Yy(1:600),X2(25),App(25),Unit(25),Dip$(25),Str(25)

DIM X(150),Y(150),H(150),Ytrend(150)

DIM Ynew(150),Ycross(150),Xdif(150),Hplot(150)

DIM Newx3(150),Hhh(1:150),Xxx(1:150)

DEG

IF Foldfile<>1 THEN INPUT "ENTER NAME OF FILE TO BE READ",Name$

INPUT "ARE ELEVATIONS IN METRIC(1) OR FEET(2)?",El

ASSIGN @File TO "FOLD"&Name$

J=0

ENTER @File;X(*),Y(*),H(*)

ON END @File GOTO 1690

LOOP

J=J+1

EXIT IF X(J)=0

END LOOP

Ndat=J-1

FOR J=1 TO Ndat

IF El=2 THEN

H(J)=H(J)*.3048
FOR J=1 TO Ndat
    H(J)=H(J)*400/(Scal)
NEXT J

J=1
INPUT "ENTER THE PLUNGE OF THE STRUCTURE", P
INPUT "IS PLUNGE TO THE NORTH OR SOUTH", Plungdir$
IF Chng=1 THEN GOTO 1880
INPUT "ENTER THE TREND OF THE STRUCTURE", T
INPUT "IS THE TREND TO THE EAST OR WEST?", Trendir$
J=1

PRINTER IS 1
FOR J=1 TO Ndat STEP 1
    IF D<F THEN Ycross(J)=TAN(R)*(X(J)-C)+D
    IF D>F THEN Ycross(J)=TAN(R)*(E-X(J))+F
    IF D=F THEN Ycross(J)=F
    IF Ycross(J)>Y(J) THEN Ynew(J)=Ycross(J)-Y(J)
    IF Ycross(J)<=Y(J) THEN Ynew(J)=Y(J)-Ycross(J)
NEXT J

B=90-T
FOR J=1 TO Ndat
    Ytrend(J)=Ynew(J)*SIN(Direc)/SIN(B+R)
    IF Ytrend(J)<0 THEN Ytrend(J)=ABS(Ytrend(J))
    Hh(J)=Ytrend(J)*TAN(P)
    IF Ycross(J)>Y(J) THEN Hh(J)=-1*Hh(J)
    Xdif(J)=Ytrend(J)*SIN(T)
    IF Y(J)>Ycross(J) THEN
        IF Trendir$="E" THEN
            Xx(J)=X(J)-Xdif(J)
        ELSE
            Xx(J)=X(J)+Xdif(J)
        END IF
    ELSE
        IF Trendir$="E" THEN
            Xx(J)=X(J)+Xdif(J)
        ELSE
            Xx(J)=X(J)-Xdif(J)
        END IF
    END IF
NEXT J

FOR J=1 TO Ndat
    Xx(J)=Xx(J)/COS(R)
NEXT J

IF Fold=1 THEN SUBEXIT
CALL Minn(Minxx,Xx(*),Ndat)
CALL Minn(Minhh,Hh(*),Ndat)
INPUT "DO YOU WANT A POLYNOMIAL FIT? Y=1", Polyn
IF Polyn=2 THEN
    CALL Ticketc(Xx(*),Hh(*),Minxx,MAX(Xx(*))-Minxx,Minhh,Scal,Ndat)
ELSE
    CALL Interval(Xx(*),Hh(*),Minxx,MAX(Xx(*))-Minxx,Minhh,Scal,Npts)
    CALL Ticketc(0(*),Oo(*),Minxx,MAX(Xx(*))-Minxx,Minhh,Scal,Npts)
END IF

LOADSUB Interval FROM "EXTRASUBS"
CALL Interval(Xx(*),Hh(*),Minxx,Minhh,Ndat,Xx(*),Hh(*))
DELSUB Interval
CALL Poly(Xxx(*),Hhh(*),Ndat,0(*),Oo(*),Npts)
CALL Ticketc(O(*),Oo(*),Minxx,MAX(Xx(*))-Minxx,Minhh,Scal,Npts)
END IF
PENUP
PEN 0
PRINT "DO YOU WANT TO CHANGE THE PLUNGE=1 OR TREND=2"
INPUT "OR BOTH=3 OR NONE=4?", Chng
IF Chng=1 THEN GOTO 1830
IF Chng=2 THEN GOTO 1860
IF Chng=3 THEN GOTO 1830
IF Chng=4 THEN GOTO 2430
2450 Topodraw:SUB Topodraw(Topofile,New$,Ntotpts,Topo,Mm,Jj(*),L)
2460 COM /Comdata/R,C,D,E,F,Direct,Scal,Xx(1:150),Hh(1:150),Oo(1500),
2470 Aa(1:40),Bb(1:40),Yy(1:600),Xz(25),App(25),Unit(25),Dip$(25)[1],Str(25)
2480 DIM Aaa(1:600),Ss(40)
2490 J=0
2500 IF Topofile<>1 THEN INPUT "ENTER NAME OF FILE CONTAINING DIGITIZED TOPOGRAPHY",New$
2510 INPUT "ENTER SCALE",Scal
2520 ASSIGN @File TO "TOPO"&New$
2530 Topofile=1
2540 ENTER @File;Aa(*),Bb(*)
2550 ON END @File GOTO 2560
2560 LOOP
2570 J=J+1
2580 EXIT IF Aa(J)=0
2590 END LOOP
2600 Ntotpts=J-1
2610 FOR J=1 TO Ntotpts
2620 IF E1=2 THEN
2630 Bb(J)=Bb(J)*.3048
2640 END IF
2650 NEXT J
2660 FOR J=1 TO Ntotpts
2670 Bb(J)=Bb(J)*400/Scal
2680 NEXT J
2690 IF Topo=1 THEN SUBEXIT
2700 CALL Minn(Minbb,Bb(*),Ntotpts)
2710 CALL Minn(Minaa,Aa(*),Ntotpts)
2720 IF Topo=2 THEN GOTO 2855
2730 INPUT "DO YOU WANT A CUBIC SPLINE FIT TO TOPOGRAPHY? Y=1",Sp
2740 IF Sp=1 THEN
2750 LOADSUB Spline FROM "EXTRASUBS"
2760 CALL Spline(Aa(*),Bb(*),Ntotpts,Ss(*),Minaa,Minbb)
2770 DELSUB Spline
2780 END IF
2790 IF Topo=2 THEN SUBEXIT
2800 IF Sp<1 THEN
2810 INPUT "ENTER PEN",Penn
2820 PEN Penn
2830 FOR J=1 TO Ntotpts
2840 IF J=1 THEN MOVE Aa(J)-Minaa,Bb(J)-Minbb
2850 DRAW Aa(J)-Minaa,Bb(J)-Minbb
2860 NEXT J
2870 END IF
2880 CSIZE 2
2890 LORG 5
2900 MOVE (MAX(Aa(*))-Minaa),(MAX(Bb(*))-Minbb)/2
2910 LINE TYPE 9
2920 DRAW (MAX(Aa(*))-Minaa),MAX(Bb(*))-Minbb
2930 LINE TYPE 1
2940 MOVE MAX(Aa(*))-Minaa+5,MAX(Bb(*))-Minbb+1
2950 LABEL INT(MAX(Bb(*))*Scal/400);"meters"
2960 MOVE MAX(Aa(*))-Minaa,MAX(Bb(*))-Minbb
2970 LINE TYPE 9
2980 DRAW MAX(Aa(*))-Minaa,.1
2990 LINE TYPE 1
3000 MOVE MAX(Aa(*))-Minaa+5,1
3010 LABEL INT(Minbb*Scal/400);"meters"
3020 GCLEAR
3030 PEN 0
3040 SUBEND
3050 "****************************************************
3060 Geodigit:SUB Geodigit(Geofile,Nice$.C)
DIM Plunge$(25)[1], Elev(25), Atype(25)
MAT Atype = (0)
PRINT "DIGITIZE GEOLOGIC DATA IN ANY ORDER."
PRINT "UNLESS YOU PLAN TO DO A HERMITE FIT, THEN DIGITIZE"
PRINT "FROM LEFT TO RIGHT AS POINTS WILL PROJECT"
PRINT "STRIKE IS NORTH (ASSUMED) THEN SOMETHING"
PRINT "EAST OR WEST"
PRINT "DIP IS EITHER EAST OR WEST"
J = 0
LOOP
J = J + 1
PRINT "IS THIS A WELL(1), PROJECTED ALONG STRIKE(2), PROJECTED ALONG "
PRINT "PLUNGE(3), OR END OF DATA SET(4)"
INPUT Sym
EXIT IF Sym = 4
INPUT "ENTER STRIKE", Str(J)
INPUT "EAST OR WEST?", Strike$(J)
IF Sym = 1 THEN
    Dp(J) = 100
    GOTO 3420
END IF
INPUT "ENTER DIP", Dp(J)
INPUT "EAST OR WEST?", Dip$(J)
INPUT "ENTER NUMBER OF UNIT", Unit(J)
IF Sym = 2 THEN
    Tr(J) = 100
    GOTO 3420
END IF
INPUT "ENTER TREND", Tr(J)
INPUT "ENTER TREND DIRECTION", Trend$(J)
INPUT "ENTER PLUNGE", Pl(J)
INPUT "ENTER PLUNGE DIRECTION", Plunge$(J)
INPUT "ENTER ELEVATION", Elev(J)
END LOOP
Ndat = J - 1
PRINT "YOU HAVE"; Ndat; "POINTS TO DIGITIZE"
FOR J = 1 TO Ndat
    PRINT "DIGITIZE LOCATION OF POINT"; J, Str(J), Strike$(J), Dp(J), Dip$(J)
    DIGITIZE X1(J), Y1(J)
    IF X1(J) = X1(J - 1) THEN
        BEEP 651, 1
        PRINT "YOU DIGITIZED A POINT TWICE, PRESS 1 IF THIS WAS INCORRECT"
        Totimes = 1
    ELSE GOTO 3460
    IF Y1(J) = Y1(J - 1) THEN
        BEEP 651, 1
        PRINT "YOU DIGITIZED A POINT TWICE, PRESS 1 IF THIS WAS INCORRECT"
        Totimes = 1
    ELSE
        PRINT "CONTINUE THEN"
    END IF
    ELSE BEEP
    END IF
NEXT J
PRINT "WHAT WOULD YOU LIKE TO CALL THE NEW FILE?", Nice$
CREATE BDAT "GEO" & Nice$, 1, 2800
ASSIGN @File TO "GEO" & Nice$
OUTPUT @File; X1(*), Y1(*), Str(*), Strike$(*), Dp(*), Dip$(*), Unit(*), Tr(*), Trend$(*), Pl(*), Plunge$(*), Elev(*), Atype(*)
Geofile = 1
SUBEND
3730 COM /Comdata/ R,C,D,E,F,Direc,Scal,Xx(1:150),Hh(1:150),Oo(1500),
   As(1:40),Bb(1:40),Yy(1:600),Aa(25),App(25),Unit(25),Dip$(25)[1],Str(25)
3740 COM /Geodata/ X1(25),Y1(25),Strike$(25)[1],Dp(25),Tr(25),Trend$(25)[1],Pr
   vdepth$[1,1],Elev(25),Atype(25)
3750 DEG
3760 DIM X5(25),Array(25)
3770 DIM Crossy(25),Newy(25),Trendy(25),Difx(25),B(25),Ang(25),Array2(25)
3780 IF Geo=0 THEN GOTO 5110
3790 IF Geofile<>1 THEN
3800 INPUT "ENTER NAME OF FILE CONTAINING GEOLOGIC DATA",Nice$
3810 Geofile=1
3820 ASSIGN @Road TO "GEO"&Nice$
3830 ON END @Road 3840
3840 J=0
3850 LOOP
3860 J=J+1
3870 EXIT IF X1(J)=0
3880 END LOOP
3890 Npts=J-1
3900 J=0
3910 Prvdepth=10000000
3920 LOOP
3930 J=J+1
3940 IF Dp(J)=100 THEN Wells=1
3950 IF Wells=1 THEN
3960 INPUT "ENTER WELL FILE NAME",Wellname$
3970 LOADSUB Well3 FROM "EXTRASUBS"
3980 CALL Well3(Wellname$,Mindepth,Dip$(25),Dp(25),J,Npts,Scal,X1(J),Y1(J),
   Unit(25),Str(J),Strike$(J),Atype(25),Plunge$(25),Pl(J),Trend$(25),Tr(J))
3990 DELSUB Well3
4000 Wells=0
4010 IF Mindepth<Prvdepth THEN Prvdepth=Mindepth
4020 END IF
4030 EXIT IF J=Npts
4040 END LOOP
4050 IF Prvdepth=10000000 THEN Prvdepth=0
4060 FOR J=1 TO Npts
4070 IF Strike$(J)="E" THEN Tick=1
4080 IF Trend$(J)="E" THEN Tock=1
4090 IF Tick=1 THEN Clock=1
4100 IF Tock=1 THEN Clock=1
4110 IF D£F THEN Crossy(J)=TAN(R)*(X1(J)-C)+D
4120 IF D£F THEN Crossy(J)=TAN(R)*(E-X1(J))+F
4130 IF D£F THEN Crossy(J)=F
4140 IF Crossy(J)>Y1(J) THEN Newy(J)=Crossy(J)-Y1(J)
4150 IF Crossy(J)<Y1(J) THEN Newy(J)=Y1(J)-Crossy(J)
4160 IF Tr(J)=100 THEN
4170 B(J)=90-Str(J)
4180 ELSE
4190 B(J)=90-Tr(J)
4200 END IF
4210 IF D>£F THEN
4220 IF Clock=1 THEN
4230 Trendy(J)=Newy(J)*SIN(Direc)/SIN(B(J)+R)
4240 ELSE
4250 Trendy(J)=Newy(J)*SIN(90+R)/SIN(Direc-Str(J))
4260 END IF
4270 ELSE
4280 IF Clock=1 THEN
4290 Trendy(J)=Newy(J)*SIN(90+R)/SIN(B(J)-R)
4300 ELSE
4310 Trendy(J)=Newy(J)*SIN(Direc)/SIN(B(J)+R)
END

IF Trendy(J)<0 THEN Trendy(J)=ABS(Trendy(J))
Difx(J)=Trendy(J)*COS(B(J))
IF Y1(J)>Crossy(J) THEN
  IF Clock=1 THEN
    X2(J)=X1(J)-Difx(J)
  ELSE
    X2(J)=X1(J)+Difx(J)
  END IF
ELSE
  IF Clock=1 THEN
    X2(J)=X1(J)+Difx(J)
  ELSE
    X2(J)=X1(J)-Difx(J)
  END IF
END IF

END IF
NEXT J
FOR J=1 TO Npts
X2(J)=X2(J)/COS(R)
END FOR J=1 TO Npts

IF D>F THEN
  IF Strike$(J)="W" THEN Ang(J)=90-Str(J)-R
  IF Strike$(J)="E" THEN
    IF R<Str(J) THEN
      Ang(J)=90+R-Str(J)
    ELSE
      Ang(J)=90+Str(J)-R
    END IF
  ELSE
    Ang(J)=90+R-Str(J)
  END IF
END IF

ELSE
  IF Strike$(J)="E" THEN Ang(J)=90-Str(J)-R
  IF Strike$(J)="W" THEN
    IF Str(J)<R THEN
      Ang(J)=90+R-Str(J)
    ELSE
      Ang(J)=90+R-Str(J)
    END IF
  ELSE
    Ang(J)=90+R-Str(J)
  END IF
END IF

NEXT J
FOR J=1 TO Npts
  IF Dp(J)>90 THEN App(J)=90
  IF Ang(J)>90 THEN GOTO 4870
  IF Dp(J)<90 THEN App(J)=ATN(TAN(Dp(J))*SIN(ABS(Ang(J))))
END FOR J=1 TO Npts

IF Wow=1 THEN GOTO 5110
FOR J=1 TO Npts
  IF Elev(J)<0 THEN
    Elev(J)=Elev(J)*400/Scal
  IF Crossy(J)>Y1(J) THEN
    IF Plunge$(J)="N" THEN
      Cc(J)=Elev(J)-ATN(SIN(Pl(J))/Trendy(J))
    ELSE
      Cc(J)=Elev(J)+ATN(SIN(Pl(J))/Trendy(J))
    END IF
END IF
END FOR J=1 TO Npts
4990 IF Plunge$(J)="N" THEN  
5000 Cc(J)=Elev(J)+ATN(SIN(P1(J))/Trendy(J))  
5010 ELSE  
5020 Cc(J)=Elev(J)-ATN(SIN(P1(J))/Trendy(J))  
5030 END IF  
5040 END IF  
5050 ELSE  
5060 IF Atype(J)=0 THEN Cc(J)=Yy(INT(X2(J)-C+1))  
5070 IF Atype(J)<0 THEN Cc(J)=Atype(J)  
5080 END IF  
5090 NEXT J  
5100 IF Geo=1 THEN SUBEXIT  
5110 INPUT "ENTER PEN FOR GEOLOGIC SYMBOLS",Colour  
5120 PEN Colour  
5130 LORG 5  
5140 CSIZE 1  
5150 FOR J=1 TO Npts  
5160 MOVE X2(J)-C,Cc(J)-Ultraymin+10  
5170 LABEL "="  
5180 MOVE X2(J)-C,Cc(J)-Ultraymin+10  
5190 IF Dip$(J)="E" THEN  
5200 DRAW X2(J)-C+2*SIN(90-App(J)),Cc(J)-Ultraymin-2*COS(90-App(J))+10  
5210 ELSE  
5220 DRAW X2(J)-C-2*SIN(90-App(J)),Cc(J)-Ultraymin-2*COS(90-App(J))+10  
5230 END IF  
5240 IF X2(J)=X2(J-1) THEN  
5250 MOVE X2(J)-C,Cc(J-1)-Ultraymin+10  
5260 DRAW X2(J)-C,Cc(J)-Ultraymin+10  
5270 IF X2(J)<X2(J+1) THEN  
5280 LINE TYPE 9  
5290 DRAW X2(J)-C,Cc(J)-Ultraymin+10  
5300 LINE TYPE 1  
5310 END IF  
5320 END IF  
5330 NEXT J  
5340 INPUT "DO YOU WANT HERMITE FIT? Y=1",Her  
5350 IF Her=1 THEN CALL Hermite(X2(*),App(*),Unit(*),Npts,Dip$(*),Cc(*),C,Min height,0,Ultraymin)  
5360 INPUT "DO YOU WANT TO PUNT ANY OF THE GEOLOGIC SYMBOLS? Y=1",Punt  
5370 IF Punt<1 THEN GOTO 5620  
5380 INPUT "ENTER PEN TO LABEL GEOLOGIC SYMBOLS",Penn  
5390 PEN Penn  
5400 CSIZE 1  
5410 LORG 5  
5420 FOR J=1 TO Npts  
5430 MOVE X2(J)-C,Cc(J)-Ultraymin+11  
5440 LABEL J  
5450 NEXT J  
5460 PEN 0  
5470 INPUT "ENTER POINT NUMBER TO BE REMOVED",J  
5480 FOR K=J TO Npts  
5490 X2(K)=X2(K+1)  
5500 Cc(K)=Cc(K+1)  
5510 App(K)=App(K+1)  
5520 Unit(K)=Unit(K+1)  
5530 Dip$(K)=Dip$(K+1)  
5540 NEXT K  
5550 Npts=Npts-1  
5560 INPUT "DO YOU WANT TO REMOVE MORE? Y=1 ",Punt  
5570 IF Punt=1 THEN  
5580 GOTO 5380  
5590 ELSE  
5600 GOTO 5110  
5610 END IF  
5619 PEN 0
Cross:SUB Crs(Ntot,Ndat,Npts,Fld,Ge,Xxx(*),Hhh(*),G,Ii(*),Jj(*),L)

DIM Cc(25)

INPUT "WILL CROSSSECTION INCLUDE FOLD? Y=1",Fld
INPUT "WILL CROSSSECTION INCLUDE GEOLOGIC DATA? Y=1",Ge
INPUT "WILL CROSSSECTION INCLUDE TOPOGRAPHY?",Tp

!DEAL WITH FOLD FIRST
IF Fld=1 THEN
   CALL Folddraw(Foldfile,Name$,Ndat,Fld)
   CALL Minn(Minxx,Xx(*),Ndat)
   CALL Minn(Minhh,Hh(*),Ndat)
   LOADSUB Interval FROM "EXTRASUBS"
   CALL Interval(Xx(*),Hh(*),Minxx,Minhh,Ndat,Xxx(*),Hhh(*))
   DELSUB Interval
   GCLEAR
   CALL Poly(Xxx(*),Hhh(*),Ndat,0(*),Oo(*),Ndats)
   CALL Minn(Minoo,Oo(*),Ndats)
END IF

!DEAL WITH TOPO
IF Tp=1 THEN
   CALL Topodraw(Topofile,New$,Ntot,Tp,Mm,Ii(*),Jj(*),L)
   CALL Minn(Minbb,Bb(*),Ntot)
   CALL Minn(Minaa,Aa(*),Ntot)
   FOR J=1 TO Ntot
      IF Minaa<C THEN
         Aa(J)=Aa(J)-(Minaa-C)
      ELSE
         Aa(J)=Aa(J)+(C-Minaa)
      END IF
   NEXT J
END IF

!DEAL WITH GEO
IF Ge=1 THEN
   LOADSUB Spacing FROM "EXTRASUBS"
   CALL Spacing(Aa(*),Bb(*),Ntot,Yy(*),M)
   DELSUB Spacing
   CALL Geodraw(Nice$,Npts,1,Ultraymin,Geofile,Prvdepth,Cc(*),Minaa)
   CALL Hermite(X2(*),App(*),Unit(*),Npts,Dip$(*),Cc(*),C,Minheight,1,Ultraymin)
END IF

Minimum(1)=Minhh
Minimum(2)=Minbb
Minimum(3)=Minheight
Minimum(4)=Prvdepth*400/Scal
CALL Minn(Ultraymin,Minimum(*),4)

IF Ge=1 THEN CALL Geodraw(Nice$,Npts,0,Ultraymin,Geofile,Mindepth,Cc(*),Minaa)

IF Fld=1 THEN CALL Tick_etc(O(*),Oo(*),C,G,Ultraymin,Scal,Ndats)

INPUT "DO YOU WANT A CUBIC SPLINE FIT TO YOUR TOPOGRAPHY Y=1",Poly
IF Poly=1 THEN
   CALL Topodraw(Topofile,New$,Ntot,2,Mm,Ii(*),Jj(*),L)
   CALL Tick_etc(Ii(*),Jj(*),C,G,Ultraymin,Scal,L)
ELSE
   CALL Tick_etc(Aa(*),Bb(*),C,G,Ultraymin,Scal,Ntot)
END IF

END IF

Tp=0
Ge=0
Fld=0
SUBEND

!*************************************************************************
FOR Gg=1 TO Numb
    IF G(Gg)=0 THEN GOTO 6300
    IF Ming=0 THEN Ming=G(Gg)
    IF G(Gg)<Ming THEN Ming=G(Gg)
NEXT Gg

SUBEND

!**************************************************************************
Poly:SUB Poly(Ww(*),Uu(*),Ndat,O(*),Oo(*),M)
PRINT "ENTER ORDER OF POLYNOMIAL"
INPUT "",N
PRINT "THIS WILL TAKE A WHILE"
DIM Ab(200,200),Aprime(20,20),Solution(0:20,1)
DIM Aaa(20,20),A_inv(20,20),B(20,1),Z(15000)
REDIM Ab(1:Ndat,O:N-1)
REDIM Aprime(0:N-1,1:Ndat)
REDIM Solution(O:N-1,1)
REDIM Aaa(0:N-1,0:N-1)
REDIM A_inv(0:N-1,0:N-1)
REDIM Ww(1:Ndat)
REDIM Uu(1:Ndat)
REDIM B(0:N-1,1)
FOR J=1 TO Ndat
    FOR K=0 TO N-1
        Ab(J,K)=(Ww(J))^K
    NEXT K
    MAT Aprime=TRN(Ab)
    MAT Aaa=Aprime*Ab
    MAT A_inv=INV(Aaa)
    MAT B=Aprime*Uu
    MAT Solution=A_inv*B
    FOR Q=.25 TO MAX(Ww(*)) STEP .25
        Z(Q)=0
        FOR S=1 TO N
            Z(Q)=Solution(S-1,0)*Q^(S-1)+Z(Q)
        NEXT S
        Z(Q*100)=Z(Q)
    NEXT Q
    J=0
    FOR Q=MIN(Ww(*)) TO MAX(Ww(*)) STEP .25
        M=M+1
        O(M)=Q
        Oo(M)=Z(Q*100)
    NEXT Q
PRINT M
SUBEND

Axial:SUB Axial(Scal,Foldfile,Name$,O(*),Oo(*))
AXIAL PROJECTION
DIM X(150),Y(150),H(150),Ytrend(150),Aa(150),Bb(150)
DIM Ynew(150),Ycross(150),Xdif(150),Hplot(150)
DIM New(150),Xx(1500),Hh(1500),Hhh(1:150),Xxx(1:150)
DEG
IF Foldfile=1 THEN GOTO 6810
INPUT "ENTER NAME OF FILE TO BE READ",Name$
INPUT "ENTER THE SCALE OF THE MAP IN FORM (1:____)",Scal
ASSIGN @File TO "FOLD"&Name$
J=0
ENTER @File;X(*),Y(*),H(*)
ON END @File GOTO 6910
FOR J=1 TO 150
    IF X(J)=0 THEN
        Ndat=J-1
    GOTO 6970
---
FOR J=1 TO Ndat
    IF El=2 THEN
        H(J)=.3048*H(J)
    END IF
NEXT J
FOR J=1 TO Ndat
    H(J)=H(J)*400/(Scal)
NEXT J
J=1
INPUT "ENTER THE PLUNGE OF THE STRUCTURE",P
INPUT "IS THE PLUNGE TO THE NORTH OR SOUTH?",Plungdir$
INPUT "ENTER THE TREND OF THE STRUCTURE",T
INPUT "IS THE TREND TO THE EAST OR WEST?",Trendir$
Slope=COS(T)/SIN(T)
Newslope=-1*1/Slope
IF Trendir$="E" THEN
    Cc=0
    Ff=0
    Ee=150
    Dd=-1*(E-C)*Newslope
ELSE
    Cc=0
    Dd=0
    Ee=150
    Ff=(E-C)*Newslope+D
END IF
FOR J=1 TO Ndat STEP 1
    IF Dd<Ff THEN Ycross(J)=TAN(Newslope)*(X(J)-Cc)+Dd
    IF Dd>Ff THEN Ycross(J)=TAN(Newslope)*(Ee-X(J))+Ff
    IF Dd=Ff THEN Ycross(J)=Ff
    IF Ycross(J)>=Y(J) THEN Ynew(J)=Ycross(J)-Y(J)
    IF Ycross(J)<-Y(J) THEN Ynew(J)=Y(J)-Ycross(J)
NEXT J
B=90-T
FOR J=1 TO Ndat
    Ytrend(J)=Ynew(J)*COS(T)
    IF Ytrend(J)<0 THEN Ytrend(J)=ABS(Ytrend(J))
    Xdif(J)=Ytrend(J)*SIN(T)
    IF Y(J)>Ycross(J) THEN Ynew(J)=Ycross(J)-Y(J)
    IF Y(J)<Ycross(J) THEN Ynew(J)=Y(J)+Ycross(J)
NEXT J
CALL Minn(Minxx,Xx(*),Ndat)
CALL Minn(Minhh,Hh(*),Ndat)
INPUT "DO YOU WANT A POLYNOMIAL FIT? Y=1",Polyn
IF Polyn=2 THEN
    FOR J=1 TO Ndat
        IF J=1 THEN MOVE Xx(1)-Minxx,Hh(1)-Minhh
ELSE
LOADSUB Interval FROM "EXTRASUBS"
CALL Interval(Xx(*),Hh(*),Minx,Minh,Ndat,Xxx(*),Hhh(*))
DELSUB Interval
CALL PLSUB(Interval(*),Interval(*),Ndat,0(*),0o(*),Npts)
CALL Tick etc(0(*),0o(*),Minx,MAX(Xx(*))-Minx,Minh,Scal,Npts)
FOR J=1 TO Npts
IF J=1 THEN MOVE O(1)-Minx,0o(1)-Minhh
DRAW O(J)-Minxx,0o(J)-Minhh
NEXT J
END IF
SUBEND

!**************************************************************************
Tick:SUB Ticketc(Xarray(*),Yarray(*),C,G,Minyarray,Scal,Ndat)
Hard=0
PLOTTER IS 3,"INTERNAL"
LOOP
INPUT "ENTER LINE TYPE NUMBER ",Lin
LINE TYPE Lin
IF Hard=1 THEN
INPUT "ENTER PEN NUMBER",Penn
PEN Penn
ELSE
PEN 1
END IF
FOR J=1 TO Ndat
IF J=1 THEN MOVE Xarray(J)-C,(Yarray(J)-Minyarray)+10
DRAW Xarray(J)-C,(Yarray(J)-Minyarray)+10
NEXT J
LINE TYPE 1
IF Hard=0 THEN
CSIZE 5
ELSE
CSIZE 2
END IF
LORG 5
!DRAW AXIS HERE
MOVE 0,MAX(Yarray(*))-Minyarray+10
DRAW 0,0
DRAW G,0
DRAW G,MAX(Yarray(*))-Minyarray+10
LABEL AXIS HERE
MOVE G+8,MAX(Yarray(*))-Minyarray+10
LABEL INT(MAX(Yarray(*)))*Scal/400);"M"
MOVE G+8,10
LABEL INT((Minyarray)*Scal/400);"M"
!TICK MARKS DRAWN HERE
FOR K=1 TO (G)/4
MOVE K*4,.3
DRAW K*4,0
NEXT K
FOR K=1 TO (MAX(Yarray(*))-Minyarray+10)/4
MOVE G+.4*K
DRAW G+.3,4*K
NEXT K
!OPTION FOR HARDCOPY HERE
EXIT IF Hard=1
INPUT "DO YOU WANT HARD COPY? Y=1",Hard
EXIT IF Hard<>1
PLOTTER IS 705,"HPGL"
END LOOP
PEN 0
GCLEAR
SUBEND
Hermite:SUB Hermite(X2(*),App(*),Unit(*),Total,Dip$(*) ,Cc(*),Minheight,Click,Ultraymin)

PLOTTER IS 705,"HPGL"
DIM U(-1:100),Slp(25),Height(6000),Yunit(25)
MAT Yunit= Unit
MAT SORT Yunit
FOR J=1 TO 25
IF Yunit(J)>Yunit(J-1) THEN
Unum=Unum+1
ELSE
Unum=Unum
END IF
NEXT J
FOR J=1 TO Total
Slp(J)=COS(90-App(J))/SIN(90-App(J))
IF Dip$(J)="E" THEN Slp(J)=-1*Slp(J)
NEXT J
FOR J=1 TO Unum
IF Click=1 THEN GOTO 8440
PRINT "ENTER PEN COLOR FOR HERMITE FIT ON UNIT";
INPUT Colour
PEN Colour
L=0
FOR K=1 TO Total
IF Unit(K)=J THEN
L=L+1
Slope(L)=Slp(K)
Cc(L)=Cc(K)
X3(L)=X2(K)
END IF
NEXT K
FOR M=1 TO L-1
Dist=X3(M+1)-X3(M)
Na=Cc(M)
Nc=Cc(M+1)
Na=Na/Dist
Nc=Nc/Dist
Nb=Slope(M)
Nd=Slope(M+1)
FOR X=0 TO 1 STEP .1
Y_of_x=Na*(X-1)^2*(2*X+1)+Nb*(X-1)^2*X+Nc*X^2*3-2*X)+Nd*X^2*(X-1)
IF Click=1 THEN GOTO 8690
IF X=0 THEN
IF M=1 THEN MOVE X3(M)-C,Na*Dist+10-Ultraymin
END IF
DRAW X3(M)-C+X*Dist,Y_of_x*Dist+10-Ultraymin
Height(K)=Y_of_x*Dist
K=K+1
NEXT X
NEXT M
NEXT J
PENUP
PEN 0
CALL Minn(Minheight,Height(*),K-1)
SUBEND
**SUB Well13(Wellname$, Mindepth, Dip$(*) , Dp(*), J, Npts, Scal, X1(*), Y1(*), Unit(*), Str(*), Strike$(*) , Atype(*), Plunge$(*) , Pl(*), Trend$(*) , Tr(*))**

**DIM Wellunit(25), Depth(25), Dipp(25), Dipdir$(25)[1], X4(25), Y14(25), Dip4$(25)[1], Unit4(25), Dp4(25), Str4(25), Strike4$(25)[1], Atype4(25), Plunge4$(25)[1], Pl4(25), Trend4$(25)[1], Tr4(25)**

**ASSIGN @Thisway TO "WELL" & Wellname$**

**ENTER @Thisway; Wellunit(*), Depth(*), Dipp(*), Dipdir$(*)**

**K=0**

**LOOP**

**K=K+1**

**EXIT IF Depth(K)=0**

**END LOOP**

**N=K-1**

**MAT X4=X1**

**MAT Y14=Y1**

**MAT Str4=Str**

**MAT Dip4$=Strike$**

**MAT Atype4=Atype**

**MAT Dp4=Dp**

**MAT Dip4$=Dip$**

**MAT Unit4=Unit**

**MAT Plunge4$=Plunge$**

**MAT Pl4=P1**

**MAT Trend4$=Trend$**

**MAT Tr4=Tr**

**FOR K=J TO Npts**

**X4(K+N-1)=X1(K)**

**Y14(K+N-1)=Y1(K)**

**Str4(K+N-1)=Str(K)**

**Strike4$(K+N-1)=Strike$(K)**

**Dip4$(K+N-1)=Dip$(K)**

**Dp4(K+N-1)=Dp(K)**

**Unit4(K+N-1)=Unit(K)**

**Atype4(K+N-1)=Atype(K)**

**Plunge4$(K+N-1)=Plunge$(K)**

**Trend4$(K+N-1)=Trend$(K)**

**P14(K+N-1)=P1(K)**

**Tr4(K+N-1)=Tr(K)**

**NEXT K**

**FOR K=0 TO N-1**

**X4(J+K)=X1(J)**

**Str4(J+K)=Str(J)**

**Strike4$(J+K)=Strike$(J)**

**Y14(J+K)=Y1(K+1)**

**Dp4(J+K)=Dip(K+1)**

**Dip4$(J+K)=Dipdir$(K+1)**

**Unit4(J+K)=Wellunit(K+1)**

**Atype4(J+K)=Depth(K+1)*400/Scal**

**Plunge4$(J+K)=" "**

**Trend4$(J+K)=" "**

**P14(J+K)=0**

**Tr4(J+K)=0**

**NEXT K**

**MAT X1=X4**
MAT up = up4
MAT Str = Str4
MAT Strike$ = Strike4$
MAT Atype = Atype4
MAT Dip$ = Dip4$
MAT Unit = Unit4
MAT Trend$ = Trend4$
MAT Plunge$ = Plunge4$
MAT Tr = Tr4
MAT Pl = Pl4
Npts = Npts + N - 1
J = J + N - 1
CALL Minn(Mindepth, Depth(*), J)

SUBEND

SUB Wellfile1
DIM Wellunit(25), Depth(25), Dipp(25), Dipdir$(25)
INPUT "ENTER NAME FOR WELL FILE", Wellname$
CREATE BDAT "WELL" & Wellname$, 4, 200
ASSIGN @Route TO "WELL" & Wellname$
J = 1
LOOP
INPUT "ENTER UNIT NAME", Wellunit(J)
EXIT IF Wellunit(J) = 0
INPUT "ENTER DEPTH OF HORIZON", Depth(J)
INPUT "ENTER DIP", Dipp(J)
INPUT "ENTER DIP DIRECTION", Dipdir$(J)
J = J + 1
END LOOP
N = J - 1
OUTPUT @Route; Wellunit(*), Depth(*), Dipp(*), Dipdir$(*)
SUBEND

SUB Spacing(X(*), Y(*), Ndat, Yy(*), M)
DIM Slope(150), Xx(150)
FOR J = 1 TO Ndat
    Xx(J) = INT(X(J))
NEXT J
FOR J = 1 TO Ndat - 1
    Xdist = Xx(J + 1) - Xx(J)
    Ydist = Y(J + 1) - Y(J)
    Slope(J) = Ydist / Xdist
NEXT J
M = 0
FOR J = 1 TO Ndat - 1
    FOR K = 0 TO Xx(J + 1) - Xx(J) - 1
        M = M + 1
        Yy(M) = Y(J) + Slope(J) * K
    NEXT K
NEXT J
L = 4
FOR L1 = M + 1 TO M + 4
    Yy(L1) = Y(J - 1) + Slope(J - 1) * (K - 1)
NEXT L1
SUBEND

SUB Interval(Xx(*), Hh(*), Minxx, Minhh, Ndat, Xxx(*), Hhh(*))
FOR J = 1 TO Ndat
    MOVE Xx(J) - Minxx, Hh(J) - Minhh
LABEL "*
NEXT J
FOR J = 1 TO Ndat
1250 \text{Xxx}(J)=\text{INT}(\text{Xx}(J))
1260 \text{ELSE}
1270 \quad \text{IF } \text{FRACT}(\text{Xx}(J))<.4 \text{ THEN}
1280 \quad \text{Xxx}(J)=\text{INT}(\text{Xx}(J))+.25
1290 \text{ELSE}
1300 \quad \text{IF } \text{FRACT}(\text{Xx}(J))<.60 \text{ THEN}
1310 \quad \text{Xxx}(J)=\text{INT}(\text{Xx}(J))+.5
1320 \text{ELSE}
1330 \quad \text{IF } \text{FRACT}(\text{Xx}(J))<.80 \text{ THEN}
1340 \quad \text{Xxx}(J)=\text{INT}(\text{Xx}(J))+1
1350 \text{ELSE}
1360 \quad \text{Xxx}(J)=\text{INT}(\text{Xx}(J))+1
1370 \text{END IF}
1380 \text{END IF}
1390 \text{END IF}
1400 \text{END IF}
1410 \text{Hhh}(J)=\text{Hh}(J)
1420 \text{NEXT J}
1430 \text{SUBEND}
1440 \text{H************************************************}
1450 \text{Spline:}
1460 \text{SUB Spline(}X(*),Y(*),Ndat,Var(*),C,Ultraymin\text{)}
1470 \text{REDIM }X(\text{Ndat})
1480 \text{REDIM }Y(\text{Ndat})
1490 \text{N=Ndat-1}
1500 \text{DIM }H(\text{40})
1510 \text{FOR }J=1 \text{ TO }N
1520 \quad H(J)=X(J)-X(J-1)
1530 \text{NEXT J}
1540 \text{DIM }A(\text{40},40),\text{A inv}(\text{40},40)
1550 \text{REDIM }A(0:N,0:N)
1560 \text{REDIM }A_inv(0:N,0:N)
1570 \text{FOR }J=0 \text{ TO }N
1580 \quad \text{FOR }I=0 \text{ TO }N
1590 \quad \text{IF }I=J \text{ THEN}
1600 \quad \quad \text{IF }J=0 \text{ THEN}
1610 \quad \quad A(J,I)=2/H(J+1)
1620 \quad \quad A(J+1,I)=1/H(J+1)
1630 \quad \quad \text{GOTO 1760}
1640 \quad \quad \text{END IF}
1650 \quad \quad \text{IF }J=N \text{ THEN}
1660 \quad \quad A(J,I)=2/H(J)
1670 \quad \quad A(J-1,I)=1/H(J)
1680 \quad \quad \text{GOTO 1760}
1690 \quad \quad \text{END IF}
1700 \quad \quad \text{END IF}
1710 \quad \quad \text{IF }I=J \text{ THEN}
1720 \quad \quad A(J,I)=(2/H(J+1))+(2/H(J))
1730 \quad \quad A(J-1,I)=1/H(J)
1740 \quad \quad A(J+1,I)=1/H(J+1)
1750 \quad \quad \text{END IF}
1760 \quad \text{NEXT I}
1770 \text{NEXT J}
1780 \text{DIM }B(40)
1790 \text{REDIM }B(0:N)
1800 \text{FOR }J=0 \text{ TO }N
1810 \quad \text{IF }J=0 \text{ THEN}
1820 \quad \quad B(J)=(Y(J+1)-Y(J))/H(J+1)^2
1830 \quad \quad \text{GOTO 1900}
1840 \quad \quad \text{END IF}
1850 \quad \quad \text{IF }J=N \text{ THEN}
1860 \quad \quad B(J)=(Y(J)-Y(J-1))/H(J)^2
1870 \quad \quad \text{ELSE}
1880 \quad \quad B(J)=(Y(J+1)-Y(J))/(H(J+1)^2)+(Y(J)-Y(J-1))/H(J)^2
1900 \quad \text{END IF}

1910 REDIM Var(0:N)
1920 MAT A_inv = INV(A)
1930 FOR J=0 TO N
1940 B(J) = 3*B(J)
1950 NEXT J
1960 MAT Var = A_inv*B
1970 REDIM X(1:Ndat)
1980 REDIM Y(1:Ndat)
1990 REDIM Var(1:Ndat)
2000 DIM U(-1:100), Slope(40), Coc(40), X3(40)
2010 L = 0
2020 FOR K = 1 TO Ndat
2030 L = L + 1
2040 Slope(L) = Var(K)
2050 Coc(L) = Y(K)
2060 X3(L) = X(K)
2070 NEXT K
2080 K = 1
2090 INPUT "ENTER PEN", Penn
2100 PEN Penn
2110 FOR M = 1 TO L - 1
2120 Dist = X3(M + 1) - X3(M)
2130 Na = Coc(M)
2140 Nc = Coc(M + 1)
2150 Na = Na/Dist
2160 Nc = Nc/Dist
2170 Nb = Slope(M)
2180 Nd = Slope(M + 1)
2190 FOR Xx = 0 TO 1 STEP .1
2200 Y_of_x = Na*(Xx-1)^2*(2*Xx+1)+Nb*(Xx-1)^2*Xx+Nc*Xx^2*(3-2*Xx)+Nd*Xx^3:
2210 IF Xx = 0 THEN
2220 IF M = 1 THEN MOVE X3(M) - C, Na*Dist-Ultraymin
2230 END IF
2240 DRAW X3(M) - C+Xx*Dist,Y_of_x*Dist-Ultraymin
2250 K = K + 1
2260 NEXT Xx
2270 NEXT M
2280 PENUP
2290 SUBEND
**STOP***

**BALANCE**

**STOP***

10 ! **INPUT "ENTER SCALE",Scal**
20 ! **PRINTER IS 1**
30 ! **GINIT**
40 ! **PLOTTER IS 705,"HPGL"**
50 ! **GRAPHICS ON**
60 ! **GRAPHICS INPUT IS 705,"HPGL"**
70 ! **PEN 0**
80 ! **PROGRAM IS DIMENSIONED FOR 200 POINTS.**
90 ! **A FAULT CAN HAVE 20 DIGITIZED POINTS**
100 ! **THERE CAN BE 25 DIGITIZED POINTS BETWEEN ALL SEGMENTS.**
110 ! **UP TO 4 LAYERS AND 8 FAULTS**
120 DIM X(200),Y(200),Bigarea(25),Layerarea(2200),Subarea(25),Ax(8,500)
130 DIM Faulttrace(8),Fx(8,5,2),Fy(8,5,2),Faultarea(8,4),Bx(8,500)
140 DIM Mm(1:25),Nn(1:25),Xs(2200),Ys(2200),Newx(150),Newy(150)
150 DIM Aa(500),Bb(500),B(20),C(20),Aa(20),Bb(20),Porder(48,6),Zzz(48,6)
160 DIM Sectionfault(48),Aa(8,20),Bb(8,20),Nodigitpts(10)
170 PRINT "DO YOU WANT TO DIGITIZE A CROSS SECTION IN ORDER TO "
180 PRINT "CHECK FOR BALANCE (1) OR DRAW A CROSS SECTION FROM AN"
190 PRINT "EXISTING FILE (2)?"
200 IF Drawonly=2 THEN
210 GOSUB 4090
220 FOR J=1 TO Ndat
230 Xx(J)=X(J)
240 Yy(J)=Y(J)
250 NEXT J
260 FOR J=1 TO Ndat
270 IF X(J)=30000 THEN Fault=1+Fault
280 IF X(J)=20000 THEN Conta=Conta+1
290 NEXT J
300 ! **LAYERS=CONTA**
310 ! **NFAULTS=FAULT/CONTA**
320 FOR J=1 TO Nfaults
330 IF Bz(J,Nodigitpts(J)-1)<Bz(J,Nodigitpts(J)) THEN
340 Faulttrace(J)=2
350 ELSE
360 Faulttrace(J)=1
370 END IF
380 NEXT J
390 J=0
400 ! **FAULT=1**
410 FOR Layer=1 TO Nlayers
420 IF J=Ndat+1 THEN GOTO 670
430 J=J+1
440 IF X(J)=30000 THEN
450 IF Fault=Nfaults THEN GOTO 590
460 Fault=1+Fault
470 END IF
480 GOTO 480
490 Fault=1
500 NEXT Layer
510 GOTO 480
520 END IF
530 IF Drawonly=1 THEN
650 GOSUB 2690
660 END IF
670 PRINT "DO YOU WANT TO FIND POLYNOMIAL FITS TO IMPROVE THE"
680 INPUT "CROSSSECTION BALANCING? Y=1", Polyn
690 IF Polyn=1 THEN GOSUB 4770
700 J=0
710 LOOP
720 EXIT IF Polyn=1
730 J=J+1
740 EXIT IF J=Ndat+1
750 Xxx(J)=X(J)
760 Yyy(J)=Y(J)
770 M=Ndat
780 Sds=Ndat
790 END LOOP
800 FOR J=1 TO 8
810 FOR K=1 TO 20
820 Ax(J,K)=Az(J,K)
830 Bx(J,K)=Bz(J,K)
840 NEXT K
850 NEXT J
860 IF Polyn=1 THEN GOSUB 5660
870 INPUT "DO YOU WANT A PLOT OF WHAT WAS JUST DIGITIZED? Y=1", Digit
880 IF Digit=1 THEN GOSUB 4330
890 INPUT "DO YOU WANT TO CHECK FOR AREA BALANCE (1) OR NOT?(2)" , Balance_it
900 IF Balance_it=2 THEN
910 Nlayers=Nlayers-1
920 GOTO 2520
930 END IF
940 Sectionno=0
950 Fault=0
960 G=0
970 E=0
980 FOR N=1 TO Nlayers
990 Fault=0
1000 Totalayer=0
1010 A=0
1020 Totalfault=0
1030 Totalarea=0
1040 Fault=Fault+1
1050 G=G+1
1060 IF Xxx(G)>20000 THEN GOTO 1050
1070 Sectionno=1+Sectionno
1080 GOTO 1050
1090 END IF
1100 IF Xxx(G+1)>20000 THEN GOTO 2220
1110 IF Xxx(G+1)>30000 THEN
1120 Sectionno=1+Sectionno
1130 GOTO 1050
1140 END IF
1150 IF Xxx(G)>30000 THEN GOTO 1210
1160 GOTO 1210
1170 ELSE
1180 Layerea(G)=.5*(Yyy(G)*Scal/400+Yyy(G+1)*Scal/400)*(Xxx(G+1)*Scal/400-xx(G)*Scal/400)
1190 GOTO 1050
1200 END IF
1210 A=1+A
1220 IF Fy(A,N+1,1)=0 THEN GOTO 2210
1230 FOR Side=1 TO 2
1240 FOR Layer=1 TO Nlayers
1250 FOR Point=1 TO Nodigitpts(Fault)
1260 IF Faulttrace(Fault)=2 THEN
1270 IF Fx(Fault,Layer+1,Side)<Ax(Fault,Point) THEN
ELSE
GOTO 1480
END IF

ELSE
GOTO 1480
END IF

ELSE
IF Fx(Fault,Layer+1,Side)>Ax(Fault,Point) THEN
IF Ax(Fault,Point)>Fx(Fault,Layer,Side) THEN
P(Fault,Layer,Point,Side)=100
ELSE
GOTO 1480
END IF
ELSE
GOTO 1480
END IF
GOTO 1480
END IF
GOTO 1480

NEXT Point
NEXT Layer
NEXT Side
MAT Faultarea = (0)
FOR Side=1 TO 2
Times=0
Point=0
Point=1+Point
IF Faulttrace(Fault)=2 THEN
IF P(Fault,N,Point,Side) = 100 THEN
IF Times=0 THEN
Faultarea(Fault,Side) = (Fy(Fault,N+1,Side) + Bx(Fault,Point)) * 0.5 * Ax(Fault,Point) + Fx(Fault,N+1,Side)
Times=1+Times
GOTO 1550
ELSE
Faultarea(Fault,Side) = Faultarea(Fault,Side) + (Bx(Fault,Point) + Bx(Fault,Point-1)) * 0.5 * (-1 * Ax(Fault,Point) + Ax(Fault,Point-1))
Times=1+Times
GOTO 1550
END IF
ELSE
IF P(Fault,N,Point-1,Side) = 100 THEN
IF Times=0 THEN
Faultarea(Fault,Side) = (Fy(Fault,N+1,Side) + Bx(Fault,Point)) * 0.5 * (-1 * Fx(Fault,N,Side) + Ax(Fault,Point-1))
GOTO 1980
ELSE
Faultarea(Fault,Side) = Faultarea(Fault,Side) + (Bx(Fault,Point) + Bx(Fault,Point-1)) * 0.5 * (Ax(Fault,Point) - Ax(Fault,Point-1))
Times=1+Times
GOTO 1550
END IF
END IF
ELSE
IF P(Fault,N,Point,Side) = 100 THEN
IF Times=0 THEN
Faultarea(Fault,Side) = (Fy(Fault,N,Side) + Bx(Fault,Point)) * 0.5 * (Ax(Fault,Point) - Fx(Fault,N,Side))
Times=1+Times
GOTO 1550
ELSE
Faultarea(Fault,Side) = Faultarea(Fault,Side) + (Bx(Fault,Point) + Bx(Fault,Point-1)) * 0.5 * (Ax(Fault,Point) - Ax(Fault,Point-1))
Times=1+Times
GOTO 1550
END IF
END IF
END IF

Faultarea(Fault,Side) = Faultarea(Fault,Side) + (Bx(Fault,Point) + Bx(Fault,Point-1)) * 0.5 * (Ax(Fault,Point) - Ax(Fault,Point-1))
Times=1+Times
GOTO 1550
1900 GOTO 1980
1910 ELSE
1920 IF Fy(Fault,N+1,Side)>Bx(Fault,Point) THEN GOTO 1970
1930 GOTO 1550
1940 END IF
1950 END IF
1960 END IF
1970 IF Faultarea(Fault,Side)=0 THEN Faultarea(Fault,Side)=(Fy(Fault,N,Side)+Fy(Fault,N+1,Side))*0.5*(Fx(Fault,N+1,Side)-Fx(Fault,N,Side))
1980 IF Faulttrace(Fault)=1 THEN
1990 IF Faultarea(Fault,1)<0 THEN Faultarea(Fault,1)=1*Faultarea(Fault,1)
2000 IF Faultarea(Fault,2)>0 THEN Faultarea(Fault,2)=1*Faultarea(Fault,2)
2010 ELSE
2020 IF Faultarea(Fault,1)>0 THEN Faultarea(Fault,1)=-1*Faultarea(Fault,1)
2030 IF Faultarea(Fault,2)<0 THEN Faultarea(Fault,2)=-1*Faultarea(Fault,2)
2040 END IF
2050 NEXT Side
2060 IF Xxy(G+2)=20000 THEN
2070 Faultarea(Fault,2)=0
2080 GOTO 2180
2090 END IF
2100 IF Xxy(G-2)=0 THEN
2110 Faultarea(Fault,1)=0
2120 ELSE
2130 IF Xxy(G-2)=20000 THEN
2140 PRINT "AT START"
2150 Faultarea(Fault,1)=0
2160 END IF
2170 END IF
2180 Areafault(N)=Faultarea(Fault,1)+Faultarea(Fault,2)+Areafault(N)
2190 Sectionfault(Sectionno)=Faultarea(Fault,1)+Sectionfault(Sectionno)
2200 Sectionfault(Sectionno+1)=Faultarea(Fault,2)+Sectionfault(Sectionno+1)
2210 GOTO 1040
2220 FOR F=E+1 TO G
2230 Totallayer=Totallayer+Layerarea(F)
2240 NEXT F
2250 E=G
2260 Totarea=Totallayer+Areafault(N)*Scal^2/160000
2270 Bigarea(N)=Totarea
2280 Subarea(N-1)=Bigarea(N-1)-Totallayer
2290 NEXT N
2300 Nlayers=Nlayers-1
2310 IF Prtout=1 THEN PRINTER IS 702
2320 PRINT "DATA FROM CROSS SECTION "; "BAL" & Name$
2330 PRINT
2340 PRINT "LISTING OF LAYER AREAS AND TOTAL CROSS SECTION AREAS"
2350 PRINT "-----------------------------------------------"
2360 FOR T=1 TO Nlayers
2370 PRINT "(";T;"), Subarea(T); "SQUARE METERS"
2380 NEXT T
2390 FOR M=1 TO Nlayers
2400 Totalarea=Totalarea+Subarea(M)
2410 NEXT M
2420 PRINT "TOTAL AREA=";Totalarea; "SQUARE METERS"
2430 PRINTER IS 1
2440 Sarah: ! SARAH'S ANALYSIS
2450 PRINT "SARAH'S ANALYSIS: USER INPUTS AN ORIGINAL THICKNESS FOR EACH"
2460 PRINT "LAYER. ORIGINAL LENGTH IS COMPUTED AND A STATISTICAL"
2470 PRINT "ANALYSIS IS DONE TO DETERMINE IF SECTION IS AREA BALANCED."
2480 INPUT "DO YOU WANT TO USE THIS ANALYSIS? Y=1, N=2", Sarah
t(*),Sectionno,Layera(*),Sds,Prtout)
2510 PRINTER IS 
2520 INPUT "DO YOU WANT TO DO A LAYER LENGTH BALANCE TEST?",Layerlength 
2530 IF Layerlength=1 THEN GOSUB 6160 
2540 PRINT "NOW THAT YOU HAVE CHECKED FOR BALANCE, DO YOU WANT TO MAKE CHANGES 
2550 PRINT "FROM THE EXISTING FILE? Y=1 N=2" 
2560 INPUT Chng 
2570 IF Chng=1 THEN GOSUB 6520 
2580 PRINT "NOW THAT YOU HAVE CHECKED FOR BALANCE ARE YOU INTERESTED IN 
2590 INPUT "SAVING THE DIGITIZED DATA FOR THREEDIM? Y=1",Make_file 
2600 IF Make_file=1 THEN 
2610 GOSUB 3910 
2620 GOSUB 4020 
2630 END IF 
2640 PRINT "PROGRAM IS FINISHED" 
STOP 
2660!****************************************************************************************************************** 
2670!****************************************************************************************************************** 
2680 Digit:*DIGITIZING SUBROUTINE 
2690 INPUT "ENTER NUMBER OF LAYER CONTACTS TO BE DIGITIZED",Nlayers 
2700 INPUT "ENTER NUMBER OF FAULTS",Nfaults 
2710 IF Nfaults=0 THEN GOTO 2780 
2720 FOR N=1 TO Nfaults 
2730 PRINT "DOES FAULT NUMBER ";N:"DIP TO THE LEFT OR RIGHT?" 
2740 INPUT "RIGHT-1, LEFT-2",Tracedir 
2750 Faulttrace(N)=Tracedir 
2760 NEXT N 
2770 IF Nfaults<>0 THEN GOSUB 3500 
2780 !THE DIGITIZING LOOP NOW STARTS. 
2790 PRINT "DIGITIZE THE POINTS ON EACH LAYER FROM LEFT TO RIGHT." 
2800 Ppp=0 
2810 Totarea=0 
2820 Nlayer=0 
2830 N=0 
2840 G=0 
2850 J=0 
2860 E=0 
2870 J=1+J 
2880 IF X(J-1)-30000 THEN GOTO 2870 
2890 Layerend=0 
2900 PRINT "START OR CONTINUE DIGITIZING ALONG CONTACT";Nlayer+1 
2910 DIGITIZE X(J),Y(J) 
2920 IF X(J)<0 THEN 
2930 PRINT "MOVE YOUR CROSS SECTION, START OVER, DIGITIZED VALUE IS LESS THAN ZERO" 
2940 BEEP 750,1.5 
2950 STOP 
2960 ELSE 
2970 IF Y(J)<0 THEN 
2980 PRINT "MOVE YOUR CROSS SECTION, START OVER, DIGITIZED VALUE IS LESS THAN ZERO" 
2990 BEEP 750,1.5 
3000 STOP 
3010 END IF 
3020 END IF 
3030 IF X(J-1)=X(J) THEN Xequal=1 
3040 IF Y(J-1)=Y(J) THEN Yequal=1 
3050 IF Xequal AND Yequal THEN 
3060 BEEP 1000,.5 
3070 INPUT "END OF A CONTACT(1), FAULT(2) OR ERROR(3)";Layerend 
3080 IF Layerend=1 THEN 
3090 X(J)=20000 
3100 Y(J)=20000 
3110 PRINT "WE ARE AT END OF LAYER";Nlayer+1
3140 Yequal=0
3150 IF Nlayer=Nlayers THEN GOTO 3460
3160 GOTO 2870
3170 ELSE
3180 IF Layerend=2 THEN
3190 Ppp=Ppp+1
3200 IF Ppp>Nfaults THEN Ppp=1
3210 Fx(Ppp,Nlayer+1,1)=X(J-1)
3220 Fy(Ppp,Nlayer+1,1)=Y(J-1)
3230 X(J)=30000
3240 Y(J)=30000
3250 PRINT "RESUME DIGITIZING ON OTHER SIDE OF FAULT";Ppp
3260 DIGITIZE X(J+1),Y(J+1)
3270 BEEP
3280 Fx(Ppp,Nlayer+1,2)=X(J+1)
3290 Fy(Ppp,Nlayer+1,2)=Y(J+1)
3300 Xequal=0
3310 Yequal=0
3320 GOTO 2870
3330 ELSE
3340 IF Layerend=3 THEN
3350 J=J-1
3360 GOTO 3420
3370 END IF
3380 END IF
3390 END IF
3400 ELSE
3410 BEEP
3420 Xequal=0
3430 Yequal=0
3440 GOTO 2870
3450 END IF
3460 Ndat=J
3470 GOSUB 3780
3480 RETURN
3490!********************************************************************
3500 Fault_digit: !THIS SUBROUTINE DIGITIZES ALONG THE FAULTS IF IN
3510 !FACT THERE ARE ANY.
3520 FOR Z=1 TO Nfaults
3530 PRINT "START DIGITIZING ALONG FAULT";Z;"FROM LEFT TO RIGHT"
3540 Last=0
3550 M=0
3560 M=M+1
3570 PRINT "DIGITIZE POINT ";M
3580 DIGITIZE Az(Z,M),Bz(Z,M)
3590 IF Az(Z,M)=Az(Z,M-1) THEN Azequal=1
3600 IF Bz(Z,M)=Bz(Z,M-1) THEN Bzequal=1
3610 IF Azequal AND Bzequal THEN
3620 Azequal=0
3630 Bzequal=0
3640 Az(Z,M)=0
3650 Bz(Z,M)=0
3660 BEEP 1000,.5
3670 Nodigitpts(Z)=M-1
3680 M=M-1
3690 GOTO 3740
3700 ELSE
3710 BEEP
3720 GOTO 3560
3730 END IF
3740 NEXT Z
3750 GOSUB 3850
3760 RETURN
CALL FOR DRAWING PURPOSES.

CREATE BDAT "BAL"&Name$,1,3220
ASSIGN @Path TO "BAL"&Name$
OUTPUT @Path;X(*),Y(*)
RETURN

!*-----------------------------------------------------------------------*

Two$="II"
CREATE BDAT Two$&Name$,2,4000
ASSIGN @File TO Two$&Name$
OUTPUT @File;Nfaults,Nodigitpts(*),Az(*),Bz(*)
RETURN

!*-----------------------------------------------------------------------*

Make a file2:
INPUT "ENTER NAME FOR POLYFIT FILE",Hello$
CREATE BDAT "POLY"&Hello$,2,25000
ASSIGN @Way TO "POLY"&Hello$
OUTPUT @Way;Ixx(*)
OUTPUT @Way;Iyy(*)
CREATE BDAT "INFL"&Hello$,1,2744
ASSIGN @Myway TO "INFL"&Hello$
OUTPUT @Myway;Porder(*)
RETURN

!*-----------------------------------------------------------------------*

Make a file3:
Two$="FAULT"
CREATE BDAT Two$&Hello$,2,40000
ASSIGN @Way2 TO Two$&Hello$
OUTPUT @Way2;Nfaults,Nodigitpts(*),Ax(*),Bx(*)
RETURN

!*-----------------------------------------------------------------------*

Read a file: !THIS SUBROUTINE READS AN EXISTING FILE SO THAT IT
!CAN BE DRAWN
INPUT "ENTER NAME OF FILE TO BE READ",Name$
ASSIGN @Path TO "BAL"&Name$
J=0
ON END @Path GOTO 4160
ENTER @Path;X(*),Y(*)
LOOP
J=J+1
EXIT IF X(J)=0
END LOOP
Ndat=J-1
F=0
FOR J=1 TO Ndat
IF X(J)=30000 THEN F=1
NEXT J
IF F=0 THEN GOTO 4310
Two$="II"
ASSIGN @File TO Two$&Name$
J=0
ON END @File GOTO 4310
ENTER @File;Nfaults,Nodigitpts(*),Az(*),Bz(*)
RETURN

!*-----------------------------------------------------------------------*

Plotting: !THIS SUBROUTINE DRAWS THE CROSS SECTION WHICH HAS PREVIOUSLY
!BEEN DIGITIZED AND MAY OR MAY NOT HAVE BEEN STORED.
INPUT "CRT(.1) OR PLOTTER(2)",Dumb
IF Dumb=2 THEN
PLOTTER IS 705,"HPGL"
INPUT "MAKE SURE PLOTTER ARM IS FREE, PRESS CONT(INUE)",Arm
INPUT "WHAT PEN COLOR?",Colour
PEN Colour
ELSE
PLOTTER IS 3,"INTERNAL"
END IF
4450 IF R=1 THEN MOVE Xx(1),Yy(1)
4460 IF R=Sds THEN GOTO 4610
4470 IF Xx(R)=20000 THEN
4480 MOVE Xx(R+1),Yy(R+1)
4490 GOTO 4600
4500 ELSE
4510 GOTO 4530
4520 END IF
4530 IF Xx(R)=30000 THEN
4540 MOVE Xx(R+1),Yy(R+1)
4550 GOTO 4600
4560 ELSE
4570 GOTO 4590
4580 END IF
4590 PLOT Xx(R),Yy(R)
4600 NEXT R
4610 PEN 0
4620 IF Nfaults=0 THEN GOTO 4720
4630 IF Dumb=2 THEN INPUT "ENTER PEN COLOR FOR FAULT TRACE",Colour
4640 PEN Colour
4650 FOR Z=1 TO Nfaults
4660 MOVE Ax(Z,1),Bx(Z,1)
4670 FOR M=1 TO Nodigitpts(Z)
4680 DRAW Ax(Z,M),Bx(Z,M)
4690 NEXT M
4700 NEXT Z
4710 PEN 0
4720 IF Dumb=1 THEN INPUT "PRESS 1 IF YOU WANT TO REMOVE PLOT FROM SCREEN",Remove
4730 IF Remove=1 THEN GCLEAR
4740 RETURN
4750!*************************************************************************
4760 !SUBROUTINE TO FIT POLYNOMIAL CURVES TO DIGITIZED POINTS
4770 Layer=0
4780 R=0
4790 Sds=0
4800 G=0
4810 Layer=Layer+1
4820 E=G+1
4830 IF Layer=Nlayers THEN RETURN
4840 IF E=0 THEN E=1
4850 G=G+1
4860 IF X(G)=20000 THEN
4870 Xx(Sds)=20000
4880 Yy(Sds)=20000
4890 GOTO 4820
4900 END IF
4910 IF X(G+1)=20000 THEN GOTO 5030
4920 IF X(G)=30000 THEN
4930 Xx(Sds)=30000
4940 Yy(Sds)=30000
4950 GOTO 4830
4960 GOTO 4830
4970 END IF
4980 IF X(G+1)=30000 THEN
4990 GOTO 5030
5000 ELSE
5010 GOTO 4860
5020 END IF
5030 R=R+1
5040 PRINT "DO YOU WANT A POLYNOMIAL FIT FOR SECTION";R;"YES=1"
5050 INPUT Polynom
5060 IF Polynom=1 THEN GOTO 5310
5070 IF FRACT(X(E))<.20 THEN
5080 !*************************************************************************

IF FRACT(X(E))<.4 THEN
    X(E)=INT(X(E))+.25
ELSE
    IF FRACT(X(E))<.60 THEN
        X(E)=INT(X(E))+.5
    ELSE
        IF FRACT(X(E))<.80 THEN
            X(E)=INT(X(E))+.75
        ELSE
            X(E)=INT(X(E))+1
        END IF
    END IF
END IF

FOR J=E TO G
    Sds=Sds+1
    Xxx(Sds)=X(J)
    Yyy(Sds)=Y(J)
    NEXT J
Sds=Sds+1
GOTO 4860

FOR J=E TO G
    Mm(J-E+1)=X(J)
    Nn(J-E+1)=Y(J)
    NEXT J
CALL Minn(Minmm,Mm(*),G-E+1)
CALL Minn(Minnn,Nn(*),G-E+1)
REDIM Mmm(1:25)
REDIM Nnn(1:25)
CALL Interval(Mm(*),Nn(*),Minmm,Minnn,G-E+1,Mm(*),Nn(*))
PLOTTER IS 3,"INTERNAL"
FOR J=E TO G
    MOVE X(J),Y(J)
    CSIZE 3
    LORG 5
    PEN 1
    LABEL "*
    NEXT J
FOR Z=1 TO Nfaults
    M(1)=Z
    CALL Poly(Mm(*),Nn(*),G-E+1,Xx(*),Yy(*),M,R,Porder(*))
FOR J=1 TO M
    IF J=1 THEN MOVE Xx(J),Yy(J)
    DRAW Xx(J),Yy(J)
    NEXT J
INPUT "DO YOU LIKE THIS OR WANT TO TRY AGAIN? TRYAGAIN=1",Tryit
GCLEAR
IF Tryit=1 THEN GOTO 5400
K=0
FOR J=Sds+1 TO M+Sds
    K=K+1
    Xxx(J)=Xx(K)
    Yyy(J)=Yy(K)
    NEXT J
Sds=M+1+Sds
GOTO 4860
RETURN

!**********************************************************************
!SUBROUTINE TO FIT CURVES TO FAULTS
FOR Z=1 TO Nfaults
MAT C= (0)
MAT B= (0)
Sarah=0
FOR J=1 TO Nodigitpts(Z)
    Sarah=1+Sarah
    C(Sarah)=Az(Z,J)
    NEXT J
**********************************************************************

PLOTTER IS 3,"INTERNAL"
CLEAR
FOR J=1 TO Nodigitpts(Z)
    MOVE Az(Z,J),Bz(Z,J)
    CSIZE 2
    LORG 5
    LABEL ":=
NEXT J
REDIM Aa(1:20)
REDIM Bb(1:20)
CALL Minn(Mina,C(*),Nodigitpts(Z))
CALL Minb(B(*),Nodigitpts(Z))
CALL Interval(C(*),B(*),Mina,Minb,Nodigitpts(Z),Aa(*),Bb(*))
PRINT "DO YOU WANT A POLY FIT FOR FAULT";Z;":\nY=1"
INPUT Toocurly
IF Toocurly<2 THEN
    FOR J=1 TO Nodigitpts(Z)
        Ax(Z,J)=Aa(J)
        Bx(Z,J)=Bb(J)
    NEXT J
    GOTO 6130
END IF
PRINT "FOR FAULT",Z;":\n"
CALL Poly(Aa(*),Bb(*),Nodigitpts(Z),Aaa(*),Bbb(*),Nopts,J,zzz(*))
PEN 1
FOR J=1 TO Nopts
    IF J=1 THEN MOVE Aaa(J),Bbb(J)
    DRAW Aaa(J),Bbb(J)
NEXT J
INPUT "DO YOU LIKE THIS FIT OR WANT TO TRY AGAIN? TRY_AGAIN=1",Try_again
IF Try_again=1 THEN GOTO 5760
GCLEAR
FOR J=1 TO Nopts
    Ax(Z,J)=Aaa(J)
    Bx(Z,J)=Bbb(J)
NEXT J
Nodigitpts(Z)=Nopts
NEXT Z
RETURN
Layerlength():SUBROUTINE TO BALANCE BY COMPARING BED LENGTHS
!NUMBER OF LAYERS IN NLAYERS
J=0
FOR N=1 TO Nlayers+1
    J=J+1
    IF Xx(J)=30000 THEN GOTO 6200
    IF Xx(J+1)=30000 THEN GOTO 6200
    IF Xx(J+1)=20000 THEN GOTO 6270
    IF Xx(J)=20000 THEN GOTO 6200
    Templength(N)=Templength(N)+SQRT((Xx(J+1)-Xx(J))^2+(Yy(J+1)-Yy(J))^2)
    GOTO 6200
    NEXT N
K=1
Finish=600
FOR J=1 TO Sds
    IF J>1 THEN
        IF Finish=Xx(J-2) THEN Start=Xx(J)
        END IF
    END IF
    IF Prtout=1 THEN PRINTER IS 702
    IF J=1 THEN Start=Xx(J)
    IF Xx(J)=20000 THEN Finish=Xx(J-1)
    IF Xx(J)=20000 THEN
        Dist=Finish-Start
        Strain=(Dist-Templength(K))/Templength(K)
6420 END IF
6430 NEXT J
6440 PRINT "BED LAYER LENGTH TECHNIQUE"
6450 PRINT ""
6460 FOR N=1 TO Nlayers+1
6470 PRINT "LENGTH OF LAYER";N:"IS";Templength(N)*Scal/400;"METERS"
6480 NEXT N
6490 PRINTER IS 1
6500 RETURN
6510
6520 Redigit!: TO REEDIT DIGITIZE PARTS OF CROSSSECT
6530 GOSUB 4330
6540 CSIZE 2
6550 INPUT "ENTER PEN COLOR FOR LABELS",Penn
6560 PEN Penn
6570 FOR J=1 TO Ndat
6580 IF X(J)=20000 THEN GOTO 6650
6590 IF X(J)=30000 THEN GOTO 6650
6600 MOVE X(J),Y(J)
6610 LORG 5
6620 LABEL "*"
6630 LORG 4
6640 LABEL J
6650 NEXT J
6660 PEN 0
6670 INPUT "DO YOU WANT TO ADD POINTS(1),MOVE POINTS(2) OR NONE(4)",Redid
6680 IF Redid=4 THEN GOTO 7140
6690 IF Redid>1 THEN 7010
6700 INPUT "WHAT 2 END POINTS ARE TO BE HELD CONSTANT?",Enda,Endb
6710 PRINT "HOW MANY POINTS DO YOU PLAN TO DIGITIZE BETWEEN";Enda,Endb
6720 INPUT Numadded
6730 FOR J=1 TO Numadded
6740 PRINT "DIGITIZE POINT #";J;"TO BE ADDED"
6750 DIGITIZE Addx(J),Addy(J)
6760 BEEP
6770 NEXT J
6780 FOR J=Endb TO Ndat
6790 Newx(J+Numadded)=X(J)
6800 Newy(J+Numadded)=Y(J)
6810 NEXT J
6820 FOR J=1 TO Enda
6830 Newx(J)=X(J)
6840 Newy(J)=Y(J)
6850 NEXT J
6860 R=Enda
6870 FOR J=1 TO Numadded
6880 R=R+1
6890 Newx(R)=Addx(J)
6900 Newy(R)=Addy(J)
6910 NEXT J
6920 IF Redid=1 THEN
6930 Ndat=Ndat+Numadded
6940 Sds=Ndat
6950 MAT Xxx= Newx
6960 MAT Yyy= Newy
6970 MAT X= Newx
6980 MAT Y= Newy
6990 GOTO 7100
7000 END IF
7010 IF Redid=2 THEN
7020 INPUT "ENTER POINT TO BE MOVED",Moved
7030 PRINT "DIGITIZE NEW LOCATION"
7040 DIGITIZE X(Moved),Y(Moved)
7050 BEEP
Sds-Ndat
END
PRINT "DO YOU WANT PLOT OF CHANGES? Y=1 N=2"
INPUT Chng
IF Chng=1 THEN GOTO 6520
IF Chng=2 THEN GOTO 6670
INPUT "DO YOU WANT TO MAKE A FILE OF THIS DATA?,Y=1",Refile
IF Refile=1 THEN GOSUB 3780
RETURN
END

------------------------------------------------------------------------
Poly:SUB Poly(Ww(*),Uu(*),Dat,O(*),Oo(*),M,R,Porder(*))
M=0
PRINT "ENTER ORDER OF POLYNOMIAL"
INPUT "\"N\"
IF N>Dat THEN GOTO 7220
DIM Ab(16,8),Aprime(8,16),Solution(0:8,1)
DIM Aaa(8,8),A inv(8,8),B(8,1),Z(0:19000),Ll(-3:190)
REDIM Ab(1:Dat,0:N-1)
REDIM Aprime(0:N-1,1:Dat)
REDIM Solution(0:N-1,1)
REDIM Aaa(0:N-1,0:N-1)
REDIM A inv(0:N-1,0:N-1)
REDIM Ww(1:Dat)
REDIM Uu(1:Dat)
REDIM B(0:N-1,1)
FOR J=1 TO Dat
   FOR K=0 TO N-1
      Ab(J,K)=(Ww(J))^K
   NEXT K
NEXT J
MAT Aprime= TRN(Ab)
MAT Aaa= Aprime*Ab
MAT A inv= INV(Aaa)
MAT B= Aprime*Uu
MAT Solution= A inv*B
FOR Q=.25 TO MAX(Ww(*)) STEP .25
   Z(Q)=0
   Ll(Q)=0
   FOR S=1 TO N
      Z(Q)=Solution(S-1,0)*Q^(S-1)+Z(Q)
      Ll(Q)=(S-1)*(S-2)*Solution(S-1,0)*Q^(S-3)+Ll(Q)
   NEXT S
   Z(Q*100)=Z(Q)
NEXT Q
J=0
FOR Q=MIN(Ww(*)) TO MAX(Ww(*)) STEP .25
   M=M+1
   Q(M)=Q
   Oo(M)=Z(Q*100)
NEXT Q
I=1
IF N<4 THEN GOTO 7800
FOR Q=.25 TO MAX(Ww(*)) STEP .25
   IF Ll(Q)>0 THEN
      IF Ll(Q-.25)<0 THEN
         Porder(R,I)=Q
         I=I+1
      END IF
   END IF
END IF
IF Ll(Q)<0 THEN
   IF Ll(Q-.25)>0 THEN
      Porder(R,I)=Q
   END IF
END IF
NEXT Q
7740 END IF
7750 IF L1(Q)=0 THEN
7760 Porder(R,I)=Q
7770 I=I+1
7780 END IF
7790 NEXT Q
7800 SUBEND

7810!**************************************************************************
7820 Interval:SUB Interval(Xy(*),Hh(*),Minxx,Minhh,Ndat,Xyx(*),Hhh(*))
7830 PLOTTER IS 3,"INTERNAL"
7840 LORG 5
7850 CSIZE 3
7860 FOR J=1 TO Ndat
7870 IF FRACT(Xy(J))<.20 THEN
7880 Xyx(J)=INT(Xy(J))
7890 ELSE
7900 IF FRACT(Xy(J))<.4 THEN
7910 Xyx(J)=INT(Xy(J))+.25
7920 ELSE
7930 IF FRACT(Xy(J))<.60 THEN
7940 Xyx(J)=INT(Xy(J))+.5
7950 ELSE
7960 IF FRACT(Xy(J))<.80 THEN
7970 Xyx(J)=INT(Xy(J))+.75
7980 ELSE
7990 Xyx(J)=INT(Xy(J))+1
8000 END IF
8010 END IF
8020 END IF
8030 END IF
8040 Hhh(J)=Hh(J)
8050 NEXT J
8060 SUBEND

8070!**************************************************************************
8080 Minn:SUB Minn(Ming,G(*),Numb)
8090 Ming=G(1)
8100 FOR Gg=1 TO Numb
8110 IF G(Gg)=0 THEN GOTO 8140
8120 IF Ming=0 THEN Ming=G(Gg)
8130 IF G(Gg)<Ming THEN Ming=G(Gg)
8140 NEXT Gg
8150 SUBEND

8160!**************************************************************************
8170 Reconst:SUB Reconstruct(Nlayers,Scal,Subarea(*),Nfaults,Xxx(*),Yyy(*),Sectio
8180 nfault(*),Sectionno,Layera(*),Sds,Prtout)
8190 DIM Subtotal(48),Carea(48),Rarea(48),Errorarea(48)
8200 PRINT "DO YOU WANT TO INPUT LAYER THICKNESSES (1) OR DIGITIZE LAYERS"
8210 PRINT "FOR A MORE ACCURATE ANALYSIS (2)?"
8220 INPUT Acc
8230 IF Acc=1 THEN GOTO 8380
8240 INPUT "ENTER # OF SPOTS YOU ARE GETTING THICKNESS INFO FROM FOR EACH LAY
8250 ER",Spot
8260 FOR Layer=1 TO Nlayers
8270 Thickness=0
8280 FOR Place=1 TO Spot
8290 PRINT "DIGITIZE TOP OF LAYER";Layer
8300 DIGITIZE A,Br
8310 BEEP
8320 PRINT "DIGITIZE BOTTOM OF LAYER";Layer
8330 DIGITIZE Cr,D
8340 BEEP
8350 Thickness=SQR((Br-D)^2+(A-Cr)^2)*Scal/400+Thickness
8360 NEXT Place
8370 Thickness=SQR((Br-D)^2+(A-Cr)^2)*Scal/400+Thickness
FOR N=1 TO Nlayers
  PRINT "ENTER ASSUMED THICKNESS OF LAYER"; N; "IN METERS"
  INPUT Thickness
  Yyy(N)=Thickness
NEXT N
IF Prtout=1 THEN PRINTER IS 702
PRINT "LISTING OF LAYER LENGTHS AND THICKNESSES"
FOR N=1 TO Nlayers
  IF Subarea(N)=0 THEN GOTO 8540
  Xxxx(N)=Subarea(N)/Yyy(N)
  IF Hardcop=2 THEN GOTO 8520
  Xthick=DROUND(Xxxx(N),3)
  Ythick=DROUND(Yyy(N),3)
  PRINT "UNDEFORMED LENGTH OF LAYER"; N; "WAS"; Xthick; "METERS"
  PRINT "THICKNESS"; Ythick; "METERS"
NEXT N
PRINT "DO YOU WANT TO CHANGE SLIGHTLY THE LAYER THICKNESSES? Y=1, N=2", Ee
k
IF Eek=1 THEN GOTO 8380
!STATISTICAL ANALYSIS
FOR N=1 TO Nlayers
  X_sum=X_sum+Xxxx(N)
NEXT N
Meanlength=X_sum/Nlayers
Meanlength=DROUND(Meanlength,3)
PRINT "AVERAGE LAYER LENGTH IS"; Meanlength; "METERS"
PRINT "DO YOU WANT TO USE A DIFFERENT MEANLENGTH?"
INPUT "YES=1, NO=2", Newmean
IF Newmean=2 THEN GOTO 8690
INPUT "ENTER DESIRED MEANLENGTH", Meanlength
PRINT "MEAN LENGTH IS"; Meanlength; "METERS"
PRINT "SAMPLE STANDARD DEVIATION=", Stddev; "METERS"
Maxit=MAX(Xxxx(M))
CALL Minn(Minit,Xxxx(*),Nlayers)
Diff=Maxit-Minit
Errorit=Diff/Meanlength
PRINT "PERCENT ERROR=", Errorit*100; "%"
PRINT "DO YOU WANT A PLOT OF UNDEFORMED LAYERS? Y=1, N=2", Plt
IF Plt=2 THEN GOTO 9830
PRINTER IS 1
FOR N=1 TO Nlayers
  IF Xxxx(N)*400/(Nratio*Scal)>150 THEN
    PRINT "WILL NOT FIT WITH CURRENT SIZE ON PLOTTER"
    GOTO 5090
  ELSE
    GOTO 8960
END IF
FOR N=1 TO Nlayers
  Qq(N)=Xxxx(N)
  Xxxx(N)=Xxxx(N)*400/Scal
INPUT "DO YOU WANT PLOTTER(1) OR CRT(2) PLOT?", Quest
INPUT "DO YOU WANT MEAN LENGTH PLOTTED FOR ALL? Y=1,N=2", Mn
IF Mn=1 THEN MAT Xxxx = (Meanlength*400/Scal)
IF Quest=1 THEN
PLOTTER IS 705,"HPGL"
ELSE
PLOTTER IS 3,"INTERNAL"
END IF
INPUT "ENTER PEN FOR WEDGE", Penn
PEN Penn
MOVE 0,0+10
FOR N=Nlayers TO 1 STEP -1
IF N=Nlayers THEN
DRAW Xxxx(N),0+10
ELSE
DRAW Xxxx(N),Y_tot+10
END IF
DRAW Xxxx(N),Yyyy(N)+Y_tot+10
Y_tot=Yyyy(N)+Y_tot
NEXT N
MOVE 0,0+10
DRAW 0,Y_tot+10
IF Nfaults=0 THEN GOTO 9140
IF Quest=1 THEN
INPUT "ENTER PEN FOR FAULTS", Penn
PEN Penn
END IF
Palin:
Section=1
Click=1
FOR J=1 TO Sds
IF Layerea(J)=0 THEN
Section=1+Section
J=J+1
GOTO 9430
END IF
IF Layerea(J)<0 THEN
Subtotal(Section)=Subtotal(Section)+Layerea(J)
END IF
NEXT J
FOR J=1 TO Section-Nfaults-2 STEP 1
Carea(J)=Subtotal(J)-Subtotal(J+1+Nfaults)+Sectionfault(J)*Scal^2/1603000
Carea(J)=Carea(J)*160000/Scal^2
NEXT J
Dave=Y_tot
DIM R(IO), Dist(IO)
Nosections=Section-Nfaults-2
Num_layer=Nosections/Nlayers
FOR J=1 TO Nlayers
R(J)=Y_tot-Yyyy(J)/2
Y_tot=Y_tot-Yyyy(J)
NEXT J
FOR J=1 TO Nlayers
Aratio(J)=Meanlength/Qq(J)
NEXT J
I=0
FOR J=1 TO Num_layer
I=I+1
H=I
FOR K=1 TO Nlayers
Dist(K)=Aratio(K)*(Carea(H)/Yyyy(K))+Dist(K)
NEXT K
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
NEXT J
DIM Geek(100), Yuck(100), S(100), T(100), C(10,10)
CALL Interval(R(*), Dist(*), 0, Dave, Nlayers, Geek(*), Yuck(*))
FOR K = 0 TO Nlayers - 1
    Geek(K) = Geek(K + 1)
    Yuck(K) = Yuck(K + 1)
NEXT K
CALL Poly(Geek(*), Yuck(*), Nlayers, S(*), T(*), A, B, C(*))
REDIM Geek(100)
REDIM Yuck(100)
PLOTTER IS 705,"HPGL"
FOR L = 1 TO A
    IF L = 1 THEN MOVE T(L), S(L) + 10
    DRAW T(L), S(L) + 10
NEXT L
NEXT J
SUBEND
1 ! *******************
10 ! * THREEDIM *
11 ! *******************
20 GINIT
30 PLOTTER IS 3,"INTERNAL"
40 GRAPHICS ON
50 DIM X(50),Y(50),H(50),Xx(50),Yy(50)
60 DIM Tags$(50)[7],Nx(2200),Ny(2200),Az(8,500),Bz(8,500)
70 DIM Nodigitpts(7),Forder(30,6),Click(50)
80 DEG
90 GOSUB 720
100 GOSUB 810
110 GOSUB 1220
120 PRINT "DO YOU WANT LOCATION OF ANY INFLECTION POINTS? Y=1"
130 INPUT Inflex
140 IF Inflex=1 THEN GOSUB 1510
150 GOSUB 620
160 Create file: !
170 !THIS PROGRAM SEGMENT DIGITIZES A CROSS SECTION AND PLACES IT
180 !IN A THREE DIMENSIONAL GRID, CORRECTLY. IT PUTS THIS INFORMATION INTO
190 !A FILE.
200 !FIRST POINT IS REFERENCE POINT. CHOOSE IT SO THAT IT IS LEFT
210 !MOST POINT.
220 CALL Echoing(X(*),Y(*),Ndat,Tags$(*)Struct,Click(*))
230 Refx=X(1)
240 Refy=Y(1)
250 FOR J=1 TO Struct
260 IF Click(J)=0 THEN GOTO 280
270 Y(J)=Refy-Y(J)
280 NEXT J
290 H(1)=Refelev
300 PRINT X(1),H(1)
310 FOR J=2 TO Struct
320 IF Click(J)=0 THEN GOTO 350
330 H(J)=H(1)-(Y(J)/400)*Scal
340 PRINT X(J),H(J)
350 NEXT J
360 FOR J=1 TO Struct
370 IF Click(J)=0 THEN GOTO 450
380 X(J)=X(J)-Refx
390 Xx(J)=COS(Sectionangle)*X(J)+Ax
400 IF Ss<0 THEN
410 Yy(J)=Ay-(Xx(J)-Ax)*ABS(Ss)
420 ELSE
430 Yy(J)=Ay+(Xx(J)-Ax)*ABS(Ss)
440 END IF
450 NEXT J
460 PRINT "XCOORD","YCOORD","ELEVATION"
470 FOR J=1 TO Struct
480 IF Click(J)=0 THEN GOTO 500
490 PRINT Tags$(J),Xx(J),Yy(J),H(J)

INPUT "ENTER NAME FOR THIS FILE",Filename$
CREATE BDAT "3_D"&Filename$,20
ASSIGN @Path TO "3_D"&Filename$
OUTPUT @Path;Ax,Ay,Cx,Cy
OUTPUT @Path;Click(*)
OUTPUT @Path;Xx(*)
OUTPUT @Path;Yy(*)
OUTPUT @Path;H(*)
OUTPUT @Path;Tags$(*)
STOP

Mapp: !***************************************************************************
INPUT "ENTER ELEVATION OF FIRST POINT IN METERS",Refelev
PRINT "DIGITIZE LEFT CROSS SECTION ENDPOINT (REF POINT)ON MAP"
GRAPHICS INPUT IS 705,"HPGL"
DIGITIZE Ax,Ay
PRINT "DIGITIZE OTHER END POINT OF CROSS SECTION"
DIGITIZE Cx,Cy
Ss=(Cy-Ay)/(Cx-Ax)
Sectionangle=ATN(ABS(Ss))
RETURN

Labeler: !***************************************************************************
INPUT "ENTER NUMBER OF STRUCTURAL FEATURES TO BE PROJECTED TOTAL",Struct
FOR I=1 TO Struct
PRINT "ENTER LABEL (7 LETTERS OR LESS) FOR STRUCTURE";I
INPUT Tags$(I)
NEXT I
RETURN

Read: !***************************************************************************
INPUT "IF POLYNOMIALS NOT FIT TO DATA, PRESS HERE(1)",Eek
INPUT "ENTER NAME OF FILE TO BE READ",Name$
IF Eek=1 THEN
ASSIGN @Path TO "BAL"&Name$
ON END @Path GOTO 930
ENTER @Path;Wx,Wy
J=J+1
Nx(J)=Wx
Ny(J)=Wy
GOTO 880
Sds=J-1
GOTO 1200
END IF
ASSIGN @Path TO "POLY"&Name$
ON END @Path GOTO 1000
ENTER @Path;Nx(*)
ENTER @Path;Ny(*)
J=1
IF Ny(J)<=0 THEN
J=J+1
GOTO 1010
J=J+1
GOTO 1030
GOTO 1010
END IF
Sds=J-1
FOR J=1 TO Sds
IF Nx(J)=30000 THEN Check=1
NEXT J
IF Check=0 THEN
Nfaults=0
GOTO 1200
END IF
Two$="FAULT"
ASSIGN @File TO Two$&Name$
1100 \texttt{\textbackslash \textbackslash @file;Nfaults}
1110 \texttt{\textbackslash @file;Nodigitpts(*)}
1120 \texttt{\textbackslash @file;Az(*)}
1130 \texttt{\textbackslash @file;Bz(*)}
1140 \texttt{RETURN}
1150 !********************************************************************
1160 Draw:!
1170 \texttt{PLOTTER IS 3,"INTERNAL"}
1180 \texttt{FOR R=1 TO Sds}
1190 \quad \texttt{IF R=1 \then MOVE \textit{Nx(1)},\textit{Ny(1)}}
1200 \quad \texttt{IF R=Sds \then GOTO 1410}
1210 \quad \texttt{IF \textit{Nx(R)}=20000 \then}
1220 \quad \quad \texttt{MOVE \textit{Nx(R+1)},\textit{Ny(R+1)}}
1230 \quad \texttt{GOTO 1400}
1240 \quad \texttt{ELSE}
1250 \quad \quad \texttt{GOTO 1330}
1260 \quad \texttt{END IF}
1270 \quad \texttt{IF \textit{Nx(R)}=30000 \then}
1280 \quad \quad \texttt{MOVE \textit{Nx(R+1)},\textit{Ny(R+1)}}
1290 \quad \texttt{GOTO 1400}
1300 \quad \texttt{ELSE}
1310 \quad \quad \texttt{GOTO 1390}
1320 \quad \texttt{END IF}
1330 \quad \texttt{IF \textit{Nx(R)}-30000 \then}
1340 \quad \quad \texttt{MOVE \textit{Nx(R+1)},\textit{Ny(R+1)}}
1350 \quad \quad \texttt{GOTO 1400}
1360 \quad \quad \texttt{ELSE}
1370 \quad \quad \quad \texttt{PLOT \textit{Nx(R)},\textit{Ny(R)}}
1380 \quad \quad \quad \texttt{NEXT R}
1390 \quad \quad \texttt{IF Nfaults=0 \then GOTO 1500}
1400 \quad \quad \texttt{NOW THE FAULTS ARE DRAWN IN (IF THERE ARE ANY)}
1410 \quad \texttt{FOR Z=1 TO Nfaults}
1420 \quad \quad \texttt{FOR M=1 TO Nodigitpts(Z)}
1430 \quad \quad \quad \texttt{IF Az(Z,M)=0 \then GOTO 1480}
1440 \quad \quad \quad \texttt{IF Az(Z,M-1)=0 \then MOVE Az(Z,M),Bz(Z,M)}
1450 \quad \quad \quad \texttt{DRAW Az(Z,M),Bz(Z,M)}
1460 \quad \quad \quad \texttt{NEXT M}
1470 \quad \quad \texttt{NEXT Z}
1480 \quad \texttt{RETURN}
1490 \texttt{Inf lex:!}
1500 \texttt{ASSIGN \texttt{\textbackslash @Myway} TO "INFL"&Name$}
1510 \texttt{ENTER \texttt{\textbackslash Myway;Porder(*)}}
1520 \quad \texttt{INPUT "ENTER SECTION NUMBER YOU WANT INFLECTION POINT (NONE=0)",Sectionno}
1530 \quad \texttt{IF Sectionno=0 \then GOTO 1850}
1540 \quad \texttt{Times=0}
1550 \texttt{FOR K=1 TO Sds}
1560 \quad \texttt{IF \textit{Nx(K)}=20000 \then GOTO 1610}
1570 \quad \texttt{IF \textit{Nx(K)}=30000 \then GOTO 1610}
1580 \quad \texttt{GOTO 1700}
1590 \quad \texttt{PRINT \textit{Nx(K)}}
1600 \quad \texttt{Times=+Times}
1610 \quad \texttt{IF \textit{Times}=\textit{Sectionno}-1 \then Stpt=K+1}
1620 \quad \texttt{IF \textit{Sectionno}=1 \then Stpt=1}
1630 \quad \texttt{IF \textit{Times}=\textit{Sectionno} \then}
1640 \quad \quad \texttt{Endpt=K-1}
1650 \quad \quad \texttt{PRINT Endpt}
1660 \quad \quad \texttt{GOTO 1710}
1670 \quad \texttt{END IF}
1680 \texttt{NEXT K}
1690 \texttt{I=1}
1700 \texttt{FOR J=Stpt TO Endpt}
1710 \quad \texttt{PRINT J}
1720 \quad \texttt{Lx=Porder(Sectionno,I)}
1730 \quad \texttt{IF \textit{Nx(J)}=Lx \then}
1740 \quad \quad \texttt{It=J}
1750 \quad \quad \texttt{MOVE Lx,Ny(It)}
1760 \quad \quad \texttt{@SIZE 4}
1770 \quad \quad \texttt{LONG 5}
1780 \quad \quad \texttt{return "",}
SUB Echoing(Aa(*), Bb(*), J, Tags$(*), Struct, Click(*))

Xx=50
Yy=50

SET ECHO Xx, Yy

ON KEY 5 LABEL "LEFT" GOTO 1990
ON KEY 6 LABEL "RIGHT" GOTO 2020
ON KEY 7 LABEL "UP" GOTO 2050
ON KEY 8 LABEL "DOWN" GOTO 2080
ON KEY 9 LABEL "MENU" GOTO 2450

GOTO Spin1

X=-.5
Y=0
GOTO Spin2
X=.5
Y=0
GOTO Spin2
X=0
Y=.5
GOTO Spin2
X=0
Y=-.5
GOTO Spin2

ON KNOB .01 GOTO 2150
GOTO 2130
SET ECHO Xx, Yy
Xx=Xx+X
Yy=Yy+Y
ON KBD GOTO Spin3

INPUT "DO YOU WANT TO DIGITIZE A POINT HERE? Y=1, STOP=2", Digit
IF Digit=1 THEN
INPUT "ENTER STRUCTURE IDENTIFIER CODE LETTER", Code$
FOR I=64 TO 64+Struct
IF Code$=CHR$(I) THEN J=I-64
NEXT I
PRINT J
Click(J)=1
Aa(J)=Xx-X
Bb(J)=Yy-Y
PEN 1
LORG 5
CSIZE 4
MOVE Aa(J), Bb(J)
LABEL " +"
MOVE Aa(J), Bb(J)+3
LABEL Code$
END IF
IF Digit=2 THEN
GRAPHICS OFF
GOTO 2520
END IF
GOTO Spin1
PRINT 
FOR I=1 TO Struct
RETURN
2470 NEXT I
2480 PRINT "
2490 INPUT "PRESS 1 TO REMOVE THE MENU", R
2500 IF R=1 THEN ALPHA OFF
2510 GOTO Spini
2520 SUBEND
2530!***************************************************************************
* PROJECTION *

**DEG**

**GINIT**

**PLOTTER IS 3,"INTERNAL"**

**GRAPHICS ON**

**N=0**

**MAT Yarray= (1000000)**

**PRINTER IS 1**

**DIM Z(50),Y(50),X(50),Xxx(50),Yyy(50),Yarray(5,50)**

**DIM R(50),Xdist(50),Ydist(50),X1(50),Y1(50),Z1(50),H(50),Xarray(5,50)**

**DIM Radi(5,50),Z2(50),Name$(50),Click(50),Thecount(50)**

**DIM Tag$(50),Lab$(50),Rtotal(50)**

**DIM Weighteddev(10,50),Xcoord(50),Ycoord(50)**

**INPUT "ENTER SCALE IN THE FORM 1: (FILL IN BLANK)",Scal**

**INPUT "DO YOU WANT TO DIGITIZE ON PLOTTER(1) OR CRT(2)?",Digdev**

**GOSUB 3110**

**IF Digdev=1 THEN**

**GRAPHICS INPUT IS 705,"HPGL"**

**DIGITIZE A,B**

**PRINT "DIGITIZE OTHER ENDPOINT"**

**DIGITIZE C,D**

**PRINT "DIGITIZING FINISHED"**

**END IF**

**IF Digdev=2 THEN**

**CALL Echoing(A,B,C,D)**

**INPUT "PRESS 1 TO CLEAR SCREEN",M**

**IF M=1 THEN GRAPHICS OFF**

**END IF**

**IF A=C THEN**

**Ss2=0**

**ELSE**

**Ss2=(D-B)/(C-A)**

**END IF**

**Sectionangle2=ATN(ABS(Ss2))**

**N=0**

**REPEAT**

**N=N+1**

**PRINT "THIS IS CROSS SECTION";N,"WHICH IS CALLED",Name$(N)**

**ASSIGN @Path1 TO "3D"&Name$(N)**

**ON END @Path1 GOTO 370**

**ENTER @Path1;Ax,Ay,Cx,Cy,Click(*),X(*),Y(*),Z(*),Tag$(*)**

**J=0**

**FOR K=1 TO 50**

**IF Click(K)=1 THEN J=J+1**

**NEXT K**

**Ndat=J**

**PRINT Ndat**

**Numb(N)=Ndat**

**PRINT "DIGITIZED DATA POINTS"**

**PRINT "XCOORD","YCOORD","ELEVATION"**

**FOR J=1 TO 50**

**IF Click(J)=0 THEN GOTO 480**

**PRINT X(J),Y(J),Z(J),Tag$(J)**

**NEXT J**

**MAT Z1= (O)**

**FOR J=1 TO 50**

**IF Click(J)=0 THEN GOTO 530**

**GOSUB 1300**

**GOSUB 1950**

**GOSUB 2180**

**NEXT J**

**INPUT "DO YOU WANT TO VIEW THIS PROJECTION ON THE SCREEN? Y=1. N=2",View**
IF `viewit=1` THEN GOSUB 2300

FOR J=1 TO 50
IF `Click(J)=0` THEN GOTO 573
Thecount(J)=1+Thecount(J)
NEXT J
MAT X=(0)
MAT Y=(0)
MAT Z=(0)
UNTIL N=Ncross

Weighted_avg:
!THIS PROGRAM OPTION DEDUCES THE WEIGHTED
!AVERAGE POINT OF A SCATTER OF PROJECTED POINTS
!FROM VARIOUS PROJECTED CROSS SECTIONS.
GOSUB 3390
INPUT "DO YOU WANT TO PLOT THE LEAST SQUARE FIT?, Y=1",Ls
IF `Ls=2` THEN GOTO 1290
INPUT "ENTER PEN NUMBER",Penn
PRINT "ENTER THE POINT LETTER THAT YOU WISH TO PERFORM A WEIGHTED"
PRINT "AVERAGE ON"
INPUT "FIN=ALL DONE",J$
IF J$="FIN" THEN GOTO 1270
FOR K=0 TO 49
IF `CHR$(65+K)=J$` THEN I=K
NEXT
I=I+1
MAT Weighted_dev=(0)
Rtot=0
!RTOT IS THE SUM OF THE ALL R's FOR ONE SPECIFIC POINT.
J=I
Nn=0
LOOP
Nn=Nn+1
EXIT IF Nn=Ncross+1
IF `Xarray(Nn,J)=0 AND Yarray(Nn,J)=0` THEN GOTO 850
Rtot=Rtot+(Radi(Nn,J))
END LOOP
Rtotal(J)=Rtot
Xtot=0
Ytot=0
!WEIGHTED_DEV REPRESENTS THE INACCURACY FOR A SPECIFIC
!POINT IN COMPARISON WITH OTHERS.
Nn=0
LOOP
Nn=Nn+1
EXIT IF Nn=Ncross+1
IF `Xarray(Nn,J)=0 AND Yarray(Nn,J)=0` THEN GOTO 970
IF Radi(Nn,J)=0 THEN GOTO 970
Weighted_dev(Nn,J)=Rtotal(J)/Radi(Nn,J)
Xtot=Weighted_dev(Nn,J)*Xarray(Nn,J)+Xtot
Ytot=Weighted_dev(Nn,J)*Yarray(Nn,J)+Ytot
END LOOP
Sumweighted_dev=0
Nn=0
LOOP
Nn=Nn+1
EXIT IF Nn=Ncross+1
IF `Xarray(Nn,J)=0 AND Yarray(Nn,J)=0` THEN GOTO 1070
Sumweighted_dev=Weighted_dev(Nn,J)+Sumweighted_dev
END LOOP
!XCOORD AND YCOORD ARE THE DETERMINED AVERAGE POINTS
!AND ARE THE PLOTTED VALUES.
Xcoord(J)=Xtot/Sumweighted_dev
Ycoord(J)=Ytot/Sumweighted_dev
PRINT "WEIGHTED AVERAGE OF PROJECTED POINT";J
PRINT "HAS COORDINATES X=";Xcoord(J);"Y=";Ycoord(J)
PRINT "DDTVTD CC 1"
MOVE Xcoord(J),(Ycoord(J)-Minelev)*400/Scal

LABEL "*

GOTO 760

INPUT "DO YOU WANT TO LABEL PLOT ETC.? Y=1",Labe

IF Labe=1 THEN GOSUB 2670

STOP

1300 Trend: THIS SUBROUTINE TAKES THE TREND OF THE STRUCTURE INTO ACCOUNT

1310 !IT PROJECTS THE DIGITIZED POINTS DIRECTLY ALONG THE TREND

1320 !AND DETERMINES THE RESULTING INTERSECTION POINT WITH THE

1330 !DESIRED CROSS SECTION.

1340 !XXX AND YYY ARE THE INTERSECTION POINTS OF X AND Y WHEN THEY ARE

1350 !PROJECTED ALONG A PERPENDICULAR PATH TO THE DESIRED CROSS SECTION.

1351 INPUT "ENTER TREND",T

1352 INPUT "EAST(E) OR WEST(W)",Trendir$

1360 Xxx(J)=(A*Ss2^2-Ss2*(B-Y(J))+X(J))/(1+Ss2^2)

1370 IF Ss2=0 THEN

1380 Yyy(J)=A

1390 GOTO 1530

1400 ELSE

1410 GOTO 1430

1420 END IF

1430 Yyy(J)=(Xxx(J)-X(J))/(-1*Ss2)+Y(J)

1450 IF Trendir$="W" THEN

1460 IF Ss2<0 THEN

1470 Newang=ABS(T+Sectionangle2)

1480 ELSE

1490 Newang=ABS(T-Sectionangle2)

1500 END IF

1510 ELSE

1520 IF Ss2<0 THEN

1530 Newang=ABS(T-Sectionangle2)

1540 ELSE

1550 Newang=ABS(T+Sectionangle2)

1560 END IF

1570 END IF

1580 !R AND M REPRESENT ORTHOGONAL DISTANCES TO THE PROJECTED POINT.

1590 R(J)=SQR((Xxx(J)-X(J))^2+(Yyy(J)-Y(J))^2)

1600 M=R(J)*TAN(Newang)

1610 Xdist(J)=M*COS(Sectionangle2)

1620 Ydist(J)=M*SIN(Sectionangle2)

1630 IF Yyy(J)>Y(J) THEN Xdist(J)--1*Xdist(J)

1640 IF X(J)>Xxx(J) THEN Ydist(J)--1*Ydist(J)

1650 !X1 AND Y1 ARE THE ACTUAL INTERSECTIONS OF THE PROJECTED POINTS.

1660 !AT THIS POINT THEY HAVE BEEN PROJECTED ALONG A HORIZONTAL PATH

1670 IF Trendir$="E" THEN

1680 IF Ss2>0 THEN

1690 X1(J)=Xxx(J)-Xdist(J)

1700 Y1(J)=Yyy(J)-Ydist(J)

1710 ELSE

1720 IF T>Sectionangle2 THEN

1730 X1(J)=Xxx(J)-Xdist(J)

1740 Y1(J)=Yyy(J)+Ydist(J)

1750 ELSE

1760 X1(J)=Xxx(J)+Xdist(J)

1770 Y1(J)=Yyy(J)+Ydist(J)

1780 END IF

1790 END IF

1800 ELSE

1810 IF Ss2<0 THEN

1820 X1(J)=Xxx(J)+Xdist(J)

1830 Y1(J)=Yyy(J)+Ydist(J)

1840 ELSE

1850 IF T>Sectionangle2 THEN

1860 X1(J)=Xxx(J)+Xdist(J)
```plaintext
1890 XI(J)=Xxx(J)-Xdist(J)
1900 Y1(J)=Yyy(J)-Ydist(J)
1910 END IF
1920 END IF
1930 END IF
1940 RETURN
1950 Plunge:  NOW WE ARE GOING TO TAKE PLUNGE INTO ACCOUNT
1960 INPUT "ENTER PLUNGE",P1
1970 !RR IS THE HORIZONTAL DISTANCE THE PROJECTED POINT TRAVELS.
1980 Rr(J)=SQR((XI(J)-X(J))^2+(Y1(J)-Y(J))^2)*Scal/400
1990 Radi(N,J)=Rr(J)/COS(P1)
2000 !H IS THE CHANGE IN ELEVATION AS A RESULT OF THE PROJECTION
2010 !WITH PLUNGE TAKEN INTO ACCOUNT.
2020 H(J)=Rr(J)*TAN(P1)
2030 !Z1 IS THE RESULTANT ELEVATION WITH PLUNGE TAKEN INTO ACCOUNT.
2040 IF Plungdir$="N" THEN
2050 IF Y1(J)<Y(J) THEN
2060 Z1(J)=H(J)+Z(J)
2070 ELSE
2080 Z1(J)=Z(J)-H(J)
2090 END IF
2100 ELSE
2110 IF Y1(J)<Y(J) THEN
2120 Z1(J)=Z(J)-H(J)
2130 ELSE
2140 Z1(J)=H(J)+Z(J)
2150 END IF
2160 END IF
2170 RETURN
2180 Rotation:  NOW WE HAVE TO ROTATE INTERMEDIATE CROSS SECTION TO BE PARALLEL
2190 !TO THE PLOTTER BASE IN ORDER TO DO A LEAST SQUARES REGRESSION.
2200 !ALL XI POINTS MUST BE ROTATED AND THEIR VALUES ADJUSTED.
2210 !X2 IS THE NEW POINT ON THE ROTATED CROSS SECTION.
2220 !L IS THE DISTANCE FROM XI TO THE END OF THE CROSS SECTION.
2230 L=SQR((XI(J)-A)^2+(Y1(J)-B)^2)
2240 Xx2(J)=A+L
2250 Xarray(N,J)=Xx2(J)
2260 Yarray(N,J)=Z1(J)
2270 RETURN
2280 Viewit:  THIS SUBROUTINE PLOTS THE RESULTING CROSS SECTION ON THE
2290 !SCREEN.  IT PLOTS THE TICK-MARKS AND ELEVATION AND AXIS.
2300 GINIT
2310 PLOTTER IS 3,"INTERNAL"
2320 GRAPHICS ON
2330 GINIT 5
2340 CALL Minn(Minz, Z1(*), 50)
2350 FOR J=1 TO 50
2360 IF Click(J)<>0 THEN GOTO 2460
2370 Z1(J)=400*Z1(J)/Scal
2380 Z1(J)=Z1(J)-400*Minz/Scal
2390 MOVE Xx2(J),Z1(J)+10
2400 LABEL "+"
2410 MOVE Xx2(J),Z1(J)+15
2420 LABEL CHR$(64+J)
2430 NEXT J
2440 MOVE Xx2(1),10
2450 DRAW MAX(Xx2(*))+3,10
2460 DRAW MAX(Xx2(*))+3,MAX(Z1(*))+10
2470 FOR K=1 TO (MAX(Xx2(*))-Xx2(1))/4
2480 MOVE K*4+Xx2(1),5+10
2490 DRAW K*4+Xx2(1),0+10
2500 NEXT K
2510 FOR K=1 TO (MAX(Z1(*))/4)
```

NEXT K
MOVE MAX(Xx2(*))+20,MAX(Z1(*))+10
LABEL INT((MAX(Z1(*))*Scal/400+Minz));"Meters"
MOVE MAX(Xx2(*))+20,1+10
LABEL INT(Minz);"Meters"
PENUP
PRINT "PRESS 1 IF YOU WANT TO REMOVE PLOT FROM THE SCREEN"
INPUT Clr
IF Clr=1 THEN GCLEAR
RETURN
Tickmark:INPUT "ENTER THE PEN NUMBER YOU WANT FOR LABELLING",Colour
PEN Colour
MOVE Xcoord(1),0
DRAW MAX(Xcoord(*))+5,0
DRAW MAX(Xcoord(*))+5,(MAX(Ycoord(*))-Minelev)*400/Scal
!THIS SUBROUTINE DRAWS THE TICK MARKS
!IT IS USED FOR WEIGHTED AVERAGE PLOT ONLY
FOR K=1 TO (MAX(Xcoord(*))-Xcoord(1))/4
MOVE K*4+Xcoord(1),0
NEXT K
FOR K=1 TO (MAX(Ycoord(*))-Minelev)*100/Scal
MOVE MAX(Xcoord(*))+4,4*K
DRAW MAX(Xcoord(*))+5,4*K
NEXT K
MOVE MAX(Xcoord(*))+15,1
LABEL INT(Minelev);"METERS"
MOVE MAX(Xcoord(*))+15,(MAX(Ycoord(*))-Minelev)*400/Scal
LABEL INT(MAX(Ycoord(*)));"METERS"
PEN 0
RETURN
Tickmarkii:!THIS SUB DRAW T TICK MARKS AND AXES FOR INDIVIDUAL
PEN Colour
MOVE Xarray(N,1),0
DRAW MAX(Xarray(*))+4,0
DRAW MAX(Xarray(*))+4,(MAX(Z1(*))-Minelev)*400/Scal
FOR K=1 TO (MAX(Xarray(*))-Xarray(N,1))/4
MOVE K*4+Xarray(N,1),.5
DRAW K*4+Xarray(N,1),0
NEXT K
FOR K=1 TO (MAX(Z1(*))-Minelev)*100/Scal
MOVE MAX(Xarray(*))+4,4*K
DRAW MAX(Xarray(*))+3.5,4*K
NEXT K
CSIZE 2
MOVE MAX(Xarray(*))+13,.5
LABEL INT(Minelev);"METERS"
MOVE MAX(Xarray(*))+13,(MAX(Z1(*))-Minelev)*400/Scal
LABEL INT(MAX(Z1(*)));"METERS"
PEN 0
RETURN
!***********************************************************************
N=1
LOOP
INPUT "ENTER CROSS SECTION NAME FOR PROJECTING PURPOSES, FIN=DONE",Nam
e$(N)
EXIT IF Name$(N)="FIN"
ASSIGN @Path1 TO Name$(N)
GOSUB 3220
N=N+1
END LOOP
Ncross=N-1
LET "ENTER @Path1; Ax; Ay; Cx; Cy"
3240 LABEL CHR$(64+N)
3250 MOVE Ax, Ay
3260 DRAW Cx, Cy
3270 MOVE Cx, Cy+4
3280 LABEL CHR$(64+N);'"'
3290 ASSIGN @Path1 TO *
3300 RETURN
3390 Picture: THIS IS A PROGRAM OPTION WHICH ALLOWS PROJECTED CROSS SECTIONS TO BE PLOTTED ON THE PLOTTER
3400 GINIT
3410 PLOTTER IS 705,"HPGL"
3420 GRAPHICS ON
3430 CSIZE 2
3440 INPUT "ENTER PEN COLOR NUMBER", Colour
3450 Minelev=MIN(Yarray(*)
3460 INPUT "ENTER THE CROSS SECTION NUMBER THAT YOU WANT TO PLOT", U
3470 PEN Colour
3480 LORG 5
3490 FOR J=1 TO 50
3500 IF Yarray(U,J)-1000000 THEN GOTO 3550
3510 MOVE Xarray(U,J),(Yarray(U,J)-Minelev)*400/Scal
3520 LABEL J
3530 NEXT J
3540 PEN 0
3550 INPUT "DO YOU WANT TO PLOT ANOTHER ONE? Y=1 N=2", Again
3560 IF Again=1 THEN GOTO 3470
3570 PRINT "DO YOU WANT TO PLOT TICKMARKS AND AXES? Y=1"
3580 INPUT Tick
3590 IF Tick=1 THEN GOSUB 2880
3600 RETURN
3610 END
3620 Echoo:SUB Echoing(A,B,C,D)
3630 Xx=50
3640 Yy=50
3650 SET ECHO Xx,Yy
3660 Spin1: !
3661 ON KEY 5 LABEL "LEFT" GOTO 3653
3662 ON KEY 6 LABEL "RIGHT" GOTO 3656
3663 ON KEY 7 LABEL "UP" GOTO 3659
3664 ON KEY 8 LABEL "DOWN" GOTO 3662
3665 ON KEY 9 LABEL "MENU" GOTO 3702
3666 GOTO Spin1
3667 X=-.5
3668 Y=0
3669 GOTO Spin2
3670 X=.5
3671 Y=0
3672 GOTO Spin2
3673 X=0
3674 Y=.5
3675 GOTO Spin2
3676 X=0
3677 Y=-.5
3678 GOTO Spin2
3679 X=-.5
3680 Y=0
3681 GOTO Spin2
3682 X=0
3683 Y=-.5
3684 GOTO Spin2
3685 Spin2: !
3686 LOOP
3687 ON KNOB .01 GOTO 3669
3688 GOTO 3667
3689 SET ECHO Xx,Yy
3670 Xx=Xx+X
3671 Yy=Yy+Y
3672 ON KBD GOTO Spin3
INPUT "DO YOU WANT TO DIGITIZE A POINT HERE? Y=1, STOP=2", Digit

IF Digit=1 THEN

IF Times=1 THEN

A=Xx-X
B=Yy-Y
MOVE A,B
LORG 5
LABEL "+"
GOTO Spin1
ELSE
C=Xx-X
D=Yy-Y
MOVE C,D
LORG 5
LABEL "+"
GOTO Spin1
END IF
END IF
SUBEND

Minn: SUB Minn(Ming, G(*), Numb)

FOR Gg=1 TO Numb
IF G(Gg)=0 THEN GOTO 3770
IF Ming=0 THEN Ming=G(Gg)
IF G(Gg)<Ming THEN Ming=G(Gg)
NEXT Gg
SUBEND

!**************************************************************************