

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 _____

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

Certified by _____

Kip Hodges
Thesis Supervisor

Accepted by _____

Chairman, Department Committee on Graduate Students

WITHDRAWN
FROM
MIT LIBRARIES
MASS. INST. TECH.
JUN 17 1986
LIBRARIES

Lindgren

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

| | page |
|--|------|
| Acknowledgements _____ | 4 |
| Introduction _____ | 5 |
| SECTION _____ | 8 |
| BALANCE _____ | 22 |
| THREEDIM _____ | 35 |
| PROJECTION _____ | 39 |
| Example from the Canadian Rockies _____ | 42 |
| Conclusions _____ | 53 |
| References _____ | 54 |
| Appendix 1 | |
| SECTION and EXTRASUBS program listings _____ | 56 |
| Appendix 2 | |
| BALANCE program listing _____ | 74 |
| Appendix 3 | |
| THREEDIM program listing _____ | 90 |
| Appendix 4 | |
| PROJECTION program listing _____ | 95 |

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 alot 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.

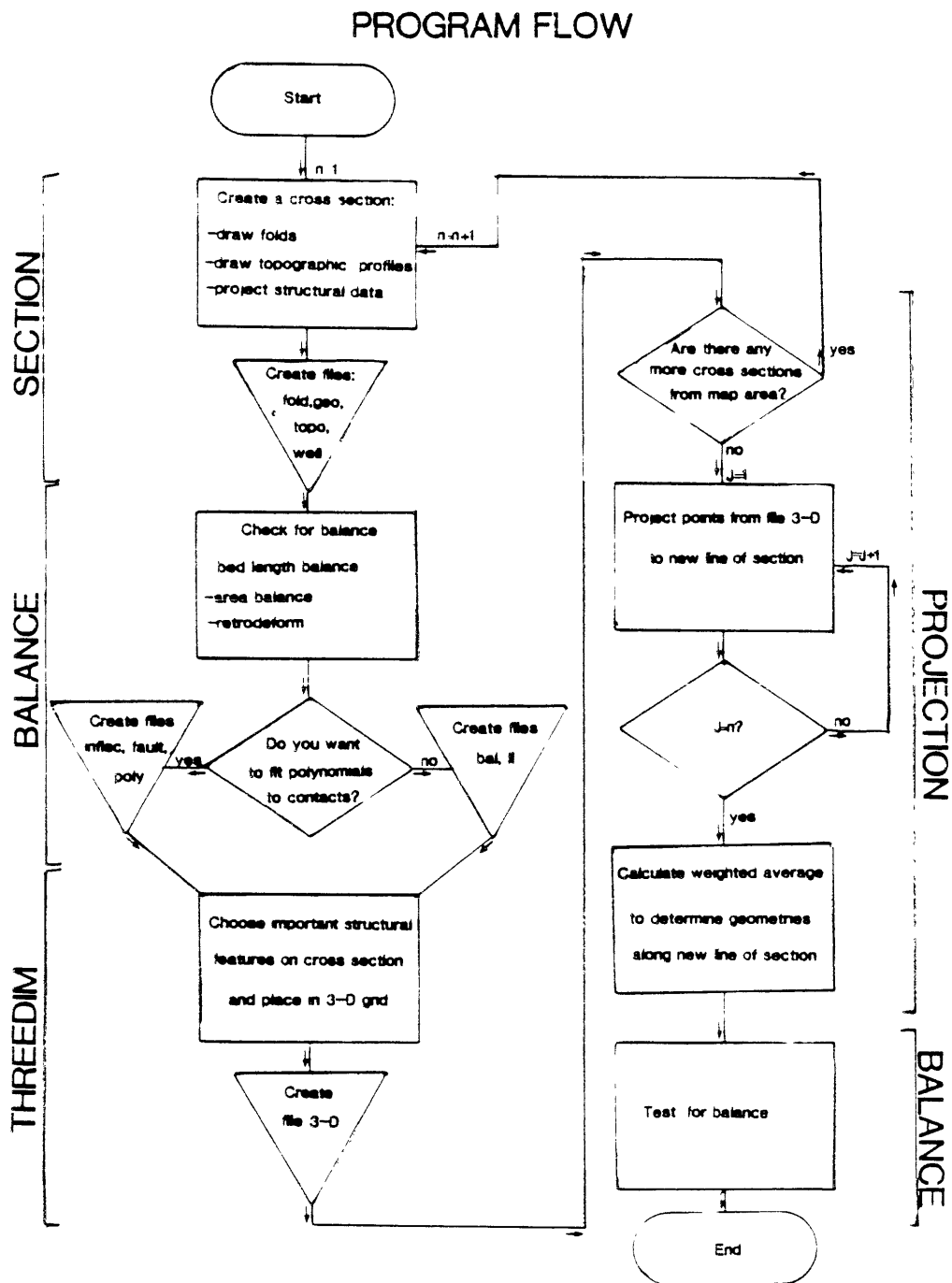


Figure 1 Flowchart illustrating general sequence of programs.

SECTION

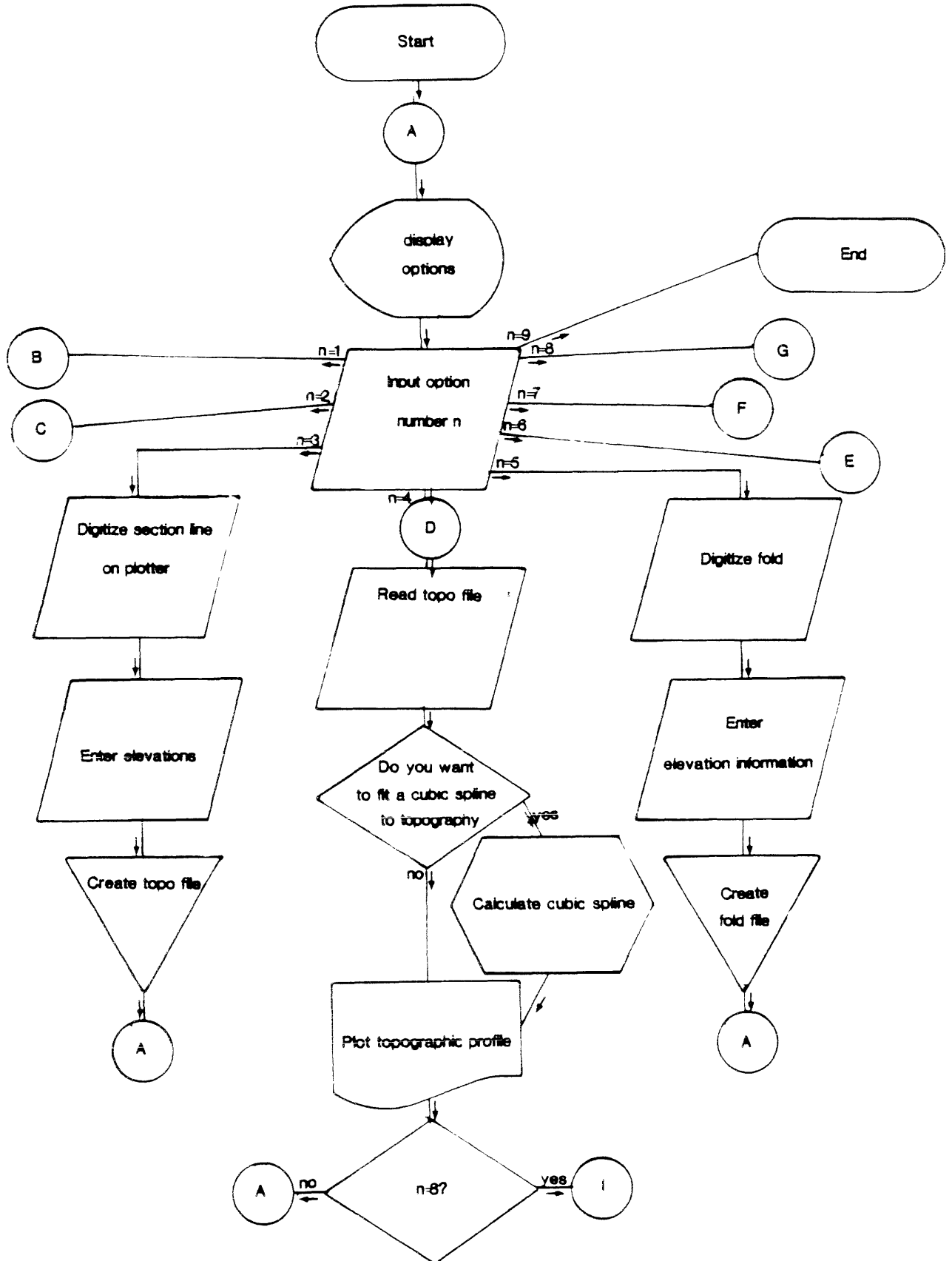
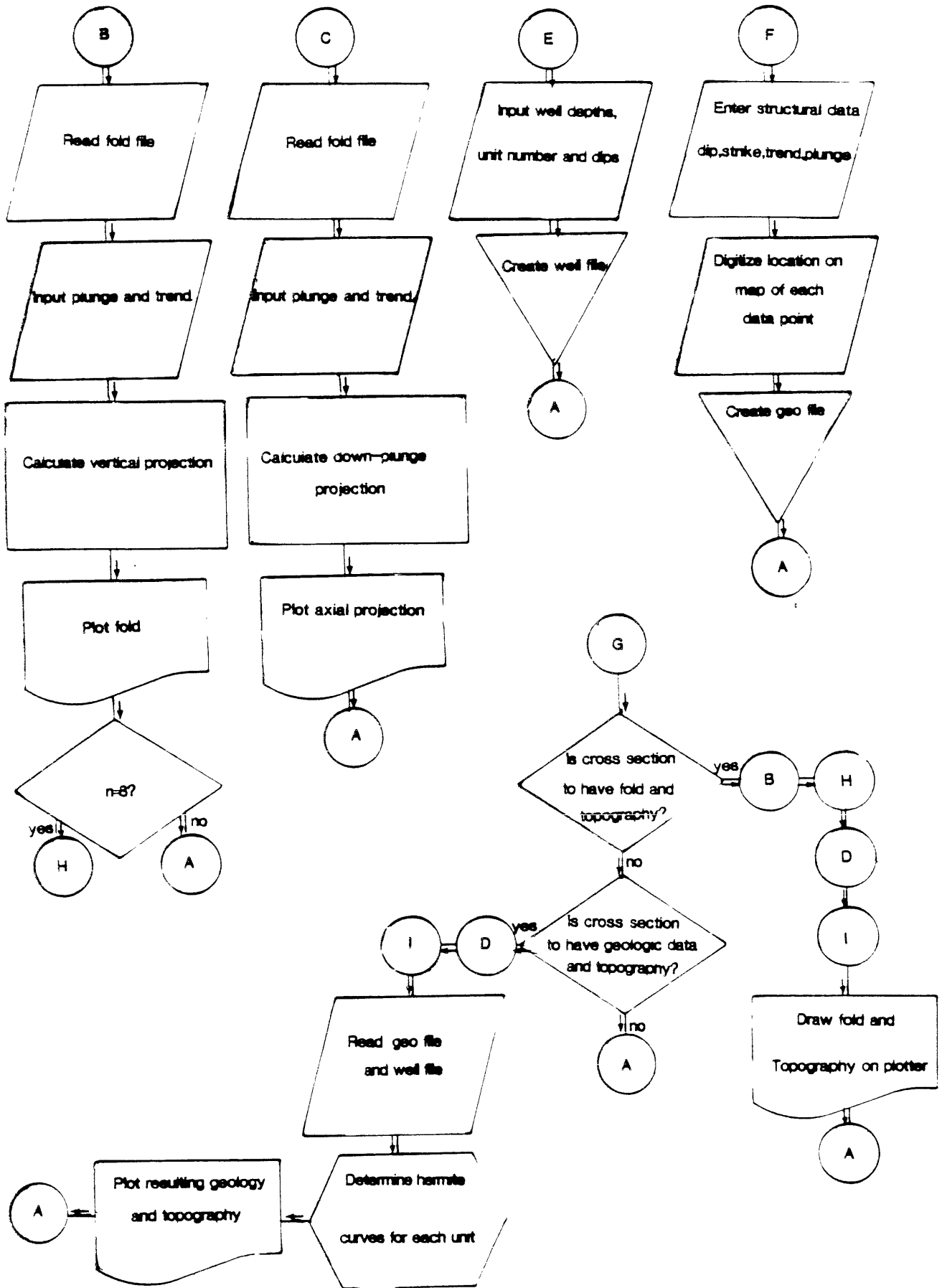


Figure 2 Flowchart for program SECTION.



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 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, $A=Y_0$ $B=S_0$ $C=Y_1$ $D=S_1$,

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.

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.

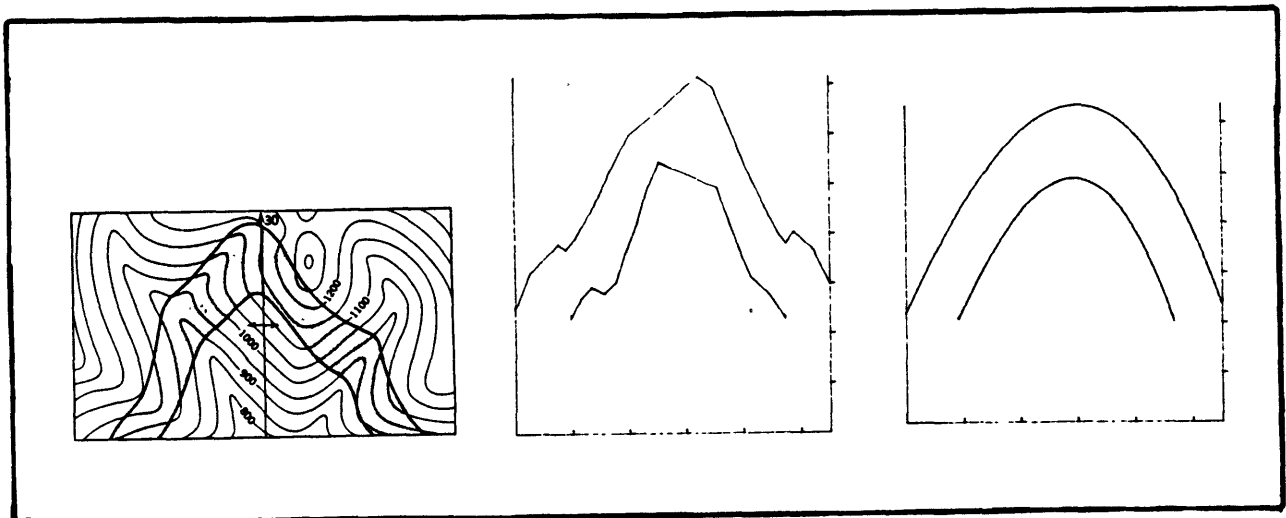


Figure 4 Straight line fit and polynomial fit to axial profile.

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:

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n \quad (3)$$

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:

$$A^T A C = A^T B \quad (4)$$

where,

$$A = \begin{bmatrix} 1 & x_0 & \cdot & \cdot & \cdot & x_0^{n-1} \\ 1 & x_1 & \cdot & \cdot & \cdot & x_1^{n-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_{m-1} & \cdot & \cdot & \cdot & x_{m-1}^{n-1} \end{bmatrix} \quad C = \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \cdot \\ \cdot \\ \cdot \\ a_0 \end{bmatrix} \quad B = \begin{bmatrix} Y_0 \\ Y_1 \\ \cdot \\ \cdot \\ \cdot \\ Y_{m-1} \end{bmatrix}$$

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.

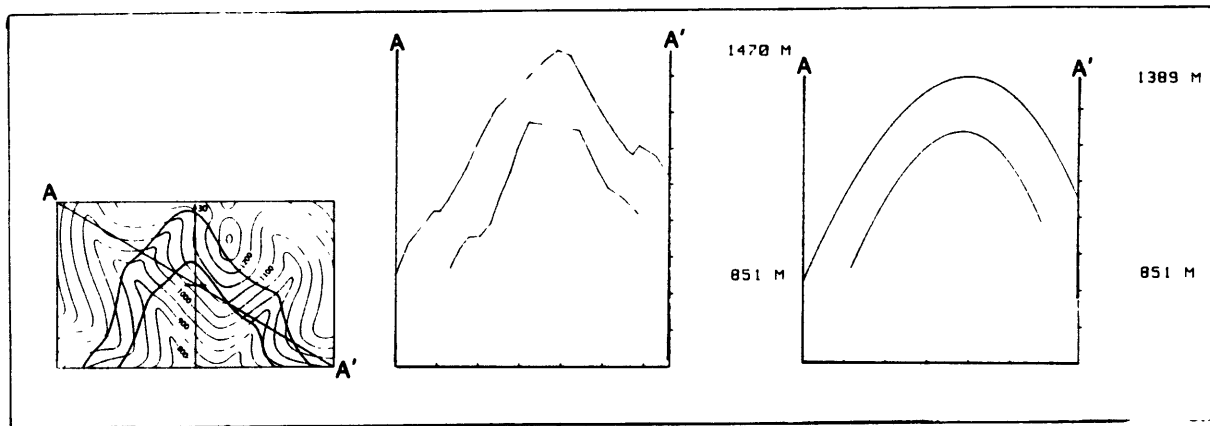


Figure 5 Vertical projection using FOLDDRAW.

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)

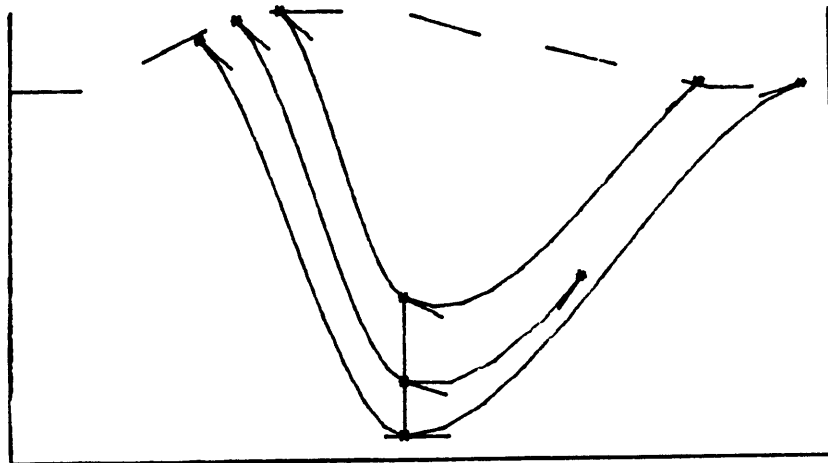


Figure 6 Hermite fit to geologic data.

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 WELLFILE1 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.

BALANCE

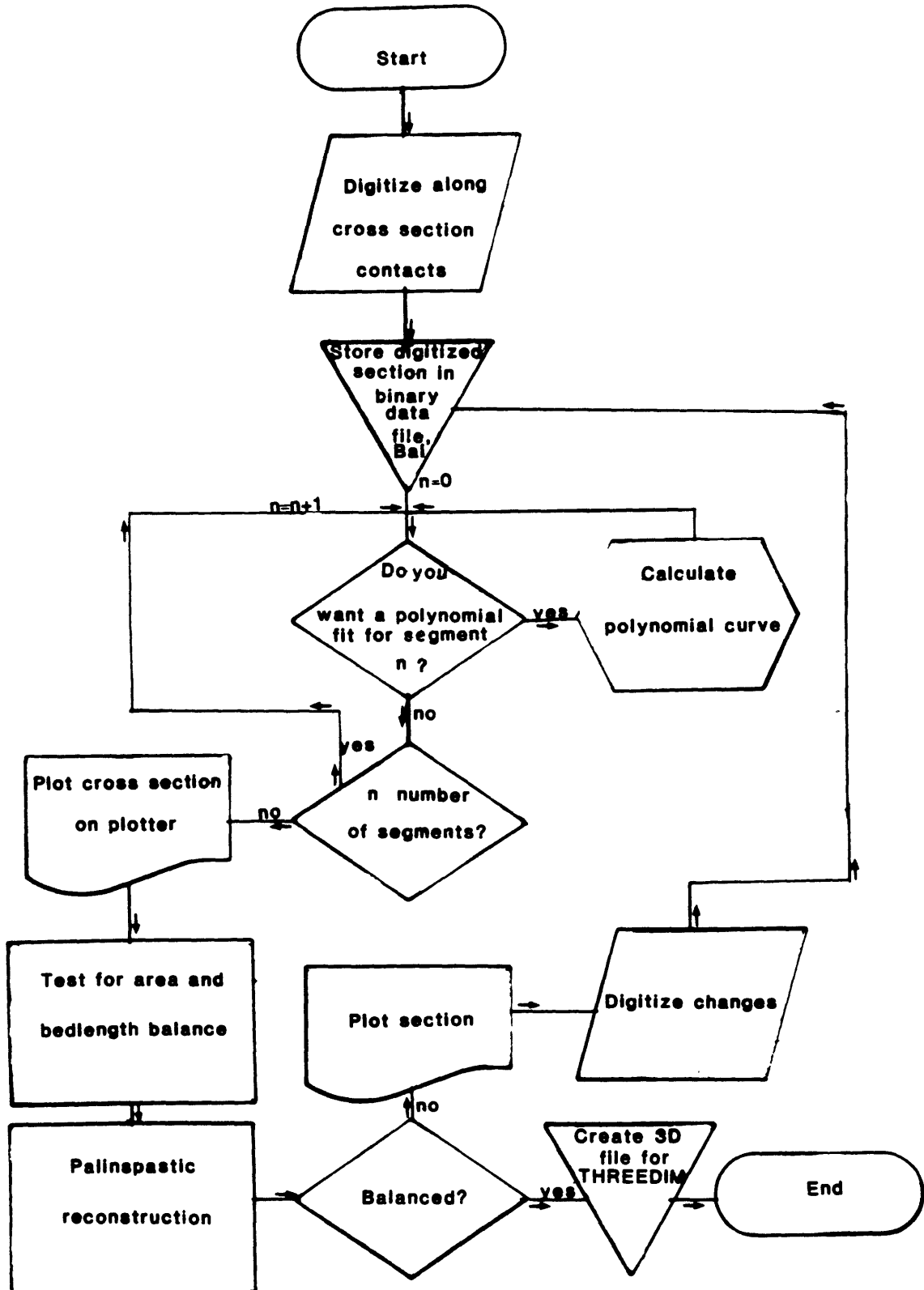


Figure 7 Flowchart for program BALANCE

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.

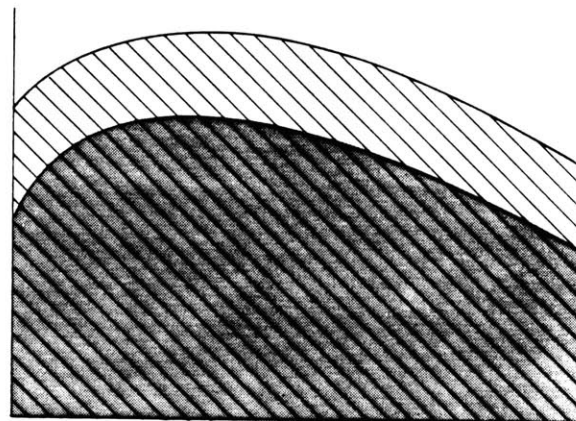
The polynomial curve which is generated is not actually a curve, but rather a collection of short connected line segments, each segment being 1/16 cm long in the x-direction. The result appears as a continuous curve. If folds have been digitized with care, for example hinge and trough points have been 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).



$$\text{Layer area} = \text{[diagonal lines]} - \text{[dense diagonal lines]}$$

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

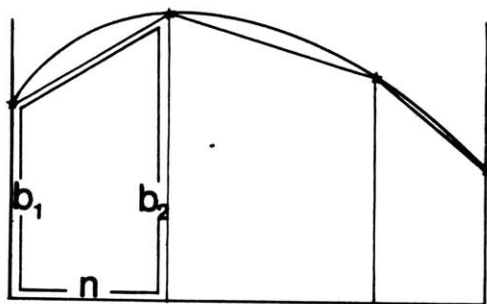


Figure 9 Trapezoidal areas.

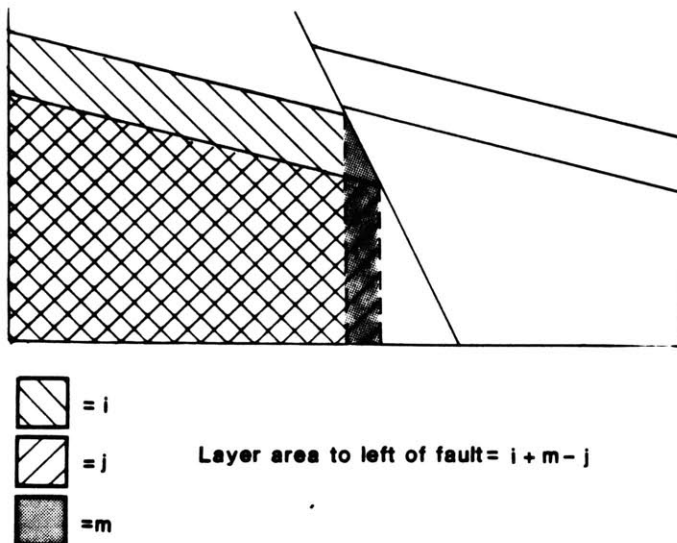


Figure 10 Determining fault areas.

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:

$$\%confidence = \frac{\text{maximum bed length} - \text{minimum bed length}}{\text{mean bed length}} \times 100 \quad (10)$$

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 (ϵ) is calculated independently with the formula:

$$\epsilon = \frac{L-L_0}{L_0} , \quad (11)$$

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 be by stepping faults (Fig. 11a). Instead, to give the fault a curved surface, a polynomial function is fit which goes through a

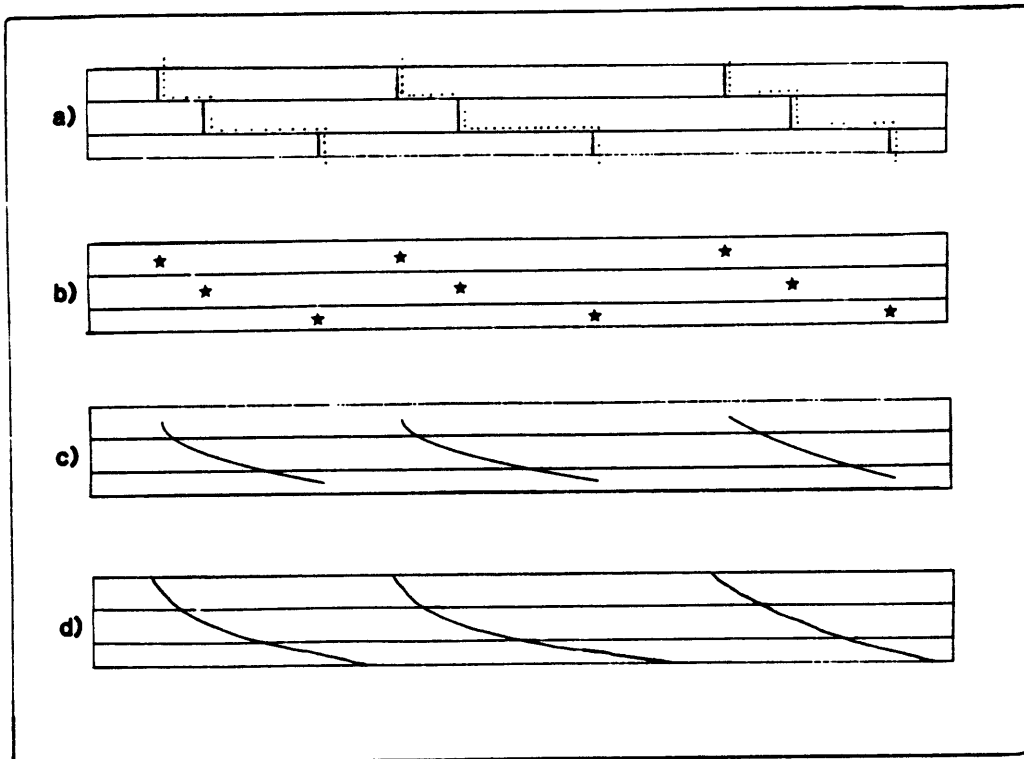


Figure 11 Refer to text.

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.

THREEDIM

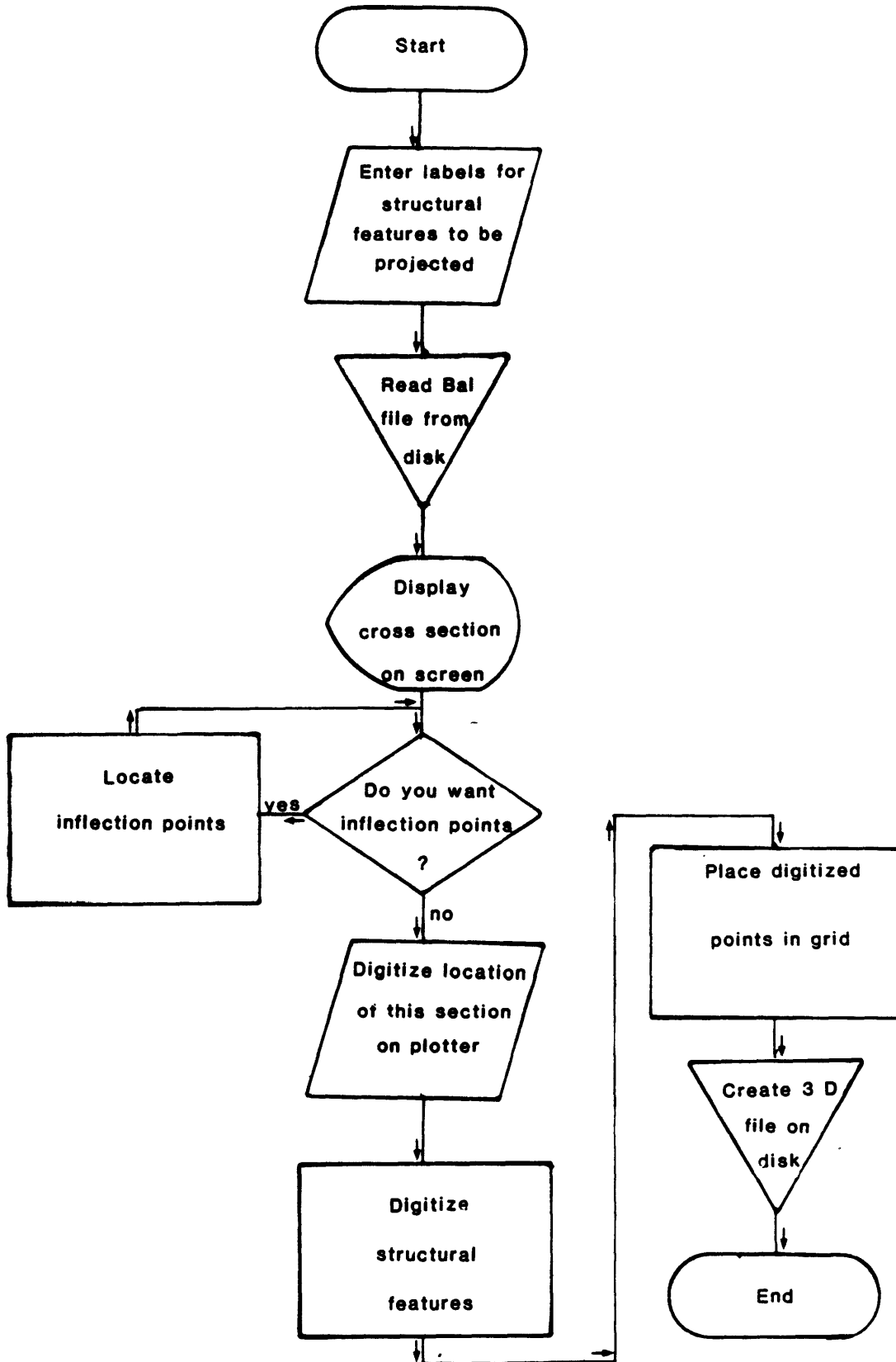


Figure 12 Flowchart for THREEDIM.

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.

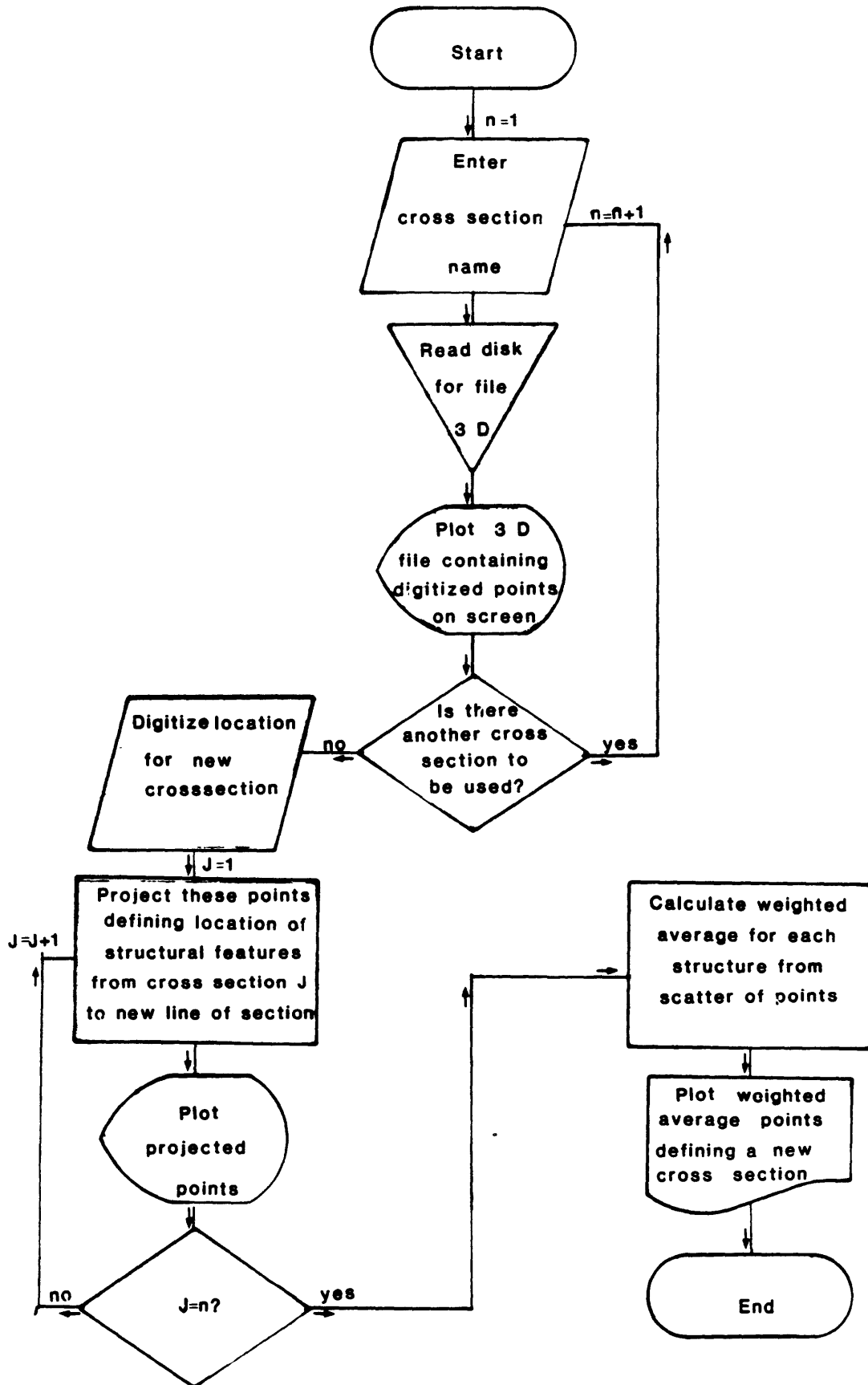


Figure 13 Flowchart for PROJECTION

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.

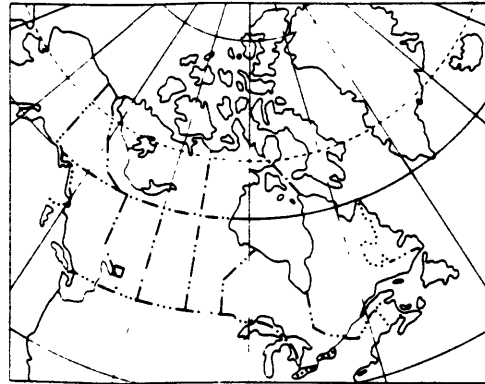
For each point which has been projected from a number of cross sections, a best fit point must be determined for the intermediate section. Therefore, the projected distance of each and every point from its cross section to the intermediate section is calculated. The average x and y coordinate value for 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.

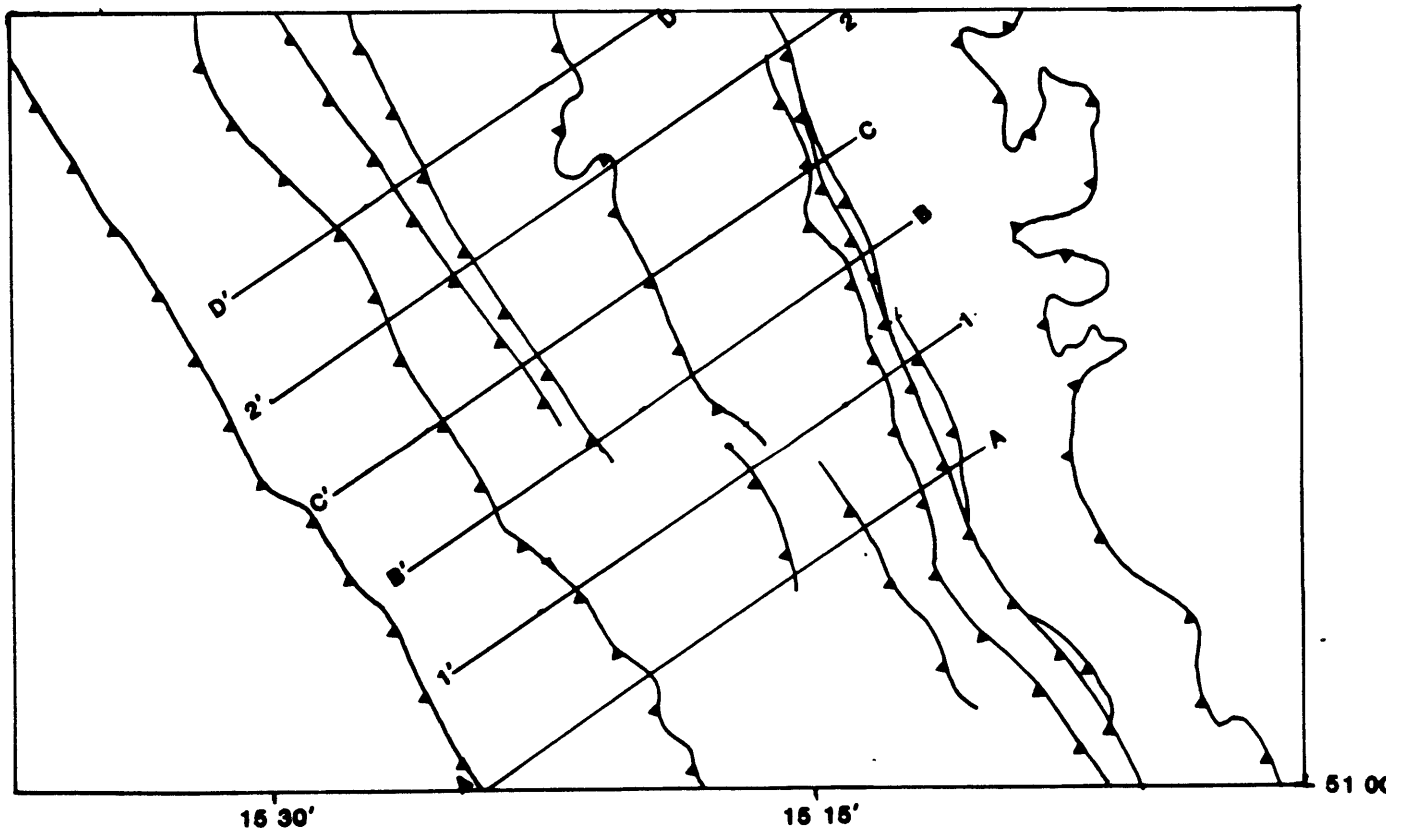
Example From the Canadian Front Ranges

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.



INDEX MAP

**Figure 14** Map showing location of cross sections.

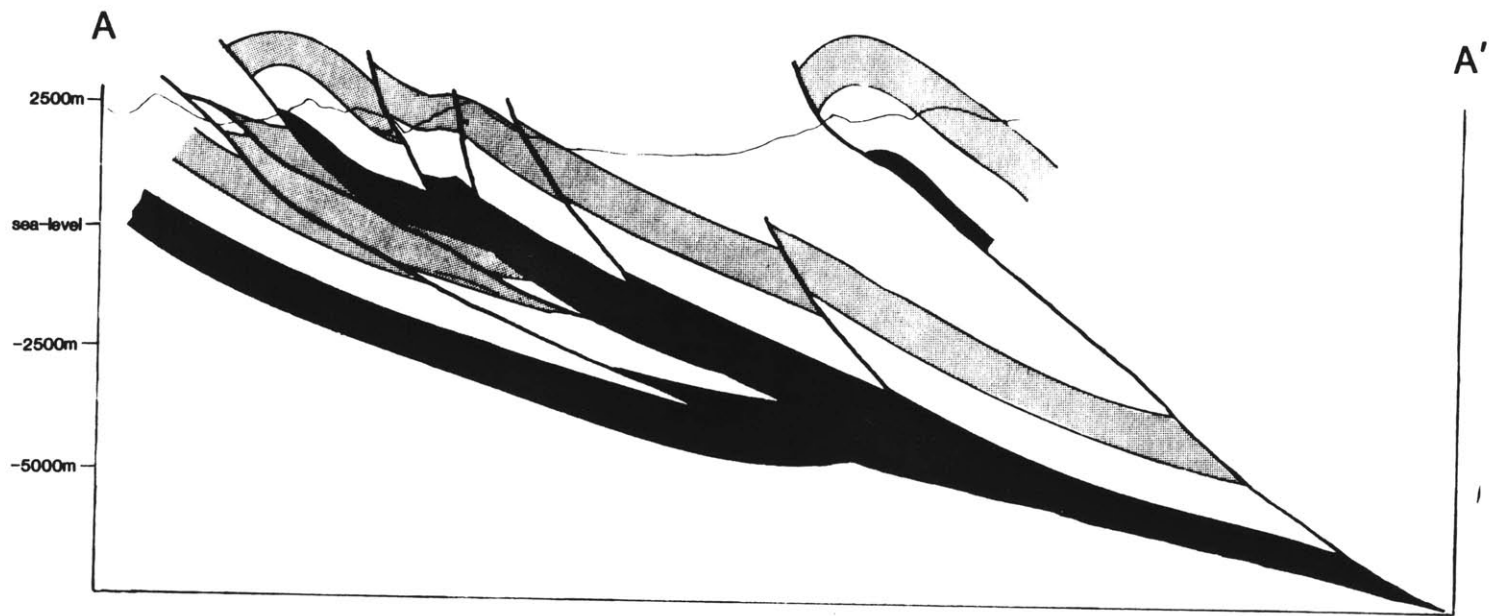


Figure 15a

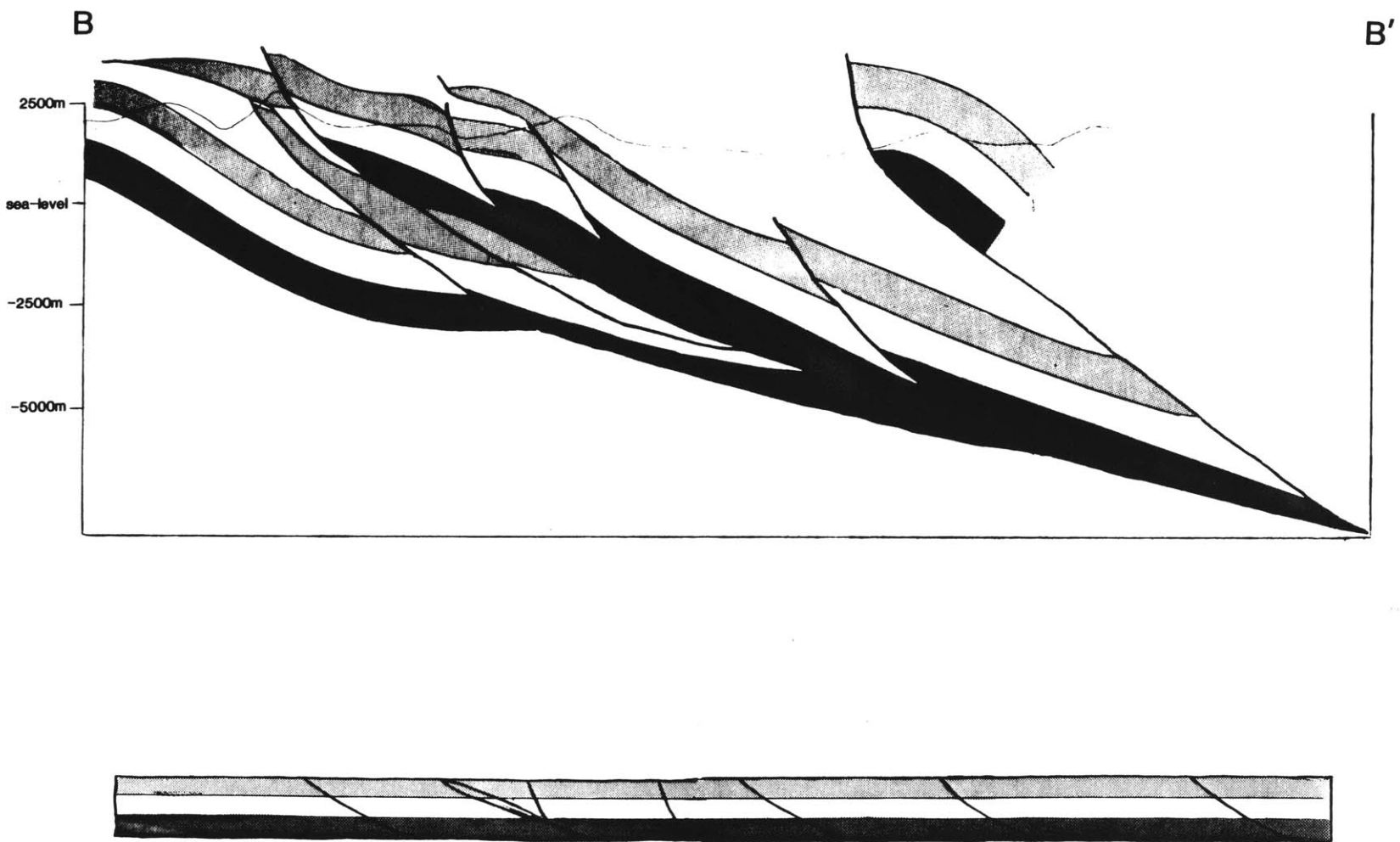


Figure 15b

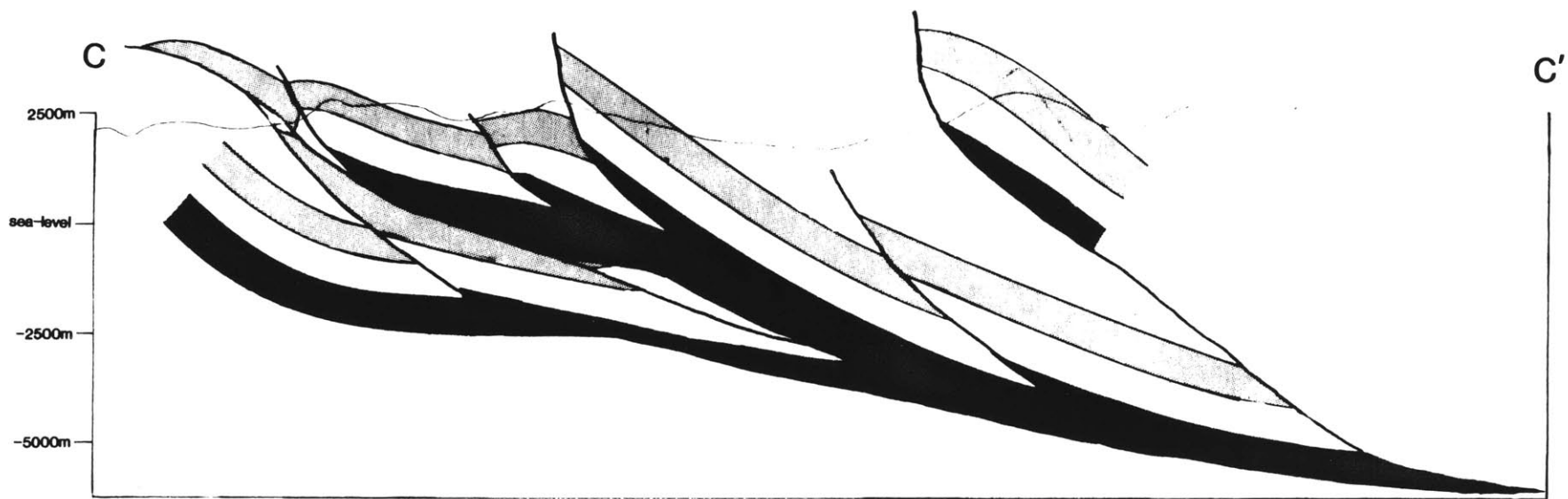


Figure 15c

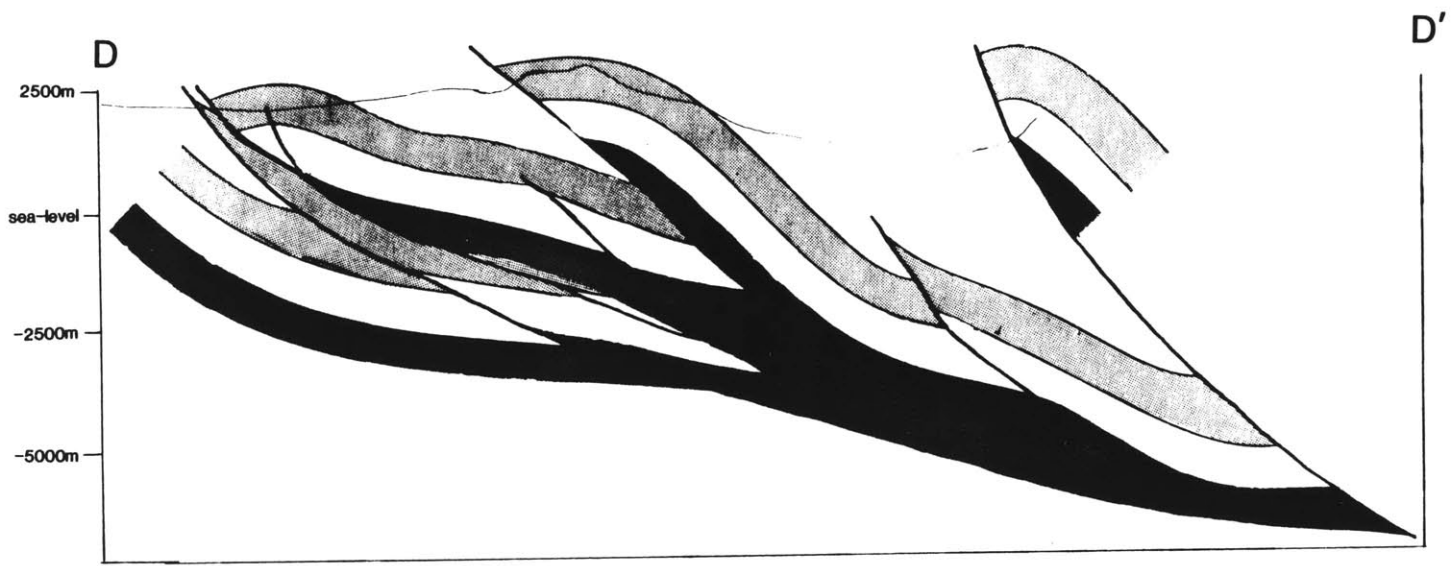


Figure 15d

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 "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.

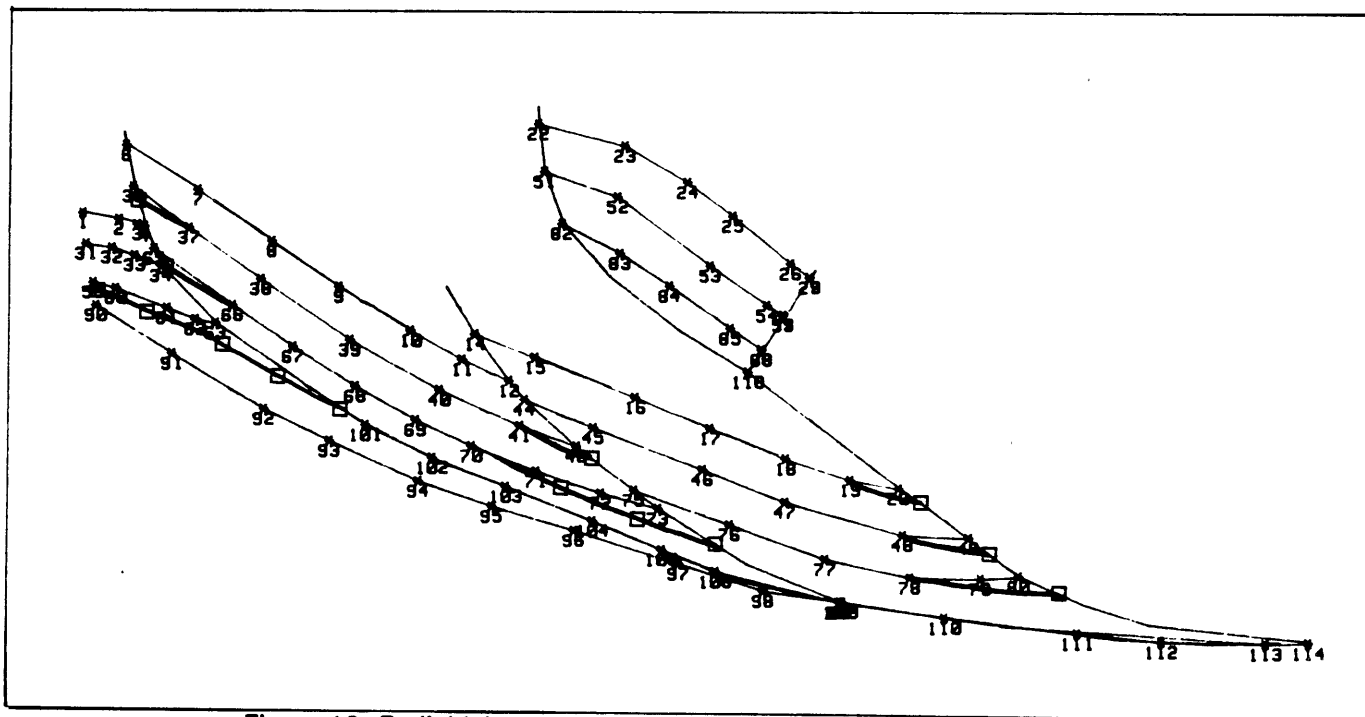


Figure 16 Redigitizing process- squares represent redigitized points.

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.

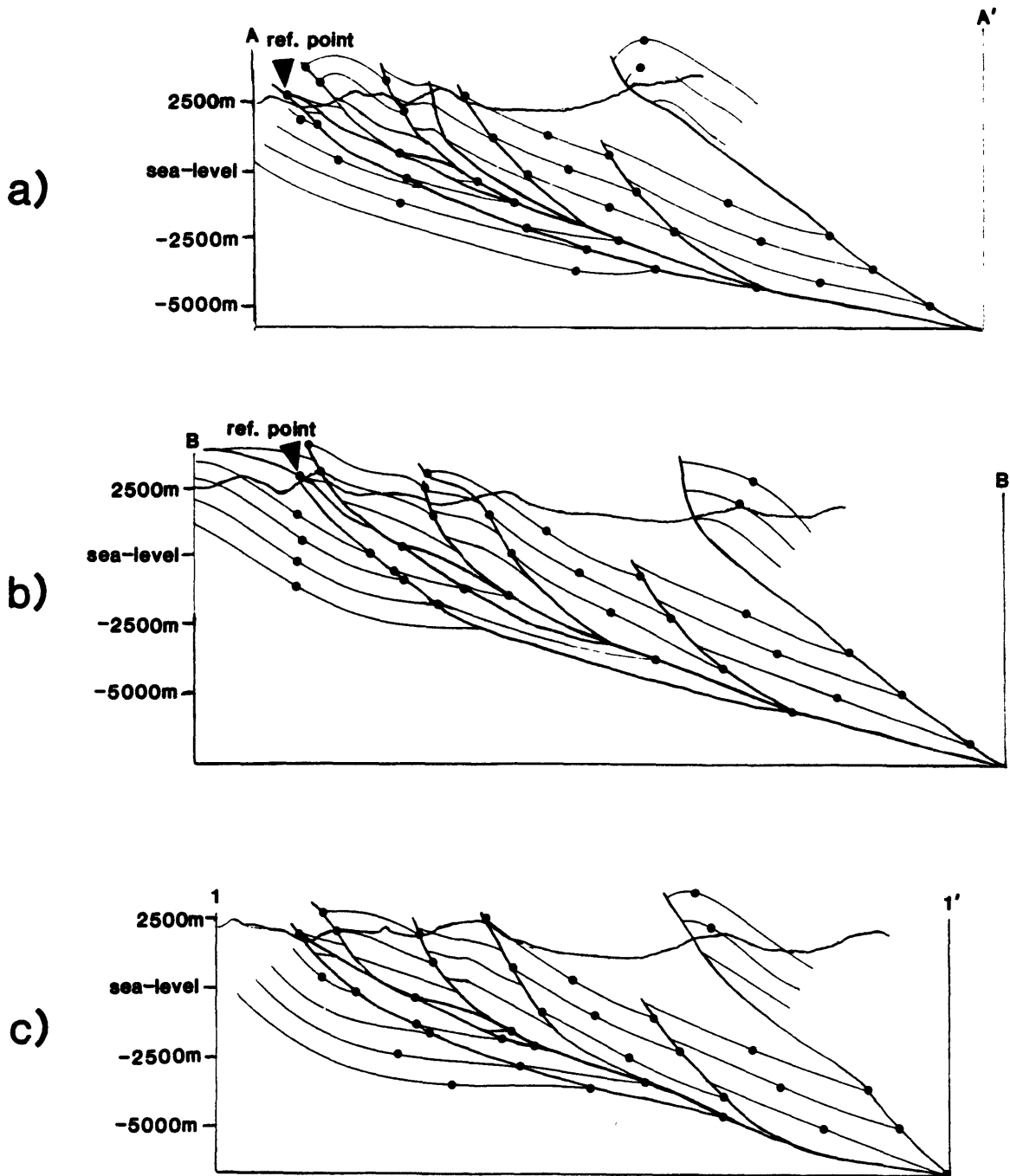
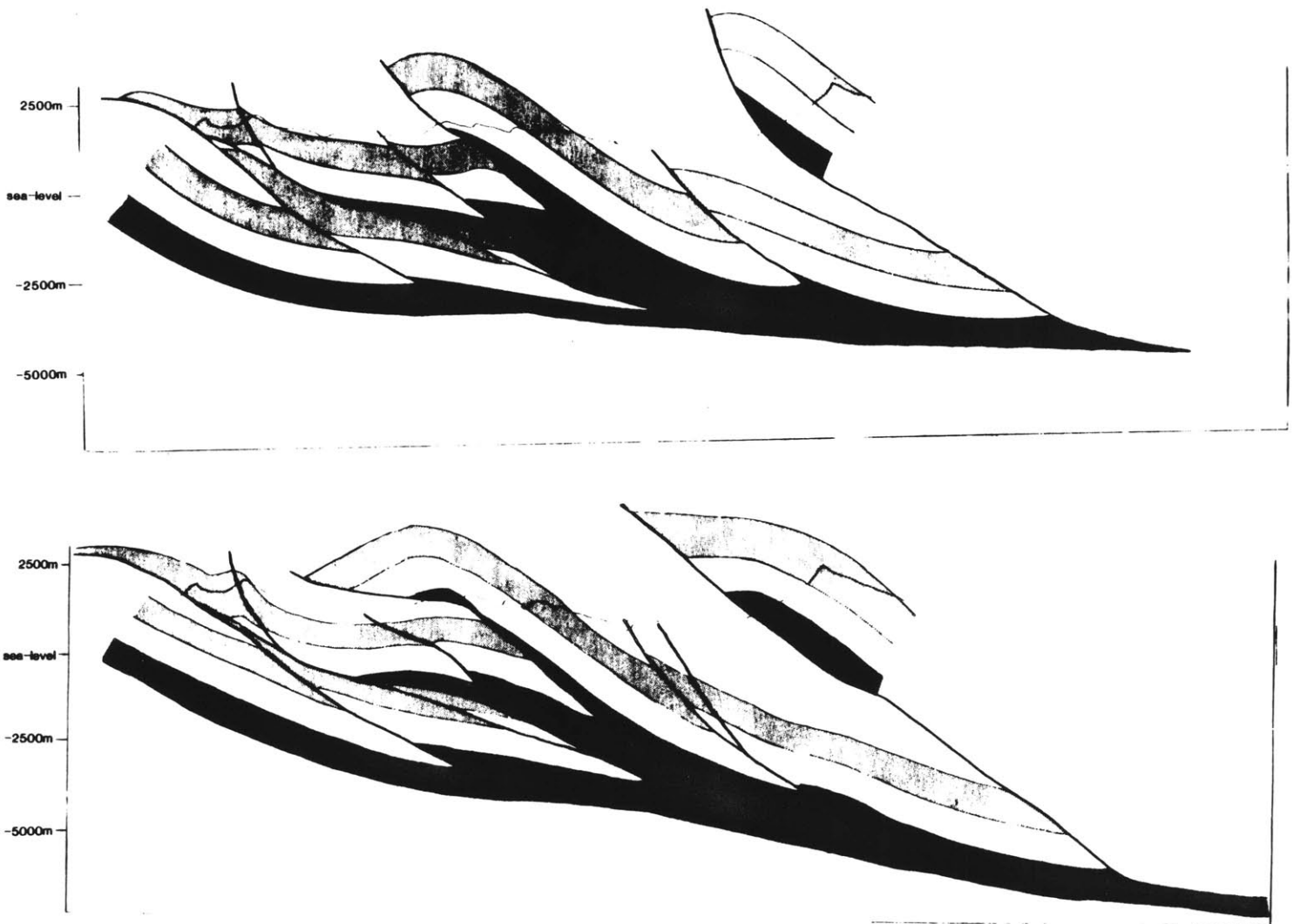


Figure 18 a&b) points projected using THREEDIM. c) resulting cross section.

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.
- Billings, M.P., 1954, *Structural Geology*: Prentice-Hall, Inc., Englewood Cliffs, N.J., 514 p.
- Boyer S.E., and Elliot, D., 1983, Thrust Systems: AAPG, v. 66, p.1196-1230.
- Bruce, C.J. and Frey, F.R., 1982, C.S.P.G. Trip No. 3, *Geology of the Southern Rocky Mountains Field Trip Guidebook*: Can. Soc. Pet. Geo., Calgary, 263 p.p.
- Dahlstrom, C.D.A., 1969, Balanced cross sections, *Can. Jour. Earth Sci.*, 6, p. 743-757.
- Elliot, D., 1983, The Construction of Balanced Cross sections: *Journal of structural Geology*, v.5, p. 101.
- Hobbs, Means, and Williams, 1976, *An Outline of Structural Geology*; John Wiley and Sons, Inc., New York, 571 p.
- Hossack, J.R., 1979, The use of balanced cross-sections in the calculation of orogenic contraction: a Review, *Journal Geol. Soc. Lond.*, 136, p. 705-711.
- Ollerenshaw, N.C., 1978, *Geology, Calgary, West of Fifth Meridian, Alberta and British Columbia*: Geologic Survey of Canada map 1457A.
- Price and Mountjoy, 1970, *Geology, Cranmore, West Half, Alberta*: Geological Survey of Canada map 1266A (1:50000 with structure sections)
- Price and Mountjoy, 1970, *Geology, Cranmore, East Half, Alberta*: Geological Survey of Canada map 1265A (1:50000 with structure sections)
- Price and Mountjoy, 1970, *Geologic Structure of the Canadian Rocky Mountains between Bow and Athabasca Rivers-A Progress Report*: *Spec. Pap. Geol. Soc. Can.*, v.6, p.7-25.
- Price and Fermor, 1985, *Structure Section of the Cordilleran Foreland Thrust and Fold Belt West of Calgary, Alberta*: Geological Survey of Canada Paper84-14.
- Ragan, D.L., 1973, *Structural Geology, An Introduction to Geometrical Techniques*, John Wiley and Sons, Inc., New York, 208 p.
- Ramsay, John G., 1967, *Folding and Fracturing of Rocks*: McGraw-Hill, New York, 568 p.

Riess R. D. and Johnson, L.W., 1982, Numerical Analysis: Addison-Wesley, Massachusetts, 563 p.p.

Strang, Gilbert, 1985, Introduction to Applied Mathematics:
Wellesley-Cambridge Press, Wellesley Mass., 758 p.p.

Suppe, J., 1985, Principles of structural geology: Prentice-Hall, Inc.
Englewood Cliffs, N.J., 537 p.

```

10                                     !*****
20                                     !***   SECTION   ***
30                                     !*****
40   DIM Hhh(1:150),Xxx(1:150),Ii(300),Jj(300)
50   COM /Comdata/ R,C,D,E,F,Dirac,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 "*"                                     *                                     *"
170  PRINT "          OPTION ONE                    *          OPTION TWO                    *"
180  PRINT "*"                                     *                                     *"
190  PRINT "    DIGITIZE TOPOGRAPHY (0)          *    DRAW TOPOGRAPHY (5)          *"
200  PRINT "    DIGITIZE GEOLOGY (1)            *    DRAW FOLD (6)            *"
210  PRINT "    DIGITIZE FOLD (2)               *    DRAW AXIAL PROF (7)          *"
220  PRINT "    CREATE WELL FILE (3)            *    DRAW CROSS SECTION (8)         *"
230  PRINT "*****"
240  Spinl: !
250  ON KEY 0 LABEL "TOPODIGIT" GOTO 390
260  ON KEY 1 LABEL "GEODIGIT" GOTO 450
270  ON KEY 2 LABEL "FOLDDIGIT" GOTO 480
280  ON KEY 3 LABEL "WELLFILE" GOTO 620
290  ON KEY 4 LABEL " " " GOTO 270
300  ON KEY 5 LABEL "TOPODRAW" GOTO 420
310  ON KEY 6 LABEL "FOLDDRAW" GOTO 510
320  ON KEY 7 LABEL "AXIAL" GOTO 590
330  ON KEY 8 LABEL "CROSSSECTION" GOTO 550
340  ON KEY 9 LABEL " " " GOTO 270
350  GOTO Spinl
360  CALL Topodigit(Topofile,New$)
370  GOTO 130
380  !
390  CALL Topodraw(Topofile,New$,Ntotpts,Topo,Mm,Ii(*),Jj(*),L)
400  GOTO 130
410  !
420  CALL Geodigit(Geofile,Nice$,C)
430  GOTO 130
440  !
450  CALL Geodigit(Geofile,Nice$,C)
460  GOTO 130

```

```

700 CALL Topodigit(Topofile,Name$)
490 GOTO 130
500 !
510 GOSUB 670
520 CALL Folddraw(Foldfile,Name$,Ndat,Fold)
530 GOTO 130
540 !
550 GOSUB 670
560 CALL Crs(Ntotpts,Ndat,Npts,Fold,Topo,Geo,Xxx(*),Hhh(*),G,Ii(*),Jj(*),L)
570 GOTO 130
580 !
590 CALL Axial(Scal,Foldfile,Name$,O(*),Oo(*))
600 GOTO 130
610 !
620 LOADSUB Wellfile1 FROM "EXTRASUBS"
630 CALL Wellfile1
640 DELSUB Wellfile1
650 GOTO 130
660 STOP
670 DEG
680 !*****
690 PRINT "DIGITIZE LEFT ENDPOINT"
700 DIGITIZE C,D
710 BEEP
720 PRINT "DIGITIZE RIGHT ENDPOINT"
730 DIGITIZE E,F
740 BEEP
750 PRINT "ENTER SCALE OF MAP AND CROSS SECTION IN THE FORM"
760 PRINT "1:_____ (FILL IN THE BLANK)"
770 INPUT Scal
780 IF C=E THEN
790   R=0
800 ELSE
810   R=ATN((ABS(F-D)/ABS(C-E)))
820 END IF
830 Direc=90-R
840 G=SQR((F-D)^2+(C-E)^2)
850 RETURN
860 END
870 !*****
880 Topodigit:SUB Topodigit(Topofile,New$)
890 DIM X(40),X2(1:40),H(1:40),H2(40),Y(40)
900 PRINT "START DIGITIZING TOPOGRAPHY FROM LEFT TO RIGHT"
910 PRINT "WHEN DONE DIGITIZING PRESS 1 FOR THE ELEVATION"
920 M=1
930 LOOP
940 PRINT "ENTER ELEVATION OF POINT",M
950 INPUT H(M)
960 EXIT IF H(M)=1
970 M=M+1
980 END LOOP
990 M=M-1
1000 Ndat=M
1010 FOR M=1 TO Ndat
1020 PRINT "DIGITIZE POINT",M,"ELEVATION IS";H(M)
1030 DIGITIZE X(M),Y(M)
1040 BEEP
1050 NEXT M
1060 R=ATN((ABS(Y(Ndat)-Y(1))/ABS(X(1)-X(Ndat))))
1070 FOR M=1 TO Ndat
1080 X2(M)=X(M)/COS(R)
1090 NEXT M
1100 INPUT "WHAT WOULD YOU LIKE TO CALL THE NEW FILE?",New$
1110 CREATE BDAT "TOPO"&New$,1,700
1120 ASSIGN @File TO "TOPO"&New$

```

```

1170 Foldfile=1
1150 SUBEND
1160 !*****
1170 Folddigit:SUB Folddigit(Foldfile,Name$)
1180 J=0
1190 INPUT "WHAT WOULD YOU LIKE TO CALL THE NEW FILE?",Name$
1200 CREATE BDAT "FOLD"&Name$,1,40000
1210 ASSIGN @File TO "FOLD"&Name$
1220 DIM X(150),Y(150),H(150),O(150),Oo(105),X1(150),Y1(150),H1(150)
1230 J=1
1240 PRINT "START DIGITIZING FOLD. WHEN DONE DIGITIZING, PRESS 1 FOR ELEVATI
ON"
1250 LOOP
1260 PRINT "ENTER ELEVATION OF POINT",J
1270 INPUT H(J)
1280 EXIT IF H(J)=1
1290 PRINT "POINT";J;"ELEVATION";H(J);"METERS"
1300 J=J+1
1310 END LOOP
1320 J=J-1
1330 Ndat=J
1340 FOR J=1 TO Ndat
1350 PRINT "DIGITIZE POINT";J;"ELEVATION IS";H(J)
1360 DIGITIZE X(J),Y(J)
1370 IF J=1 THEN GOTO 1480
1380 IF X(J)=X(J-1) THEN
1390 IF Y(J)=Y(J-1) THEN
1400 PRINT "YOU JUST DIGITIZED THIS POINT TWICE!"
1410 BEEP 651,1
1420 PRINT "DIGITIZE THE CORRECT POINT THIS TIME"
1430 GOTO 1360
1440 ELSE
1450 BEEP
1460 END IF
1470 ELSE
1480 BEEP
1490 END IF
1500 PRINT " LOCATION OF POINT";J;"IS",X(J),Y(J)
1510 NEXT J
1520 PRINT "TOTAL NUMBER OF POINTS IS";Ndat
1530 OUTPUT @File;X(*),Y(*),H(*)
1540 Foldfile=1
1550 SUBEND
1560 !*****
1570 Folddraw:SUB Folddraw(Foldfile,Name$,Ndat,Fold)
1580 COM /Comdata/ R,C,D,E,F,Dirac,Scal,Xx(1:150),Hh(1:150),O(1500),Oo(1500),
Aa(1:40),Bb(1:40),Yy(1:600),X2(25),App(25),Unit(25),Dip$(25)[1],Str(25)
1590 DIM X(150),Y(150),H(150),Ytrend(150)
1600 DIM Ynew(150),Ycross(150),Xdif(150),Hplot(150)
1610 DIM Newx3(150),Hhh(1:150),Xxx(1:150)
1620 DEG
1630 IF Foldfile<>1 THEN INPUT "ENTER NAME OF FILE TO BE READ",Name$
1640 INPUT "ARE ELEVATIONS IN METRIC(1) OR FEET(2)?",E1
1650 ASSIGN @File TO "FOLD"&Name$
1660 J=0
1670 ENTER @File;X(*),Y(*),H(*)
1680 ON END @File GOTO 1690
1690 LOOP
1700 J=J+1
1710 EXIT IF X(J)=0
1720 END LOOP
1730 Ndat=J-1
1740 FOR J=1 TO Ndat
1750 IF E1=2 THEN
1760 H(J)=H(J)*.3048

```

```

1790     FOR J=1 TO Ndat
1800         H(J)=H(J)*400/(Scal)
1810     NEXT J
1820     J=1
1830     INPUT "ENTER THE PLUNGE OF THE STRUCTURE",P
1840     INPUT "IS PLUNGE TO THE NORTH OR SOUTH",Plungdir$
1850     IF Chng=1 THEN GOTO 1880
1860     INPUT "ENTER THE TREND OF THE STRUCTURE",T
1870     INPUT "IS THE TREND TO THE EAST OR WEST?",Trendir$
1880     J=1
1890     PRINTER IS 1
1900     FOR J=1 TO Ndat STEP 1
1910         IF D<F THEN Ycross(J)=TAN(R)*(X(J)-C)+D
1920         IF D>F THEN Ycross(J)=TAN(R)*(E-X(J))+F
1930         IF D=F THEN Ycross(J)=F
1940         IF Ycross(J)>=Y(J) THEN Ynew(J)=Ycross(J)-Y(J)
1950         IF Ycross(J)<=Y(J) THEN Ynew(J)=Y(J)-Ycross(J)
1960     NEXT J
1970     B=90-T
1980     FOR J=1 TO Ndat
1990         Ytrend(J)=Ynew(J)*SIN(Direc)/SIN(B+R)
2000         IF Ytrend(J)<0 THEN Ytrend(J)=ABS(Ytrend(J))
2010         Hh(J)=Ytrend(J)*TAN(P)
2020         IF Ycross(J)>Y(J) THEN Hh(J)=-1*Hh(J)
2030         Hh(J)=Hh(J)+H(J)
2040         Xdif(J)=Ytrend(J)*SIN(T)
2050         IF Y(J)>Ycross(J) THEN
2060             IF Trendir$="E" THEN
2070                 Xx(J)=X(J)-Xdif(J)
2080             ELSE
2090                 Xx(J)=X(J)+Xdif(J)
2100             END IF
2110         ELSE
2120             IF Trendir$="E" THEN
2130                 Xx(J)=X(J)+Xdif(J)
2140             ELSE
2150                 Xx(J)=X(J)-Xdif(J)
2160             END IF
2170         END IF
2180     NEXT J
2190     FOR J=1 TO Ndat
2200         Xx(J)=Xx(J)/COS(R)
2210     NEXT J
2220     IF Fold=1 THEN SUBEXIT
2230     CALL Minn(Minxx,Xx(*),Ndat)
2240     CALL Minn(Minhh,Hh(*),Ndat)
2250     INPUT "DO YOU WANT A POLYNOMIAL FIT? Y=1",Polyn
2260     IF Polyn=2 THEN
2270         CALL Tick_etc(Xx(*),Hh(*),Minxx,MAX(Xx(*))-Minxx,Minhh,Scal,Ndat)
2280     ELSE
2290         LOADSUB Interval FROM "EXTRASUBS"
2300         CALL Interval(Xx(*),Hh(*),Minxx,Minhh,Ndat,Xxx(*),Hhh(*))
2310         DELSUB Interval
2320         CALL Poly(Xxx(*),Hhh(*),Ndat,O(*),Oo(*),Npts)
2330         CALL Tick_etc(O(*),Oo(*),Minxx,MAX(Xx(*))-Minxx,Minhh,Scal,Npts)
2340     END IF
2350     PENUP
2360     PEN 0
2370     PRINT "DO YOU WANT TO CHANGE THE PLUNGE=1 OR TREND=2"
2380     INPUT "OR BOTH=3 OR NONE=4?",Chng
2390     IF Chng=1 THEN GOTO 1830
2400     IF Chng=2 THEN GOTO 1860
2410     IF Chng=3 THEN GOTO 1830
2420     IF Chng=4 THEN GOTO 2430

```

```

*****
2450 Topodraw:SUB Topodraw(Topofile,New$,Ntotpts,Topo,Mm,Ii(*),Jj(*),L)
2460 COM /Comdata/ R,C,D,E,F,Dirac,Scal,Xx(1:150),Hh(1:150),O(1500),Oo(1500),
Aa(1:40),Bb(1:40),Yy(1:600),X2(25),App(25),Unit(25),Dip$(25)[1],Str(25)
2470 DIM Aaa(1:600),Ss(40)
2480 J=0
2490 IF Topofile<>1 THEN INPUT "ENTER NAME OF FILE CONTAINING DIGITIZED TOPOG
RAPHY",New$
2500 INPUT "ENTER SCALE",Scal
2510 INPUT "ARE ELEVATIONS IN METRIC(1) OR FEET(2)",E1
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)

```

```

3080 DIM Dp(25),Dips(25)[1],Unit(25),Pl(25),Tr(25),Trend$(25)[1]
3090 DIM Plunge$(25)[1],Elev(25),Atype(25)
3100 MAT Atype= (0)
3110 PRINT "DIGITIZE GEOLOGIC DATA IN ANY ORDER."
3120 PRINT "UNLESS YOU PLAN TO DO A HERMITE FIT, THEN DIGITZE"
3130 PRINT "FROM LEFT TO RIGHT AS POINTS WILL PROJECT"
3140 PRINT "STRIKE IS NORTH (ASSUMED) THEN SOMETHING"
3150 PRINT "EAST OR WEST"
3160 PRINT "DIP IS EITHER EAST OR WEST"
3170 J=0
3180 LOOP
3190   J=J+1
3200   PRINT "IS THIS A WELL(1),PROJECTED ALONG STRIKE(2),PROJECTED ALONG "
3210   PRINT "PLUNGE(3), OR END OF DATA SET(4)"
3220   INPUT Sym
3230   EXIT IF Sym=4
3240   INPUT "ENTER STRIKE",Str(J)
3250   INPUT "EAST OR WEST?",Strike$(J)
3260   IF Sym=1 THEN
3270     Dp(J)=100
3280     GOTO 3420
3290   END IF
3300   INPUT "ENTER DIP",Dp(J)
3310   INPUT "EAST OR WEST?",Dip$(J)
3320   INPUT "ENTER NUMBER OF UNIT",Unit(J)
3330   IF Sym=2 THEN
3340     Tr(J)=100
3350     GOTO 3420
3360   END IF
3370   INPUT "ENTER TREND ",Tr(J)
3380   INPUT "ENTER TREND DIRECTION",Trend$(J)
3390   INPUT "ENTER PLUNGE",Pl(J)
3400   INPUT "ENTER PLUNGE DIRECTION",Plunge$(J)
3410   INPUT "ENTER ELEVATION",Elev(J)
3420 END LOOP
3430 Ndat=J-1
3440 PRINT "YOU HAVE";Ndat;"POINTS TO DIGITIZE"
3450 FOR J=1 TO Ndat
3460   PRINT "DIGITIZE LOCATION OF POINT";J,Str(J);Strike$(J);Dp(J);Dip$(J)
3470   DIGITIZE X1(J),Y1(J)
3480   IF X1(J)=X1(J-1) THEN
3490     IF Y1(J)=Y1(J-1) THEN
3500       BEEP 651,1
3510       INPUT "YOU DIGITIZED A POINT TWICE, PRESS 1 IF THIS WAS INCORRECT"
3520       ,Totimes
3530       IF Totimes=1 THEN
3540         J=J-1
3550         GOTO 3460
3560       ELSE
3570         PRINT "CONTINUE THEN"
3580       END IF
3590     ELSE
3600       BEEP
3610     END IF
3620   ELSE
3630     BEEP
3640   END IF
3650 NEXT J
3660 INPUT "WHAT WOULD YOU LIKE TO CALL THE NEW FILE?",Nice$
3670 CREATE BDAT "GEO"&Nice$,1,2800
3680 ASSIGN @File TO "GEO"&Nice$
3690 OUTPUT @File;X1(*),Y1(*),Str(*),Strike$(*),Dp(*),Dip$(*),Unit(*),Tr(*),T
rend$(*),Pl(*),Plunge$(*),Elev(*),Atype(*)
3700 SUBEND

```

```

3720 Geodraw:SUB Geodraw(Nice$,Npts,Geo,Ultraymin,Geofile,Prvdepth,Cc(*),Minaa)
3730   COM /Comdata/ R,C,D,E,F,Dirac,Scal,Xx(1:150),Hh(1:150),O(1500),Oo(1500),
Aa(1:40),Bb(1:40),Yy(1:600),X2(25),App(25),Unit(25),Dip$(25)[1],Str(25)
3740   COM /Geodata/ X1(25),Y1(25),Strike$(25)[1],Dp(25),Tr(25),Trend$(25)[1],P
1(25),Plunge$(25)[1],Elev(25),Atype(25)
3750   DEG
3760   DIM X5(25),Array(25)
3770   DIM Crossy(25),Newy(25),Trendy(25),Difx(25),B(25),Ang(25),Array2(25)
3780   IF Geo=0 THEN GOTO 5110
3790   IF Geofile<>1 THEN INPUT "ENTER NAME OF FILE CONTAINING GEOLOGIC DATA",N
ice$
3800   Geofile=1
3810   ASSIGN @Road TO "GEO"&Nice$
3820   ON END @Road GOTO 3840
3830   ENTER @Road;X1(*),Y1(*),Str(*),Strike$(*),Dp(*),Dip$(*),Unit(*),Tr(*),Tr
end$(*),Pl(*),Plunge$(*),Elev(*),Atype(*)
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 Well13 FROM "EXTRASUBS"
3980       CALL Well13(Wellname$,Mindepth,Dip$(*),Dp(*),J,Npts,Scal,X1(*),Y1(*),
Unit(*),Str(*),Strike$(*),Atype(*),Plunge$(*),Pl(*),Trend$(*),Tr(*))
3990       DELSUB Well13
4000       Wells=0
4010       IF Mindepth<Prvdepth THEN Prvdepth=Mindepth
4020     END IF
4030   EXIT IF J=Npts
4040   END LOOP
4050   IF Prvdepth=10000000 THEN Prvdepth=0
4060   FOR J=1 TO Npts
4070     IF Strike$(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(Dirac)/SIN(B(J)+R)
4240       ELSE
4250         Trendy(J)=Newy(J)*SIN(90+R)/SIN(Dirac-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(Dirac)/SIN(R(J)+R)

```

```

4330     END IF
4340     IF Trendy(J)<0 THEN Trendy(J)=ABS(Trendy(J))
4350     Difx(J)=Trendy(J)*COS(B(J))
4360     IF Y1(J)>Crossy(J) THEN
4370         IF Clock=1 THEN
4380             X2(J)=X1(J)-Difx(J)
4390         ELSE
4400             X2(J)=X1(J)+Difx(J)
4410         END IF
4420     ELSE
4430         IF Clock=1 THEN
4440             X2(J)=X1(J)+Difx(J)
4450         ELSE
4460             X2(J)=X1(J)-Difx(J)
4470         END IF
4480     END IF
4490 NEXT J
4500 FOR J=1 TO Npts
4510     X2(J)=X2(J)/COS(R)
4520 NEXT J
4530 FOR J=1 TO Npts
4540     IF Minaa>C THEN
4550         X2(J)=X2(J)-(Minaa-C)
4560     ELSE
4570         X2(J)=X2(J)+(C-Minaa)
4580     END IF
4590 NEXT J
4600 IF X2(1)<C THEN X2(1)=C
4610 FOR J=1 TO Npts
4620     IF D>F THEN
4630         IF Strike$(J)="W" THEN Ang(J)=90-Str(J)-R
4640         IF Strike$(J)="E" THEN
4650             IF R<Str(J) THEN
4660                 Ang(J)=90+R-Str(J)
4670             ELSE
4680                 Ang(J)=90+Str(J)-R
4690             END IF
4700         END IF
4710     ELSE
4720         IF Strike$(J)="E" THEN Ang(J)=90-Str(J)-R
4730         IF Strike$(J)="W" THEN
4740             IF Str(J)<R THEN
4750                 Ang(J)=90+Str(J)-R
4760             ELSE
4770                 Ang(J)=90+R-Str(J)
4780             END IF
4790         END IF
4800     END IF
4810 NEXT J
4820 FOR J=1 TO Npts
4830     IF Dp(J)=90 THEN App(J)=90
4840     IF Ang(J)=90 THEN App(J)=Dp(J)
4850     IF Ang(J)=90 THEN GOTO 4870
4860     IF Dp(J)<90 THEN App(J)=ATN(TAN(Dp(J))*SIN(ABS(Ang(J))))
4870 NEXT J
4880 IF Wow=1 THEN GOTO 5110
4890 FOR J=1 TO Npts
4900     IF Elev(J)<>0 THEN
4910         Elev(J)=Elev(J)*400/Scal
4920         IF Crossy(J)>Y1(J) THEN
4930             IF Plunge$(J)="N" THEN
4940                 Cc(J)=Elev(J)-ATN(SIN(P1(J))/Trendy(J))
4950             ELSE
4960                 Cc(J)=Elev(J)+ATN(SIN(P1(J))/Trendy(J))
4970             END IF

```

```

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
5620

```

```

5640 !*****
5650 Cross:SUB Crs(Ntot,Ndat,Npts,Fld,Tp,Ge,Xxx(*),Hhh(*),G,Ii(*),Jj(*),L)
5660   COM /Comdata/ R,C,D,E,F,Dirac,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$(25)[1],Str(25)
5670   COM /Geodata/ X1(25),Y1(25),Strike$(25)[1],Dp(25),Tr(25),Trend$(25)[1],P
1(25),Plunge$(25)[1],Elev(25),Atype(25)
5680   DIM Cc(25)
5690   INPUT "WILL CROSSSECTION INCLUDE FOLD? Y=1",Fld
5700   INPUT "WILL CROSSSECTION INCLUDE GEOLOGIC DATA? Y=1",Ge
5710   INPUT "WILL CROSSSECTION INCLUDE TOPOGRAPHY?",Tp
5720   !DEAL WITH FOLD FIRST
5730   IF Fld=1 THEN
5740     CALL Folddraw(Foldfile,Name$,Ndat,Fld)
5750     CALL Minn(Minxx,Xx(*),Ndat)
5760     CALL Minn(Minhh,Hh(*),Ndat)
5770     LOADSUB Interval FROM "EXTRASUBS"
5780     CALL Interval(Xx(*),Hh(*),Minxx,Minhh,Ndat,Xxx(*),Hhh(*))
5790     DELSUB Interval
5800     GCLEAR
5810     CALL Poly(Xxx(*),Hhh(*),Ndat,O(*),Oo(*),Ndats)
5820     CALL Minn(Minoo,Oo(*),Ndats)
5830   END IF
5840   !DEAL WITH TOPO
5850   IF Tp=1 THEN
5860     CALL Topodraw(Topofile,New$,Ntot,Tp,Mm,Ii(*),Jj(*),L)
5870     CALL Minn(Minbb,Bb(*),Ntot)
5880     CALL Minn(Minaa,Aa(*),Ntot)
5890     FOR J=1 TO Ntot
5900       IF Minaa<C THEN
5910         Aa(J)=Aa(J)-(Minaa-C)
5920       ELSE
5930         Aa(J)=Aa(J)+(C-Minaa)
5940       END IF
5950     NEXT J
5960   END IF
5970   !DEAL WITH GEO
5980   IF Ge=1 THEN
5990     LOADSUB Spacing FROM "EXTRASUBS"
6000     CALL Spacing(Aa(*),Bb(*),Ntot,Yy(*),M)
6010     DELSUB Spacing
6020     CALL Geodraw(Nice$,Npts,1,Ultraymin,Geofile,Prvdepth,Cc(*),Minaa)
6030     CALL Hermite(X2(*),App(*),Unit(*),Npts,Dip$(*),Cc(*),C,Minheight,1,Ult
raymin)
6040   END IF
6050   Minimum(1)=Minhh
6060   Minimum(2)=Minbb
6070   Minimum(3)=Minheight
6080   Minimum(4)=Prvdepth*400/Scal
6090   CALL Minn(Ultraymin,Minimum(*),4)
6100   IF Ge=1 THEN CALL Geodraw(Nice$,Npts,0,Ultraymin,Geofile,Mindepth,Cc(*),
Minaa)
6110   IF Fld=1 THEN CALL Tick_etc(O(*),Oo(*),C,G,Ultraymin,Scal,Ndats)
6120   INPUT "DO YOU WANT A CUBIC SPLINE FIT TO YOUR TOPOGRAPHY Y=1",Poly
6130   IF Poly=1 THEN
6140     CALL Topodraw(Topofile,New$,Ntot,2,Mm,Ii(*),Jj(*),L)
6150     CALL Tick_etc(Ii(*),Jj(*),C,G,Ultraymin,Scal,L)
6160   ELSE
6170     CALL Tick_etc(Aa(*),Bb(*),C,G,Ultraymin,Scal,Ntot)
6180   END IF
6190   Tp=0
6200   Ge=0
6210   Fld=0
6220 SUBEND
6230 !*****

```

```

6260   FOR Gg=1 TO Numb
6270     IF G(Gg)=0 THEN GOTO 6300
6280     IF Ming=0 THEN Ming=G(Gg)
6290     IF G(Gg)<Ming THEN Ming=G(Gg)
6300   NEXT Gg
6310   SUBEND
6320 !*****
6330 Poly:SUB Poly(Ww(*),Uu(*),Ndat,O(*),Oo(*),M)
6340   PRINT "ENTER ORDER OF POLYNOMIAL"
6350   INPUT " ",N
6360   PRINT "THIS WILL TAKE A WHILE"
6370   DIM Ab(200,20),Aprime(20,200),Solution(0:20,1)
6380   DIM Aaa(20,20),A_inv(20,20),B(20,1),Z(15000)
6390   REDIM Ab(1:Ndat,0:N-1)
6400   REDIM Aprime(0:N-1,1:Ndat)
6410   REDIM Solution(0:N-1,1)
6420   REDIM Aaa(0:N-1,0:N-1)
6430   REDIM A_inv(0:N-1,0:N-1)
6440   REDIM Ww(1:Ndat)
6450   REDIM Uu(1:Ndat)
6460   REDIM B(0:N-1,1)
6470   FOR J=1 TO Ndat
6480     FOR K=0 TO N-1
6490       Ab(J,K)=(Ww(J))^K
6500     NEXT K
6510   NEXT J
6520   MAT Aprime= TRN(Ab)
6530   MAT Aaa= Aprime*Ab
6540   MAT A_inv= INV(Aaa)
6550   MAT B= Aprime*Uu
6560   MAT Solution= A_inv*B
6570   FOR Q=.25 TO MAX(Ww(*)) STEP .25
6580     Z(Q)=0
6590     FOR S=1 TO N
6600       Z(Q)=Solution(S-1,0)*Q^(S-1)+Z(Q)
6610     NEXT S
6620     Z(Q*100)=Z(Q)
6630   NEXT Q
6640   J=0
6650   FOR Q=MIN(Ww(*)) TO MAX(Ww(*)) STEP .25
6660     M=M+1
6670     O(M)=Q
6680     Oo(M)=Z(Q*100)
6690   NEXT Q
6700   PRINT M
6710   SUBEND
6720   SUB Axial(Scal,Foldfile,Name$,O(*),Oo(*))
6730 !*****
6740 Axial:!AXIAL PROJECTION
6750   DIM X(150),Y(150),H(150),Ytrend(150),Aa(150),Bb(150)
6760   DIM Ynew(150),Ycross(150),Xdif(150),Hplot(150)
6770   DIM Newx3(150),Xx(1500),Hh(1500),Hhh(1:150),Xxx(1:150)
6780   DEG
6790   IF Foldfile=1 THEN GOTO 6810
6800   INPUT "ENTER NAME OF FILE TO BE READ",Name$
6810   INPUT "ENTER THE SCALE OF THE MAP IN FORM (1:_____)",Scal
6820   ASSIGN @File TO "FOLD"&Name$
6830   J=0
6840   ENTER @File;X(*),Y(*),H(*)
6850   ON END @File GOTO 6910
6860   FOR J=1 TO 150
6870     IF X(J)=0 THEN
6880       Ndat=J-1
6890       GOTO 6970

```

```

6920 FOR J=1 TO Ndat
6930   IF E1=2 THEN
6940     H(J)=.3048*H(J)
6950   END IF
6960 NEXT J
6970 FOR J=1 TO Ndat
6980   H(J)=H(J)*400/(Scal)
6990 NEXT J
7000 J=1
7010 INPUT "ENTER THE PLUNGE OF THE STRUCTURE",P
7020 INPUT "IS THE PLUNGE TO THE NORTH OR SOUTH?",Plungdir$
7030 INPUT "ENTER THE TREND OF THE STRUCTURE",T
7040 INPUT "IS THE TREND TO THE EAST OR WEST?",Trendir$
7050 Slope=COS(T)/SIN(T)
7060 Newslope=-1*1/Slope
7070 IF Trendir$="E" THEN
7080   Cc=0
7090   Ff=0
7100   Ee=150
7110   Dd=-1*(E-C)*Newslope
7120 ELSE
7130   Cc=0
7140   Dd=0
7150   Ee=150
7160   Ff=(E-C)*Newslope+D
7170 END IF
7180 FOR J=1 TO Ndat STEP 1
7190   IF Dd<Ff THEN Ycross(J)=TAN(Newslope)*(X(J)-Cc)+Dd
7200   IF Dd>Ff THEN Ycross(J)=TAN(Newslope)*(Ee-X(J))+Ff
7210   IF Dd=Ff THEN Ycross(J)=Ff
7220   IF Ycross(J)>=Y(J) THEN Ynew(J)=Ycross(J)-Y(J)
7230   IF Ycross(J)<=Y(J) THEN Ynew(J)=Y(J)-Ycross(J)
7240 NEXT J
7250 B=90-T
7260 FOR J=1 TO Ndat
7270   Ytrend(J)=Ynew(J)*COS(T)
7280   IF Ytrend(J)<0 THEN Ytrend(J)=ABS(Ytrend(J))
7290   Xdif(J)=Ytrend(J)*SIN(T)
7300   IF Y(J)>Ycross(J) THEN
7310     IF Trendir$="E" THEN
7320       Xx(J)=X(J)-Xdif(J)
7330     ELSE
7340       Xx(J)=X(J)+Xdif(J)
7350     END IF
7360   ELSE
7370     IF Trendir$="E" THEN
7380       Xx(J)=X(J)+Xdif(J)
7390     ELSE
7400       Xx(J)=X(J)-Xdif(J)
7410     END IF
7420   END IF
7430 NEXT J
7440 FOR J=1 TO Ndat
7450   Hh(J)=Ytrend(J)*SIN(P)
7460   IF Ycross(J)>Y(J) THEN Hh(J)=-1*Hh(J)
7470   Hh(J)=H(J)+Hh(J)
7480   Xx(J)=Xx(J)/COS(T)
7490 NEXT J
7500 CALL Minn(Minxx,Xx(*),Ndat)
7510 CALL Minn(Minhh,Hh(*),Ndat)
7520 INPUT "DO YOU WANT A POLYNOMIAL FIT? Y=1",Polyn
7530 IF Polyn=2 THEN
7540   FOR J=1 TO Ndat
7550     IF J=1 THEN MOVE Xx(1)-Minxx,Hh(1)-Minhh

```

```

7570      NEXT J
7580  ELSE
7590      LOADSUB Interval FROM "EXTRASUBS"
7600      CALL Interval(Xx(*),Hh(*),Minxx,Minhh,Ndat,Xxx(*),Hhh(*))
7610      DELSUB Interval
7620      CALL Poly(Xxx(*),Hhh(*),Ndat,O(*),Oo(*),Npts)
7630      CALL Tick_etc(O(*),Oo(*),Minxx,MAX(Xx(*))-Minxx,Minhh,Scal,Npts)
7640      FOR J=1 TO Npts
7650          IF J=1 THEN MOVE O(1)-Minxx,Oo(1)-Minhh
7660          DRAW O(J)-Minxx,Oo(J)-Minhh
7670      NEXT J
7680  END IF
7690  SUBEND
7700  !*****
7710  Tick:SUB Tick_etc(Xarray(*),Yarray(*),C,G,Minyarray,Scal,Ndat)
7720      Hard=0
7730      PLOTTER IS 3,"INTERNAL"
7740      LOOP
7750          INPUT "ENTER LINE TYPE NUMBER ",Lin
7760          LINE TYPE Lin
7770          IF Hard=1 THEN
7780              INPUT "ENTER PEN NUMBER",Penn
7790              PEN Penn
7800          ELSE
7810              PEN 1
7820          END IF
7830          FOR J=1 TO Ndat
7840              IF J=1 THEN MOVE Xarray(J)-C,(Yarray(J)-Minyarray)+10
7850              DRAW Xarray(J)-C,(Yarray(J)-Minyarray)+10
7860          NEXT J
7870          LINE TYPE 1
7880          IF Hard=0 THEN
7890              CSIZE 5
7900          ELSE
7910              CSIZE 2
7920          END IF
7930          LORG 5
7940          !DRAW AXIS HERE
7950          MOVE 0,MAX(Yarray(*))-Minyarray+10
7960          DRAW 0,0
7970          DRAW G,0
7980          DRAW G,MAX(Yarray(*))-Minyarray+10
7990          !LABEL AXIS HERE
8000          MOVE G+8,MAX(Yarray(*))-Minyarray+10
8010          LABEL INT(MAX(Yarray(*))*Scal/400);"M"
8020          MOVE G+8,10
8030          LABEL INT((Minyarray)*Scal/400);"M"
8040          !TICK MARKS DRAWN HERE
8050          FOR K=1 TO (G)/4
8060              MOVE K*4,.3
8070              DRAW K*4,0
8080          NEXT K
8090          FOR K=1 TO (MAX(Yarray(*))-Minyarray+10)/4
8100              MOVE G,4*K
8110              DRAW G+.3,4*K
8120          NEXT K
8130          !OPTION FOR HARDCOPY HERE
8140          EXIT IF Hard=1
8150          INPUT "DO YOU WANT HARD COPY? Y=1",Hard
8160          EXIT IF Hard<>1
8170          PLOTTER IS 705,"HPGL"
8180          END LOOP
8190          PEN 0
8200          GCLEAR
8210  SUBEND

```

```

8220 !*****
8230 Hermite:SUB Hermite(X2(*),App(*),Unit(*),Total,Dip$(*),Cc(*),C,Minheight,Click,Ultraymin)
8240 PLOTTER IS 705,"HPGL"
8250 DIM U(-1:100),Slp(25),Height(6000),Yunit(25)
8260 MAT Yunit= Unit
8270 MAT SORT Yunit
8280 FOR J=1 TO 25
8290 IF Yunit(J)<>Yunit(J-1) THEN
8300 Unum=Unum+1
8310 ELSE
8320 Unum=Unum
8330 END IF
8340 NEXT J
8350 FOR J=1 TO Total
8360 Slp(J)=COS(90-App(J))/SIN(90-App(J))
8370 IF Dip$(J)="E" THEN Slp(J)=-1*Slp(J)
8380 NEXT J
8390 FOR J=1 TO Unum
8400 IF Click=1 THEN GOTO 8440
8410 PRINT "ENTER PEN COLOR FOR HERMITE FIT ON UNIT";J
8420 INPUT Colour
8430 PEN Colour
8440 L=0
8450 FOR K=1 TO Total
8460 IF Unit(K)=J THEN
8470 L=L+1
8480 Slope(L)=Slp(K)
8490 Coc(L)=Cc(K)
8500 X3(L)=X2(K)
8510 END IF
8520 NEXT K
8530 K=1
8540 FOR M=1 TO L-1
8550 Dist=X3(M+1)-X3(M)
8560 Na=Coc(M)
8570 Nc=Coc(M+1)
8580 Na=Na/Dist
8590 Nc=Nc/Dist
8600 Nb=Slope(M)
8610 Nd=Slope(M+1)
8620 FOR X=0 TO 1 STEP .1
8630 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)
8640 IF Click=1 THEN GOTO 8690
8650 IF X=0 THEN
8660 IF M=1 THEN MOVE X3(M)-C,Na*Dist+10-Ultraymin
8670 END IF
8680 DRAW X3(M)-C+X*Dist,Y_of_x*Dist+10-Ultraymin
8690 Height(K)=Y_of_x*Dist
8700 K=K+1
8710 NEXT X
8720 NEXT M
8730 NEXT J
8740 PENUP
8750 PEN 0
8760 CALL Minn(Minheight,Height(*),K-1)
8770 SUBEND

```

```

10                                     !*****
20                                     !***   EXTRASUBS   ***
30                                     !*****
40   END
50   !*****
60   SUB Well13(Wellname$,Mindepth,Dip$(*),Dp(*),J,Npts,Scal,X1(*),Y1(*),Unit(*)
,Str(*),Strike$(*),Atype(*),Plunge$(*),P1(*),Trend$(*),Tr(*))
70     DIM Wellunit(25),Depth(25),Dipp(25),Dipdir$(25)[1],X4(25),Y14(25),Dip4$(
25)[1],Unit4(25),Dp4(25),Str4(25),Strike4$(25)[1],Atype4(25),Plunge4$(25)[1]
80     DIM P14(25),Trend4$(25)[1],Tr4(25)
90     ASSIGN @Thisway TO "WELL"&Wellname$
100    ENTER @Thisway;Wellunit(*),Depth(*),Dipp(*),Dipdir$(*)
110    K=0
120    LOOP
130      K=K+1
140      EXIT IF Depth(K)=0
150    END LOOP
160    N=K-1
170    MAT X4= X1
180    MAT Y14= Y1
190    MAT Str4= Str
200    MAT Strike4$= Strike$
210    MAT Atype4= Atype
220    MAT Dp4= Dp
230    MAT Dip4$= Dip$
240    MAT Unit4= Unit
250    MAT Plunge4$= Plunge$
260    MAT P14= P1
270    MAT Trend4$= Trend$
280    MAT Tr4= Tr
290    FOR K=J TO Npts
300      X4(K+N-1)=X1(K)
310      Y14(K+N-1)=Y1(K)
320      Str4(K+N-1)=Str(K)
330      Strike4$(K+N-1)=Strike$(K)
340      Dip4$(K+N-1)=Dip$(K)
350      Dp4(K+N-1)=Dp(K)
360      Unit4(K+N-1)=Unit(K)
370      Atype4(K+N-1)=Atype(K)
380      Plunge4$(K+N-1)=Plunge$(K)
390      Trend4$(K+N-1)=Trend$(K)
400      P14(K+N-1)=P1(K)
410      Tr4(K+N-1)=Tr(K)
420    NEXT K
430    FOR K=0 TO N-1
440      X4(J+K)=X1(J)
450      Str4(J+K)=Str(J)
460      Strike4$(J+K)=Strike$(J)
470      Y14(J+K)=Y1(K+1)
480      Dp4(J+K)=Dipp(K+1)
490      Dip4$(J+K)=Dipdir$(K+1)
500      Unit4(J+K)=Wellunit(K+1)
510      Atype4(J+K)=Depth(K+1)*400/Scal
520      Plunge4$(J+K)=" "
530      Trend4$(J+K)=" "
540      P14(J+K)=0
550      Tr4(J+K)=0
560    NEXT K
570    MAT X1= X4

```

```

590     MAT Up= Up+
600     MAT Str= Str4
610     MAT Strike$= Strike4$
620     MAT Atype= Atype4
630     MAT Dip$= Dip4$
640     MAT Unit= Unit4
650     MAT Trend$= Trend4$
660     MAT Plunge$= Plunge4$
670     MAT Tr= Tr4
680     MAT Pl= Pl4
690     Npts=Npts+N-1
700     J=J+N-1
710     CALL Minn(Mindepth,Depth(*),J)
720 SUBEND
730 !*****
740 SUB Wellfile1
750     DIM Wellunit(25),Depth(25),Dipp(25),Dipdir$(25)[1]
760     INPUT "ENTER NAME FOR WELL FILE",Wellname$
770     CREATE BDAT "WELL"&Wellname$,4,200
780     ASSIGN @Route TO "WELL"&Wellname$
790     J=1
800     LOOP
810         INPUT "ENTER UNIT NAME",Wellunit(J)
820     EXIT IF Wellunit(J)=0
830         INPUT "ENTER DEPTH OF HORIZON",Depth(J)
840         INPUT "ENTER DIP",Dipp(J)
850         INPUT "ENTER DIP DIRECTION",Dipdir$(J)
860         J=J+1
870     END LOOP
880     N=J-1
890     OUTPUT @Route;Wellunit(*),Depth(*),Dipp(*),Dipdir$(*)
900 SUBEND
910 !*****
920 SUB Spacing(X(*),Y(*),Ndat,Yy(*),M)
930     DIM Slope(150),Xx(150)
940     FOR J=1 TO Ndat
950         Xx(J)=INT(X(J))
960     NEXT J
970     FOR J=1 TO Ndat-1
980         Xdist=Xx(J+1)-Xx(J)
990         Ydist=Y(J+1)-Y(J)
1000        Slope(J)=Ydist/Xdist
1010    NEXT J
1020    M=0
1030    FOR J=1 TO Ndat-1
1040        FOR K=0 TO Xx(J+1)-Xx(J)-1
1050            M=M+1
1060            Yy(M)=Y(J)+Slope(J)*K
1070        NEXT K
1080    NEXT J
1090    L=4
1100    FOR L1=M+1 TO M+4
1110        Yy(L1)=Y(J-1)+Slope(J-1)*(K-1)
1120    NEXT L1
1130 SUBEND
1140 !*****
1150 Interval:SUB Interval(Xx(*),Hh(*),Minxx,Minhh,Ndat,Xxx(*),Hhh(*))
1160 PLOTTER IS 3,"INTERNAL"
1170     LOG 5
1180     CSIZE 3
1190     FOR J=1 TO Ndat
1200         MOVE Xx(J)-Minxx,Hh(J)-Minhh+10
1210         LABEL "*"
1220     NEXT J
1230     FOR I=1 TO Ndat

```

```

1250     Xxx(J)=INT(Xx(J))
1260     ELSE
1270     IF FRACT(Xx(J))<.4 THEN
1280     Xxx(J)=INT(Xx(J))+.25
1290     ELSE
1300     IF FRACT(Xx(J))<.60 THEN
1310     Xxx(J)=INT(Xx(J))+.5
1320     ELSE
1330     IF FRACT(Xx(J))<.80 THEN
1340     Xxx(J)=INT(Xx(J))+.75
1350     ELSE
1360     Xxx(J)=INT(Xx(J))+1
1370     END IF
1380     END IF
1390     END IF
1400     END IF
1410     Hhh(J)=Hh(J)
1420     NEXT J
1430     SUBEND
1440 !*****
1450 Spline:
1460 SUB Spline(X(*),Y(*),Ndat,Var(*),C,Ultraymin)
1470 REDIM X(Ndat)
1480 REDIM Y(Ndat)
1490 N=Ndat-1
1500 DIM H(40)
1510 FOR J=1 TO N
1520     H(J)=X(J)-X(J-1)
1530 NEXT J
1540 DIM A(40,40),A_inv(40,40)
1550 REDIM A(0:N,0:N)
1560 REDIM A_inv(0:N,0:N)
1570 FOR J=0 TO N
1580     FOR I=0 TO N
1590         IF I=J THEN
1600             IF J=0 THEN
1610                 A(J,I)=2/H(J+1)
1620                 A(J+1,I)=1/H(J+1)
1630                 GOTO 1760
1640             END IF
1650             IF J=N THEN
1660                 A(J,I)=2/H(J)
1670                 A(J-1,I)=1/H(J)
1680                 GOTO 1760
1690             END IF
1700         END IF
1710         IF I=J THEN
1720             A(J,I)=(2/H(J+1))+(2/H(J))
1730             A(J-1,I)=1/H(J)
1740             A(J+1,I)=1/H(J+1)
1750         END IF
1760     NEXT I
1770 NEXT J
1780 DIM B(40)
1790 REDIM B(0:N)
1800 FOR J=0 TO N
1810     IF J=0 THEN
1820         B(J)=(Y(J+1)-Y(J))/H(J+1)^2
1830         GOTO 1900
1840     END IF
1850     IF J=N THEN
1860         B(J)=(Y(J)-Y(J-1))/H(J)^2
1870     ELSE
1880         B(J)=(Y(J+1)-Y(J))/(H(J+1)^2)+(Y(J)-Y(J-1))/H(J)^2
1890     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),Slp(40),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
(Xx-1)
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

```

```

10      !                               *****
20      !                               *   BALANCE   *
30      !                               *****
40      INPUT "ENTER SCALE",Scal
50      PRINTER IS 1
60      GINIT
70      PLOTTER IS 705,"HPGL"
80      GRAPHICS ON
90      GRAPHICS INPUT IS 705,"HPGL"
100     PEN 0
110     !PROGRAM IS DIMENSIONED FOR 200 POINTS.
120     !A FAULT CAN HAVE 20 DIGITIZED POINTS
130     !THERE CAN BE 25 DIGITIZED POINTS BETWEEN ALL SEGMENTS.
140     !UP TO 4 LAYERS AND 8 FAULTS
150     DIM X(200),Y(200),Bigarea(25),Layera(2200),Subarea(25),Ax(8,500)
160     DIM Faulttrace(8),Fx(8,5,2),Fy(8,5,2),Faultarea(8,4),Bx(8,500)
170     DIM P(8,-1:4,500,2),Rr(201),Mm(1:25),Nn(1:25),Xx(500),Yy(500)
180     DIM Mmm(1:25),Nnn(1:25),Xxx(2200),Yyy(2200),Newx(150),Newy(150)
190     DIM Aaa(500),Bbb(500),B(20),C(20),Aa(20),Bb(20),Porder(48,6),Zzz(48,6)
200     DIM Sectionfault(48),Az(8,20),Bz(8,20),Nodigitpts(10)
210     PRINT "DO YOU WANT TO DIGITIZE A CROSS SECTION IN ORDER TO "
220     PRINT "CHECK FOR BALANCE (1) OR DRAW A CROSS SECTION FROM AN"
230     PRINT "EXISTING FILE (2)?"
240     INPUT Draw_only
250     INPUT "DO YOU WANT PRINTOUT OF DATA RESULTS?",Prtout
260     IF Draw_only=2 THEN
270         GOSUB 4090
280         FOR J=1 TO Ndat
290             Xxx(J)=X(J)
300             Yyy(J)=Y(J)
310         NEXT J
320         FOR J=1 TO Ndat
330             IF X(J)=30000 THEN Faul=1+Faul
340             IF X(J)=20000 THEN Conta=Conta+1
350         NEXT J
360         Nlayers=Conta
370         Nfaults=Faul/Conta
380         FOR J=1 TO Nfaults
390             IF Bz(J,Nodigitpts(J)-1)<Bz(J,Nodigitpts(J)) THEN
400                 Faulttrace(J)=2
410             ELSE
420                 Faulttrace(J)=1
430             END IF
440         NEXT J
450         J=0
460         Fault=1
470         FOR Layer=1 TO Nlayers
480             IF J=Ndat+1 THEN GOTO 670
490             J=J+1
500             IF X(J)=30000 THEN
510                 Fx(Fault,Layer,1)=X(J-1)
520                 Fx(Fault,Layer,2)=X(J+1)
530                 Fy(Fault,Layer,1)=Y(J-1)
540                 Fy(Fault,Layer,2)=Y(J+1)
550             IF Fault=Nfaults THEN GOTO 590
560             Fault=1+Fault
570         END IF
580         GOTO 480
590         Fault=1
600     NEXT Layer
610     GOTO 670
620     END IF
630     IF Draw_only=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        Totallayer=0
1010        A=0
1020        Totalfault=0
1030        Totarea=0
1040        Fault=Fault+1
1050        G=G+1
1060        IF Xxx(G)=20000 THEN
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
1160            GOTO 1210
1170        ELSE
1180            Layarea(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

```

```

1300         ELSE
1310             GOTO 1480
1320         END IF
1330         ELSE
1340             GOTO 1480
1350         END IF
1360         ELSE
1370             IF Fx(Fault,Layer+1,Side)>Ax(Fault,Point) THEN
1380                 IF Ax(Fault,Point)>Fx(Fault,Layer,Side) THEN
1390                     P(Fault,Layer,Point,Side)=100
1400                 ELSE
1410                     GOTO 1480
1420                 END IF
1430             ELSE
1440                 GOTO 1480
1450             END IF
1460             GOTO 1480
1470         END IF
1480     NEXT Point
1490 NEXT Layer
1500 NEXT Side
1510 MAT Faultarea= (0)
1520 FOR Side=1 TO 2
1530     Times=0
1540     Point=0
1550     Point=1+Point
1560     IF Faulttrace(Fault)=2 THEN
1570         IF P(Fault,N,Point,Side)=100 THEN
1580             IF Times=0 THEN
1590                 Faultarea(Fault,Side)=(Fy(Fault,N+1,Side)+Bx(Fault,Point))*0.5*(
1600 1*Ax(Fault,Point)+Fx(Fault,N+1,Side))
1610                 Times=1+Times
1620                 GOTO 1550
1630             ELSE
1640                 Faultarea(Fault,Side)=Faultarea(Fault,Side)+(Bx(Fault,Point)+Bx
1650 Fault,Point-1))*0.5*(-1*Ax(Fault,Point)+Ax(Fault,Point-1))
1660                 Times=1+Times
1670                 GOTO 1550
1680             END IF
1690         ELSE
1700             IF P(Fault,N,Point-1,Side)=100 THEN
1710                 Faultarea(Fault,Side)=Faultarea(Fault,Side)+(Bx(Fault,Point-1)+
1720 y(Fault,N,Side))*0.5*(-1*Fx(Fault,N,Side)+Ax(Fault,Point-1))
1730                 GOTO 1980
1740             ELSE
1750                 IF Fy(Fault,N+1,Side)<Bx(Fault,Point) THEN GOTO 1970
1760                 GOTO 1550
1770             END IF
1780         END IF
1790     ELSE
1800         IF P(Fault,N,Point,Side)=100 THEN
1810             IF Times=0 THEN
1820                 Faultarea(Fault,Side)=(Fy(Fault,N,Side)+Bx(Fault,Point))*0.5*(Ax
1830 Fault,Point)-Fx(Fault,N,Side))
1840                 Times=1+Times
1850                 GOTO 1550
1860             ELSE
1870                 Faultarea(Fault,Side)=Faultarea(Fault,Side)+(Bx(Fault,Point)+Bx
1880 Fault,Point-1))*0.5*(Ax(Fault,Point)-Ax(Fault,Point-1))
1890                 Times=1+Times
1900                 GOTO 1550
1910             END IF
1920         ELSE
1930             IF P(Fault,N,Point-1,Side)=100 THEN

```

```

, (Fault, N+1, Side) - Ax(Fault, Point-1))
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,
)
2000         IF Faultarea(Fault, 2) > 0 THEN Faultarea(Fault, 2) = -1*Faultarea(Fault,
)
2010     ELSE
2020         IF Faultarea(Fault, 1) > 0 THEN Faultarea(Fault, 1) = -1*Faultarea(Fault,
)
2030         IF Faultarea(Fault, 2) < 0 THEN Faultarea(Fault, 2) = -1*Faultarea(Fault,
)
2040     END IF
2050 NEXT Side
2060 IF Xxx(G+2) = 20000 THEN
2070     Faultarea(Fault, 2) = 0
2080     GOTO 2180
2090 END IF
2100 IF Xxx(G-2) = 0 THEN
2110     Faultarea(Fault, 1) = 0
2120 ELSE
2130     IF Xxx(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 + Layerea(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,Layerea(*),Sds,Prtout)
2510  PRINTER IS 1
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"
2650  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 THA
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 TH
AN 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

```

```

3130      Xequal=0
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
3770!*****

```

```

3770          !ONLY FOR DRAWING PURPOSES.
3800 CREATE BDAT "BAL"&Name$,1,3220
3810 ASSIGN @Path TO "BAL"&Name$
3820 OUTPUT @Path;X(*),Y(*)
3830 RETURN
3840 !*****
3850 Two$="II"
3860 CREATE BDAT Two$&Name$,2,4000
3870 ASSIGN @File TO Two$&Name$
3880 OUTPUT @File;Nfaults,Nodigitpts(*),Az(*),Bz(*)
3890 RETURN
3900 !*****
3910 Make_a_file2:
3920 INPUT "ENTER NAME FOR POLYFIT FILE",Hello$
3930 CREATE BDAT "POLY"&Hello$,2,25000
3940 ASSIGN @Way TO "POLY"&Hello$
3950 OUTPUT @Way;Xxx(*)
3960 OUTPUT @Way;Yyy(*)
3970 CREATE BDAT "INFL"&Hello$,1,2744
3980 ASSIGN @Myway TO "INFL"&Hello$
3990 OUTPUT @Myway;Porder(*)
4000 RETURN
4010 !*****
4020 Make_a_file3:
4030 Two$="FAULT"
4040 CREATE BDAT Two$&Hello$,2,40000
4050 ASSIGN @Way2 TO Two$&Hello$
4060 OUTPUT @Way2;Nfaults,Nodigitpts(*),Ax(*),Bx(*)
4070 RETURN
4080 !*****
4090 Read_a_file: !THIS SUBROUTINE READS AN EXISTING FILE SO THAT IT
4100             !CAN BE DRAWN
4110 INPUT "ENTER NAME OF FILE TO BE READ",Name$
4120 ASSIGN @Path TO "BAL"&Name$
4130 J=0
4140 ON END @Path GOTO 4160
4150 ENTER @Path;X(*),Y(*)
4160 LOOP
4170     J=J+1
4180 EXIT IF X(J)=0
4190 END LOOP
4200 Ndat=J-1
4210 F=0
4220 FOR J=1 TO Ndat
4230     IF X(J)=30000 THEN F=1
4240 NEXT J
4250 IF F=0 THEN GOTO 4310
4260 Two$="II"
4270 ASSIGN @File TO Two$&Name$
4280 J=0
4290 ON END @File GOTO 4310
4300 ENTER @File;Nfaults,Nodigitpts(*),Az(*),Bz(*)
4310 RETURN
4320 !*****
4330 Plotting: !THIS SUBROUTINE DRAWS THE CROSS SECTION WHICH HAS PREVIOUSLY
4340 !BEEN DIGITIZED AND MAY OR MAY NOT HAVE BEEN STORED.
4350 INPUT "CRT(1) OR PLOTTER(2)",Dumb
4360 IF Dumb=2 THEN
4370     PLOTTER IS 705,"HPGL"
4380     INPUT "MAKE SURE PLOTTER ARM IS FREE, PRESS CONT(INUE)",Arm
4390     INPUT "WHAT PEN COLOR?",Colour
4400     PEN Colour
4410 ELSE
4420     PLOTTER IS 3,"INTERNAL"
4430 END IF

```

```

4450 IF R=1 THEN MOVE Xxx(1),Yyy(1)
4460 IF R=Sds THEN GOTO 4610
4470 IF Xxx(R)=20000 THEN
4480 MOVE Xxx(R+1),Yyy(R+1)
4490 GOTO 4600
4500 ELSE
4510 GOTO 4530
4520 END IF
4530 IF Xxx(R)=30000 THEN
4540 MOVE Xxx(R+1),Yyy(R+1)
4550 GOTO 4600
4560 ELSE
4570 GOTO 4590
4580 END IF
4590 PLOT Xxx(R),Yyy(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 Fit_curve:!
4770 !SUBROUTINE TO FIT POLYNOMIAL CURVES TO DIGITIZED POINTS
4780 Layer=0
4790 R=0
4800 Sds=0
4810 G=0
4820 Layer=Layer+1
4830 E=G+1
4840 IF Layer=Nlayers+1 THEN RETURN
4850 IF E=0 THEN E=1
4860 G=G+1
4870 IF X(G)=20000 THEN
4880 Xxx(Sds)=20000
4890 Yyy(Sds)=20000
4900 GOTO 4820
4910 END IF
4920 IF X(G+1)=20000 THEN GOTO 5030
4930 IF X(G)=30000 THEN
4940 Xxx(Sds)=30000
4950 Yyy(Sds)=30000
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 V/E\ _TMT(V/E\

```

```

5100   IF FRACT(X(E))<.4 THEN
5110     X(E)=INT(X(E))+.25
5120   ELSE
5130     IF FRACT(X(E))<.60 THEN
5140       X(E)=INT(X(E))+.5
5150     ELSE
5160       IF FRACT(X(E))<.80 THEN
5170         X(E)=INT(X(E))+.75
5180       ELSE
5190         X(E)=INT(X(E))+1
5200       END IF
5210     END IF
5220   END IF
5230 END IF
5240 FOR J=E TO G
5250   Sds=Sds+1
5260   Xxx(Sds)=X(J)
5270   Yyy(Sds)=Y(J)
5280 NEXT J
5290 Sds=Sds+1
5300 GOTO 4860
5310 FOR J=E TO G
5320   Mm(J-E+1)=X(J)
5330   Nn(J-E+1)=Y(J)
5340 NEXT J
5350 CALL Minn(Minmm,Mm(*),G-E+1)
5360 CALL Minn(Minmm,Nn(*),G-E+1)
5370 REDIM Mmm(1:25)
5380 REDIM Nnn(1:25)
5390 CALL Interval(Mm(*),Nn(*),Minmm,Minnn,G-E+1,Mmm(*),Nnn(*))
5400 PLOTTER IS 3,"INTERNAL"
5410 FOR J=E TO G
5420   MOVE X(J),Y(J)
5430   CSIZE 3
5440   LORG 5
5450   PEN 1
5460   LABEL "*"
5470 NEXT J
5480 CALL Poly(Mmm(*),Nnn(*),G-E+1,Xx(*),Yy(*),M,R,Porder(*))
5490 FOR J=1 TO M
5500   IF J=1 THEN MOVE Xx(J),Yy(J)
5510   DRAW Xx(J),Yy(J)
5520 NEXT J
5530 INPUT "DO YOU LIKE THIS OR WANT TO TRY AGAIN? TRYAGAIN=1",Tryit
5540 GCLEAR
5550 IF Tryit=1 THEN GOTO 5400
5560 K=0
5570 FOR J=Sds+1 TO M+Sds
5580   K=K+1
5590   Xxx(J)=Xx(K)
5600   Yyy(J)=Yy(K)
5610 NEXT J
5620 Sds=M+1+Sds
5630 GOTO 4860
5640 RETURN
5650! *****
5660 !SUBROUTINE TO FIT CURVES TO FAULTS
5670 FOR Z=1 TO Nfaults
5680   MAT C= (0)
5690   MAT B= (0)
5700   Sarah=0
5710   FOR J=1 TO Nodigitpts(Z)
5720     Sarah=1+Sarah
5730     C(Sarah)=Az(Z,J)

```

```

5760 PLOTTER IS 3,"INTERNAL"
5770 GCLEAR
5780 FOR J=1 TO Nodigitpts(Z)
5790     MOVE Az(Z,J),Bz(Z,J)
5800     CSIZE 2
5810     LORG 5
5820     LABEL "*"
5830 NEXT J
5840 REDIM Aa(1:20)
5850 REDIM Bb(1:20)
5860 CALL Minn(Mina,C(*),Nodigitpts(Z))
5870 CALL Minn(Minb,B(*),Nodigitpts(Z))
5880 CALL Interval(C(*),B(*),Mina,Minb,Nodigitpts(Z),Aa(*),Bb(*))
5890 PRINT "DO YOU WANT A POLY FIT FOR FAULT";Z;"Y=1"
5900 INPUT Toocurly
5910 IF Toocurly=2 THEN
5920     FOR J=1 TO Nodigitpts(Z)
5930         Ax(Z,J)=Aa(J)
5940         Bx(Z,J)=Bb(J)
5950     NEXT J
5960     GOTO 6130
5970 END IF
5980 PRINT "FOR FAULT",Z;":"
5990 CALL Poly(Aa(*),Bb(*),Nodigitpts(Z),Aaa(*),Bbb(*),Nopts,J,Zzz(*))
6000 PEN 1
6010 FOR J=1 TO Nopts
6020     IF J=1 THEN MOVE Aaa(J),Bbb(J)
6030     DRAW Aaa(J),Bbb(J)
6040 NEXT J
6050 INPUT "DO YOU LIKE THIS FIT OR WANT TO TRY AGAIN? TRY_AGAIN=1",Try_again
6060 IF Try_again=1 THEN GOTO 5760
6070 GCLEAR
6080 FOR J=1 TO Nopts
6090     Ax(Z,J)=Aaa(J)
6100     Bx(Z,J)=Bbb(J)
6110 NEXT J
6120 Nodigitpts(Z)=Nopts
6130 NEXT Z
6140 RETURN
6150 !*****
6160 Layerlength:!SUBROUTINE TO BALANCE BY COMPARING BED LENGTHS
6170     !NUMBER OF LAYERS IN NLAYERS
6180     J=0
6190     FOR N=1 TO Nlayers+1
6200         J=J+1
6210         IF Xxx(J)=30000 THEN GOTO 6200
6220         IF Xxx(J+1)=30000 THEN GOTO 6200
6230         IF Xxx(J+1)=20000 THEN GOTO 6270
6240         IF Xxx(J)=20000 THEN GOTO 6200
6250         Templength(N)=Templength(N)+SQR((Xxx(J+1)-Xxx(J))^2+(Yyy(J+1)-Yyy(J))^2)
6260         GOTO 6200
6270     NEXT N
6280     K=1
6290     Finish=600
6300     FOR J=1 TO Sds
6310         IF J>1 THEN
6320             IF Finish=Xxx(J-2) THEN Start=Xxx(J)
6330         END IF
6340         IF Prtout=1 THEN PRINTER IS 702
6350         IF J=1 THEN Start=Xxx(J)
6360         IF Xxx(J)=20000 THEN Finish=Xxx(J-1)
6370         IF Xxx(J)=20000 THEN
6380             Dist=Finish-Start
6390             Strain=(Dist-Templength(K))/Templength(K)

```

```

0+10      N=N+1
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 REDIGITIZE 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

```

```

7080   Sds=Ndat
7090 END IF
7100 PRINT "DO YOU WANT PLOT OF CHANGES? Y=1 N=2"
7110 INPUT Chng
7120 IF Chng=1 THEN GOTO 6520
7130 IF Chng=2 THEN GOTO 6670
7140 INPUT "DO YOU WANT TO MAKE A FILE OF THIS DATA?,Y=1",Refile
7150 IF Refile=1 THEN GOSUB 3780
7160 RETURN
7170 END
7180!*****
7190!*****
7200 Poly:SUB Poly(Ww(*),Uu(*),Dat,O(*),Oo(*),M,R,Porder(*))
7210   M=0
7220   PRINT "ENTER ORDER OF POLYNOMIAL"
7230   INPUT " ",N
7240   IF N>Dat THEN GOTO 7220
7250   DIM Ab(16,8),Aprime(8,16),Solution(0:8,1)
7260   DIM Aaa(8,8),A_inv(8,8),B(8,1),Z(0:19000),L1(-3:190)
7270   REDIM Ab(1:Dat,0:N-1)
7280   REDIM Aprime(0:N-1,1:Dat)
7290   REDIM Solution(0:N-1,1)
7300   REDIM Aaa(0:N-1,0:N-1)
7310   REDIM A_inv(0:N-1,0:N-1)
7320   REDIM Ww(1:Dat)
7330   REDIM Uu(1:Dat)
7340   REDIM B(0:N-1,1)
7350   FOR J=1 TO Dat
7360     FOR K=0 TO N-1
7370       Ab(J,K)=(Ww(J))^K
7380     NEXT K
7390   NEXT J
7400   MAT Aprime= TRN(Ab)
7410   MAT Aaa= Aprime*Ab
7420   MAT A_inv= INV(Aaa)
7430   MAT B= Aprime*Uu
7440   MAT Solution= A_inv*B
7450   FOR Q=.25 TO MAX(Ww(*)) STEP .25
7460     Z(Q)=0
7470     L1(Q)=0
7480     FOR S=1 TO N
7490       Z(Q)=Solution(S-1,0)*Q^(S-1)+Z(Q)
7500       L1(Q)=(S-1)*(S-2)*Solution(S-1,0)*Q^(S-3)+L1(Q)
7510     NEXT S
7520     Z(Q*100)=Z(Q)
7530   NEXT Q
7540   J=0
7550   FOR Q=MIN(Ww(*)) TO MAX(Ww(*)) STEP .25
7560     M=M+1
7570     O(M)=Q
7580     Oo(M)=Z(Q*100)
7590   NEXT Q
7600   I=1
7610   IF N<4 THEN GOTO 7800
7620   FOR Q=.25 TO MAX(Ww(*)) STEP .25
7630     IF L1(Q)>0 THEN
7640       IF L1(Q-.25)<0 THEN
7650         Porder(R,I)=Q
7660         I=I+1
7670       END IF
7680     END IF
7690     IF L1(Q)<0 THEN
7700       IF L1(Q-.25)>0 THEN
7710         Porder(R,I)=0

```

```

7730         END IF
7740     END IF
7750     IF Ll(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     LOG 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(*),Secti
onfault(*),Sectionno,Layerea(*),Sds,Prtout)
8180     DIM Subtotal(48),Carea(48),Rarea(48),Errorarea(48)
8190     PRINT "DO YOU WANT TO INPUT LAYER THICKNESSES (1) OR DIGITIZE LAYERS "
8200     PRINT "FOR A MORE ACCURATE ANALYSIS (2)?"
8210     INPUT Acc
8220     IF Acc=1 THEN GOTO 8380
8230     INPUT "ENTER # OF SPOTS YOU ARE GETTING THICKNESS INFO FROM FOR EACH LAY
ER",Spot
8240     FOR Layer=1 TO Nlayers
8250         Thickness=0
8260         FOR Place=1 TO Spot
8270             PRINT "DIGITIZE TOP OF LAYER";Layer
8280             DIGITIZE A,Br
8290             BEEP
8300             PRINT "DIGITIZE BOTTOM OF LAYER";Layer
8310             DIGITIZE Cr,D
8320             BEEP
8330             Thickness=SQR((Br-D)^2+(A-Cr)^2)*Scal/400+Thickness
8340         NEXT Place
8350     NEXT Layer
vxxx(Layer)=Thickness/Spot

```

```

8370 GOTO 8460
8380 FOR N=1 TO Nlayers
8390 PRINT "ENTER ASSUMED THICKNESS OF LAYER";N;"IN METERS"
8400 INPUT " ",Thickness
8410 Yyyy(N)=Thickness
8420 NEXT N
8430 IF Prtout=1 THEN PRINTER IS 702
8440 PRINT "LISTING OF LAYER LENGTHS AND THICKNESSES"
8450 PRINT " "
8460 FOR N=1 TO Nlayers
8470 IF Subarea(N)=0 THEN GOTO 8540
8480 Xxxx(N)=Subarea(N)/Yyyy(N)
8490 IF Hardcop=2 THEN GOTO 8520
8500 Xthick=DROUND(Xxxx(N),3)
8510 Ythick=DROUND(Yyyy(N),3)
8520 PRINT "UNDEFORMED LENGTH OF LAYER";N;"WAS";Xthick;"METERS"
8530 PRINT " " THICKNESS";Ythick;"METERS"
8540 NEXT N
8550 PRINTER IS 1
8560 INPUT "DO YOU WANT TO CHANGE SLIGHTLY THE LAYER THICKNESSES? Y=1,N=2",Ee
k
8570 IF Eek=1 THEN GOTO 8380
8580 !STATISTICAL ANALYSIS
8590 FOR N=1 TO Nlayers
8600 X_sum=X_sum+Xxxx(N)
8610 NEXT N
8620 Meanlength=X_sum/Nlayers
8630 Meanlength=DROUND(Meanlength,3)
8640 PRINT "AVERAGE LAYER LENGTH IS";Meanlength;"METERS"
8650 PRINT "DO YOU WANT TO USE A DIFFERENT MEANLENGTH?"
8660 INPUT "YES=1,NO=2",Newmean
8670 IF Newmean=2 THEN GOTO 8690
8680 INPUT "ENTER DESIRED MEANLENGTH",Meanlength
8690 !SAMPLE STANDARD DEVIATION
8700 FOR M=1 TO Nlayers
8710 S(M)=(Xxxx(M)-Meanlength)^2
8720 S_sum=S(M)+S_sum
8730 NEXT M
8740 Stddev=S_sum/(Nlayers-1)
8750 Stddev=DROUND(Stddev,3)
8760 IF Prtout=1 THEN PRINTER IS 702
8770 PRINT "MEAN LENGTH IS";Meanlength;"METERS"
8780 PRINT "SAMPLE STANDARD DEVIATION=";Stddev;"METERS"
8790 Maxit=MAX(Xxxx(M))
8800 CALL Minn(Minit,Xxxx(*),Nlayers)
8810 Diff=Maxit-Minit
8820 Errorit=Diff/Meanlength
8830 PRINT "PERCENT ERROR=";Errorit*100;"%"
8840 PRINTER IS 1
8850 !KIP'S SUGGESTED PLOT
8860 INPUT "DO YOU WANT A PLOT OF UNDEFORMED LAYERS?Y=1,N=2",Plt
8870 IF Plt=2 THEN GOTO 9830
8880 PRINTER IS 1
8890 FOR N=1 TO Nlayers
8900 IF Xxxx(N)*400/(Nratio*Scal)>150 THEN
8910 PRINT "WILL NOT FIT WITH CURRENT SIZE ON PLOTTER"
8920 GOTO 5090
8930 ELSE
8940 GOTO 8960
8950 END IF
8960 NEXT N
8970 FOR N=1 TO Nlayers
8980 Qq(N)=Xxxx(N)
8990 Xxxx(N)=Xxxx(N)*400/Scal

```

```

9020 INPUT "DO YOU WANT PLOTTER(1) OR CRT(2) PLOT?",Quest
9030 INPUT "DO YOU WANT MEAN LENGTH PLOTTED FOR ALL? Y=1,N=2",Mn
9040 IF Mn=1 THEN MAT Xxxx= (Meanlength*400/Scal)
9050 IF Quest=1 THEN
9060     PLOTTER IS 705,"HPGL"
9070     INPUT "MAKE SURE PLOTTER ARM IS FREE FOR A PEN, PRESS CONT(INUE)",F
9080     ELSE
9090         PLOTTER IS 3,"INTERNAL"
9100     END IF
9110 INPUT "ENTER PEN FOR WEDGE",Penn
9120 PEN Penn
9130 MOVE 0,0+10
9140 FOR N=Nlayers TO 1 STEP -1
9150     IF N=Nlayers THEN
9160         DRAW Xxxx(N),0+10
9170     ELSE
9180         DRAW Xxxx(N),Y_tot+10
9190     END IF
9200     DRAW Xxxx(N),Yyyy(N)+Y_tot+10
9210     DRAW 0,Yyyy(N)+Y_tot+10
9220     Y_tot=Yyyy(N)+Y_tot
9230 NEXT N
9240 MOVE 0,0+10
9250 DRAW 0,Y_tot+10
9260 IF Nfaults=0 THEN GOTO 9140
9270 IF Quest=1 THEN
9280     INPUT "ENTER PEN FOR FAULTS",Penn
9290     PEN Penn
9300 END IF
9310 Palin: !
9320 Section=1
9330 Click=1
9340 FOR J=1 TO Sds
9350     IF Layerea(J)=0 THEN
9360         Section=1+Section
9370         J=J+1
9380         GOTO 9430
9390     END IF
9400     IF Layerea(J)<>0 THEN
9410         Subtotal(Section)=Subtotal(Section)+Layerea(J)
9420     END IF
9430 NEXT J
9440 FOR J=1 TO Section-Nfaults-2 STEP 1
9450     Carea(J)=Subtotal(J)-Subtotal(J+1+Nfaults)+Sectionfault(J)*Scal^2/1600
9460     Carea(J)=Carea(J)*160000/Scal^2
9470 NEXT J
9480 Dave=Y_tot
9490 DIM R(10),Dist(10)
9500 Nosections=Section-Nfaults-2
9510 Num_lay=Nosections/Nlayers
9520 FOR J=1 TO Nlayers
9530     R(J)=Y_tot-Yyyy(J)/2
9540     Y_tot=Y_tot-Yyyy(J)
9550 NEXT J
9560 FOR J=1 TO Nlayers
9570     Aratio(J)=Meanlength/Qq(J)
9580 NEXT J
9590 I=0
9600 FOR J=1 TO Num_lay
9610     I=I+1
9620     H=I
9630     FOR K=1 TO Nlayers
9640         Dist(K)=Aratio(K)*(Carea(H)/Yyyy(K))+Dist(K)

```

```
9670     NEXT K
9680     DIM Geek(100),Yuck(100),S(100),T(100),C(10,10)
9690     CALL Interval(R(*),Dist(*),0,Dave,Nlayers,Geek(*),Yuck(*))
9700     FOR K=0 TO Nlayers-1
9710         Geek(K)=Geek(K+1)
9720         Yuck(K)=Yuck(K+1)
9730     NEXT K
9740     CALL Poly(Geek(*),Yuck(*),Nlayers,S(*),T(*),A,B,C(*))
9750     REDIM Geek(100)
9760     REDIM Yuck(100)
9770     PLOTTER IS 705,"HPGL"
9780     FOR L=1 TO A
9790         IF L=1 THEN MOVE T(L),S(L)+10
9800         DRAW T(L),S(L)+10
9810     NEXT L
9820     NEXT J
9830 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$(50)[7],Nx(2200),Ny(2200),Az(8,500),Bz(8,500)
70     DIM Nodigitpts(7),Porder(30,6),Click(50)
80     DEG
90     GOSUB 720
100    GOSUB 810
110    GOSUB 1220
120    PRINT "DO YOU WANT LOCATION OF ANY INFLECTION POINTS? Y=1"
130    INPUT Inflex
140    IF Inflex=1 THEN GOSUB 1510
150    GOSUB 620
160    Create_file: !
170    !THIS PROGRAM SEGMENT DIGITIZES A CROSS SECTION AND PLACES IT
180    !IN A THREE DIMENSIONAL GRID,CORRECTLY. IT PUTS THIS INFORMATION INTO
190    !A FILE.
200    !FIRST POINT IS REFERENCE POINT. CHOOSE IT SO THAT IT IS LEFT
210    !MOST POINT.
220    CALL Echoing(X(*),Y(*),Ndat,Tags$(*),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)

```

```

510  FRANKLIN IS 1
520  INPUT "ENTER NAME FOR THIS FILE",Filename$
530  CREATE BDAT "3_D"&Filename$,20
540  ASSIGN @Path TO "3_D"&Filename$
550  OUTPUT @Path;Ax,Ay,Cx,Cy
560  OUTPUT @Path;Click(*)
570  OUTPUT @Path;Xx(*)
580  OUTPUT @Path;Yy(*)
590  OUTPUT @Path;H(*)
600  OUTPUT @Path;Tags$(*)
610  STOP
620  Mapp: !*****
630  INPUT "ENTER ELEVATION OF FIRST POINT IN METERS",Refelev
640  PRINT "DIGITIZE LEFT CROSS SECTION ENDPOINT (REF POINT)ON MAP"
650  GRAPHICS INPUT IS 705,"HPGL"
660  DIGITIZE Ax,Ay
670  PRINT "DIGITIZE OTHER END POINT OF CROSS SECTION"
680  DIGITIZE Cx,Cy
690  Ss=(Cy-Ay)/(Cx-Ax)
700  Sectionangle=ATN(ABS(Ss))
710  .RETURN
720  Labeler: !*****
730  !THIS SUBROUTINE DETERMINES LABELS
740  INPUT " ENTER NUMBER OF STRUCTURAL FEATURES TO BE PROJECTED TOTAL",Struct
750  FOR I=1 TO Struct
760  PRINT "ENTER LABEL (7 LETTERS OR LESS) FOR STRUCTURE";I
770  INPUT Tags$(I)
780  NEXT I
790  RETURN
800  !*****
810  Read: !THIS SUBROUTINE READS AN EXISTING FILE SO THAT IT
820  !CAN BE DRAWN
830  INPUT "IF POLYNOMIALS NOT FIT TO DATA, PRESS HERE(1)",Eek
840  INPUT "ENTER NAME OF FILE TO BE READ",Name$
850  IF Eek=1 THEN
860  ASSIGN @Path TO "BAL"&Name$
870  ON END @Path GOTO 930
880  ENTER @Path;Wx,Wy
890  J=J+1
900  Nx(J)=Wx
910  Ny(J)=Wy
920  GOTO 880
930  Sds=J-1
940  GOTO 1200
950  END IF
960  ASSIGN @Path TO "POLY"&Name$
970  ON END @Path GOTO 1000
980  ENTER @Path;Nx(*)
990  ENTER @Path;Ny(*)
1000 J=1
1010 IF Ny(J)<>0 THEN
1020 J=J+1
1030 GOTO 1010
1040 END IF
1050 Sds=J-1
1060 FOR J=1 TO Sds
1070 IF Nx(J)=30000 THEN Check=1
1080 NEXT J
1090 IF Check=0 THEN
1100 Nfaults=0
1110 GOTO 1200
1120 END IF
1130 Two$="FAULT"
1140 ASSIGN @File TO Two$&Name$

```

```

1160 ENTER @File;Nfaults
1170 ENTER @File;Nodigitpts(*)
1180 ENTER @File;Az(*)
1190 ENTER @File;Bz(*)
1200 RETURN
1210 !*****
1220 Draw: !
1230 PLOTTER IS 3,"INTERNAL"
1240 FOR R=1 TO Sds
1250   IF R=1 THEN MOVE Nx(1),Ny(1)
1260   IF R=Sds THEN GOTO 1410
1270   IF Nx(R)=20000 THEN
1280     MOVE Nx(R+1),Ny(R+1)
1290     GOTO 1400
1300   ELSE
1310     GOTO 1330
1320   END IF
1330   IF Nx(R)=30000 THEN
1340     MOVE Nx(R+1),Ny(R+1)
1350     GOTO 1400
1360   ELSE
1370     GOTO 1390
1380   END IF
1390   PLOT Nx(R),Ny(R)
1400 NEXT R
1410 IF Nfaults=0 THEN GOTO 1500
1420 !NOW THE FAULTS ARE DRAWN IN (IF THERE ARE ANY)
1430 FOR Z=1 TO Nfaults
1440   FOR M=1 TO Nodigitpts(Z)
1450     IF Az(Z,M)=0 THEN GOTO 1480
1460     IF Az(Z,M-1)=0 THEN MOVE Az(Z,M),Bz(Z,M)
1470     DRAW Az(Z,M),Bz(Z,M)
1480   NEXT M
1490 NEXT Z
1500 RETURN
1510 Inflex: !
1520 ASSIGN @Myway TO "INFL"&Name$
1530 ENTER @Myway;Porder(*)
1540 INPUT "ENTER SECTION NUMBER YOU WANT INFLECTION POINT (NONE=0)",Sectionno
1550 IF Sectionno=0 THEN GOTO 1850
1560 Times=0
1570 FOR K=1 TO Sds
1580   IF Nx(K)=20000 THEN GOTO 1610
1590   IF Nx(K)=30000 THEN GOTO 1610
1600   GOTO 1700
1610   PRINT Nx(K)
1620   Times=1+Times
1630   IF Times=Sectionno-1 THEN Stpt=K+1
1640   IF Sectionno=1 THEN Stpt=1
1650   IF Times=Sectionno THEN
1660     Endpt=K-1
1670     PRINT Endpt
1680     GOTO 1710
1690   END IF
1700 NEXT K
1710 I=1
1720 FOR J=Stpt TO Endpt
1730   PRINT J
1740   Lx=Porder(Sectionno,I)
1750   IF Nx(J)=Lx THEN
1760     It=J
1770     MOVE Lx,Ny(It)
1780     CSIZE 4
1790     LOG 5
1800     I AREF "I"

```

```

1820     END IF
1830 NEXT J
1840 GOTO 1540
1850 RETURN
1860 END
1870 !*****
1880 Echoo:SUB Echoing(Aa(*),Bb(*),J,Tags$(*),Struct,Click(*))
1890     Xx=50
1900     Yy=50
1910     SET ECHO Xx,Yy
1920 Spin1: !
1930     ON KEY 5 LABEL "LEFT" GOTO 1990
1940     ON KEY 6 LABEL "RIGHT" GOTO 2020
1950     ON KEY 7 LABEL "UP" GOTO 2050
1960     ON KEY 8 LABEL "DOWN" GOTO 2080
1970     ON KEY 9 LABEL "MENU" GOTO 2450
1980     GOTO Spin1
1990     X=-.5
2000     Y=0
2010     GOTO Spin2
2020     X=.5
2030     Y=0
2040     GOTO Spin2
2050     X=0
2060     Y=.5
2070     GOTO Spin2
2080     X=0
2090     Y=-.5
2100     GOTO Spin2
2110 Spin2: !
2120     LOOP
2130         ON KNOB .01 GOTO 2150
2140         GOTO 2130
2150         SET ECHO Xx,Yy
2160         Xx=Xx+X
2170         Yy=Yy+Y
2180         ON KBD GOTO Spin3
2190     END LOOP
2200 Spin3: !
2210     INPUT "DO YOU WANT TO DIGITIZE A POINT HERE? Y=1,STOP=2",Digit
2220     IF Digit=1 THEN
2230         INPUT "ENTER STRUCTURE IDENTIFIER CODE LETTER",Code$
2240         FOR I=64 TO 64+Struct
2250             IF Code$=CHR$(I) THEN J=I-64
2260         NEXT I
2270         PRINT J
2280         Click(J)=1
2290         Aa(J)=Xx-X
2300         Bb(J)=Yy-Y
2310         PEN 1
2320         LORG 5
2330         CSIZE 4
2340         MOVE Aa(J),Bb(J)
2350         LABEL "+"
2360         MOVE Aa(J),Bb(J)+3
2370         LABEL Code$
2380     END IF
2390     IF Digit=2 THEN
2400         GRAPHICS OFF
2410         GOTO 2520
2420     END IF
2430     GOTO Spin1
2440     PRINT "*****"
2450     FOR I=1 TO Struct

```

```
2470     NEXT I
2480     PRINT "
2490     INPUT "PRESS 1 TO REMOVE THE MENU",R
2500     IF R=1 THEN ALPHA OFF
2510     GOTO Spin1
2520     SUBEND
2530 !*****
```

```

1      ! *****
2      ! *           PROJECTION           *
3      ! *****
10     DEG
11     GINIT
12     PLOTTER IS 3,"INTERNAL"
13     GRAPHICS ON
30     N=0
31     MAT Yarray= (1000000)
40     PRINTER IS 1
50     DIM Z(50),Y(50),X(50),Xxx(50),Yyy(50),Rr(50),Xx2(50),Yarray(5,50)
60     DIM R(50),Xdist(50),Ydist(50),X1(50),Y1(50),Z1(50),H(50),Xarray(5,50)
70     DIM Radi(5,50),Z2(50),Name$(50)[8],Click(50),Thecount(50)
80     DIM Tag$(50)[7],Lab$(50)[7],Rtotal(50)
81     DIM Weighted_dev(10,50),Xcoord(50),Ycoord(50)
90     INPUT "ENTER SCALE IN THE FORM 1:_____ (FILL IN BLANK)",Scal
110    INPUT "DO YOU WANT TO DIGITIZE ON PLOTTER(1) OR CRT(2)?",Digdev
111    GOSUB 3110
120    IF Digdev=1 THEN
130        GRAPHICS INPUT IS 705,"HPGL"
140        DIGITIZE A,B
150        PRINT "DIGITIZE OTHER ENDPOINT"
160        DIGITIZE C,D
170        PRINT "DIGITIZING FINISHED"
180    END IF
190    IF Digdev=2 THEN
200        CALL Echoing(A,B,C,D)
201        INPUT "PRESS 1 TO CLEAR SCREEN",M
202        IF M=1 THEN GRAPHICS OFF
210    END IF
220    IF A=C THEN
230        Ss2=0
240    ELSE
250        Ss2=(D-B)/(C-A)
260    END IF
270    Sectionangle2=ATN(ABS(Ss2))
290    N=0
300    REPEAT
310        N=N+1
320        PRINT "THIS IS CROSS SECTION";N,"WHICH IS CALLED",Name$(N)
330        ASSIGN @Path1 TO "3_D"&Name$(N)
340        ON END @Path1 GOTO 370
360        ENTER @Path1;Ax,Ay,Cx,Cy,Click(*),X(*),Y(*),Z(*),Tag$(*)
370        J=0
380        FOR K=1 TO 50
390            IF Click(K)=1 THEN J=J+1
410    NEXT K
420        Ndat=J
421        PRINT Ndat
430        Numb(N)=Ndat
440        PRINT "    DIGITIZED DATA POINTS"
450        PRINT "XCOORD","YCOORD","ELEVATION"
460        FOR J=1 TO 50
461            IF Click(J)=0 THEN GOTO 480
470            PRINT X(J),Y(J),Z(J),Tag$(J)
480        NEXT J
481        MAT Z1= (0)
490        FOR J=1 TO 50
492            IF Click(J)=0 THEN GOTO 530
500                GOSUB 1300
510                GOSUB 1950
520                GOSUB 2180
530        NEXT J
550    INPUT "DO YOU WANT TO VIEW THIS PROJECTION ON THE SCREEN? Y=1. N=2".View

```

```

500   IF ViewIt=1 THEN GOSUB 2300
570   FOR J=1 TO 50
571   IF Click(J)=0 THEN GOTO 573
572   Thecount(J)=1+Thecount(J)
573   NEXT J
620   MAT X= (0)
630   MAT Y= (0)
640   MAT Z= (0)
650   UNTIL N=Ncross
730   Weighted_avg:           !THIS PROGRAM OPTION DEDUCES THE WEIGHTED
740                           !AVERAGE POINT OF A SCATTER OF PROJECTED POINTS
750                           !FROM VARIOUS PROJECTED CROSS SECTIONS.
751                           GOSUB 3390
752   INPUT "DO YOU WANT TO PLOT THE LEAST SQUARE FIT?, Y=1",Ls
753   IF Ls=2 THEN GOTO 1290
755   INPUT "ENTER PEN NUMBER",Penn
760   PRINT "ENTER THE POINT LETTER THAT YOU WISH TO PERFORM A WEIGHTED"
770   PRINT "AVERAGE ON"
780   INPUT "FIN= ALL DONE",J$
781   IF J$="FIN" THEN GOTO 1270
783   FOR K=0 TO 49
784   IF CHR$(65+K)=J$ THEN It=K
785   NEXT K
786   I=It+1
790   MAT Weighted_dev= (0)
800   Rtot=0
810                           !RTOT IS THE SUM OF THE ALL R's FOR ONE SPECIFIC POINT.
820   J=I
830   Nn=0
840   LOOP
850   Nn=Nn+1
851   EXIT IF Nn=Ncross+1
860   IF Xarray(Nn,J)=0 AND Yarray(Nn,J)=0 THEN GOTO 850
870   Rtot=Rtot+(Radi(Nn,J))
890   END LOOP
900   Rtotal(J)=Rtot
910   Xtot=0
920   Ytot=0
930                           !WEIGHTED_DEV REPRESENTS THE INACCURACY FOR A SPECIFIC
940                           !POINT IN COMPARISON WITH OTHERS.
950   Nn=0
960   LOOP
970   Nn=Nn+1
971   EXIT IF Nn=Ncross+1
980   IF Xarray(Nn,J)=0 AND Yarray(Nn,J)=0 THEN GOTO 970
981   IF Radi(Nn,J)=0 THEN GOTO 970
990   Weighted_dev(Nn,J)=Rtotal(J)/Radi(Nn,J)
1000  Xtot=Weighted_dev(Nn,J)*Xarray(Nn,J)+Xtot
1010  Ytot=Weighted_dev(Nn,J)*Yarray(Nn,J)+Ytot
1030  END LOOP
1040  Sumweighted_dev=0
1050  Nn=0
1060  LOOP
1070  Nn=Nn+1
1071  EXIT IF Nn=Ncross+1
1080  IF Xarray(Nn,J)=0 AND Yarray(Nn,J)=0 THEN GOTO 1070
1090  Sumweighted_dev=Weighted_dev(Nn,J)+Sumweighted_dev
1110  END LOOP
1120                           !XCOORD AND YCOORD ARE THE DETERMINED AVERAGE POINTS
1130                           !AND ARE THE PLOTTED VALUES.
1140  Xcoord(J)=Xtot/Sumweighted_dev
1150  Ycoord(J)=Ytot/Sumweighted_dev
1160  PRINT "WEIGHTED AVERAGE OF PROJECTED POINT";J
1170  PRINT "HAS COORDINATES X=";Xcoord(J);"Y=";Ycoord(J)
1180  PRINT " "

```

```

1230      MOVE Xcoord(J),(Ycoord(J)-Minelev)*400/Scal
1240      LABEL "*"
1261      GOTO 760
1270      INPUT "DO YOU WANT TO LABEL PLOT ETC.? Y=1",Labe
1280      IF Labe=1 THEN GOSUB 2670
1290      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$="W" THEN
1460              IF Ss2<0 THEN
1470                  Newang=ABS(T+Sectionangle2)
1480              ELSE
1490                  Newang=ABS(T-Sectionangle2)
1500              END IF
1510          ELSE
1520              IF Ss2<0 THEN
1530                  Newang=ABS(T-Sectionangle2)
1540              ELSE
1550                  Newang=ABS(T+Sectionangle2)
1560              END IF
1570          END IF
1580          !R AND M REPRESENT ORTHAGONAL DISTANCES TO THE PROJECTED POINT.
1590          R(J)=SQR((Xxx(J)-X(J))^2+(Yyy(J)-Y(J))^2)
1600          M=R(J)*TAN(Newang)
1610          Xdist(J)=M*COS(Sectionangle2)
1620          Ydist(J)=M*SIN(Sectionangle2)
1630          IF Yyy(J)>Y(J) THEN Xdist(J)=-1*Xdist(J)
1640          IF X(J)>Xxx(J) THEN Ydist(J)=-1*Ydist(J)
1650          !X1 AND Y1 ARE THE ACTUAL INTERSECTIONS OF THE PROJECTED POINTS.
1660          !AT THIS POINT THEY HAVE BEEN PROJECTED ALONG A HORIZONTAL PATH
1670          IF Trendir$="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)

```

```

1890         X1(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
1961     INPUT "NORTH(N) OR SOUTH(S) ?",Plungdir$
1970 !RR IS THE HORIZONTAL DISTANCE THE PROJECTED POINT TRAVELS.
1980     Rr(J)=SQR((X1(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$="N" THEN
2050         IF Y1(J)<Y(J) THEN
2060             Z1(J)=H(J)+Z(J)
2070         ELSE
2080             Z1(J)=Z(J)-H(J)
2090         END IF
2100     ELSE
2110         IF Y1(J)<Y(J) THEN
2120             Z1(J)=Z(J)-H(J)
2130         ELSE
2140             Z1(J)=H(J)+Z(J)
2150         END IF
2160     END IF
2170     RETURN
2180 Rotation: !NOW WE HAVE TO ROTATE INTERMEDIATE CROSS SECTION TO BE PARALLEL
2190 !TO THE PLOTTER BASE IN ORDER TO DO A LEAST SQUARES REGRESSION.
2200 !ALL X1 POINTS MUST BE ROTATED AND THEIR VALUES ADJUSTED.
2210 !X2 IS THE NEW POINT ON THE ROTATED CROSS SECTION.
2230 !L IS THE DISTANCE FROM X1 TO THE END OF THE CROSS SECTION.
2240     L=SQR((X1(J)-A)^2+(Y1(J)-B)^2)
2250     Xx2(J)=A+L
2270     Xarray(N,J)=Xx2(J)
2280     Yarray(N,J)=Z1(J)
2290     RETURN
2300 Viewit: !THIS SUBROUTINE PLOTS THE RESULTING CROSS SECTION ON THE
2310 !SCREEN. IT PLOTS THE TICK-MARKS AND ELEVATION AND AXIS.
2320     GINIT
2330     PLOTTER IS 3,"INTERNAL"
2340     GRAPHICS ON
2350     LORG 5
2360     CALL Minn(Minz,Z1(*),50)
2390     FOR J=1 TO 50
2391     IF Click(J)=0 THEN GOTO 2460
2400         Z1(J)=400*Z1(J)/Scal
2410         Z1(J)=Z1(J)-400*Minz/Scal
2420         MOVE Xx2(J),Z1(J)+10
2430         LABEL "*"
2440         MOVE Xx2(J),Z1(J)+15
2450         LABEL CHR$(64+J)
2460     NEXT J
2470     MOVE Xx2(1),10
2480     DRAW MAX(Xx2(*))+3,10
2490     DRAW MAX(Xx2(*))+3,MAX(Z1(*))+10
2500     FOR K=1 TO (MAX(Xx2(*))-Xx2(1))/4
2510         MOVE K*4+Xx2(1),.5+10
2520         DRAW K*4+Xx2(1),0+10
2530     NEXT K
2540     FOR K=1 TO (MAX(Z1(*)/4)

```

```

2570     NEXT K
2580     MOVE MAX(Xx2(*)+20,MAX(Z1(*)+10
2590     LABEL INT((MAX(Z1(*))*Scal/400+Minz));"Meters"
2600     MOVE MAX(Xx2(*)+20,1+10
2610     LABEL INT(Minz);"Meters"
2620     PENUP
2630     PRINT "PRESS 1 IF YOU WANT TO REMOVE PLOT FROM THE SCREEN"
2640     INPUT C1r
2650     IF C1r=1 THEN GCLEAR
2660     RETURN
2670 Tickmark:INPUT "ENTER THE PEN NUMBER YOU WANT FOR LABELLING",Colour
2680     PEN Colour
2690     MOVE Xcoord(1),0
2700     DRAW MAX(Xcoord(*)+5,0
2710     DRAW MAX(Xcoord(*)+5,(MAX(Ycoord(*)-Minelev)*400/Scal
2720         !THIS SUBROUTINE DRAWS THE TICK MARKS
2730         !IT IS USED FOR WEIGHTED AVERAGE PLOT ONLY
2740     FOR K=1 TO (MAX(Xcoord(*)-Xcoord(1))/4
2750         MOVE K*4+Xcoord(1),.5
2760         DRAW K*4+Xcoord(1),0
2770     NEXT K
2780     FOR K=1 TO (MAX(Ycoord(*)-Minelev)*100/Scal
2790         MOVE MAX(Xcoord(*)+4,4*K
2800         DRAW MAX(Xcoord(*)+5,4*K
2810     NEXT K
2820     MOVE MAX(Xcoord(*)+15,1
2830     LABEL INT(Minelev);"METERS"
2840     MOVE MAX(Xcoord(*)+15,(MAX(Ycoord(*)-Minelev)*400/Scal
2850     LABEL INT(MAX(Ycoord(*)));"METERS"
2860     PEN 0
2870     RETURN
2880 Tickmarkii:!THIS SUB DRAWS TICK MARKS AND AXES FOR INDIVIDUAL
2890         !CROSS SECTIONS
2900     INPUT "ENTER PEN COLOR NUMBER ",Colour
2910     PEN Colour
2920     MOVE Xarray(N,1),0
2930     DRAW MAX(Xarray(*)+4,0
2940     DRAW MAX(Xarray(*)+4,(MAX(Z1(*)-Minelev)*400/Scal
2950     FOR K=1 TO (MAX(Xarray(*)-Xarray(N,1)+4)/4
2960         MOVE K*4+Xarray(N,1),.5
2970         DRAW K*4+Xarray(N,1),0
2980     NEXT K
2990     FOR K=1 TO (MAX(Z1(*)-Minelev)*100/Scal
3000         MOVE MAX(Xarray(*)+4,4*K
3010         DRAW MAX(Xarray(*)+3.5,4*K
3020     NEXT K
3030     CSIZE 2
3040     MOVE MAX(Xarray(*)+13,.5
3050     LABEL INT(Minelev);"METERS"
3060     MOVE MAX(Xarray(*)+13,(MAX(Z1(*)-Minelev)*400/Scal
3070     LABEL INT(MAX(Z1(*)));"METERS"
3080     PEN 0
3090     RETURN
3100     !*****
3110     N=1
3120     LOOP
3130     INPUT "ENTER CROSS SECTION NAME FOR PROJECTING PURPOSES, FIN=DONE",Name$(N)
3140     EXIT IF Name$(N)="FIN"
3150     ASSIGN @Path1 TO Name$(N)
3160     GOSUB 3220
3170     N=N+1
3180     END LOOP
3190     Ncross=N-1

```

```

3220 ENTER @Path1;Ax;Ay;Cx;Cy
3230 MOVE Ax,Ay+4
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
3400 !TO BE PLOTTED ON THE PLOTTER
3430 GINIT
3440 PLOTTER IS 705,"HPGL"
3450 GRAPHICS ON
3460 CSIZE 2
3470 INPUT "ENTER PEN COLOR NUMBER",Colour
3480 Minelev=MIN(Yarray(*))
3490 INPUT "ENTER THE CROSS SECTION NUMBER THAT YOU WANT TO PLOT",U
3500 PEN Colour
3510 LORG 5
3520 FOR J=1 TO 50
3521 IF Yarray(U,J)=1000000 THEN GOTO 3550
3530 MOVE Xarray(U,J),(Yarray(U,J)-Minelev)*400/Scal
3540 LABEL J
3550 NEXT J
3560 PEN 0
3570 INPUT "DO YOU WANT TO PLOT ANOTHER ONE? Y=1 N=2",Again
3580 IF Again=1 THEN GOTO 3470
3590 PRINT "DO YOU WANT TO PLOT TICKMARKS AND AXES? Y=1"
3600 INPUT Tick
3610 IF Tick=1 THEN GOSUB 2880
3620 RETURN
3630 END
3640 Echoo:SUB Echoing(A,B,C,D)
3643 Xx=50
3644 Yy=50
3645 SET ECHO Xx,Yy
3646 Spin1: !
3647 ON KEY 5 LABEL "LEFT" GOTO 3653
3648 ON KEY 6 LABEL "RIGHT" GOTO 3656
3649 ON KEY 7 LABEL "UP" GOTO 3659
3650 ON KEY 8 LABEL "DOWN" GOTO 3662
3651 ON KEY 9 LABEL "MENU" GOTO 3702
3652 GOTO Spin1
3653 X=-.5
3654 Y=0
3655 GOTO Spin2
3656 X=.5
3657 Y=0
3658 GOTO Spin2
3659 X=0
3660 Y=.5
3661 GOTO Spin2
3662 X=0
3663 Y=-.5
3664 GOTO Spin2
3665 Spin2: !
3666 LOOP
3667 ON KNOB .01 GOTO 3669
3668 GOTO 3667
3669 SET ECHO Xx,Yy
3670 Xx=Xx+X
3671 Yy=Yy+Y
3672 ON KBD GOTO Spin3

```

```

3074 spins: !
3675 Times=1+Times
3676 INPUT "DO YOU WANT TO DIGITIZE A POINT HERE? Y=1,STOP=2",Digit
3677 IF Digit=1 THEN
3678 IF Times=1 THEN
3684 A=Xx-X
3694 B=Yy-Y
3695 MOVE A,B
3696 LORG 5
3698 LABEL "+"
3699 GOTO Spin1
3700 ELSE
3701 C=Xx-X
3702 D=Yy-Y
3703 MOVE C,D
3704 LORG 5
3706 LABEL "+"
3707 END IF
3708 END IF
3709 SUBEND
3710 !*****
3711 Minn:SUB Minn(Ming,G(*),Numb)
3720 Ming=G(1)
3730 FOR Gg=1 TO Numb
3740 IF G(Gg)=0 THEN GOTO 3770
3750 IF Ming=0 THEN Ming=G(Gg)
3760 IF G(Gg)<Ming THEN Ming=G(Gg)
3770 NEXT Gg
3780 SUBEND
3790 !*****

```