DANIEL F. MERRIAM, Editor

COMPUTER PROGRAMS FOR AUTOMATIC CONTOURING

By

DONALD B. McINTYRE
Pomona College

DAVID D. POLLARD
Stanford University

and

ROGER SMITH
Oberlin College

COMPUTER CONTRIBUTION 23
State Geological Survey
The University of Kansas, Lawrence
1968

Editor's Remarks

Geologists, long accustomed to visual displays, will welcome this series of computer programs by D.B. McIntyre, D.D. Pollard, and Roger Smith. The programs represent a considerable step forward in resolving proglems of geologic contouring. Computer programs for automatic contouring is the largest and most involved program package yet made available through the COMPUTER CONTRIBUTION series. The package should prove to be one of the most popular and widely applied of the series.

The authors state that "the purpose of the programs is to plot contoured maps with irregularly spaced data, trend surfaces, and residuals." Use of the approach as described in the publication should be obvious. In addition to being extremely rapid, automatic contouring has the advantage of reproducibility and consistency.

The programs will find utilization in both research and routine applications in the many different aspects of earth science.

For a limited time, the Geological Survey will make available the program on magnetic tape for $40.00. Because of the size of the program and restrictions on mailing card decks, an extra charge of $10.00 must be made if a punched-card deck is required. An up-to-date list of COMPUTER CONTRIBUTIONS and related publications may be obtained by writing the Editor, COMPUTER CONTRIBUTIONS, Kansas Geological Survey, The University of Kansas, Lawrence, Kansas 66044.

Plans are now completed for the formation of an International Association for Mathematical Geology (IAMG). The organization's statutes have been drafted and include affiliation with the International Union of Geological Sciences. Information on the IAMG can be obtained from Prof. R.A. Reyment, Department of Historical Geology and Paleontology, University of Uppsala, Uppsala, Sweden.

The first issues of the GEOCOM Bulletin are now available. GEOCOM is a monthly international publication for geoscientists using computers and mathematical methods. Requests for information should be addressed to Mr. Graham Lea, Editor, P.O. Box 1024, Calgary 2, Alberta, Canada.

The North American Branch of the Classification Society is now being formed. The Society's aim is to promote cooperation and interchange of views among those interested in the principles and practice of classification. Information about the Society may be obtained from Dr. D.F. Merriam, Chairman of the Membership Committee, Kansas Geological Survey, The University of Kansas, Lawrence, Kansas 66044.

COMPUTER PROGRAMS FOR AUTOMATIC CONTOURING

By

DONALD B. McINTYRE, DAVID D. POLLARD, and ROGER SMITH

INTRODUCTION

Geologists are interested in contouring because it is a traditional and highly efficient technique for presenting information visually. Hand contouring, however, is a subjective procedure, susceptible to all pitfalls and hidden biases of other methods of subjective description. Empirically defined mechanical procedures, such as those used in the accompanying programs, are not necessarily better than other approaches, but they have the advantages of reproducibility and consistency. They also have the very real advantage of extreme rapidity of preparation.

The purpose of the programs is to plot contoured maps with irregularly spaced data, trend surfaces, and residuals. The contour program listed here is based on a square grid, the points of which are determined by second-degree trend surfaces fitted to the nearest stations. The package has been thoroughly tested and used for production work for several years.

Other publications of interest include those by Batcha and Reese (1964); Krumbein (1959); Ojakangas and Basham (1964); and Surkan, Denny, and Batcha (1964).

Acknowledgments.- David D. Pollard and Roger Smith contributed to development of the programs while holding NSF Undergraduate Research Participation grants; some of the programs were developed as part of a project supported by NSF grant GA-686. We are grateful to Dianne Willis of IBM for helping with the coding of subroutine DRAFT.

Computer time was provided generously by Western Data Processing Center, University of California at Los Angeles. From 1965 Pomona College provided time on its IBM System/360 and Calcomp plotter. Part of the off-line plotting was paid for by a grant from the Shell Assists Fund, Shell Companies Foundation.

METHOD

The first difficulty encountered in automatic contouring is that a plotter can move the pen only in either the up or down position along a straight line. Subroutine PLOT uses the coordinates of the new point to interpolate control points on the most direct path possible. A curved line must therefore be subdivided into straight-line segments, each short enough to give a satisfactory approximation. It is true that rounded contours can be drawn with an analog plotter (such as EAI model 3440) by taking advantage of the momentum of the pen and ignoring some of the computed points. Although this sort of filtering is subjective, the results are pleasing. But this is a technique employed by the operator of an off-line plotter, and the task of the computer programmer is unchanged.

Now, if the contours are to be subdivided into straight-line segments, the area must be dissected into triangles within which the surface is taken as planar. Ideally the data points should be at the corners of equilateral triangles forming a hexagonal pattern; but we have chosen instead a square grid, with four isosceles triangles meeting in the center of each square. Subroutine DRAFT works alternately down and up the columns of squares, determining as it goes the coordinates of the end points of each straight line segment. Because these points lie on the sides and diagonals of the squares, discontinuities in the contours are not possible. The output can be conceived of as a matrix with as many rows as there are straight-line segments, and whose four columns specify the end points (x_1, y_1) and (x_2, y_2).

After plotting begins, the matrix can be scanned to see if there is another line to be drawn without raising the pen; if there is not, the pen is lifted to the nearest remaining segment. This logic is incorporated in subroutine SIFT. Because the pen moves at the same speed whether it is up or down, this procedure saves considerable time, and is, of course, essential if the smoothing possibilities of an analog plotter are of interest. If the plotter is off-line, the size of the matrix to be sifted is determined by the available memory and the relative costs of computer and plotter time. But it is a disadvantage to have an on-line digital plotter idle while the sifting process is going on, so the relative sizes of the plotter and sift buffers should be adjusted to keep the plotter usefully employed. The buffer sizes given in the listings are for an IBM System 360/40 with 32K bytes and an on-line Calcomp model 565 plotter.

Data rarely are distributed on a grid, but a great simplification is effected if we can construct a grid that is an acceptable substitute. Provided that points which are close together or coincident have similar values attached to them, a very fine grid can come close to honoring every data point. If this condition does not exist, the data are described as noisy, for there is significant within-stations variance. The trouble is, of course, that the amount of computer time increases fourfold when the side of a grid-square is halved, and there will be many more individual line-segments to be sifted. Moreover, in order that the fine grid may honor every data point, spurious jogs will appear in the contour lines. Geological data usually are noisy, so a certain amount of smoothing is an advantage. If we are trying to contour mountainous terrain on the basis of a small number of first-order triangulation stations, our object being to portray the area as a whole, it is perhaps not necessary or even advisable to insist that the contours should exactly fit each data point. If the distribution had been better, data of lower individual quality might have been preferable. This is a sampling problem that has often been ignored. It should be remembered that each data point has to support the surface over a polygonal area extending outwards towards the adjoining points, and it is a waste of effort to determine elevation to a fraction of an inch at one point when there is an unrecorded change in elevation of a hundred feet only a few steps away.

In constructing the grid we use a least-squares criterion and fit a polynomial surface to the points closest to each grid intersection. This is subroutine SGRID (phase DBMC12). A plane can be fitted to three points; but then no degrees of freedom remain, the surface cannot rise above or fall below the control points, and meaningless results are obtained if the points are approximately collinear. To avoid this problem a quadratic surface is fitted. Because six coefficients are to be determined we must use at least six points, and because these may be distributed poorly we insist on a minimum of eight. A circle is constructed such that the probable number of points contained within it is ten, and if fewer than eight points are found, the radius is incremented and a second count is made. The number and magnitude of the increments can be changed, but experience shows that if more than one increment is allowed the result is likely to be unsatisfactory. Points are weighted inversely as the square of their distance from the grid intersection, and the surface is fitted. On this basis a value is assigned to the grid point. This value may be unsatisfactory because of the distribution of the data points, but instead of a time consuming test of the distribution we use the arbitrary criterion that no predicted value may lie outside the range of the values used to compute it by more than 20 percent of that range. If for any reason the program is not able to assign a satisfactory value to a grid point, it is flagged 999.9 and is ignored during plotting.

If too few data points are used in the definition of a grid point, the loss of degrees of freedom will obviously result in a poor prediction. If too many are used the result also may be bad. A second-degree surface may be an unsatisfactory fit over the larger area containing more numerous data points. The necessary compromise is not easily achieved, because it depends in part on the density of data points relative to the complexity of the actual surface. If the density or distribution of data points is poor, or the data are noisy, it is best to try for a smoothed surface. In this instance, the points should not be weighted (in DBMC12 remove cards DBM2-4,5 and DBM2-152 through 164, and add "WT(I) = 1.0" after statement #613, DBM2-174). The process then becomes that of fitting a moving average trend surface of the second degree.

It is sometimes convenient to consider the distribution of a variable as the superposition of local effects on a regional trend, and such data can be analyzed by studying the residuals from a least-squares surface. Areas of geologic interest are sometimes outlined more clearly by residuals than by the data in their original form. If the number of polynomial coefficients is increased, the sum of squares of the residuals will diminish until the deviation from the surface approaches the within-stations variance. The surface then represents the data with the noise eliminated. Many measurements thus are condensed into a small number of coefficients, and the noise level may be estimated even if the within-stations variance has not been measured independently. In 1959, Krumbein described a method of analysis applicable if the spacing of data points is irregular. His program was divided into parts, the output from one being the input for the next, and the largest matrix inverted was the 10x10 for a third-degree surface. We extended Krumbein's method to include eighth-degree surfaces, and published the program in 1963 (see McIntyre, 1963). It was written as a single unit for operation on the 7090/7094 system at Western Data Processing Center (WDPC). Being in FORTRAN, it used an inefficient method of building the large matrix. We later replaced this portion by a subroutine written in FAP, with considerable savings of space and time. The contouring program was developed to supplement the trend-surface program, but the two were kept separate because it proved advantageous to review the output of the first before using computer and plotter time for the second. For several years both programs were operated at WDPC, and the tapes were sent to service bureaus for off-line plotting. To prevent wasteful processing of bad tapes, status reports were printed while the plotter tapes were being written. This feature is retained in the version given here, although in its present form the program is for an on-line plotter. Pomona College took delivery of an IBM System 360/40 in 1965, and a Calcomp 565 plotter was attached on-line. Our original programs were modified for this configuration. Because the memory of the System 360 was only about one quarter that of the 7094, the program had to be broken into phases that are successively fetched for execution and overlay one another in core.

NOTES ON PROGRAM LISTINGS

These notes are intended to serve as an introduction and supplement to the numerous comment statements included in the listings.

The programs were run on an IBM System 360/40 with 32K bytes, under Disc Operating System Version 2, Release 11, and a 6K Supervisor. Three tape drives are required, as well as two other devices, which may be tapes or disc files. The card file for input is data set reference number 1, and printed output is data set reference number 3. Alphameric literals and logical IF statements are not available in this version of FORTRAN IV.

Twelve phases are catalogued in the core image library, from which they can be FETCHED by subroutine CHAIN or by an EXEC control card. To permit positioning of tape files and the use of variable formats we have written subroutines FILE and FORMAT, which like CHAIN should be catalogued in the relocatable library. We also find it advantageous to include the sorting routines FSORTA and ISORTA. The matrix inversion program DINVRT was written to take advantage of the hardware of System/360, and it too should be in the relocatable library.

The phases are as follows:

1. TREND: This phase reads control cards giving device assignments, the number of data points (stations), the instructions for scaling data, etc. It also accepts title cards and variable formats. Limits of U (along the paper) and V (across the paper) and the increments are defined, as are minimum and maximum values permitted for the trend surface. Data cards then are read, and the necessary summations are computed.

2. TREND2: Because instructions to build a large matrix require a lot of space, this phase relies on three ASSEMBLY language subroutines to build the matrix on tape. These three subroutines, PLACE1, PLACE2, and PLACE3, are not included, but a FORTRAN program which generates them is included along with the necessary data. The output from this FORTRAN program is a data set which is defined as input to a job that executes ASSEMBLY and punches object decks for the three subroutines required. It should be observed that this FORTRAN program can also create matrix building routines for hypersurfaces in U, V, W. It requires the function subprogram KTRAIL for converting binary integers into alphameric form. Although only 26,111 bytes of memory are available to the user, it is possible by this method to construct an eighth-degree matrix occupying 16,200 bytes, and then to invert this large matrix.

3. TREND3: This phase picks up the matrix of normal equations, built on tape by TREND2, and places it in core where it is ready for inversion.

4. TREND4: In this phase the different matrices are scaled and inverted. Because this is the step involving the greatest computational effort, it should be efficient, and to achieve effective use of the System/360 hardware, it is written in ASSEMBLY language.

5. TREND5: This phase generates the coefficients defining the surfaces of various degree. It uses subroutine TREAD, which reads tape by means of Physical Input-Output Control System (PIOCS).

6. TREND6: The coefficients being available, it is now possible to generate the different trend surfaces and residuals that were specified in the original data. The results are printed (DSR 3 = SYSLST) and also written on tape for contouring.

7. DBMCI1: It is of course possible to set up a deck so that, within a single job, contouring will immediately follow execution of TREND6; but it is advisable to pause at this stage and examine the printed output, checking on the goodness of fit before deciding what is to be plotted. When this decision has been made, phase DBMCI1 can be executed. It will operate on the data stored on tape by TREND6. The file called for may contain raw data, a trend surface, or residuals, and a check is first made to verify that the correct type of file has been found. Checks also are made to ensure that the various scaling factors asked for will yield a possible plot, and that all stations are within the bounds specified. This verification is particularly important for off-line plotting. Subroutines SDATA, SRESID, PDATA, CHECK1, and CHECK2 are used, in addition to those supplied by the plotter manufacturer.

8. DBMCI2: If raw data or residuals are to be plotted, this phase is fetched by DBMCI1 to prepare the grid.

9. DBMCI3: This phase processes a trend surface file, which is of course already in the form of a grid. Subroutine DRAFT is used to compute the line segments.

10. DBMCI4: This is fetched by DBMCI2 after the grid is constructed, and it too uses subroutine DRAFT.

11. DBMCI5: When the line segments have been computed by subroutine DRAFT, this phase first organizes them (subroutine SIFT) so that contour lines can be drawn continuously across the map, and then it invokes the plotter routines.

12. DBMCID: This phase is a modification of DBMCI1 that permits contouring of raw data without first passing through the trend surface program. It uses subroutines GETDTA and CHECK1 as well as the plotter routines.

We think that by segmenting the program and providing many comment statements considerable flexability is achieved. If a larger core is available, phases can be combined or larger matrices manipulated. Individual subprograms can either be eliminated if not needed, or lifted for use in other programs.

TEST DATA FOR TREND

Data for the trend part of the program are identified in the listing by the sequence codes DATA 0 through 117. In the following description cross references are given in parentheses to the comment and READ statements. For further information regarding the test data see the discussion and contoured maps in McIntyre (1967).

DATA 0: (TRND 55-56) Format 16I4
Output for plotting and contouring is data set reference 10. That is, it is to be written on a device with logical number 10; the physical assignment is made by the operator, and it should be magnetic tape. DSR 9, 8, 7 specify work files (TRND 36-37).

There are 100 stations with 1 reading at each (TRND 39-40). No transformations are to be made (TRND 40-41). Trend surface coefficients are to be computed through eighth degree. When trend surfaces are printed there will be 14 values per line. Trend surface coefficients are to be printed (TRND 44-45).

DATA 1: (TRND 65) Format 18A4
The first 72 characters are used as a title for printed output and for identification of plots.

DATA 2-3: (TRND 65) Formats 18A4
Comments to be included in printed output.

DATA 4-7: (TRND 66) Formats 18A4
These four cards define the formats for input and output (TRND 51-53). In order to maintain significant figures the values of the coordinates U and V, which will be raised to high powers, must be as close to unity as possible. If decimal points are not punched, scaling can be effected by the variable format for input.

DATA 8-15: (TRND 76-89) Formats I2, 6F6.4
Control cards for each degree up to the maximum called for (in this instance eighth degree).

Code 99 specifies that degrees 1, 2, 4, 6, and 7 are to be ignored.

Code 1 on the third card specifies that we are to compute residuals, but not the trend surface, for third degree.

Code -1 on the fifth card specifies that we are to compute the trend surface, but not the residuals, for the fifth degree. Both U and V are to go from 0 to 1.0 by increments of 0.05.

Code 0 on the eighth card specifies that both trend surface and residuals are to be computed for the eighth degree.

DATA 16-17: (TRND 90-91) Variable format, DATA 4.
Two dummy data cards specify the smallest and largest values permitted in the trend surface. Any values outside of this range will be set to 999.9, and will be ignored by the contouring program.

DATA 18-117: (TRND 112) Variable format, DATA 4.
One data card for each station. To save time in plotting, the stations should be sequenced in such a way that the pen works across the paper without waste motion. This can be done with a card sorter, or the computer can be programmed to do the sort. It is not necessary to have the origin at the center of the plot, and we usually place it at the bottom left-hand corner.

TEST DATA FOR CONTOURING AFTER TREND

Data for contouring after computing trend surfaces are identified by sequence codes PLOT 0-19. Examples of data for five different plots are included, each requiring four data cards.

PLOT 0 (also 4, 8, 12, 16): (DBM1 75) Format 3I2
DSR 7, 8, 9 are defined as three work files.

PLOT 1 (also 5, 9, 13, 17): (DBM1 47-58, 77-78) Format 8F6.2, 3I2
A blank in the first field indicates that subroutine FACTOR is not to change the scaling factor. The plotting is to be done on a Calcomp digital plotter with 0.01" step; if the plotter has a 0.005" step, this field should contain 2.00.

Both U and V go from 0 at the origin to maxima of 1.0.

The plot is not to exceed 10.1" in either U or V. This information permits a check on the scaling values that are specified.

Lettering and symbols are to have a height of 0.07". This is the minimum possible on this plotter, but a larger size (e.g. 0.10") should be used if possible.

The 3 in the next field specifies that the locations are to be marked by a +; a zero in this field (PLOT 9) causes suppression of plotting of the stations. It should be remembered that the time to plot the stations is likely to exceed the time required for contouring.

The next field specifies the number of decimal places to be used for writing data values on the plotted output. If the number is negative (PLOT 5), the values will not be written by the plotter.

The output from the trend surface is here defined as DSR 10; it should be written on magnetic tape.

PLOT 2 (also 6, 10, 14, 18): (DBM1 60-71, 94) Format 2I2
In this example, the output from the trend surface program was written as five files as follows:

File	Contents	Label
0	Data	1
1	third-degree residuals	3
2	fifth-degree trend	2
3	eighth-degree trend	2
4	eighth-degree residuals	3

Hence "0 1" (PLOT 2, 6, 10) locates the data file; "3 2" (PLOT 14) locates the file with the eighth-degree trend; and "4 3" (PLOT 18) locates the file with eighth-degree residuals. In each instance the label will be checked to ensure that the correct type of file has been found. It should be noted that the first file, which always contains the original data, is numbered 0.

PLOT 3 (also 7, 11, 15, 19): (SDTA 14-18, 27: SRSD 14-15, 28; DBM1 150) Format 3F7.3, I2
The first field specifies that U and V are to be multiplied by 7.5 to convert the units to inches for plotting.

The second field gives the side of the grid in inches (0.75"). For a trend-surface file (PLOT 15) this is not relevant.

The third field gives the contour interval. If this is zero then the values at the stations, or merely the positions of the stations, can be plotted without contouring.

The fourth field (relevant only for a data file) specifies which data value is to be used when more than one variable is recorded at each station.

TEST DATA FOR CONTOURING WITHOUT TREND

Data for contouring or plotting directly from a card file, without first passing through the trend surface program, are identified as CONT 0-105.

CONT 0: (DBMD 40-41,43) Format 3I2, I4
DSR 7, 8, 9 are defined as three work files, and there are 100 stations.

CONT 1: (DBMD 45-46) Format 8F6.2, 2I2
Identical to PLOT 1 except that no input file is specified.

CONT 2: (DBMD 58) Format 18A4
Title: Identical to DATA 1.

CONT 3: (GETD 42) Format 3F7.3
Identical to PLOT 3

CONT 4: (GETD 53) Format 20A4
Variable format for input.

CONT 5-104: Data identical to DATA 18-117

CONT 105: Blank. Only required if the program is able to cycle.

OUTPUT

The results of automatic contouring of the 100 randomly distributed points used here as test data, and also of the eighth-degree surface fitted to them, have already been reproduced (McIntyre, 1967). So, to illustrate the present paper, we include two maps contoured as before in every respect except one. Whereas previously in the construction of the grid the data points were weighted inversely as the square of their distances from the grid intersections, in Figure 1 there is no such weighting and the surface is simply a moving-average second-degree trend. The result is somewhat smoother than the map constructed by weighting (McIntyre, 1967, fig. 2), but it is perhaps less true to the original pattern (McIntyre, 1967, fig. 1). For the second example (Figure 2) the data were again weighted, but the grid squares were reduced from 1 percent to 0.5 percent of the total area and the amount of computation is, of course, nearly quadrupled. Although the data points are fitted closer than they were before, the overall result is poor and certainly not worth the extra cost. It is thus obvious that simply making the grid finer does not necessarily make the map better.

A geologist is often, and perhaps always, justified in modifying by hand and eye the output plotted by the machine because he is likely to have information not given to the computer. He may have additional data of a qualitative kind or he may know the form of surfaces describing related variables over the same area. From experience in similar instances he will probably have some judgment as to what is probable, how well the data points are to be honored, and whether jogs in the contours are real or should be smoothed. Because the geologist will always insist on having the last word, and rightly so, the programming effort and machine time required to get the computer to do much more may not be worth while.

PC112 100 RANDOM POINTS. KANSAS CONTR.12, 1967, P 45-46

RAW DATA

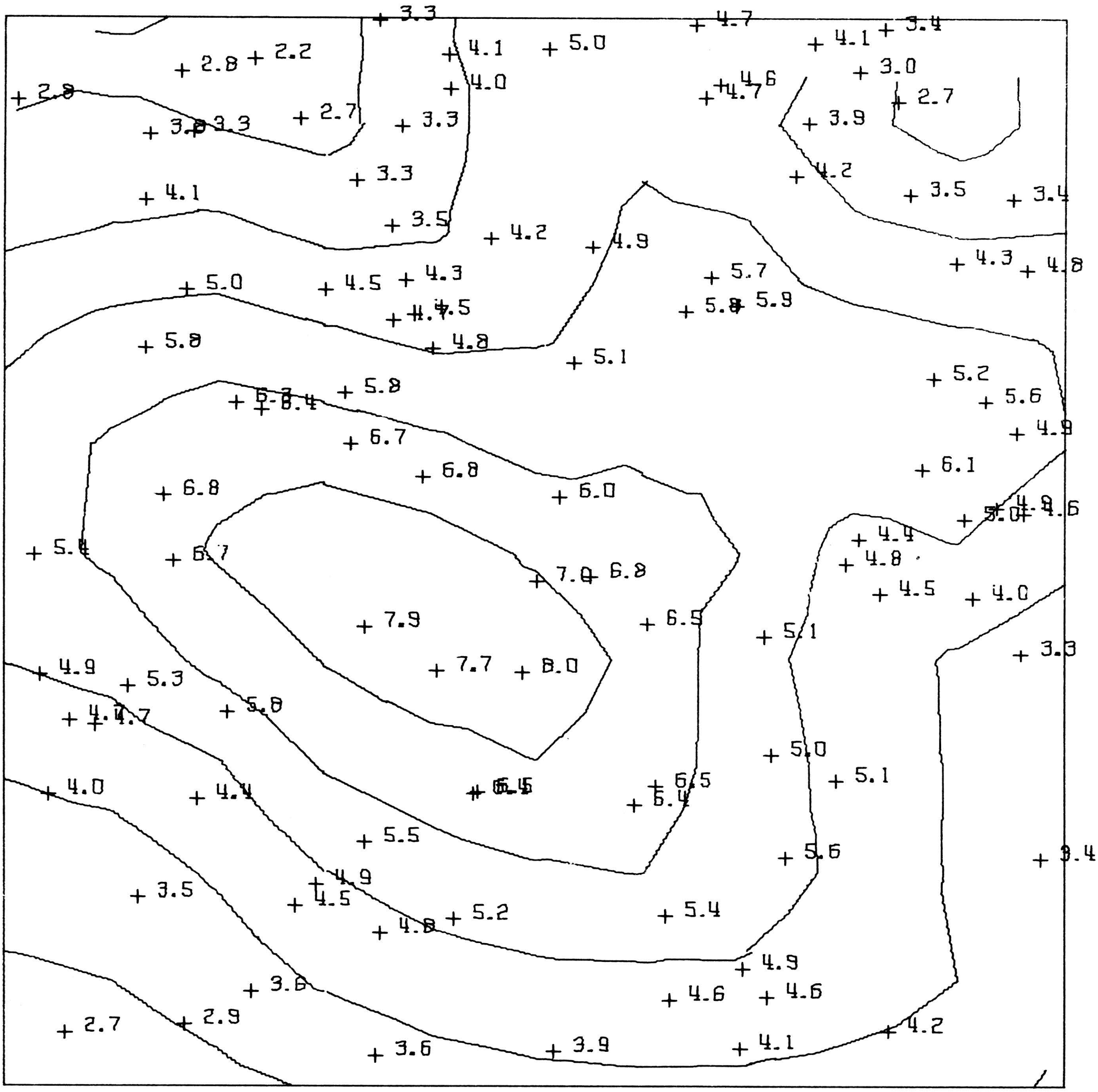

Figure 1. - Contoured as moving average, second-degree trend.

PC112 100 RANDOM PØINTS

RAW DATA

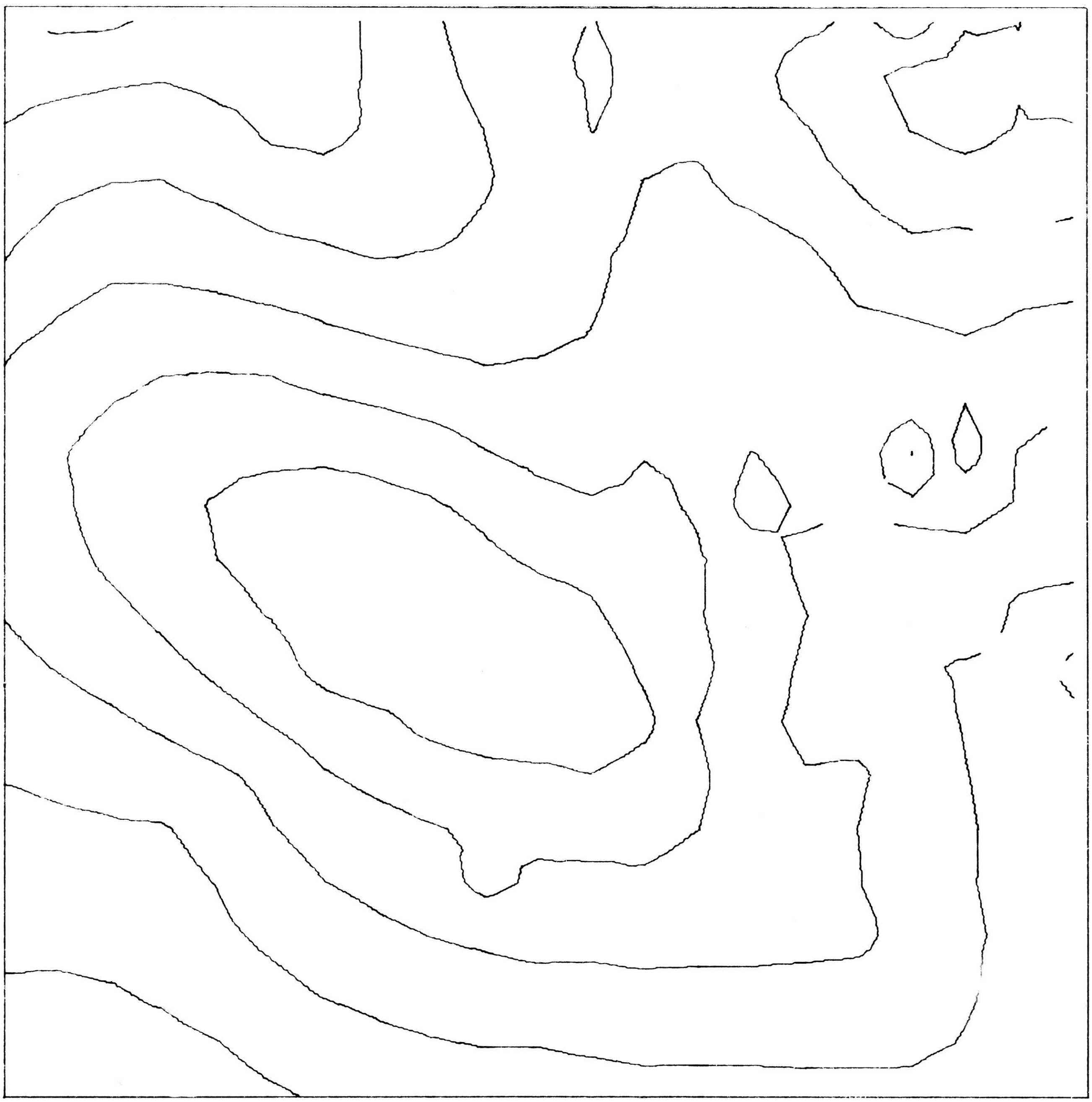

Figure 2. - Contoured using smaller grid and weighting data points. Data same as Figure 1.

PC112 100 RANDOM POINTS. KANSAS CONTR.12, 1967, P 45-46

THE COORDINATES U AND V WERE FIXED BY A RANDOM NUMBER ROUTINE
BUT THE VALUES OF Y ARE INTERPOLATED FROM A GIVEN PATTERN.

```
CONTROL NUMBERS  10   9   8   7 100   1   0   0   8  14   1   0   0   0   0   0

INPUT FORMAT IS (T21,A2,T1,F3.3,F7.3,F7.2)

PERMITTED RANGE     1.0000 TO     9.0000

COEFFICIENT VECTORS ARE

DEGREE 1

   0.5340899336D 01   0.3636383363D-01  -0.1027454834D 01

DEGREE 2

   0.2040794907D 01   0.7489761688D 01   0.1062395125D 02  -0.7980447126D 01   0.8622661216D 00  -0.1148533659D 02

DEGREE 3

  -0.2494550713D 00   0.1285279880D 02   0.2328682607D 02  -0.8433812877D 01  -0.2199470159D 02  -0.2789380213D 02
  -0.2862164770D 01   0.1008834169D 02   0.1216638643D 02   0.6517330589D 01

DEGREE 4

   0.5043057947D 00   0.1078847854D 02   0.8645035748D 01   0.9062453268D 01  -0.2858084245D 02   0.4013315820D 02
  -0.3791408066D 02   0.1679710636D 02   0.2400285215D 02  -0.1014543621D 03   0.2156622948D 02  -0.1192313479D 02
   0.1129055102D 02  -0.1426926899D 02   0.5562065897D 02

DEGREE 5

   0.4422093538D 01  -0.1409921605D 02  -0.3270333209D 02   0.7203049366D 01   0.2768429347D 03   0.1198640734D 03
   0.1085866181D 03  -0.4327314958D 03  -0.6684596806D 03  -0.7459162107D 02  -0.2060191696D 03   0.2291686840D 03
   0.7281741441D 03   0.4971798576D 03  -0.8360431600D 02   0.1080246010D 03  -0.7609987640D 02  -0.1180814108D 03
  -0.3641136946D 03  -0.8087512797D 02   0.6848233446D 02

DEGREE 6

   0.6028920620D 01  -0.2888670917D 02  -0.6236541835D 02   0.9558589780D 02   0.3343402356D 03   0.3091262215D 03
  -0.6369024937D 02  -0.6235097402D 03  -0.9266208744D 03  -0.5555617442D 03  -0.2241742073D 03   0.9792862593D 03
   0.3595210569D 03   0.1481461709D 04   0.3813137100D 03   0.4057953381D 03  -0.9463358518D 03  -0.1678097887D 03
   0.2524557194D 03  -0.1468993788D 04  -0.4476664420D 01  -0.1914747463D 03   0.3929272456D 03  -0.4884821449D 02
   0.6773553319D 02  -0.3274253756D 03   0.6466435083D 03  -0.7467593364D 02

DEGREE 7

  -0.2958171027D 01   0.4238756637D 02   0.1578008833D 03  -0.1128101631D 03  -0.1194604543D 04  -0.1043350509D 04
   0.2460771920D 03   0.3473594787D 04   0.6232258780D 04   0.3164116251D 04  -0.5680760295D 03  -0.4635072603D 04
  -0.1347346269D 05  -0.1344537287D 05  -0.4824455679D 04   0.8331566019D 03   0.2979530154D 04   0.1367810193D 05
   0.1888038255D 05   0.1479103171D 05   0.3577151640D 04  -0.5635748347D 03  -0.8999218915D 03  -0.6760344595D 04
  -0.1131564070D 05  -0.1189566487D 05  -0.8397762686D 04  -0.9924916887D 03   0.1233210937D 03   0.1852318201D 03
   0.1111690906D 04   0.2853991050D 04   0.2969462837D 04   0.2876501142D 04   0.2003692084D 04  -0.3585896308D 02

DEGREE 8

  -0.4142150128D 01  -0.1337702088D 01   0.2515390861D 03   0.1999522057D 03  -0.1069320933D 04  -0.2133184410D 04
  -0.9631409699D 02   0.9256632754D 03   0.7700040696D 04   0.8559258819D 04  -0.3769369950D 04   0.5278249444D 04
  -0.1030814575D 05  -0.2231044644D 05  -0.1891887067D 05   0.1463437578D 05  -0.2055155492D 05   0.8904509670D 04
   0.1442483615D 05   0.3751902055D 05   0.2433621545D 05  -0.2409229499D 05   0.3255922216D 05  -0.6345347581D 04
  -0.9315887070D 03  -0.1301514088D 05  -0.3669706925D 05  -0.1832117319D 05   0.1894385689D 05  -0.2491126746D 05
   0.2184315479D 04   0.8505628456D 03  -0.5782484974D 04   0.8650977429D 04   0.1899967761D 05   0.7589270643D 04
  -0.5840420609D 04   0.7804293635D 04  -0.1077868893D 04   0.7282112203D 03   0.8813110494D 03   0.2136377677D 04
  -0.2461332875D 04  -0.4044278229D 04  -0.1355900833D 04
```

3 DEGREE RESIDUAL

ID	U	V	Y	PRED	RESID
0	0.057	0.051	2.70	1.51	1.19
1	0.041	0.274	4.00	4.48	-0.48
2	0.085	0.339	4.70	5.23	-0.53
3	0.061	0.343	4.70	5.11	-0.41
4	0.034	0.385	4.90	5.16	-0.26
5	0.028	0.497	5.40	5.37	0.03
6	0.012	0.923	2.80	2.64	0.16
7	0.169	0.059	2.90	2.75	0.15
8	0.125	0.178	3.50	4.10	-0.60
9	0.182	0.270	4.40	5.35	-0.95
10	0.116	0.375	5.30	5.57	-0.27
11	0.159	0.492	6.70	5.92	0.78
12	0.150	0.553	6.80	5.79	1.01
13	0.132	0.691	5.80	5.10	0.70
14	0.171	0.745	5.00	4.82	0.18
15	0.132	0.830	4.10	3.97	0.13
16	0.136	0.891	3.80	3.36	0.44
17	0.177	0.893	3.30	3.46	-0.16
18	0.165	0.949	2.80	2.80	-0.00
19	0.235	0.962	2.20	2.87	-0.67
20	0.278	0.905	2.70	3.61	-0.91
21	0.218	0.640	6.30	5.63	0.67
22	0.241	0.633	6.40	5.73	0.67
23	0.209	0.351	5.80	5.92	-0.12
24	0.293	0.189	4.90	5.24	-0.34
25	0.274	0.169	4.50	4.96	-0.46
26	0.232	0.089	3.60	3.72	-0.12
27	0.350	0.030	3.60	3.58	0.02
28	0.353	0.144	4.80	5.06	-0.26
29	0.339	0.229	5.50	5.75	-0.25
30	0.339	0.431	7.90	6.46	1.44
31	0.394	0.571	6.80	6.25	0.55
32	0.325	0.602	6.70	6.04	0.66
33	0.320	0.649	5.80	5.79	0.01
34	0.365	0.718	4.70	5.42	-0.72
35	0.385	0.723	4.50	5.41	-0.91
36	0.301	0.745	4.50	5.12	-0.62
37	0.378	0.755	4.30	5.17	-0.87
38	0.364	0.805	3.50	4.74	-1.24
39	0.331	0.848	3.30	4.30	-1.00
40	0.373	0.899	3.30	3.90	-0.60
41	0.352	0.998	3.30	2.83	0.47
42	0.417	0.965	4.10	3.36	0.74
43	0.419	0.933	4.00	3.67	0.33
44	0.458	0.794	4.20	4.95	-0.75
45	0.403	0.691	4.80	5.64	-0.84
46	0.407	0.390	7.70	6.53	1.17
47	0.488	0.388	8.00	6.56	1.44
48	0.446	0.276	6.50	6.25	0.25
49	0.442	0.275	6.40	6.24	0.16
50	0.423	0.157	5.20	5.40	-0.20
51	0.518	0.033	3.90	4.21	-0.31
52	0.593	0.264	6.40	6.13	0.27
53	0.501	0.473	7.00	6.53	0.47
54	0.552	0.477	6.80	6.47	0.33
55	0.523	0.552	6.00	6.32	-0.32
56	0.536	0.678	5.10	5.74	-0.64
57	0.553	0.786	4.90	5.05	-0.15
58	0.512	0.971	5.00	3.52	1.48
59	0.651	0.993	4.70	3.59	1.11
60	0.674	0.937	4.60	4.00	0.60
61	0.660	0.926	4.70	4.07	0.63
62	0.666	0.758	5.70	5.13	0.57
63	0.690	0.731	5.90	5.23	0.67
64	0.641	0.726	5.80	5.35	0.45
65	0.605	0.433	6.50	6.41	0.09
66	0.614	0.282	6.50	6.15	0.35
67	0.623	0.160	5.40	5.46	-0.06
68	0.696	0.111	4.90	4.84	0.06
69	0.627	0.081	4.60	4.75	-0.15
70	0.694	0.036	4.10	4.09	0.01
71	0.719	0.084	4.60	4.50	0.10
72	0.736	0.215	5.60	5.39	0.21
73	0.784	0.287	5.10	5.42	-0.32
74	0.723	0.311	5.00	5.84	-0.84
75	0.716	0.421	5.10	6.02	-0.92
76	0.794	0.489	4.80	5.57	-0.77
77	0.746	0.852	4.20	4.47	-0.27
78	0.757	0.902	3.90	4.19	-0.29
79	0.763	0.976	4.10	3.79	0.31
80	0.830	0.990	3.40	3.72	-0.32
81	0.806	0.949	3.00	3.91	-0.91
82	0.842	0.921	2.70	3.98	-1.28
83	0.853	0.835	3.50	4.29	-0.79
84	0.897	0.771	4.30	4.32	-0.02
85	0.876	0.663	5.20	4.73	0.47
86	0.865	0.578	6.10	4.99	1.11
87	0.806	0.512	4.40	5.47	-1.07
88	0.826	0.461	4.50	5.38	-0.88
89	0.833	0.052	4.20	3.53	0.67
90	0.977	0.213	3.40	3.36	0.04
91	0.959	0.406	3.30	4.21	-0.91
92	0.914	0.457	4.00	4.69	-0.69
93	0.906	0.531	5.00	4.74	0.26
94	0.961	0.536	4.60	4.27	0.33
95	0.936	0.541	4.90	4.49	0.41
96	0.955	0.612	4.90	4.27	0.63
97	0.925	0.642	5.60	4.45	1.15
98	0.964	0.764	4.80	3.96	0.84
99	0.951	0.831	3.40	3.90	-0.50

VARIANCE OF DATA 0.1540863E 01 SD 0.1241315E 01 TOTAL SS 0.1525454E 03

RESID SS 0.4276964E 02 DF 90 MS 0.4752182E 00 SD 0.6893607E 00
71.963 PERCENT SS ACCOUNTED FOR

5 DEGREE TREND SURFACE

	INITIAL	FINAL	INCREMENT
U	0.0	0.1000000E 01	0.5000000E-01
V	0.0	0.1000000E 01	0.5000000E-01

COLUMN NUMBER 1

1.87	2.47	3.29	4.16	4.98	5.65	6.11	6.34	6.33	6.08	5.63	5.02	4.31	3.57
2.87	2.31	1.97	1.92	2.27	3.08	4.42							

COLUMN NUMBER 2

2.27	2.65	3.32	4.11	4.90	5.60	6.14	6.48	6.59	6.49	6.18	5.71	5.12	4.46
3.80	3.22	2.77	2.54	2.58	2.97	3.75							

COLUMN NUMBER 3

2.51	2.68	3.20	3.91	4.67	5.39	5.99	6.42	6.65	6.67	6.50	6.16	5.68	5.11
4.51	3.92	3.40	3.02	2.83	2.86	3.17							

COLUMN NUMBER 4

2.68	2.66	3.04	3.67	4.39	5.12	5.76	6.26	6.59	6.73	6.68	6.45	6.09	5.61
5.06	4.48	3.93	3.44	3.06	2.82	2.74							

COLUMN NUMBER 5

2.87	2.67	2.92	3.46	4.14	4.86	5.52	6.08	6.49	6.72	6.78	6.66	6.39	6.00
5.51	4.97	4.40	3.84	3.32	2.85	2.46							

COLUMN NUMBER 6

3.11	2.75	2.88	3.33	3.96	4.66	5.33	5.93	6.39	6.70	6.84	6.82	6.64	6.32
5.90	5.39	4.82	4.22	3.60	2.98	2.34							

COLUMN NUMBER 7

3.42	2.92	2.93	3.30	3.87	4.54	5.21	5.83	6.33	6.69	6.89	6.94	6.84	6.60
6.23	5.76	5.21	4.59	3.91	3.17	2.37							

COLUMN NUMBER 8

3.80	3.17	3.09	3.37	3.89	4.52	5.17	5.78	6.30	6.69	6.94	7.04	7.00	6.81
6.50	6.08	5.56	4.94	4.23	3.43	2.50							

COLUMN NUMBER 9

4.21	3.50	3.33	3.54	3.99	4.57	5.19	5.78	6.30	6.70	6.98	7.11	7.11	6.97
6.71	6.33	5.84	5.25	4.54	3.70	2.72							

COLUMN NUMBER 10

4.63	3.86	3.62	3.77	4.16	4.69	5.26	5.82	6.31	6.71	6.99	7.14	7.16	7.05
6.83	6.49	6.04	5.48	4.80	3.97	2.98							

COLUMN NUMBER 11

5.00	4.20	3.93	4.03	4.37	4.84	5.36	5.87	6.32	6.69	6.95	7.09	7.12	7.03
6.84	6.54	6.14	5.62	4.99	4.21	3.25							

COLUMN NUMBER 12

5.27	4.49	4.21	4.28	4.57	4.99	5.45	5.90	6.30	6.62	6.84	6.96	6.98	6.90
6.72	6.46	6.10	5.65	5.08	4.38	3.50							

COLUMN NUMBER 13

5.38	4.67	4.42	4.48	4.74	5.10	5.51	5.89	6.23	6.49	6.66	6.74	6.73	6.64
6.47	6.24	5.93	5.55	5.07	4.47	3.71							

COLUMN NUMBER 14

5.28	4.69	4.51	4.59	4.84	5.17	5.52	5.83	6.09	6.28	6.38	6.41	6.36	6.25
6.08	5.87	5.62	5.31	4.94	4.48	3.88							

COLUMN NUMBER 15

4.93	4.52	4.45	4.59	4.86	5.16	5.46	5.71	5.89	6.00	6.02	5.98	5.88	5.74
5.57	5.38	5.18	4.97	4.72	4.42	4.02							

COLUMN NUMBER 16

4.28	4.13	4.23	4.47	4.78	5.09	5.35	5.53	5.64	5.66	5.60	5.48	5.33	5.14
4.96	4.80	4.65	4.54	4.43	4.32	4.16							

COLUMN NUMBER 17

3.32	3.49	3.82	4.22	4.62	4.95	5.19	5.33	5.37	5.30	5.16	4.96	4.74	4.51
4.32	4.17	4.09	4.08	4.14	4.25	4.36							

COLUMN NUMBER 18

2.03	2.62	3.26	3.87	4.39	4.79	5.04	5.15	5.12	4.98	4.76	4.48	4.19	3.92
3.71	3.59	3.58	3.70	3.94	4.29	4.72							

COLUMN NUMBER 19

999.90	1.55	2.57	3.46	4.17	4.67	4.97	5.07	5.00	4.79	4.49	4.14	3.78	3.47
3.25	3.16	3.24	3.49	3.94	4.56	5.35							

COLUMN NUMBER 20

999.90	999.90	1.82	3.07	4.02	4.68	5.06	5.19	5.10	4.85	4.48	4.07	3.65	3.31
3.09	3.04	3.21	3.62	4.30	5.23	6.41							

COLUMN NUMBER 21

999.90	999.90	1.11	2.79	4.07	4.95	5.46	5.65	5.58	5.31	4.90	4.43	3.98	3.62
3.41	3.42	3.70	4.29	5.22	6.50	8.12							

8 DEGREE TREND SURFACE

```
      INITIAL            FINAL            INCREMENT
U    0.0              0.1000000E 01      0.5000000E-01
V    0.0              0.1000000E 01      0.5000000E-01
COLUMN NUMBER   1
    3.01    2.07    1.32  999.90  999.90    1.11    1.64    2.32    3.02    3.62    4.04    4.27    4.34    4.38
    4.54    4.99    5.75    6.55    6.57    4.06  999.90
COLUMN NUMBER   2
    4.89    4.61    4.34    4.22    4.31    4.59    4.99    5.42    5.79    5.98    5.95    5.67    5.20    4.65
    4.20    4.04    4.23    4.63    4.60    2.71  999.90
COLUMN NUMBER   3
    3.88    4.11    4.25    4.46    4.79    5.23    5.72    6.19    6.54    6.69    6.57    6.17    5.54    4.79
    4.08    3.61    3.50    3.67    3.67    2.32  999.90
COLUMN NUMBER   4
    2.56    3.07    3.45    3.87    4.39    4.99    5.63    6.22    6.68    6.92    6.89    6.56    5.96    5.19
    4.41    3.78    3.48    3.48    3.47    2.52  999.90
COLUMN NUMBER   5
    1.87    2.44    2.88    3.37    3.97    4.67    5.41    6.11    6.70    7.08    7.19    6.99    6.51    5.81
    5.03    4.33    3.88    3.72    3.64    2.96  999.90
COLUMN NUMBER   6
    1.93    2.38    2.73    3.16    3.74    4.45    5.23    6.01    6.69    7.19    7.45    7.40    7.06    6.47
    5.73    4.99    4.42    4.10    3.92    3.37    1.27
COLUMN NUMBER   7
    2.49    2.73    2.90    3.20    3.69    4.34    5.10    5.89    6.63    7.22    7.59    7.68    7.48    7.00
    6.33    5.59    4.93    4.47    4.16    3.66    2.08
COLUMN NUMBER   8
    3.26    3.26    3.23    3.37    3.73    4.29    4.99    5.75    6.48    7.12    7.57    7.77    7.70    7.34
    6.75    6.04    5.33    4.75    4.32    3.83    2.64
COLUMN NUMBER   9
    4.01    3.81    3.60    3.59    3.83    4.28    4.90    5.59    6.29    6.92    7.41    7.69    7.72    7.48
    6.99    6.32    5.60    4.95    4.42    3.94    3.04
COLUMN NUMBER  10
    4.60    4.29    3.97    3.86    4.00    4.36    4.88    5.48    6.11    6.69    7.17    7.49    7.59    7.44
    7.06    6.47    5.78    5.10    4.52    4.03    3.36
COLUMN NUMBER  11
    5.00    4.66    4.32    4.17    4.26    4.55    4.97    5.47    5.99    6.48    6.90    7.21    7.34    7.27
    6.99    6.50    5.87    5.21    4.61    4.13    3.62
COLUMN NUMBER  12
    5.19    4.91    4.62    4.51    4.60    4.86    5.20    5.57    5.95    6.31    6.62    6.87    7.00    6.98
    6.79    6.41    5.89    5.29    4.70    4.22    3.80
COLUMN NUMBER  13
    5.16    4.99    4.82    4.81    4.97    5.22    5.50    5.75    5.97    6.16    6.33    6.46    6.55    6.57
    6.46    6.21    5.81    5.31    4.76    4.26    3.85
COLUMN NUMBER  14
    4.90    4.85    4.83    4.98    5.25    5.55    5.79    5.93    5.99    5.99    5.98    5.98    6.01    6.04
    6.02    5.91    5.67    5.28    4.79    4.25    3.73
COLUMN NUMBER  15
    4.44    4.44    4.57    4.90    5.33    5.72    5.97    6.03    5.95    5.77    5.57    5.44    5.39    5.44
    5.52    5.58    5.52    5.28    4.83    4.20    3.42
COLUMN NUMBER  16
    3.91    3.80    4.01    4.51    5.12    5.65    5.96    6.00    5.81    5.49    5.14    4.89    4.80    4.89
    5.10    5.35    5.50    5.42    5.00    4.18    2.94
COLUMN NUMBER  17
    3.61    3.12    3.25    3.86    4.65    5.36    5.78    5.86    5.64    5.23    4.78    4.46    4.37    4.53
    4.92    5.38    5.76    5.82    5.36    4.19    2.21
COLUMN NUMBER  18
    4.18    2.85    2.62    3.17    4.08    4.96    5.54    5.71    5.51    5.08    4.60    4.26    4.21    4.49
    5.05    5.74    6.31    6.43    5.77    4.01  999.90
COLUMN NUMBER  19
    6.59    3.74    2.67    2.87    3.72    4.67    5.37    5.64    5.49    5.06    4.56    4.22    4.19    4.56
    5.26    6.08    6.71    6.70    5.55    2.75  999.90
COLUMN NUMBER  20
  999.90    6.85    4.19    3.49    3.88    4.65    5.28    5.51    5.31    4.79    4.18    3.72    3.60    3.89
    4.51    5.22    5.57    5.00    2.80  999.90  999.90
COLUMN NUMBER  21
  999.90  999.90    7.98    5.46    4.69    4.70    4.81    4.62    4.01    3.07    2.01    1.08  999.90  999.90
  999.90  999.90  999.90  999.90  999.90  999.90  999.90
```

ID	U	V	Y	PRED	RESID
0	0.057	0.051	2.70	2.67	0.03
1	0.041	0.274	4.00	4.18	-0.18
2	0.085	0.339	4.70	4.57	0.13
3	0.061	0.343	4.70	4.58	0.12
4	0.034	0.385	4.90	4.90	-0.00
5	0.028	0.497	5.40	5.37	0.03
6	0.012	0.923	2.80	2.80	-0.00
7	0.169	0.059	2.90	2.95	-0.05
8	0.125	0.178	3.50	3.43	0.07
9	0.182	0.270	4.40	4.35	0.05
10	0.116	0.375	5.30	5.28	0.02
11	0.159	0.492	6.70	6.92	-0.22
12	0.150	0.553	6.80	6.92	-0.12
13	0.132	0.691	5.80	5.80	-0.00
14	0.171	0.745	5.00	4.91	0.09
15	0.132	0.830	4.10	4.27	-0.17
16	0.136	0.891	3.80	3.73	0.07
17	0.177	0.893	3.30	3.15	0.15
18	0.165	0.949	2.80	2.83	-0.03
19	0.235	0.962	2.20	2.24	-0.04
20	0.278	0.905	2.70	2.77	-0.07
21	0.218	0.640	6.30	6.21	0.09
22	0.241	0.633	6.40	6.27	0.13
23	0.209	0.351	5.80	5.95	-0.15
24	0.293	0.189	4.90	4.75	0.15
25	0.274	0.169	4.50	4.41	0.09
26	0.232	0.089	3.60	3.76	-0.16
27	0.350	0.030	3.60	3.50	0.10
28	0.353	0.144	4.80	4.70	0.10
29	0.339	0.229	5.50	5.65	-0.15
30	0.339	0.431	7.90	7.76	0.14
31	0.394	0.571	6.80	6.70	0.10
32	0.325	0.602	6.70	6.54	0.16
33	0.320	0.649	5.80	5.85	-0.05
34	0.365	0.718	4.70	4.70	-0.00
35	0.385	0.723	4.50	4.61	-0.11
36	0.301	0.745	4.50	4.41	0.09
37	0.378	0.755	4.30	4.22	0.08
38	0.364	0.805	3.50	3.71	-0.21
39	0.331	0.848	3.30	3.31	-0.01
40	0.373	0.899	3.30	3.40	-0.10
41	0.352	0.998	3.30	3.30	0.00
42	0.417	0.965	4.10	4.07	0.03
43	0.419	0.933	4.00	3.90	0.10
44	0.458	0.794	4.20	4.07	0.13
45	0.403	0.691	4.80	5.01	-0.21
46	0.407	0.390	7.70	7.69	0.01
47	0.488	0.388	8.00	7.41	0.59
48	0.446	0.276	6.50	6.79	-0.29
49	0.442	0.275	6.40	6.77	-0.37
50	0.423	0.157	5.20	5.11	0.09
51	0.518	0.033	3.90	4.01	-0.11
52	0.593	0.264	6.40	6.34	0.06
53	0.501	0.473	7.00	7.08	-0.08
54	0.552	0.477	6.80	6.73	0.07
55	0.523	0.552	6.00	6.38	-0.38
56	0.536	0.678	5.10	5.30	-0.20
57	0.553	0.786	4.90	4.68	0.22
58	0.512	0.971	5.00	4.90	0.10
59	0.651	0.993	4.70	4.90	-0.20
60	0.674	0.937	4.60	4.69	-0.09
61	0.660	0.926	4.70	4.78	-0.08
62	0.666	0.758	5.70	5.58	0.12
63	0.690	0.731	5.90	5.81	0.09
64	0.641	0.726	5.80	5.62	0.18
65	0.605	0.433	6.50	6.45	0.05
66	0.614	0.282	6.50	6.29	0.21
67	0.623	0.160	5.40	5.40	0.00
68	0.696	0.111	4.90	4.94	-0.04
69	0.627	0.081	4.60	4.58	0.02
70	0.694	0.036	4.10	4.01	0.09
71	0.719	0.084	4.60	4.68	-0.08
72	0.736	0.215	5.60	5.47	0.13
73	0.784	0.287	5.10	5.04	0.06
74	0.723	0.311	5.00	5.28	-0.28
75	0.716	0.421	5.10	5.21	-0.11
76	0.794	0.489	4.80	4.74	0.06
77	0.746	0.852	4.20	4.53	-0.33
78	0.757	0.902	3.90	3.90	0.00
79	0.763	0.976	4.10	3.67	0.43
80	0.830	0.990	3.40	3.54	-0.14
81	0.806	0.949	3.00	3.05	-0.05
82	0.842	0.921	2.70	2.64	0.06
83	0.853	0.835	3.50	3.39	0.11
84	0.897	0.771	4.30	4.29	0.01
85	0.876	0.663	5.20	5.65	-0.45
86	0.865	0.578	6.10	5.33	0.77
87	0.806	0.512	4.40	4.85	-0.45
88	0.826	0.461	4.50	4.38	0.12
89	0.833	0.052	4.20	4.21	-0.01
90	0.977	0.213	3.40	3.39	0.01
91	0.959	0.406	3.30	3.31	-0.01
92	0.914	0.457	4.00	4.20	-0.20
93	0.906	0.531	5.00	4.86	0.14
94	0.961	0.536	4.60	4.42	0.18
95	0.936	0.541	4.90	4.84	0.06
96	0.955	0.612	4.90	5.34	-0.44
97	0.925	0.642	5.60	5.61	-0.01
98	0.964	0.764	4.80	4.49	0.31
99	0.951	0.831	3.40	3.57	-0.17

RESID SS 0.3462986E 01 DF 55 MS 0.6296337E-01 SD 0.2509250E 00
97.730 PERCENT SS ACCOUNTED FOR

REFERENCES

Batcha, J.P., and Reese, J.R., 1964, Surface determination and automatic contouring for mineral exploration, extraction, and processing: Colorado Sch. Mines Quart., v. 59, no. 4, p. 1-14.

Krumbein, W.C., 1959, Trend surface analysis of contour-type maps with irregular control-point spacing: Jour. Geophys. Res., v. 64, no. 7, p. 823-834.

McIntyre, D.B., 1963, Program for computation of trend surfaces and residuals of degree 1 through 8: Dept. Geology, Pomona College, Tech. Rept. 4, 24 p.

McIntyre, D.B., 1967, Trend-surface analysis of noisy data: Kansas Geol. Survey Computer Contr. 12, p. 45-56.

Ojakangas, D.R., and Basham, W.L., 1964, Simplified computer contouring of exploration data, in Computers in the mineral industries: Stanford Univ. Publ., Geol. Sci., v. 9, no. 2, p. 757-770.

Surkan, A.J., Denny, J.R., and Batcha, J.P., 1964, Computer contouring: a new tool for evaluation and analysis of mines: Eng. and Min. Jour., v. 165, p. 72-76.

Note the following IBM updated manuals also are useful: Grid value determination, 1620 SPS, 1620-MP-08X; Automatic grid contouring, 1620 SPS, 1620-MP-09X; Contouring by triangulation, 1620 SPS, 1620-MP-21X; and Numerical surface techniques and contour map plotting, E 20-0117.

```
// EXEC FORTRAN
C
C     TREND  1ST OF 6 PHASES.  REQUIRES FORMAT, CHAIN, + FILE.
C
      COMMON NT1,NT2,NT3,KT,MAXD,MAXC,NS,NY,KM,KC,KSV,KIM,NSC,YMIN,YMAX
      COMMON TITLE(18),KTR(8),UINL(8),UFNL(8),UINC(8),VINL(8),VFNL(8)
      COMMON VINC(8),FMT2(18),FMT3(18),FMT4(18)
C     DUMMY IS NEEDED FOR DOUBLE PRECISION ALIGNMENT
      COMMON DUMMY
      DOUBLE PRECISION D
      DOUBLE PRECISION U,V,      Q(16),QQ(16),QR(15,15),UP(16)
      DOUBLE PRECISION VP(16),YRS(25)
      COMMON Q,QQ,QR
      DIMENSION REMRK1(18),REMRK2(18),FMT(18)
      DIMENSION IPHASE(2),YSING(25)
100   FORMAT(16I4)
101   FORMAT(18A4)
102   FORMAT(I2,6F6.4)
103   FORMAT('1',24X,18A4/'0',24X,18A4/' ',24X,18A4/'0')
104   FORMAT('0INPUT FORMAT IS ',18A4)
108   FORMAT('0','CONTROL NUMBERS',16I4)
109   FORMAT(' ',6D19.10)
110   FORMAT('0','PERMITTED RANGE',F10.4,' TO',F10.4)
200   FORMAT('-',T56,'SUMMATIONS')
201   FORMAT('0')
C
C     800 FORMAT WILL BE REPLACED AT EXECUTION BY A MORE REASONABLE
C     FORMAT FOR READING THE INPUT
C
800   FORMAT(///////////////////////////////////////////////////////////////
     *///////////////////////////////////////////////////////////////////)
C
      D=0.D0
C
C     IF KT IS POSITIVE IT DEFINES A DEVICE ON WHICH OUTPUT IS
C     WRITTEN IN TREND6.   THE FILE CAN LATER BE USED FOR INPUT TO
C     PLOTTING ROUTINES.
C     NT1, NT2, AND NT3 ARE WORK FILES.  IN TREND4 IT IS ASSUMED THAT
C        NT2 AND NT3 ARE TAPES WHEN THEY ARE ADDRESSED BY PHYSICAL IOCS
C
C     NS IS NUMBER OF STATIONS.
C     NY IS NUMBER OF READINGS/STATION.  NTR IS TRANSFORMATION CONTROL.
C     NSC IS SCALE CONTROL PRIOR TO OUTPUT.  MAXD IS HIGHEST DEGREE.
C     IT IS NUMBER OF VALUES PER LINE OF OUTPUT OF TREND SURFACE.
C
C     IF THE FOLLOWING CONTROLS ARE ZERO SUPPRESS PRINTING -
C     KC FOR COEFFICIENTS.
C     KS FOR SUMMATIONS.  KM FOR UNSCALED MATRIX.
C     KSM FOR SCALED MATRIX (PRINT OUT NO LONGER OPERATIONAL).
C     KSV FOR SCALING VECTOR.  KIM FOR INVERTED MATRICES.
C
C     TITLE, REMRK1, AND REMRK2 ARE DESCRIPTIVE STATEMENTS.
C     FMT IS FORMAT FOR STATION DATA,  ID,U,V,YRS(I),I=1,NY
C     FMT2 IS OUTPUT FOR TREND SURFACE,  FMT3 FOR RESIDUALS.
C     FMT4 IS HEADINGS FOR RESIDUALS.
C
1     READ (1,100) KT,NT1,NT2,NT3,NS,NY,NTR,NSC,MAXD,IT,KC,KS,KM,KSM,
     $   KSV,KIM
C
C     ****************************************
```

```
C
C        NOTE DSR 4 IS A DISC FILE TO BE USED FOR COMMUNICATION.
C        UNDER CURRENT DOS, DSR1 IS CARD READER, AND 3 IS PRINTER.
C
C     ******************************************
C
      READ(1,101) TITLE,REMRK1,REMRK2
      READ (1,101)    FMT,FMT2,FMT3,FMT4
      WRITE (3,103) TITLE,REMRK1,REMRK2
      WRITE (3,108) KT,NT1,NT2,NT3,NS,NY,NTR,NSC,MAXD,IT,KC,KS,KM,
     $     KSM,KSV,KIM
      WRITE (3,104)    FMT
      CALL FORMAT(FMT)
      WRITE (3,800)
C     THE ABOVE WRITE IS A DUMMY AND WILL NOT BE EXECUTED
C     TWO WRONGS DO NOT MAKE A WRITE
C
C     CONTROL CARDS FOR TREND SURFACES AND RESIDUALS ARE INPUT IN
C     SEQUENCE BEGINNING WITH 1ST DEGREE AND CONTINUING TO HIGHEST
C     DEGREE REQUIRED.
C     IF KTR IS NEGATIVE, COMPUTE TREND ONLY.
C     IF KTR IS ZERO, COMPUTE BOTH TREND AND RESIDUAL.
C     IF KTR IS 1, COMPUTE RESIDUALS ONLY.  IF KTR EXCEEDS 1, NEITHER
C       TRENDS OR RESIDUALS WILL BE COMPUTED.
C     UINL IS INITIAL VALUE OF U, UFNL IS FINAL VALUE, AND UINC IS
C       INCREMENT OF U FOR TREND SURFACE.  SIMILARLY FOR V.
C
C     YMIN AND YMAX ARE THE LIMITS TO BE USED IN PLOTTING TREND SURFACE
C
      READ (1,102) ((KTR(I),UINL(I),UFNL(I),UINC(I),VINL(I),VFNL(I),
     $       VINC(I)),I=1,MAXD)
      READ (1,800) DUMMY,DUMMY,DUMMY,YMIN
      READ (1,800) DUMMY,DUMMY,DUMMY,YMAX
      WRITE (3,110) YMIN,YMAX
C
C     MAXC IS NUMBER OF COEFFICIENTS IN POLYNOMIAL OF DEGREE MAXD
C
      MAXC=1+((3+MAXD)*MAXD)/2
      DO 23 I=1,16
      Q(I)=D
      QQ(I)=D
23    CONTINUE
      DO 29 J=1,15
      DO 29 I=1,15
29    QR(I,J)=D
      IMAX=MAXD+MAXD
      JMAX=IMAX-1
      DO 71 ICNT=1,NS
C
C     READ STATION DATA,  EFFECT TRANSFORMATIONS, AND STORE.
C     PREPARE THE NECESSARY U-V SUMMATIONS.
C     Q GIVES THE POWERS OF U, QQ THE POWERS OF V, AND QR THE CROSS PROD
C
      READ (1,800) ID,U,V,(YRS(I),I=1,NY)
      IF(KT) 35,35,31
C
C     USING AND VSING ARE SINGLE PRECISION EQUIVALENTS OF U AND V.
C
 31   USING=U
      VSING=V
      DO 30 I=1,NY
```

```
 30    YSING(I)=YRS(I)
C
C      DSR 4 IS A DISC FILE.  STORE DATA AND PICK UP IN TREND6 FOR KT
C      NTR CONTROLS THE TRANSFORMATIONS.
C
       WRITE (4 ) ID,USING,VSING,(YSING(I),I=1,NY)
35     IF(NTR) 34,34,311
311    IF(NTR-1) 34,32,33
32     DO 320 I=1,NY
320    YRS(I)=ALOG10(YRS(I))
       GO TO 34
33     DO 321 I=1,NY
321    YRS(I)=SQRT(YRS(I))
34     WRITE (NT1) ID,U,V,(YRS(I),I=1,NY)
       UP(1)=U
       VP(1)=V
       DO 39 I=2,IMAX
       UP(I)=UP(I-1)*U
39     VP(I)=VP(I-1)*V
       DO 42 I=1,IMAX
       Q(I)=Q(I)+UP(I)
42     QQ(I)=QQ(I)+VP(I)
       DO 46 J=1,JMAX
       II=IMAX-J
       DO 46 I=1,II
46     QR(I,J)=QR(I,J)+UP(I)*VP(J)
71     CONTINUE
       IF(KS) 48,49,48
48     WRITE (3,200)
       WRITE (3,109) (Q(I),I=1,IMAX)
       WRITE (3,201)
       WRITE (3,109) (QQ(I),I=1,IMAX)
       WRITE (3,201)
       DO 65 J=1,JMAX
       II=IMAX-J
65     WRITE (3,109) (QR(I,J),I=1,II)
49     IF(KT) 89,89,50
 50    REWIND 4
 89    CALL FILE(NT1,0)
       REWIND NT1
C
C      CALL IN TREND2
C
       IPHASE(1)=-472267307
       IPHASE(2)=-990756800
       CALL CHAIN (IPHASE)
       END
```

```
// EXEC FORTRAN
C                                                                       TRN2   0
C     TREND2   REQUIRES PLACE1, PLACE2, PLACE3, + CHAIN.                TRN2   1
C                                                                       TRN2   2
      COMMON NT1,NT2,NT3,KT,MAXD,MAXC,NS,NY,KM,KC,KSV,KIM,NSC,YMIN,YMAX TRN2   3
      COMMON TITLE(18),KTR(8),UINL(8),UFNL(8),UINC(8),VINL(8),VFNL(8)   TRN2   4
      COMMON VINC(8),FMT2(18),FMT3(18),FMT4(18)                         TRN2   5
C     DUMMY IS NEEDED FOR DOUBLE PRECISION ALIGNMENT                    TRN2   6
      COMMON DUMMY                                                      TRN2   7
      DOUBLE PRECISION Q(16),QQ(16),QR(15,15)                           TRN2   8
      COMMON Q,QQ,QR                                                    TRN2   9
      DIMENSION IPHASE(2)                                               TRN2  10
      DOUBLE PRECISION X(45,15)                                         TRN2  11
C                                                                       TRN2  12
C     SUBROUTINES PLACE1, PLACE2, AND PLACE3 BUILD THE 3 15*45 STRIPS   TRN2  13
C     THAT CONSTITUTE THE MATRIX OF NORMAL EQUATIONS.                   TRN2  14
C                                                                       TRN2  15
      X(1,1)=NS                                                         TRN2  16
      CALL PLACE1(X,Q,QQ,QR)                                            TRN2  17
      WRITE (NT2) X                                                     TRN2  18
      CALL PLACE2(X,Q,QQ,QR)                                            TRN2  19
      WRITE (NT2) X                                                     TRN2  20
      CALL PLACE3(X,Q,QQ,QR)                                            TRN2  21
      WRITE (NT2) X                                                     TRN2  22
      REWIND NT2                                                        TRN2  23
C                                                                       TRN2  24
C     CALL IN TREND3                                                    TRN2  25
C                                                                       TRN2  26
      IPHASE(1)=-472267307                                              TRN2  27
      IPHASE(2)=-990691264                                              TRN2  28
      CALL CHAIN (IPHASE)                                               TRN2  29
      END                                                               TRN2  30
```

```
// EXEC FORTRAN
C
C     THIS CREATES THE ASSEMBLY LANGUAGE PROGRAMS PLACE1, PLACE2 PLACE3
C     THAT ARE NEEDED TO BUILD THE MATRIX OF NORMAL EQUATIONS.
C     IT IS WRITTEN IN SUCH A WAY THAT OTHER MATRICES, SUCH AS U V W,
C     CAN BE CONSTRUCTED BY MODIFYING THE DATA.   IN THIS CASE DOUBLE
C     PRECISION IS NECESSARY, BUT THIS IS AN OPTION CONTROLLED BY DATA.
C
C     NPREC IS 1 FOR SINGLE PRECISION, TWO FOR DOUBLE PRECISION
C     N IS THE SIZE OF THE MATRIX BEING CONSTRUCTED
C     M IS THE SIZE OF THE MATRIX WHICH WILL CONTAIN THE NXN MATRIX AS
C     A SUBMATRIX.   M .GT. .OR. .EQ. N
C     U,V,W ARRAYS CAN BE ANY SIZE
C     UV ARRAY DIMENSIONED (NUV,XXX)
C     UW ARRAY DIMENSIONED (NUW,XXX)
C     VW ARRAY DIMENSIONED (NVW,XXX)
C     UVW ARRAY DIMENSIONED (N1UVW,N2UVW,XXX)
C     FUNCTION KTRAIL(J) HAS AS ITS RESULT AN A4 REPRESENTATION OF J
C     INCLUDING LEADING ZEROES
C
      DIMENSION TITLE(10)
      DIMENSION MATRIX(2100)
100   FORMAT(8I3)
102   FORMAT(10A4)
108   FORMAT(' ERROR IN U,V,W ',3I13)
400   FORMAT('PLAC  TITLE ''',10A4,I2,'''',24X)
401   FORMAT('PLACE',I1,' START 0',66X/
     1  ' USING *,15',69X/
     4  ' STM 14,12,12(13)     SAVE ALL REGISTERS',41X/
     5  ' ST 13,RSAVE',68X/
     X  ' STD 0,FSAVE',68X/
     6  ' BC 15,NEXT',69X/
     X  'FSAVE DS D',70X/
     7  'RSAVE DC F''1''',67X/
     8  'NEXT DS OF',70X)
404   FORMAT(' L 10,12(0,1)  UV',63X/
     2  ' L 14,0(0,1)  NEW MATRIX',56X)
405   FORMAT(' L 9,12(0,1)  W',65X/
     1  ' L 10,16(0,1)  UV',63X/
     2  ' L 11,20(0,1)  UW',63X/
     4  ' L 12,24(0,1)  VW',63X/
     5  ' L 13,28(0,1)  UVW',62X/
     6  ' L 14,32(0,1)  NEW MATRIX',55X)
407   FORMAT(' L 7,4(0,1)  U',66X/
     1  ' L 8,8(0,1)  V',66X/
     4  ' LA 1,4095(0,0)',65X/
     5  ' LA 1,1(0,1)  4096',62X/
     6  ' LA 2,0(1,1)  8192',62X/
     7  ' LA 3,0(2,1)  3*4096',60X/
     8  ' LA 4,0(3,1)  4*4096',60X/
     9  ' LA 5,0(4,1)  5*4096',60X/
     1  ' LA 6,0(5,1)  6*4096',60X)
600   FORMAT(' ',A3,' 0,',A4,'(',I1,',',A4,')',61X)
602   FORMAT(' L 13,RSAVE',69X/
     X  ' LD 0,FSAVE',69X/
     1  ' LM 14,12,12(13)',64X/
     2  ' MVI 12(13),X''FF''',63X/
     3  ' BCR 15,14',70X/
     4  ' END PLACE',I1,69X)
C
```

```
C      NOUT IS THE DATA SET REFERENCE NUMBER OF THE OUTPUT FILE
C
       NOUT=7
       REWIND NOUT
C
       READ (1,102) TITLE
       READ (1,100) NPREC,N,M,NUV,NUW,NVW,N1UVW,N2UVW
       MATRIX(1)=0
       NSTARI=N+1
       DO 2 I=2,N
C
C      INPUT IS THE TOP ROW OF THE MATRIX
C      IGNORING THE CONSTANT TERM, WHICH IS AUTOMATICALLY INCLUDED.
C      CONSTRUCTING A 10X10 MATRIX REQUIRES INPUT OF 9 MATRIX ELEMENTS
C      FOR EACH MATRIX ELEMENT OF THE TOP ROW--
C      INPUT IS IN THE FORM        IU, IV, IW       (3I3)
C      IU IS THE POWER OF U
C      IV IS THE POWER OF V
C      IW IS THE POWER OF W
C      MAXIMUM ALLOWED IW IS 21, FOR WHICH IV MUST BE LESS THAN 48.
C      IF NO POWER OF W DIFFERENT FROM 0 IS INPUT FOR ANY ELEMENT,
C      THE GENERATED PROGRAM WILL NOT REFER TO ANY ARRAY INVOLVING W
C
       READ (1,100) IU,IV,IW
       MN=10000*(10000*IW+100*IV+IU)
       MATRIX(I)=MN+(I-1)
       MATRIX(NSTARI)=MN+M*(I-1)
2      NSTARI=NSTARI+N
       NSTARI=N+1
C
C      ONLY CONSTRUCT THE TOP HALF OF THE MATRIX SINCE IT IS SYMMETRIC
C
       DO 3 J=2,N
       DO 4 I=2,N
       IJ=I+NSTARI-1
       IF(I-J) 8,8,9
9      MATRIX(IJ)=0
       GO TO 4
8      MATRIX(IJ)=MATRIX(I)+MATRIX(NSTARI)
4      CONTINUE
3      NSTARI=NSTARI+N
       DO 299 I=2,N
299    MATRIX(I)=0
C
C      MATRIX IS SET UP,NOW DO THE SORT
C
       NSQ=N*N
       CALL ISORTA(MATRIX,NSQ)
       IF (NPREC-1) 300,301,300
301    LOAD=-742047680
       ISTORE=-488389312
       GO TO 302
300    LOAD=-742113216
       ISTORE=-488389568
302    NBYTES=4*NPREC
       ITOP=0
       K14=KTRAIL(14)
       K13=KTRAIL(13)
       K12=KTRAIL(12)
       K11=KTRAIL(11)
       K10=KTRAIL(10)
```

```
      K9=KTRAIL(9)
      K8=KTRAIL(8)
      K7=KTRAIL(7)
C
C     NOW FOR THE MAIN PART OF THE PROGRAM
C     REGISTERS USED BY PLACE WILL BE--
C     R0    0
C     R1    4096
C     R2    8192
C     R3    12288
C     R4    16384
C     R5    20480
C     R6    24576
C     R7    U
C     R8    V
C     R9    W
C     R10   UV
C     R11   UW
C     R12   VW
C     R13   UVW
C     R14   ADDRESS OF CONSTRUCTED MATRIX
C     R15   BASE REGISTER
      MISST=5400
      DO 503 MISS=1,3
C
C     CREATE LEADING MATERIAL FOR ASSEMBLY LANGUAGE PROGRAM
C
      WRITE (NOUT,400) TITLE,MISS
      WRITE (NOUT,401) MISS
      IF(MATRIX(NSQ)-100000000) 402,402,403
402   WRITE (NOUT,404)
      GO TO 406
403   WRITE (NOUT,405)
406   WRITE (NOUT,407)
      DO 500 KOUNT=2,NSQ
      KOW=MATRIX(KOUNT)
      IF(KOW) 500,500,701
701   KTOP=KOW/10000
      KLAST=(KOW-KTOP*10000)*NBYTES
      IF(KLAST-MISST) 700,800,800
800   MATRIX(KOUNT)=KOW-5400/8
      GO TO 500
C
C     ONLY ONE THIRD OF THE MATRIX IS TO BE OUTPUT THIS TIME
C
700   MATRIX(KOUNT)=0
      IF(KTOP-ITOP) 502,501,502
501   INSTR=ISTORE
      KBASE=K14
530   KREG=KLAST/4096
      KDIS=KLAST-KREG*4096
      KDIS=KTRAIL(KDIS)
      WRITE (NOUT,600) INSTR,KDIS,KREG,KBASE
      KLAST=KSAVE
      IF(KBASE-K14) 501,500,501
502   IW=KTOP/10000
      KSAVE=KLAST
      INSTR=LOAD
      ITOP=KTOP
      IVV=KTOP-IW*10000
      IV=IVV/100
```

```
      IU=IVV-IV*100                                                     PLCE 181
      IF(IW) 99,517,510                                                 PLCE 182
99    WRITE (3,108) IU,IV,IW                                            PLCE 183
      CALL EXIT                                                         PLCE 184
517   IF(IV) 99,518,520                                                 PLCE 185
520   IF(IU) 99,519,521                                                 PLCE 186
510   IF(IV) 99,512,511                                                 PLCE 187
512   IF(IU) 99,514,513                                                 PLCE 188
511   IF(IU) 99,515,516                                                 PLCE 189
518   KBASE=K7                                                          PLCE 190
      KLAST=NBYTES*(IU-1)                                               PLCE 191
      GO TO 530                                                         PLCE 192
519   KBASE=K8                                                          PLCE 193
      KLAST=NBYTES*(IV-1)                                               PLCE 194
      GO TO 530                                                         PLCE 195
514   KBASE=K9                                                          PLCE 196
      KLAST=NBYTES*(IW-1)                                               PLCE 197
      GO TO 530                                                         PLCE 198
521   KBASE=K10                                                         PLCE 199
      KLAST=NBYTES*(NUV*(IV-1)+IU-1)                                    PLCE 200
      GO TO 530                                                         PLCE 201
513   KBASE=K11                                                         PLCE 202
      KLAST=NBYTES*(NUW*(IW-1)+IU-1)                                    PLCE 203
      GO TO 530                                                         PLCE 204
515   KBASE=K12                                                         PLCE 205
      KLAST=NBYTES*(NVW*(IW-1)+IV-1)                                    PLCE 206
      GO TO 530                                                         PLCE 207
516   KBASE=K13                                                         PLCE 208
      KLAST=NBYTES*(N1UVW*(N2UVW*(IW-1)+IV-1)+IU-1)                     PLCE 209
      GO TO 530                                                         PLCE 210
500   CONTINUE                                                          PLCE 211
      WRITE (NOUT,602) MISS                                             PLCE 212
C                                                                       PLCE 213
C     SOME VERSIONS OF THE DOS SYSTEM REQUIRE TAPE MARKS BETWEEN EACH   PLCE 214
C     ASSEMBLY SOURCE PROGRAM.   A CALL TO FILE WITH ARGUMENT ZERO      PLCE 215
C     ACHIEVES THIS WITHOUT REWINDING THE TAPE.   SOME VERSIONS OF       PLCE 216
C     DOS FORTRAN REWIND WHEN THE 'END FILE' STATEMENT IS EXECUTED.     PLCE 217
C                                                                       PLCE 218
      CALL FILE(NOUT,0)                                                 PLCE 219
503   CONTINUE                                                          PLCE 220
      REWIND NOUT                                                       PLCE 221
      CALL EXIT                                                         PLCE 222
      END                                                               PLCE 223


// EXEC ASSEMBLY
         PRINT   NOGEN                                                  KTRL   0
KTRAIL   START   0                                                      KTRL   1
         USING   KTRAIL,15                                              KTRL   2
         STM     14,3,12(13)                                            KTRL   3
         ST      13,SAVE13                                              KTRL   4
         LR      3,15                                                   KTRL   5
         USING   KTRAIL,3                                               KTRL   6
         DROP    15                                                     KTRL   7
         L       1,0(0,1)                                               KTRL   8
         L       1,0(0,1)                                               KTRL   9
         CVD     1,TEMPD                                                KTRL  10
         UNPK    TEMPF,TEMPD                                            KTRL  11
         OI      TEMPF+3,X'F0'                                          KTRL  12
         L       0,TEMPF                                                KTRL  13
*        DO NOT DESTROY THE CONTENTS OF GPRO BY LM                      KTRL  14
```

```
         L        13,SAVE13                                              KTRL  15
         LM       14,15,12(13)                                           KTRL  16
         LM       1,3,24(13)                                             KTRL  17
         MVI      12(13),X'FF'                                           KTRL  18
         BR       14                                                     KTRL  19
TEMPD    DS       D                                                      KTRL  20
TEMPF    DS       F                                                      KTRL  21
SAVE13   DS       F                                                      KTRL  22
         END      KTRAIL                                                 KTRL  23
```

```
// EXEC
GENERATE 8TH DEGREE PLACE 1,2,3 PROGRAMS FOR U AND V.                    8THD   0
  2 45 45 15                                                             8THD   1
  1                                                                      8THD   2
      1                                                                  8THD   3
  2                                                                      8THD   4
  1   1                                                                  8THD   5
      2                                                                  8THD   6
  3                                                                      8THD   7
  2   1                                                                  8THD   8
  1   2                                                                  8THD   9
      3                                                                  8THD  10
  4                                                                      8THD  11
  3   1                                                                  8THD  12
  2   2                                                                  8THD  13
  1   3                                                                  8THD  14
      4                                                                  8THD  15
  5                                                                      8THD  16
  4   1                                                                  8THD  17
  3   2                                                                  8THD  18
  2   3                                                                  8THD  19
  1   4                                                                  8THD  20
      5                                                                  8THD  21
  6                                                                      8THD  22
  5   1                                                                  8THD  23
  4   2                                                                  8THD  24
  3   3                                                                  8THD  25
  2   4                                                                  8THD  26
  1   5                                                                  8THD  27
      6                                                                  8THD  28
  7                                                                      8THD  29
  6   1                                                                  8THD  30
  5   2                                                                  8THD  31
  4   3                                                                  8THD  32
  3   4                                                                  8THD  33
  2   5                                                                  8THD  34
  1   6                                                                  8THD  35
      7                                                                  8THD  36
  8                                                                      8THD  37
  7   1                                                                  8THD  38
  6   2                                                                  8THD  39
  5   3                                                                  8THD  40
  4   4                                                                  8THD  41
  3   5                                                                  8THD  42
  2   6                                                                  8THD  43
  1   7                                                                  8THD  44
      8                                                                  8THD  45
```

```
// EXEC FORTRAN
C
C     TREND3  REQUIRES CHAIN
C
      COMMON NT1,NT2,NT3,KT,MAXD,MAXC,NS,NY,KM,KC,KSV,KIM,NSC,YMIN,YMAX
      COMMON TITLE(18),KTR(8),UINL(8),UFNL(8),UINC(8),VINL(8),VFNL(8)
      COMMON VINC(8),FMT2(18),FMT3(18),FMT4(18)
C     DUMMY IS NEEDED FOR DOUBLE PRECISION ALIGNMENT
      COMMON DUMMY
      DOUBLE PRECISION X(45,15),X1(45,15),X2(45,15)
      COMMON X,X1,X2
      DIMENSION IPHASE(2)
105   FORMAT('0',T56,'MATRIX')
109   FORMAT(' ',6D19.10)
C
C     THE MATRIX EXISTS IN 3 PARTS ON DSR NT2.  READ IT.
C
      READ (NT2) X
      READ (NT2) X1
      READ (NT2) X2
      DO 1 I=1,MAXC
      DO 1 J=1,I
1     X(I,J)=X(J,I)
      IF(KM) 1100,1101,1100
1100   WRITE (3,105)
      WRITE (3,109) ((X(I,J),I=1,MAXC),J=1,MAXC)
C
C     LEAVE MATRIX IN CORE FOR INVERSION
C
C
C     CALL TREND4
C
1101  IPHASE(1)=-472267307
      IPHASE(2)=-990625728
      CALL CHAIN(IPHASE)
      END
```

```
// EXEC ASSEMBLY
*
*         TREND4  REQUIRES DINVRT
*
*                SCALES MATRIX AND WRITES SCALING VECTOR ON NT3
*                THIS PHASE INVERTS ALL MATRICES
*
*                IT WRITES ON TAPE ONE RECORD OF 360 BYTES WHICH IS THE
*                SCALING VECTOR    8BYTES*45COEFFICIENTS
*                THEN IT WRITES 3*MAXD RECORDS OF 5400 BYTES
*                IT USES PIOCS
*
                 PRINT NOGEN
TREND4           START    0
                 BALR     12,0
                 USING *,12
                 L        11,=A(NT1)     FOR ADDRESSING COMMON
                 USING    NT1,11
                 LA       13,SAVEAREA
*
*                GET LOGICAL DEVICE NUMBERS FROM DSR NUMBRS
*
                 LA       9,3
                 L        10,NT2
                 SR       10,9
                 STC      10,SYSNT2
                 L        10,NT3
                 SR       10,9
                 STC      10,SYSNT3
                 LA       7,2700
                 LA       7,0(7,7)     5400 IN GPR7
*
*                REWIND NT3
*
*
*                WRITE    SCALING VECTOR ON NT3    ALL 360 BYTES
*
                 MVC      DEVICE(1),SYSNT3
                 BAL      8,REWIND
                 LA        8,X
                 ST        8,TAPECCW
                 MVI       TAPECCW,X'01'
                 EXCP      TAPE
                 WAIT      TAPE
*
*                SET COUNT TO 5400
*
                 ST       7,TAPECCW+4     NEW COUNT
*
*                SCALE THE MATRIX
*
SCALE            LA       1,1
                 LNR      1,1
                 A        1,MAXC     MAXC-1
                 SLL      1,3
                 LR       5,1
                 LA       4,45
                 MR       4,4      360*(MAXC-1)
                 LA       3,X     ADDRESS OF MATRIX
                 AR       5,3     X+360*(MAXC-1)
```

```
         LA      4,360
         LA      0,8    INCREMENT IS 8 BYTES
         LD      0,=D'1.'
OUTER    LD      2,0(0,3)     TAKE TOP OF COLUMN
         STD     0,0(0,3)      SET IT TO 1
         LR      2,0    START WITH SECOND IN COLUMN
INNER    LD      4,0(2,3)     TAKE WORD X(I,J)
         DDR     4,2          SCALE IT
         STD     4,0(2,3)      RE-STORE IT
         BXLE    2,0,INNER
         BXLE    3,4,OUTER
*
*        WRITE SCALED MATRIX ON NT2
*
         MVC     DEVICE(1),SYSNT2
         BAL     8,REWIND
         BAL     8,WRITEX
*        INVERT SUCCESSIVE SUBMATRICES
*
         SR       2,2    DEGREE OF TREND SURFACE
         LA       3,1    SIZE OF SUBMATRIX
NEXTDEG  LA       2,1(0,2)   INCREMENT THE DEGREE.
         LA       3,1(3,2)    INCREMENT SUBMATRIX SIZE
         ST       3,SUBSIZE
*
*        USE DINVRT ROUTINE FOR DOUBLE PRECISION MATRIX INVERSION
*
         CALL    DINVRT,(X,SUBSIZE,TOTSIZE,ST1,ST2,DET)
*        WRITE ON NT3
         MVC     DEVICE(1),SYSNT3
         BAL     8,WRITEX
         C       2,MAXD    IS IT LAST SURFACE
         BE      DONE
*
*        READ IN MATRIX
*
         MVC     DEVICE(1),SYSNT2
         BAL     8,REWIND
         BAL     8,READX
         B       NEXTDEG
*
*        FINISHED
*        REWIND NT2 AND NT3 AT END OF RUN
*
DONE     MVC     DEVICE(1),SYSNT3
         BAL     8,REWIND
         MVC     DEVICE(1),SYSNT2
         BAL     8,REWIND
         FETCH   TREND5
REWIND   MVI     TAPECCW,X'07'  REWIND CODE
         EXCP    TAPE
         WAIT    TAPE
         BR      8
WRITEX   MVI     RORW,X'01'
         B       GO
READX    MVI     RORW,X'02'
GO       LA      5,3
         LA      6,X
LOOP     ST      6,TAPECCW
         MVC     TAPECCW(1),RORW
```

```
                    EXCP      TAPE
                    WAIT      TAPE
                    LA        6,0(7,6)    NEXT BLOCK
                    BCT       5,LOOP
                    BR        8
DET                 DS        D
TAPECCW             CCW       0,0,0,360 BYTES
TAPE                CCB       SYS001,TAPECCW  THIS TOO WILL CHANGE
RORW                DS        C
SYSNT2              DS        C
SYSNT3              DS        C
SUBSIZE             DS        F
TOTSIZE             DC        F'45'
ST1                 DS        45F
ST2                 DS        45F
SAVEAREA            DS        9D
DEVICE              EQU       TAPE+7
                    LTORG
*                   TREND4 COMMON   PICKED UP FROM TREND3
                    COM
NT1                 DS        F
NT2                 DS        F
NT3                 DS        F
KT                  DS        F
MAXD                DS        F
MAXC                DS        F
NS                  DS        F
NY                  DS        F
KM                   DS       F
KC                  DS        F
KSV                  DS       F
KIM                  DS       F
NSC                  DS       F
YMIN      DS    F
YMAX      DS    F
TITLE                DS       18F
KTR       DS    8F
UINL      DS    8F
UFNL      DS    8F
UINC      DS    8F
VINL      DS    8F
VFNL      DS    8F
VINC      DS    8F
FMT2      DS    18F
FMT3      DS    18F
FMT4      DS    18F
DUMMY               DS        F
*                   DUMMY IS NEEDED FOR ALIGNMENT.
X                   DS        2025D
                    END       TREND4
```

```
// EXEC FORTRAN
C
C       TREND5  REQUIRES TREAD AND CHAIN
C
        COMMON NT1,NT2,NT3,KT,MAXD,MAXC,NS,NY,KM,KC,KSV,KIM,NSC,YMIN,YMAX
        COMMON TITLE(18),KTR(8),UINL(8),UFNL(8),UINC(8),VINL(8),VFNL(8)
        COMMON VINC(8),FMT2(18),FMT3(18),FMT4(18)
        COMMON DUMMY
C
C       DUMMY IS NEEDED FOR DOUBLE PRECISION ALIGNMENT
C       DO IS 'D ZERO'
C
        DOUBLE PRECISION DO
        DOUBLE PRECISION U,V,YR,UP(8),VP(8),SOLN(45),C(45),YRS(25)
        DOUBLE PRECISION X(45,15)
        DOUBLE PRECISION D(45)
        DIMENSION IPHASE(2)
102     FORMAT('OSCALING VECTOR IS')
103     FORMAT('OTRANSPOSE OF INVERTED SCALED MATRIX, DEGREE',I3)
104     FORMAT('OSOLUTION VECTOR, VARIABLE NO.',I3)
105     FORMAT('OSCALED SOLUTION VECTOR, VARIABLE NO',I3)
107     FORMAT('OCOEFFICIENT VECTORS ARE')
108     FORMAT('ODEGREE',I2)
109     FORMAT(/(' ',6D19.10))
C
C       PICK UP SCALING VECTOR OFF NT3, WRITTEN BY PIOCS
C       NOTE THAT NT2 AND NT3 WERE REWOUND IN PHASE 4
C
        DO=0.DO
        CALL TREAD(NT3,D,360)
        IF(KSV) 35,36,35
35      WRITE (3,102)
        WRITE (3,109) (D(I),I=1,MAXC)
36      DO 310 I=1,MAXD
        IF(KIM) 31,32,31
31      WRITE (3,103) I
        ID=(I*(I+3))/2+1
        JD=ID
32      DO 33 J=1,3
        CALL TREAD(NT3,X,5400)
        WRITE (NT2) X
C
C       START AN OUTPUT FILE ON NT2, USING PIOCS
C       NOTE THAT X IS WRITTEN IN 3 BLOCKS
C
        IF(KIM) 34,33,34
34      IF(JD) 33,33,311
311     JQ=15
        IF(JD-JQ) 360,37,37
360     JQ=JD
37      JD=JD-15
        DO 316 JJ=1,JQ
316     WRITE (3,109) (X(II,JJ),II=1,ID)
33      CONTINUE
310     CONTINUE
        REWIND NT2
        REWIND NT3
C
C       PICK UP DATA FROM FIRST PHASE
C       TAPE WAS REWOUND
```

```
C
      DO 61 MY=1,NY
C
C     FOR ALL VARIABLES
C
      DO 63 MI=1,45
   63 SOLN(MI)=DO
      DO 50 MS=1,NS
C
C     READ ALL STATION DATA, SELECT RIGHT Y
C
      READ (NT1) ID,U,V,(YRS(I),I=1,NY)
      YR=YRS(MY)
      UP(1)=U
      VP(1)=V
      DO 1C I=2,MAXD
      UP(I)=UP(I-1)*U
   10 VP(I)=VP(I-1)*V
      SOLN(1)=SOLN(1)+YR
      SOLN(2)=SOLN(2)+YR*U
      SOLN(3)=SOLN(3)+YR*V
      SOLN(4)=SOLN(4)+YR*UP(2)
      SOLN(5)=SOLN(5)+YR*U*V
      SOLN(6)=SOLN(6)+YR*VP(2)
      SOLN(7)=SOLN(7)+YR*UP(3)
      SOLN(8)=SOLN(8)+YR*UP(2)*V
      SOLN(9)=SOLN(9)+YR*U*VP(2)
      SOLN(10)=SOLN(10)+YR*VP(3)
      IF(MAXD-3) 50,50,20
   20 SOLN(11)=SOLN(11)+YR*UP(4)
      SOLN(12)=SOLN(12)+YR*UP(3)*V
      SOLN(13)=SOLN(13)+YR*UP(2)*VP(2)
      SOLN(14)=SOLN(14)+YR*U*VP(3)
      SOLN(15)=SOLN(15)+YR*VP(4)
      IF(MAXD-4) 50,50,25
   25 SOLN(16)=SOLN(16)+YR*UP(5)
      SOLN(17)=SOLN(17)+YR*UP(4)*V
      SOLN(18)=SOLN(18)+YR*UP(3)*VP(2)
      SOLN(19)=SOLN(19)+YR*UP(2)*VP(3)
      SOLN(20)=SOLN(20)+YR*U*VP(4)
      SOLN(21)=SOLN(21)+YR*VP(5)
      IF(MAXD-5) 50,50,30
   30 SOLN(22)=SOLN(22)+YR*UP(6)
      SOLN(23)=SOLN(23)+YR*UP(5)*V
      SOLN(24)=SOLN(24)+YR*UP(4)*VP(2)
      SOLN(25)=SOLN(25)+YR*UP(3)*VP(3)
      SOLN(26)=SOLN(26)+YR*UP(2)*VP(4)
      SOLN(27)=SOLN(27)+YR*U*VP(5)
      SOLN(28)=SOLN(28)+YR*VP(6)
      IF(MAXD-6) 50,50,350
  350 SOLN(29)=SOLN(29)+YR*UP(7)
      SOLN(30)=SOLN(30)+YR*UP(6)*V
      SOLN(31)=SOLN(31)+YR*UP(5)*VP(2)
      SOLN(32)=SOLN(32)+YR*UP(4)*VP(3)
      SOLN(33)=SOLN(33)+YR*UP(3)*VP(4)
      SOLN(34)=SOLN(34)+YR*UP(2)*VP(5)
      SOLN(35)=SOLN(35)+YR*U*VP(6)
      SOLN(36)=SOLN(36)+YR*VP(7)
      IF(MAXD-7) 50,50,40
   40 SOLN(37)=SOLN(37)+YR*UP(8)
```

```
      SOLN(38)=SOLN(38)+YR*UP(7)*V
      SOLN(39)=SOLN(39)+YR*UP(6)*VP(2)
      SOLN(40)=SOLN(40)+YR*UP(5)*VP(3)
      SOLN(41)=SOLN(41)+YR*UP(4)*VP(4)
      SOLN(42)=SOLN(42)+YR*UP(3)*VP(5)
      SOLN(43)=SOLN(43)+YR*UP(2)*VP(6)
      SOLN(44)=SOLN(44)+YR*U*VP(7)
      SOLN(45)=SOLN(45)+YR*VP(8)
50    CONTINUE
      IF(KSV) 60,60,55
55    WRITE (3,104) MY
      WRITE (3,109) (SOLN(I),I=1,MAXC)
C
C     SCALE SOLUTION VECTOR
C
60    DO 65 I=1,MAXC
65    SOLN(I)=SOLN(I)/D(I)
      IF(KSV) 75,75,70
70    WRITE (3,105) MY
      WRITE (3,109) (SOLN(I),I=1,MAXC)
C
C     GENERATE COEFFICENTS FOR ALL DEGREES
C
75    IF(KC) 92,93,92
92    WRITE (3,107)
93    DO 85 IM=1,MAXD
      ID=(IM*(IM+3))/2+1
      DO 72 II=1,45
 72   C(II)=D0
      JD=ID
      JJMIN=1
      JJMAX=0
      DO 73 JP=1,3
      READ (NT2) X
      IF (JD) 73,73,748
 748  IF (JP-2) 751,750,750
 750  JJMIN=JJMIN+15
751   JQ=15
      IF(JD-JQ) 752,753,753
752   JQ=JD
753   JD=JD-15
      JJMAX=JJMAX+JQ
      JX=0
      DO 82 JJ=JJMIN,JJMAX
      JX=JX+1
      DO 80 II=1,ID
80    C(JJ)=C(JJ)+X(II,JX)*SOLN(II)
82    CONTINUE
73    CONTINUE
      IF(KC) 90,91,90
90    WRITE (3,108) IM
      WRITE (3,109) (C(II),II=1,ID)
91    CONTINUE
85    WRITE (NT3) C
      REWIND NT1
      REWIND NT2
61    CONTINUE
      REWIND NT3
C
C     CALL TREND6
C
```

```
      IPHASE(1)=-472267307                                            TRN5 180
      IPHASE(2)=-990494656                                            TRN5 181
      CALL CHAIN(IPHASE)                                              TRN5 182
      END                                                             TRN5 183
```

```
// EXEC ASSEMBLY
*                                                                     TREA    0
*              SUBROUTINE TREAD(NT,X,NBYTES)                          TREA    1
*                                                                     TREA    2
*              READ NBYTES INTO LOCATION X FROM TAPE NT               TREA    3
               PRINT NOGEN                                            TREA    4
TREAD          START    0                                             TREA    5
               USING    *,15                                          TREA    6
               STM      14,4,12(13)                                   TREA    7
               LM       2,4,0(1)     LOAD ADDRESSES                   TREA    8
               LA       0,3    FOR LOGICAL ASSIGNMENT FROM DSR        TREA    9
               LNR      0,0                                           TREA   10
               A        0,0(0,2)    DSR-3                             TREA   11
               STC      0,TAPENO                                      TREA   12
               ST       3,CCW      ADDRESS OF X IN CCW                TREA   13
               MVI      CCW,X'02'     SET COMMAND TO READ             TREA   14
               MVC      CCW+6(2),2(4)      FILL IN LENGTH             TREA   15
               EXCP     TAPE                                          TREA   16
               WAIT     TAPE                                          TREA   17
               LM       2,4,28(13)                                    TREA   18
               MVI      12(13),X'FF'                                  TREA   19
               BR       14                                            TREA   20
TAPE           CCB      SYS001,CCW                                    TREA   21
TAPENO         EQU      TAPE+7  INSERT ADDRESS OF LOGICAL UNIT        TREA   22
CCW            CCW      X'02',0,0,1                                   TREA   23
               END      TREAD                                         TREA   24
```

```
// EXEC FORTRAN
C
C      TREND6  REQUIRES POLY, FORMAT, AND FILE.
C
       COMMON NT1,NT2,NT3,KT,MAXD,MAXC,NS,NY,KM,KC,KSV,KIM,NSC,YMIN,YMAX
       COMMON TITLE(18),KTR(8),UINL(8),UFNL(8),UINC(8),VINL(8),VFNL(8)
       COMMON VINC(8),FMT2(18),FMT3(18),FMT4(18)
       COMMON DUMMY
C
C      DUMMY IS NEEDED FOR DOUBLE PRECISION ALIGNMENT
C      DO IS 'D ZERO'
C
       DOUBLE PRECISION DO
       DOUBLE PRECISION YP,U,V,YRS(25),C(45,8)
       DIMENSION IPHASE(2)
       DIMENSION YPR(100),YSING(25)
 52    FORMAT(1H1,48X,I1,21H DEGREE TREND SURFACE/1H )
 53    FORMAT(15H COLUMN NUMBER ,I3)
  55   FORMAT (1H1,52X,I1,16H DEGREE RESIDUAL/1H0)
  59   FORMAT (1H0,6X,7HINITIAL,13X,5HFINAL,12X,9HINCREMENT/1H ,1HU,2X,
     1E14.7,2(5X,E14.7)/1H ,1HV,2X,E14.7,2(5X,E14.7))
 60    FORMAT(1H0,'RESID SS 'E15.7,5X,'DF',I5,5X,'MS',E16.7,5X,
     $     'SD',E16.7/1H ,F7.3,' PERCENT SS ACCOUNTED FOR')
 61    FORMAT(17HOVARIANCE OF DATA,E15.7,6X,2HSD,E15.7,6X,8HTOTAL SS,
     1  E15.7)
 102   FORMAT('  THIS IS TO BE REPLACED BY FMT2 DURING EXECUTION
     $                                                 ')
 103   FORMAT('  THIS IS TO BE REPLACED BY FMT3 DURING EXECUTION
     $                                                 ')
 104   FORMAT('  THIS IS TO BE REPLACED BY FMT4 DURING EXECUTION
     $                                                 ')
 105   FORMAT(1H ,'TRACER ',A4,2X,2D15.4,F15.4,D15.4)
C
C      NP IS DEGREE OF POLYNOMIAL FOR TREND OR RESIDUAL.
C
C      NOTE THAT DO IS 'D ZERO'
C
       DO=0.0D0
       CALL FORMAT (FMT2)
       READ (1,102)
       CALL FORMAT (FMT3)
       READ (1,103)
       CALL FORMAT (FMT4)
       READ (1,104)
       IF (KT) 880,880,801
 801   IR1=1
C
C      USE CODE 1 TO IDENTIFY THE DATA FILE ON KT
C
       WRITE (KT) IR1,TITLE
       WRITE (KT) NS,NY
       DO 805 J=1,NS
       READ (4) ID,USING,VSING,(YSING(I),I=1,NY)
 805   WRITE (KT) ID,USING,VSING,(YSING(I),I=1,NY)
       CALL FILE(KT,0)
       REWIND 4
 880   CONTINUE
       IR2=2
       IR3=3
       DO 80 MY=1,NY
```

```
      DO 8 JC=1,MAXD
 8    READ (NT3) (C(IC,JC),IC=1,45)
      NP=0
      VAR=0.0
C
      DO 47  JCNT=1,MAXD
      NP=NP+1
      FNP=NP
      NC=1.0+1.5*FNP+0.5*FNP**2
      K=KTR(JCNT)
C
C     QUESTION - IS TREND SURFACE REQUIRED.
C
      IF (K) 10,10,28
 10   WRITE (3,52) NP
      IMAXV=1.0+(VFNL(NP)-VINL(NP))/VINC(NP)
      IMAXU=1.0+(UFNL(NP)-UINL(NP))/UINC(NP)
      WRITE (3,59) UINL(NP),UFNL(NP),UINC(NP),VINL(NP),VFNL(NP),VINC(NP)
      IF (KT) 12,12,11
C
C     USE CODE 2 TO IDENTIFY A TREND SURFACE FILE ON KT
C
 11   WRITE (KT) IR2,TITLE
      WRITE (KT) NP,IMAXU,IMAXV,UINL(NP) ,UFNL(NP),UINC(NP),
     $    VINL(NP),VFNL(NP),VINC(NP)
 12   IVV=0
      U=DO
      V=DO
      U=UINL(NP)-UINC(NP)
C
      DO 27 IU=1,IMAXU
      U=U+UINC(NP)
      IVV=IVV+1
      WRITE (3,53) IVV
      V=VINL(NP)-VINC(NP)
      DO 26 IV=1,IMAXV
      V=V+VINC(NP)
      CALL POLY (NP,U,V,C,YP)
C
C     IF NSC=0, NO SCALING.  IF NSC=1, SCALE X10 PRIOR TO OUTPUT.
C     IF NSC=2, SCALE ANTILOG.  IF NSC=3 SCALE **2
C
      IF (NSC-1) 20,201,202
 201  YP=YP*10.0
      GO TO 20
 202  IF (NSC-3) 204,206,20
 204  YP=10.00**YP
      GO TO 20
 206  YP=YP**2
C
C     IF PREDICTED VALUE IS OUTSIDE OF PERMITTED RANGE SET TO 999.9
C
 20   IF (YP-YMAX) 23,25,24
 23   IF (YP-YMIN) 24,25,25
 24   YP=999.9
C
C     REVERSE SEQUENCE IN ORDER TO PRINT COLUMNS DOWNWARDS.
C     THIS IS A RELIC OF AN OLD FORM OF PRINT OUT.
C
 25   ISTORE=IMAXV+1-IV
```

```
 26    YPR(ISTORE)=YP
       WRITE (3,102 ) (YPR(I),I=1,IMAXV)
       IF (KT) 27,27,252
 252   WRITE (KT) (YPR(I),I=1,IMAXV)
 27    CONTINUE
C
       IF (KT) 28,28,271
 271   CALL FILE(KT,0)
C
C      QUESTION - ARE RESIDUALS REQUIRED.
C
  28   IF (K) 47,30,29
  29   IF (K-1) 47,30,47
 30    WRITE (3,55) NP
       IF (KT) 32,32,31
C
C      USE CODE 3 TO IDENTIFY A FILE OF RESIDUALS ON KT
C
 31    WRITE (KT) IR3,TITLE
       WRITE (KT) NP,NS
 32    WRITE (3,104)
       SUMYR=0.0
       SUMY2=0.0
       SUMRS2=0.0
       REWIND NT1
C
C      TAKE DATA FOR STATION FROM STORAGE AND COMPUTE RESIDUAL.
C
C      NOTE THAT TAPE WAS WRITTEN IN DOUBLE PRECISION
C
       DO 43 NSTAT=1,NS
       READ (NT1) ID,U,V,(YRS(I),I=1,NY)
       YR=YRS(MY)
C
C      IF THE NEXT 3 STATEMENTS ARE REMOVED, THE PROGRAM WILL DUMP.
C
       IF (NS) 340,340,341
 340   WRITE (3,105) ID,U,V,YR,YP
 341   CONTINUE
       CALL POLY (NP,U,V,C,YP)
C
C      IF NSC=0, NO SCALING.  IF NSC=1, SCALE X10 PRIOR TO OUTPUT.
C      IF NSC=2, SCALE ANTILOG.  IF NSC=3 SCALE **2
C
       IF (NSC-1) 36,351,352
 351   YP=YP*10.0
       YR=YR*10.0
       GO TO 36
 352   IF (NSC-3) 354,356,36
 354   YP=10.0**YP
       YR=10.0**YR
       GO TO 36
 356   YP=YP**2
       YR=YR**2
  36   RES=YR-YP
       SUMRS2=SUMRS2+RES**2
C
C      COMPUTE VARIANCE AND STANDARD DEVIATION OF ORIGINAL DATA.
C
       IF (VAR) 42,39,42
 39    SUMYR=SUMYR+YR
```

```
      SUMY2=SUMY2+YR**2
C
C     CONVERT TO SINGLE PRECISION FOR PRINTING
C
   42 USING=U
      VSING=V
      YPSING=YP
      WRITE (3,103) ID,USING,VSING,YR,YPSING,RES
      IF (KT) 43,43,421
  421 WRITE (KT)USING,VSING,RES
   43 CONTINUE
C
      REWIND NT1
      DF=NS -NC
      IDF=DF
      ZMS=SUMRS2/DF
      SD=SQRT(ZMS)
      IF(VAR) 46,44,46
   44 FNS=NS
      VAR=SUMY2/(FNS-1.0)-(SUMYR/(FNS**2-FNS))*SUMYR
      SUMSQ=VAR*(FNS-1.0)
      SDMEAN=SQRT(VAR)
      WRITE (3,61) VAR,SDMEAN,SUMSQ
C
C     COMPUTE PERCENTAGE SS ACCOUNTED FOR.
C
   46 SSPCNT=(1.0-SUMRS2/SUMSQ)*100.0
      WRITE (3,60) SUMRS2,IDF,ZMS,SD,SSPCNT
      IF (KT) 47,47,461
  461 WRITE (KT) SSPCNT
      CALL FILE(KT,0)
   47 CONTINUE
   80 CONTINUE
      REWIND NT3
      CALL EXIT
      END
```

```
// EXEC FORTRAN
C
      SUBROUTINE POLY (NP,U,V,C,YP)
      DOUBLE PRECISION U,V,YP,C(45,8),PU(8),PV(8)
C
C     USE THE COEFFICIENTS TO DETERMINE THE PREDICTED VALUE.
C
      M=NP
      PU(1)=U
      PV(1)=V
      DO 64 I=2,NP
      PU(I)=PU(I-1)*U
   64 PV(I)=PV(I-1)*V
      YP=C(1,M)+C(2,M)*PU(1)+C(3,M)*PV(1)
      YP=YP+C(4,M)*PU(2)+C(5,M)*PU(1)*PV(1)+C(6,M)*PV(2)
      IF (NP-3) 80,68,68
   68 YP=YP+C(7,M)*PU(3)+C(8,M)*PU(2)*PV(1)+C(9,M)*PU(1)*PV(2)
      YP=YP+ C(10,M)*PV(3)+C(11,M)*PU(4)+C(12,M)*PU(3)*PV(1)
      YP=YP+C(13,M)*PU(2)*PV(2)+C(14,M)*PU(1)*PV(3)+C(15,M)*PV(4)
      IF (NP-5) 80,69,69
   69 YP=YP+C(16,M)*PU(5)+C(17,M)*PU(4)*PV(1)
      YP=YP+ C(18,M)*PU(3)*PV(2)+C(19,M)*PU(2)*PV(3)
      YP=YP+C(20,M)*PU(1)*PV(4)+C(21,M)*PV(5)
      IF (NP-6) 80,70,70
   70 YP=YP+C(22,M)*PU(6)+C(23,M)*PU(5)*PV(1)+C(24,M)*PU(4)*PV(2)
      YP=YP+C(25,M)*PU(3)*PV(3)+C(26,M)*PU(2)*PV(4)
      YP=YP+C(27,M)*PU(1)*PV(5)+C(28,M)*PV(6)
      IF (NP-7) 80,71,71
   71 YP=YP+C(29,M)*PU(7)+C(30,M)*PU(6)*PV(1)+C(31,M)*PU(5)*PV(2)
      YP=YP+C(32,M)*PU(4)*PV(3)+C(33,M)*PU(3)*PV(4)+C(34,M)*PU(2)*PV(5)
      YP=YP+C(35,M)*PU(1)*PV(6)+C(36,M)*PV(7)
      IF (NP-8) 80,72,72
   72 YP=YP+C(37,M)*PU(8)+C(38,M)*PU(7)*PV(1)+C(39,M)*PU(6)*PV(2)
      YP=YP+C(40,M)*PU(5)*PV(3)+C(41,M)*PU(4)*PV(4)+C(42,M)*PU(3)*PV(5)
      YP=YP+C(43,M)*PU(2)*PV(6)+C(44,M)*PU(1)*PV(7)+C(45,M)*PV(8)
   80 RETURN
      END
```

```
// EXEC FORTRAN
C                                                                       DBM1
C     DBMCI1   1ST OF 5 PHASES FOR CONTOURING                           DBM1
C                                                                       DBM1
C     THIS VERSION USES 3 CALCOMP SUBROUTINES - PLOTS (WITH ENTRY POINTSDBM1
C         FACTOR AND PLOT), SYMBOL, AND NUMBER.                         DBM1
C     THE UTILITY ROUTINES FILE AND CHAIN ARE CALLED.                   DBM1
C     SUBROUTINES SDATA, SRESID, PDATA, CHECK1, AND CHECK2 ARE USED.    DBM1
C                                                                       DBM1
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT                               DBM1
      COMMON N,NY,INY,NP,JX,JY                                          DBM1
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN                        DBM1
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL                                    DBM1
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY        DBM1
      COMMON SS,ISYM,NPLACE,SIZE                                        DBM1
      COMMON BUFFER(150)                                                DBM1
      DIMENSION TITLE(18),B(100)                                        DBM1
C                                                                       DBM1
C     MAIN PROGRAM LOCATES THE CORRECT FILE ON THE INPUT TAPE, WHICH WASDBM1
C     WRITTEN BY THE TREND SURFACE PROGRAM, AND CALLS THE SUBROUTINES.  DBM1
C                                                                       DBM1
  100 FORMAT(55H **********DATA CANNOT BE READ FROM THIS FILE**********)DBM1
  101 FORMAT(62H **********THIS FILE HAS RESIDUALS - CHECK DATA DECK****DBM1
     1******)                                                           DBM1
  102 FORMAT(66H **********THIS FILE HAS TREND SURFACE - CHECK DATA DECKDBM1
     1**********)                                                       DBM1
  103 FORMAT(61H **********THIS FILE HAS RAW DATA - CHECK DATA DECK*****DBM1
     1*****)                                                            DBM1
  104 FORMAT(8F6.2,3I2)                                                 DBM1
  105 FORMAT(2I2)                                                       DBM1
  106 FORMAT(1H /1H ,18A4)                                              DBM1
  107 FORMAT(13H DATA PLOTTED)                                          DBM1
  108 FORMAT(F7.3,7X,F7.3)                                              DBM1
  109 FORMAT(1H ,I2,25H DEGREE RESIDUALS PLOTTED)                       DBM1
  110 FORMAT(51H **********DATA PLOTTED BUT NOT CONTOURED**********)    DBM1
  111 FORMAT(43H **********DATA CANNOT BE PLOTTED**********)            DBM1
  112 FORMAT(3I2)                                                       DBM1
  113 FORMAT(11H **********,I3,45H DEGREE RESIDUALS CANNOT BE PLOTTED***DBM1
     1*******)                                                          DBM1
  114 FORMAT(11H **********,I3,53H DEGREE RESIDUALS PLOTTED BUT NOT CONTDBM1
     1OURED**********)                                                  DBM1
  115 FORMAT(11H **********,I3,41H DEGREE TREND CANNOT BE PLOTTED*******DBM1
     1***)                                                              DBM1
  502 FORMAT(1H ,'JOB COMPLETED')                                       DBM1
  521 FORMAT(12H SCALING  SF,F6.2,8H, FACTOR,F6.2)                      DBM1
  523 FORMAT(9H LIMITS U,F5.2,3H TO,F7.2,3H, V,F5.2,3H TO,F7.2,         DBM1
     1       5H.   X,F5.1,3H, Y,F5.1,7H INCHES)                         DBM1
C                                                                       DBM1
C     SUBROUTINE FACTOR WILL BE CALLED WITH ARGUMENT FACT.              DBM1
C     IF FACT=0. IT WILL BE SET EQUAL TO 1.   SET FACT=2. FOR 0.005 INC.DBM1
C     UINL IS THE VALUE OF U AT THE ORIGIN.  UFNL IS THE HIGHEST VALUE. DBM1
C     VINL IS THE VALUE OF V AT THE ORIGIN.  VFNL IS THE HIGHEST VALUE. DBM1
C     THE PLOT IS NOT TO EXCEED XMAXP INCHES ON X AXIS.                 DBM1
C     THE PLOT IS NOT TO EXCEED YMAXP INCHES ON Y AXIS.                 DBM1
C     SIZE IS HEIGHT OF LETTERS. ISYM SELECTS SYMBOL.                   DBM1
C     NPLACE IS NUMBER OF DECIMALS FOR WRITING DATA.                    DBM1
C                                                                       DBM1
C                                                                       DBM1
C     IF ISYM IS ZERO OMIT PLOT OF DATA POINTS.                         DBM1
C     IF NPLACE IS NEGATIVE DO NOT WRITE DATA VALUES.                   DBM1
```

```
C
C     KTAPE IS DATA TAPE WRITTEN IN BINARY BY TREND SURFACE PROGRAM.
C     AT START OF EACH FILE IS AN INTEGER INDICATING FILE CONTENTS-
C        1 INDICATES FILE OF RAW DATA
C        2 INDICATES FILE OF TREND SURFACE
C        3 INDICATES FILE OF RESIDUALS.
C     ITYPE = 1,2, OR 3 INDICATING TYPE OF FILE TO BE PROCESSED.
C     SUBROUTINE FILE WILL BE CALLED WITH ARGUMENT NFILE.
C     NFILE IS ZERO TO PICK UP THE THE FIRST FILE.
C     AFTER THE FIRST FILE HAS BEEN READ, NFILE=1 WILL PROCEED TO THE
C     2ND FILE.  AT START NFILE=1 WOULD HAVE INDICATED PROCEED DIRECTLY
C     TO 2ND FILE.
C     NFILE=99 INDICATES TO EMPTY BUFFER, TIDY UP AND CALL EXIT.
C
C     READ DATA SET REFERENCE NUMBERS OF 3 WORK FILES.
C
      READ (1,112) IDSR,JDSR,KDSR
C
 1    READ (1,104) FACT,ZUINL,ZUFNL,ZVINL,ZVFNL,XMAXP,YMAXP,SIZE,ISYM,
     1   NPLACE,KTAPE
C
C     'TREND SURFACE'
      NAME3=-472267307
      NAME4=-1002380572
      NAME5=-641285693
      NAME6=-985644992
C     'DEGREE'
      NAMEA=-993671207
      NAMEB=-976928704
C     **************
C
 29   IF (FACT) 30,30,333
 30   FACT=1.0
 333  CALL PLOTS(BUFFER(1),600)
      CALL FACTOR(FACT)
      READ (1,105) NFILE,ITYPE
      IF (NFILE-99) 4,99,99
 4    READ (KTAPE) DUMMY
      REWIND KTAPE
      READ (JDSR) DUMMY
      REWIND JDSR
      IF (NFILE) 3,7,3
 3    CALL FILE(KTAPE,NFILE)
 7    READ (KTAPE) IR,TITLE
C
C     NOTE THAT THE FIRST RECORD IS IR AND TITLE(1) TO TITLE(18)
C     CONFIRM THAT DATA DECK IS IN PHASE WITH DATA TAPE.
C
      IF (IR-ITYPE) 5,15,5
 5    IF (IR-3) 9,8,6
 6    WRITE(3,100)
      GO TO 99
 8    WRITE(3,101)
      GO TO 99
 9    IF (IR-1) 6,11,10
 10   WRITE (3,102)
      GO TO 99
 11   WRITE (3,103)
      GO TO 99
 15   WRITE(3,106) TITLE
```

```
C
C     PLOT TITLE TO IDENTIFY THIS OUTPUT.
C
      CALL SYMBOL(-1.0,0.5, 0.14,TITLE(1),90.0,72)
      UINL=ZUINL
      UFNL=ZUFNL
      VINL=ZVINL
      VFNL=ZVFNL
C     DEPENDING ON WHETHER IR IS 1, 2, OR 3, BRANCH TO SDATA,
C         STREND('DBMCI3'), OR SRESID, TO PROCESS DATA FILE,
C         TREND SURFACE FILE, OR RESIDUALS FILE.
C
C
      IF (IR-2) 16,17,18
C
C     DATA FILE
C
 16   CALL SDATA
      IF (IWARN-1) 40,301,300
 300  WRITE(3,110)
      GO TO 20
 301  WRITE(3,111)
 35   CALL PLOT(5.0,0.0,-3)
      GO TO 50
 40   WRITE(3,107)
      GO TO 20
C
C     TREND SURFACE FILE
C     'DBMCI3'
C
 17   READ (KTAPE) NP,JX,JY,UINL,UFNL,UINC,VINL,VFNL,VINC
      READ (1,108) SF,CONTIN
C
C     CHECK THE DIMENSIONS OF THE PLOT.
C
      CALL CHECK1
      WRITE(3,521) SF,FACT
      WRITE (3,523) UINL,UFNL,VINL,VFNL,UINCH,VINCH
      IF (IWARN) 43,43,302
 302  WRITE(3,115) NP
      CALL PLOT(5.0,0.0,-3)
      GO TO 1
 43   FNP=NP
C
C     ANNOTATE THE PLOT.
C
      CALL SYMBOL(-0.5,0.5,0.14,NAME3,90.0,13)
      CALL NUMBER(-0.5, 2.3 ,0.14,FNP,90.0,-1)
      CALL SYMBOL(-0.5,2.5,0.14,NAMEA,90.0,6)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(UINCH,0.0,2)
      CALL PLOT(UINCH,VINCH,2)
      CALL PLOT(0.0,VINCH,2)
      CALL PLOT(0.0,0.0,2)
      DO 42  I=1,JX
      READ (KTAPE) (B(K),K=1,JY)
 42   WRITE (JDSR) (B(K),K=1,JY)
      REWIND JDSR
      CALL PLOT(0.0,0.0,-3)
C
C     DBMCI3
```

```
C
      NXTPH =-993864509
      NXTPH2=-906805184
      CALL CHAIN(NXTPH)
C
C     RESIDUALS FILE
C
 18   CALL SRESID
      IF (IWARN-1) 45,303,304
 303  WRITE (3,113) NP
      GO TO 35
 304  WRITE (3,114) NP
      GO TO 20
 45   WRITE (3,109) NP
 20   CALL PLOT(WIDTH,0.0,-3)
 50   WRITE (3,521) SF,FACT
      WRITE (3,523) UINL,UFNL,VINL,VFNL,UINCH,VINCH
C     GO TO 333
C
 99   WRITE (3,502)
      CALL EXIT
      END
```

```
// EXEC FORTRAN
C
      SUBROUTINE SDATA
C
C     PROCESS DATA FILE.
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON BUFFER(150)
  100 FORMAT(3F7.3,I2)
C
C     MULTIPLY U AND V BY SF TO CONVERT TO INCHES.
C     IF SF IS ZERO SET SF = 1.0
C     GRID IS LENGTH OF SIDE OF GRID SQUARE IN INCHES.
C     CONTIN IS THE CONTOUR INTERVAL.
C     THERE ARE NY READINGS/STATION.  PROCESS THE INY'TH OF THESE.
C
C     CHECK THAT LIMITS OF AREA DEFINED ARE WITHIN MAXIMUM DIMENSIONS
C     PERMITTED.
C
C     'RAW DATA'
      NAME1=-641604032
      NAME2=-993926207
C
      READ(1,100) SF,GRID,CONTIN,INY
      IF (INY) 20,20,21
   20 INY=1
   21 CALL CHECK1
      IF (IWARN) 2,2,99
    2 READ (KTAPE) N,NY
C
C     PICK LARGEST AND SMALLEST VALUES OF DATA FUNCTION
C     RETURN IF ANY STATION IS OUTSIDE THE LIMITS OF THE AREA DEFINED.
C
      KSW=0
      CALL CHECK2(KSW)
      IF (IWARN) 3,3,99
    3 CALL SYMBOL(-0.5,0.5,0.14,NAME1,90.0,8)
C
C     OUTLINE THE AREA AND PLOT VALUES AT STATIONS.
C
      CALL PDATA
      IF (CONTIN) 99,99,10
C
C     CONTOUR THE VALUES PLOTTED.
C
C     NEXT PHASE IS DBMCI2 (ORIGINALLY SUBROUTINE SGRID)
C
   10 CALL PLOT(0.0,0.0,-3)
      NXTPH=-993864509
      NXTPH2=-906870720
      CALL CHAIN(NXTPH)
   99 RETURN
      END
```

```
// EXEC FORTRAN
C
      SUBROUTINE SRESID
C
C     PROCESS FILE OF RESIDUALS
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON BUFFER(150)
  100 FORMAT(3F7.3)
C
C     GRID IS LENGTH OF SIDE OF GRID SQUARE IN INCHES.
C     SF IS SCALING FACTOR.   CONTIN IS CONTOUR INTERVAL.
C
C     'RESIDUALS'
      NAME8=-641342775
      NAME9=-991641133
      NAME10=-499105728
C     ' DEGREE'
      NAME11=1086637511
      NAME12=-641350336
C     'PERCENT SS'
      NAME13=-674899517
      NAME14=-975838400
      NAME15=-488488896
      READ (1,100) SF,GRID,CONTIN
C
C     CONFIRM THAT DIMENSIONS OF PLOT ARE CORECT.
C
      CALL CHECK1
      IF (IWARN) 2,2,999
    2 READ (KTAPE) NP,N
C
C     CONFIRM THAT STATIONS FALL WITHIN AREA DEFINED.
C
      KSW=1
      CALL CHECK2(KSW)
      IF (IWARN) 3,3,999
    3 FNP=NP
      CALL SYMBOL(-0.5,0.5,0.14,NAME8,90.0,9)
      CALL NUMBER(-0.5, 1.8  ,0.14,FNP,90.0,-1)
      CALL SYMBOL(-0.5, 2.0 ,0.14,NAME11,90.0,7)
      CALL PDATA
      READ (KTAPE) SS
      SS=SS+0.05
      WIDTH=UINCH+4.0
      XX=WIDTH-3.0
      CALL NUMBER(XX,0.5,0.14,SS,90.0,1)
      CALL SYMBOL(XX, 1.25,0.14,NAME13,90.0,10)
      IF (CONTIN) 999,999,10
C
C     NEXT PHASES IS DBMCI2 (ORIGINALLY SUBROUTINE SGRID)
C
   10 CALL PLOT(0.0,0.0,-3)
      NXTPH=-993864509
      NXTPH2=-906870720
```

```
      CALL CHAIN(NXTPH)
  999 RETURN
      END
```

```
// EXEC FORTRAN
C
      SUBROUTINE PDATA
C
C     PLOT VALUES AT STATIONS.
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON BUFFER(150)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(UINCH,0.0,2)
      CALL PLOT(UINCH,VINCH,2)
      CALL PLOT(0.0,VINCH,2)
      CALL PLOT(0.0,0.0,2)
C
C     IF ISYM IS ZERO DO NOT PLOT DATA.
C
      IF (ISYM) 2,99,2
    2 DO 30 K=1,N
      READ (JDSR) UD,VD,DF
C
C     USE RELATIVE BASE SO THAT ORIGIN WILL BE U=0.0, V=0.0
C
      U=(UD-UINL)*SF
      V=(VD-VINL)*SF
      CALL SYMBOL(U,V,SIZE,ISYM,0.0,-1)
C
C     IF NPLACE IS NEGATIVE DO NOT WRITE DATA VALUES.
C
      IF (NPLACE) 30,28,28
   28 U=U+SIZE*1.5
C
C     ROUND PRIOR TO PLOTTING.
C
      DFTP=DF +0.5/10.0**NPLACE
      CALL NUMBER(U,V,SIZE,DFTP ,0.0,NPLACE)
   30 CONTINUE
      REWIND JDSR
   99 WIDTH=UINCH+4.0
      RETURN
      END
```

```
// EXEC FORTRAN
C
      SUBROUTINE CHECK1
C
C     CONFIRM THAT DIMENSIONS OF PLOT ARE CORRECT.
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON BUFFER(150)
  517 FORMAT(16H ********** PLOT,F6.1,25H EXCEEDS PERMITTED LENGTH,F6.1,
     1   9H ALONG X.)
  518 FORMAT(16H ********** PLOT,F6.1,25H EXCEEDS PERMITTED HEIGHT,F6.1,
     1   9H ALONG Y.)
      IWARN=0
      IF (SF) 35,35,36
   35 SF=1.0
   36 UINCH=(UFNL-UINL)*SF
      VINCH=(VFNL-VINL)*SF
      IF (UINCH-XMAXP) 331,331,330
  330 WRITE (3,517) UINCH,XMAXP
      GO TO 999
  331 IF (VINCH-YMAXP) 333,333,332
  332 WRITE (3,518) VINCH,YMAXP
  999 IWARN=1
  333 RETURN
      END
```

```
// EXEC FORTRAN
C
      SUBROUTINE CHECK2(KSW)
C
C     CONFIRM THAT STATIONS ARE WITHIN RANGE.
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON BUFFER(150)
      DIMENSION YRS(25)
  513 FORMAT(18H **********STATION,I4,2H U,E13.5,7H   UINL,E13.5,
     1    10H**********)
  514 FORMAT(18H **********STATION,I4,2H U,E13.5,7H   UFNL,E13.5,
     1    10H**********)
  515 FORMAT(18H **********STATION,I4,2H V,E13.5,7H   VINL,E13.5,
     1    10H**********)
  516 FORMAT(18H **********STATION,I4,2H V,E13.5,7H   VFNL,E13.5,
     1    10H**********)
      IWARN=0
      DO 320 K=1,N
      IF (KSW) 340,340,341
  340 READ (KTAPE) ID,UD,VD,(YRS(J),J=1,NY)
      DF=YRS(INY)
      GO TO 345
  341 READ (KTAPE) UD,VD,DF
  345 WRITE (JDSR) UD,VD,DF
      IF (K-1) 350,350,351
  350 BIGY=DF
      TINY=DF
C
      GO TO 383
  351 IF (DF-BIGY) 381,383,380
  380 BIGY=DF
      GO TO 383
  381 IF (DF-TINY) 382,383,383
  382 TINY=DF
  383 IF (UD-UINL) 313,316,314
C
  313 WRITE (3,513) K,UD,UINL
      GO TO 999
  314 IF (UFNL-UD) 315,316,316
  315 WRITE (3,514) K,UD,UFNL
      GO TO 999
  316 IF (VD-VINL) 317,320,318
  317 WRITE (3,515) K,VD,VINL
      GO TO 999
  318 IF (VFNL-VD) 319,320,320
  319 WRITE (3,516) K,VD,VFNL
  320 CONTINUE
      REWIND JDSR
      RETURN
  999 IWARN=1
      REWIND JDSR
      RETURN
      END
```

```
// EXEC FORTRAN
C
C     DBMCI2  -  FORMERLY SUBROUTINE SGRID.   REQUIRES CHAIN AND FSORTA
C
C     MAKE A SQUARE GRID BY USING LEAST SQUARES 2ND DEGREE POLYNOMIAL
C     WITH WEIGHTED DATA.
C     THIS VERSION WEIGHTS BY 1/D**2, AND THE MINIMUM NUMBER OF POINTS
C     ON WHICH A GRID POINT IS PERMITTED TO BE BASED IS  MINCPC=8.
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON B(100)
      DIMENSION W(6,7),P(6,7),S(6,7),WT(50),UU(50),VV(50),Y(50)
      DIMENSION UX(800),VX(800),DFX(800)
  525 FORMAT(17H CONTOUR INTERVAL,F7.2,6H. GRID,F6.2,8H INCHES.)
  526 FORMAT(23H DATA VALUES RANGE FROM,F10.4,3H TO,F10.4)
  527 FORMAT(11H **********,I5,30H CONTOURS CALLED FOR**********)
  529 FORMAT(20H SMALLEST GRID VALUE,F8.2,5H AT (,I4,1H,,I4,1H))
  530 FORMAT(20H  LARGEST GRID VALUE,F8.2,5H AT (,I4,1H,,I4,1H))
  531 FORMAT(1H ,I5,31H POINTS COULD NOT BE DETERMINED)
C
C     MINCPC IS MINIMUM NUMBER OF DATA POINTS PER CIRCLE PERMITTED.
C     CHOOSE RADIUS.   IF THERE ARE CPC POINTS/CIRCLE RADIUS R,
C     THEN THERE ARE CPC/PI POINTS PER GRID SQUARE SIDE R.
C     FNOS IS TOTAL NUMBER OF SQUARES AND TAREA IS TOTAL AREA COVERED.
C     TAREA/FNOS IS AREA OF A SINGLE SQUARE AND R IS LENGTH OF SIDE.
C
C     MINMAX IS THE NUMBER OF INCREMENTS PERMITTED FOR THE RADIUS OF THE
C     COUNTING CIRCLE.
C
      MINMAX=1
      MINCPC=8
C
      READ (JDSR) DUMMY
      REWIND JDSR
      READ (KDSR) DUMMY
      REWIND KDSR
C
C
      CPC=MINCPC+2
      IWARN=0
      FN=N
      R=SQRT((UFNL-UINL)*(VFNL-VINL)*CPC/(FN*3.14))
C
C     NOTE THAT R IS IN U AND V UNITS, AND NOT NECESSARILY IN INCHES.
C
      MAXU=SF*(UFNL-UINL)/GRID+1.0
      MAXV=SF*(VFNL-VINL)/GRID+1.0
      WRITE (3,526) TINY,BIGY
      WRITE (3,525) CONTIN,GRID
      NCONT=(BIGY-TINY)/CONTIN
C
C     NUMBER OF CONTOURS IS TO RANGE FROM 2 TO 100.
C
      IF (NCONT-2) 390,391,391
  390 WRITE (3,527) NCONT
```

```
      IWARN=2
C
C     WE COULD FETCH PHASE1 (DBMCI1) AND BEGIN AGAIN.
C     IN THAT CASE THE VALUE OF IWARN WOULD HAVE TO BE TESTED AT THE
C        START OF PHASE1.   BUT THE DATA DECK IS THEN LIKELY TO BE
C        OUT OF PHASE, AND IT IS SAFER TO CALL EXIT.
C
C     NXTPH =-993864509
C     NXTPH2=-906936256
C     CALL CHAIN(NXTPH)
C
      CALL EXIT
C
 391  IF (NCONT-100) 392,392,390
 392  JY=MAXV
      JX=MAXU
      KBAD=0
      KSW=0
      GAP=(VFNL-VINL)- FLOAT(MAXV-1)*GRID/SF
      DO 393 I=1,N
 393  READ (JDSR) UX(I),VX(I),DFX(I)
      REWIND JDSR
      U1=UINL-GRID/SF
      DO 50 IU=1,MAXU
      U1=U1+GRID/SF
C
C     ADJUST START NEAR TOP OF COLUMN SO THAT LAST VALUE IS AT V=0
C
      V1=VFNL+GRID/SF-GAP
      DO 49 IV=1,MAXV
      RI=R
C
C     U2 AND U3 ARE USED TO DEFINE SQUARE ENCLOSING CIRCLE RADIUS R.
C
      U2=U1+R
      U3=U1-R
      V1=V1-GRID/SF
C
C     V2 AND V3 ARE USED TO DEFINE SQUARE ENCLOSING CIRCLE RADIUS RI.
C     COUNT THE POINTS INSIDE THE CIRCLE.
C     RETURN TO THIS POINT IF THE RADIUS HAS TO BE INCREMENTED.
C
      KMINC=0
 45   V2=V1+RI
      V3=V1-RI
      RI2=RI**2
      KOUNT    =0
      DO 28 K=1,N
      U=UX(K)
      V=VX(K)
      DF=DFX(K)
      IF (U-U3) 28,22,21
 21   IF (U-U2) 22,22,28
 22   IF (V-V3) 28,24,23
 23   IF (V-V2) 24,24,28
 24   HT2=RI2-(U-U1)**2
      IF (HT2) 241,241,240
 240  H=SQRT(HT2)
      GO TO 242
 241  H=0.0
```

```
 242  IF (V-V1) 26,27,25
 25   IF (V1+H-V) 28,27,27
 26   IF (V1-H-V) 27,27,28
 27   KOUNT    =KOUNT    +1
      IF(KOUNT-50) 271,271,11
 11   KOUNT=50
      GO TO 465
 271  KKK=KOUNT
      UU(KKK)=U
      VV(KKK)=V
      Y(KKK)=DF
 28   CONTINUE
      NN=KOUNT
      FN=NN
      IF (KOUNT-MINCPC) 46,465,465
C
C     ALLOW ONLY MINMAX INCREMENTS TO THE COUNTING CIRCLE.
C
 46   IF (KMINC-MINMAX) 461,461,71
 461  KMINC=KMINC+1
      RI=RI+R
      U2=U1+RI
      U3=U1-RI
      GO TO 45
C
C     NN IS GREATER THAN OR = TO MINCPC AND LESS THAN OR = TO 50
C
 465  SUMY2=0.0
      TOTWT=0.0
      DO 5 J=1,7
      DO 5 I=1,6
      S(I,J)=0.0
 5    W(I,J)=0.0
C
C     DETERMINE THE APPROPRIATE WEIGHTS
C
      DO 7 I=1,NN
C
C     IF A GRID POINT COINCIDES WITH A DATA POINT USE THIS VALUE.
C     OTHERWISE YOU WILL DIVIDE BY ZERO.
C
      DIST2=(UU(I)-U1)**2+(VV(I)-V1)**2
      IF (DIST2) 8,8,7
 8    B(IV)=Y(I)
      GO TO 68
 7    WT(I)=1.0/DIST2
C
C     FIT 2ND DEGREE TREND SURFACE AND SOLVE FOR GRID POINT.
C
      CODE=UU(1)
      SCALE=Y(1)
      IF (CODE) 611,610,611
 610  CODE=1.0
 611  IF (SCALE) 613,612,613
 612  SCALE=1.0
 613  DO 63 I=1,NN
      UC=UU(I)/CODE
      VC=VV(I)/CODE
      YS=Y(I)/SCALE
      SUMY2=SUMY2+YS**2
C
```

```
C     IF THE WEIGHT AT A GIVEN POINT IS W, THIS IS EQUIVALENT TO
C     HAVING W POINTS WITH COINCIDENT VALUES OF U,V,Y.
C
      W(1,2)=W(1,2)+UC*WT(I)
      W(1,3)=W(1,3)+VC*WT(I)
      W(1,4)=W(1,4)+UC**2*WT(I)
      W(1,5)=W(1,5)+UC*VC*WT(I)
      W(1,6)=W(1,6)+VC**2*WT(I)
      W(2,4)=W(2,4)+UC**3*WT(I)
      W(2,5)=W(2,5)+UC**2*VC*WT(I)
      W(2,6)=W(2,6)+UC*VC**2*WT(I)
      W(3,6)=W(3,6)+VC**3*WT(I)
      W(4,4)=W(4,4)+UC**4*WT(I)
      W(4,5)=W(4,5)+UC**3*VC*WT(I)
      W(4,6)=W(4,6)+UC**2*VC**2*WT(I)
      W(5,6)=W(5,6)+UC*VC**3*WT(I)
      W(6,6)=W(6,6)+VC**4*WT(I)
      W(1,7)=W(1,7)+YS*WT(I)
      W(2,7)=W(2,7)+YS*WT(I)*UC
      W(3,7)=W(3,7)+YS*WT(I)*VC
      W(4,7)=W(4,7)+YS*WT(I)*UC**2
      W(5,7)=W(5,7)+YS*WT(I)*UC*VC
      W(6,7)=W(6,7)+YS*WT(I)*VC**2
   63 TOTWT=TOTWT+WT(I)
      W(1,1)=TOTWT
      W(2,2)=W(1,4)
      W(2,3)=W(1,5)
      W(3,3)=W(1,6)
      W(3,4)=W(2,5)
      W(3,5)=W(2,6)
      W(5,5)=W(4,6)
      DO 64 I=1,5
      JMIN=I+1
      DO 64 J=JMIN,6
   64 W(J,I)=W(I,J)
      DO 66 I=1,6
      DO 65 J=1,7
   65 P(I,J)=W(I,J)-S(1,I)*P(1,J)-S(2,I)*P(2,J)-S(3,I)*P(3,J)-S(4,I)*P(4
     1      ,J)-S(5,I)*P(5,J)
      PII=P(I,I)
      IF (PII) 641,71,641
  641 DO 66 J=1,7
   66 S(I,J)=P(I,J)/PII
      CF6=S(6,7)
      CF5=S(5,7)-CF6*S(5,6)
      CF4=S(4,7)-CF6*S(4,6)-CF5*S(4,5)
      CF3=S(3,7)-CF6*S(3,6)-CF5*S(3,5)-CF4*S(3,4)
      CF2=S(2,7)-CF6*S(2,6)-CF5*S(2,5)-CF4*S(2,4)-CF3*S(2,3)
      CF1=S(1,7)-CF6*S(1,6)-CF5*S(1,5)-CF4*S(1,4)-CF3*S(1,3)-CF2*S(1,2)
      CF6=CF6*SCALE/CODE**2
      CF5=CF5*SCALE/CODE**2
      CF4=CF4*SCALE/CODE**2
      CF3=CF3*SCALE/CODE
      CF2=CF2*SCALE/CODE
      CF1=CF1*SCALE
      B(IV)=CF1+CF2*U1+CF3*V1+CF4*U1**2+CF5*U1*V1+CF6*V1**2
C
C     B IS VALUE OF Y AT GRID POINT.
C     GET RANGE OF DATA POINTS USED FOR THIS DETERMINATION.
C     COMPUTE RANGE.  IF GRID VALUE IS 20 PERCENT BEYOND THIS RANGE
```

```
C     SET GRID VALUE TO 999.9 AS A FLAG FOR A BAD VALUE.
C
      CALL FSORTA(Y,NN)
      PERMIT=0.2*(Y(NN)-Y(1))
      YMAX=Y(NN)+PERMIT
      YMIN=Y(1)-PERMIT
      IF (B(IV)-YMAX) 70,68,71
 70   IF (B(IV)-YMIN) 71,68,68
 71   B(IV)=999.9
      KBAD=KBAD+1
      GO TO 49
C
C     FIND THE LARGEST AND SMALLEST VALUES IN THE GRID.
C
 68   IF (KSW) 601,601,602
 601  KSW=1
      BIGY2=B(IV)
      TINY2=B(IV)
      NBIGU=IU
      NBIGV=IV
      NTINYU=IU
      NTINYV=IV
      GO TO 49
 602  IF (B(IV)-BIGY2) 604,49,603
 603  BIGY2=B(IV)
      NBIGU=IU
      NBIGV=IV
      GO TO 49
 604  IF (B(IV)-TINY2) 605,49,49
 605  TINY2=B(IV)
      NTINYU=IU
      NTINYV=IV
 49   CONTINUE
      WRITE (KDSR) (B(I),I=1,MAXV)
 50   CONTINUE
      REWIND KDSR
      WRITE (3,529) TINY2,NTINYU,NTINYV
      WRITE (3,530) BIGY2,NBIGU,NBIGV
      WRITE (3,531) KBAD
C
C     FETCH NEXT PHASE - DBMCI4  (ORIGINALLY SUBROUTINES DRAFT AND SIFT
C
      NXTPH=-993864509
      NXTPH2=-906739648
      CALL CHAIN(NXTPH)
      END
```

```
// EXEC FORTRAN
C
C       DBMCI3    THIS PHASE PROCESSES A TREND SURFACE FILE.
C                 SUBROUTINES CHAIN AND DRAFT ARE REQUIRED.
C
        COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
        COMMON N,NY,INY,NP,JX,JY
        COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
        COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
        COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
        COMMON SS,ISYM,NPLACE,SIZE
        COMMON IFST,N1500,LINEK,NSET
        COMMON POINTS(4,250)
        COMMON B(100)
  525   FORMAT(17H CONTOUR INTERVAL,F7.2,6H. GRID,F6.2,7H INCHES)
        NSET=100
        GRID=UINC*SF
C
C       IT IS ASSUMED THAT UINC=VINC
C
        WRITE (3,525) CONTIN,GRID
        IFST=1
        KSW=0
        DO 50 I=1,JX
        READ (JDSR) (B(K),K=1,JY)
C
C       DETERMINE LARGEST AND SMALLEST VALUES.
C
        DO 383 K=1,JY
        IF (B(K) -999.9) 20,383,20
   20   IF (KSW) 21,21,25
   21   KSW=1
        TINY=B(K)
        BIGY=B(K)
        GO TO 383
   25   IF (B(K)-BIGY) 381,383,380
  380   BIGY=B(K)
        GO TO 383
  381   IF (B(K)-TINY) 382,383,383
  382   TINY=B(K)
  383   CONTINUE
C
C       SUBDIVIDE INTO TRIANGULAR AREAS, DETERMINE STRAIGHT LINE SEGMENTS,
C
        CALL DRAFT
   50   CONTINUE
        REWIND JDSR
        REWIND IDSR
C
C       COMPUTE WIDTH OF PLOT IN ORDER TO SPACE OVER TO NEXT MAP.
C
        WIDTH=FLOAT (JX-1)*GRID+4.0
C
C       REARRANGE INTO OPTIMAL ORDER, AND PLOT THE CONTOURS.
C       FETCH NEXT PHASE - DBMCI5  (ORIGINALLY SUBROUTINE SIFT)
C
        NXTPH =-993864509
        NXTPHB=-906674112
        CALL CHAIN(NXTPH)
        END
```

```
// EXEC FORTRAN
C
      SUBROUTINE DRAFT
C
C     TAKE EACH SQUARE OF GRID IN TURN.  DETERMINE MID-POINT, AND
C     SUBDIVIDE SQUARE INTO 4 TRIANGLES.  ASSUME SURFACE IS PLANAR
C     WITHIN ONE TRIANGLE.
C     THE SUBROUTINE DRAFT INCORPORATES LOGIC TO SCAN THE COLUMNS
C     ALTERNATELY DOWN AND UP.   THIS OPTIMIZED THE ORIGINAL PLOTTING,
C     BUT AS THE PROGRAM NOW STANDS THIS FEATURE IS MADE OBSOLETE BY
C     SUBROUTINE SIFT (DBMCI5).
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON IFST,N1500,LINEK,NSET
      COMMON POINTS(4,250)
      COMMON B(100)
      DIMENSION A(100),C(5)
C
C     CHECK FOR CONTOUR THRU PTS
C
      DO 30 I = 1,JY
      IF (B(I)-999.9) 19,30,19
 19   TEMP = B(I) / CONTIN
      KON = TEMP
      CONTR = KON
      IF ( TEMP - CONTR ) 30,20,30
C
C     THE VALUE .001 IS APPROPRIATE TO DATA AT 2 DP
C
   20  B(I) = B(I) + .001
   30 CONTINUE
C
C     TEST IF THIS IS FIRST TIME THROUGH.
C
      IF (IFST) 40,40,35
   35 IFST = 0
      READ  (IDSR) DUMMYX
      REWIND IDSR
      LINEK=0
      N1500=0
      XCORD=-GRID
      XB=0.0
      IDWN=1
      GO TO 115
C
C     SET UP SQUARE
C
 40   CONTINUE
      NEXT=NEXT
      C(1) = A(NEXT + 1)
      C(2) = B(NEXT + 1)
      C(3) = B(NEXT)
      C(4) = A(NEXT)
      C(5) = C(1)
C
```

```
C      A GRID VALUE OF 999.9 IS A FLAG FOR A BAD VALUE.
C
       SUMC=0.0
       ZP=0.0
       DO 300 I=1,4
       IF (C(I)-999.9) 308,300,308
 308   SUMC=SUMC+C(I)
       ZP=ZP+1.0
 300   CONTINUE
C
C      IGNORE THIS SQUARE IF THE NUMBER OF GOOD CORNERS IS LESS THAN 3
C
       IF (ZP-3.0) 110,328,328
C      FIND CENTER VALUE
 328   ZC=SUMC/ZP
C
C      FIRST TRIANGLE
C
       ITRI = 1
C
C      DOES CENTER LIE ON CONTOUR
C
       TEMP = ZC / CONTIN
       KON = TEMP
       CONTR = KON
       IF ( TEMP - CONTR ) 47,44,47
C
C      THE VALUE .001 IS APPROPRIATE TO DATA AT 2 DP
C
   44 ZC = ZC + .001
 47    DO 45 I=1,2
       II=ITRI+I-1
C
C      ONE VALUE FLAGGED AS 999.9 JUSTIFIES IGNORING THE TRIANGLE.
C
       IF (C(II)-999.9) 45,77,45
 45    CONTINUE
C
C      FIND ZA AND ZB
C
       IF ( ITRI - 2) 49,49,51
   49 ZA = C(ITRI)
       ZB = C(ITRI + 1)
       GO TO 53
   51 ZA = C(ITRI + 1)
       ZB = C(ITRI)
   53  AB = ZA- ZB
       CA = ZC - ZA
       CB = ZC - ZB
C
C      FIND MAX AND MIN
C
       IF (AB)54,54,55
   54 IF ( CA ) 57,56,56
   56 ZMIN = ZA
       IF ( CB ) 60,59,59
   59 ZMAX = ZC
       GO TO 70
   55 IF ( CB ) 57,58,58
   58  ZMIN = ZB
```

```
      IF ( CA) 61,61,59
   61 ZMAX = ZA
      GO TO 70
   57 ZMIN = ZC
      IF ( AB ) 60,60,61
   60 ZMAX = ZB
   70 CONTR = ZMIN / CONTIN
      KON = CONTR
      IF ( CONTR ) 72,71 ,71
   71 KON = KON + 1
   72 CONTR = KON
      CONMAX = ZMAX / CONTIN
      KON = CONMAX
      IF ( CONMAX ) 75,76,76
   75 KON = KON - 1
   76 CONMAX = KON
C
C     ANY CONTOURS IN TRIANGLE
C
      IF ( CONTR - CONMAX ) 78,78,77
   77 IF ( ITRI - 4) 79,110,110
   79 ITRI = ITRI + 1
      GO TO 47
   78 CONTR = CONTR * CONTIN
      CONMAX = CONMAX * CONTIN
      XHT = CONTR - ZA
C
C     TURN ON END PT SW
C
   80 IPT = 1
      IF( CONTR - ZB) 81,81,82
   81 IF (XHT) 100,100,83
   82 IF(XHT) 83,83,100
C
C     CALC CROSS PT ON AB
C
83    X1 = XCORD
      Y1 = YCORD
      TEMP = XHT * GRID / (- AB)
      IF ( ITRI - 2 ) 84,85,86
   86 IF ( ITRI - 3 ) 89,88,89
   88 Y1 = Y1 + GRID
   84 X1 = X1 + TEMP
      GO TO 90
   85  X1 = X1 + GRID
   89  Y1 = Y1 + TEMP
   90 IPT = 0
C
C     TEST FOR PTS ON AC
C
      IF ( CONTR - ZC ) 91,91,92
   91 IF(XHT)105,105,100
   92 IF(XHT)100,100,105
C
C     CALC PTS ON AC
C
  100 TEMP = XHT * GRID / (2. * CA )
      IF ( ITRI - 3 ) 205,206,205
  206 Y2 = YCORD + GRID - TEMP
      GO TO 207
  205 Y2 = YCORD + TEMP
```

```
  207 IF(ITRI - 2) 101,102,101
  101 X2 = XCORD + TEMP
      GO TO 103
  102 X2 = XB - TEMP
  103 IF (IPT) 107,107,104
  104  X1 = X2
      Y1 = Y2
      IPT = 0
  105 TEMP = (CONTR - ZB) * GRID / (2. * CB)
      IF ( ITRI - 1) 210,211,210
  210 Y2 = YCORD + GRID - TEMP
      GO TO 212
  211 Y2 = YCORD + TEMP
212   IF (ITRI - 4 ) 102,101,102
C
C     DETERMINE LINE SEGMENT X1,Y1 TO X2,Y2
C
  107 IF (X1-X2) 108,109,108
  109 IF (Y1-Y2) 108,106,108
C
C     COUNT NUMBER OF LINE SEGMENTS IN THIS BATCH.
C     WHEN A BATCH HAS  NSET SEGMENTS, STORE BATCH ON IDSR
C
 108  LINEK=LINEK+1
      POINTS(1,LINEK)=X1
      POINTS(2,LINEK)=Y1
      POINTS(3,LINEK)=X2
      POINTS(4,LINEK)=Y2
      IF (LINEK- NSET) 106,114,114
C
C     COUNT THE NUMBER OF BATCHES OF  NSET SEGMENTS WRITTEN ON IDSR.
C
 114  N1500=N1500+1
      LINEK=0
      WRITE (IDSR) POINTS
C
C     INCREMENT CONTOUR
C
  106 CONTR = CONTR + CONTIN
      XHT = XHT + CONTIN
C
C     MORE CONTOURS
C
      IF(CONTR - CONMAX ) 80,80,77
  110 IF (IDWN) 111,111,112
  111 NEXT = NEXT - 1
      YCORD = YCORD + GRID
      GO TO 113
  112 NEXT = NEXT + 1
      YCORD = YCORD - GRID
  113 IF(NEXT - IEND ) 40,115,40
  115 DO 116 I = 1,JY
  116 A(I) = B(I)
      IF (IDWN) 117,117,118
  117 IDWN = 1
      NEXT = 1
      IEND = JY
      TEMP = JY - 2
      YCORD = TEMP * GRID
      GO TO 119
  118 IDWN = 0
```

```
      NEXT = JY - 1
      IEND  = 0
      YCORD = 0.
  119 XB = XB + GRID
      XCORD = XCORD + GRID
      RETURN
      END
```

```
// EXEC FORTRAN
C
C     DBMCI4     REQUIRES SUBROUTINES CHAIN AND DRAFT.
C     SUBROUTINE DRAFT USED BY DBMCI4 IS IDENTICAL TO THAT IN DBMCI3.
C                GRID DATA IS AVAILABLE ON KDSR.
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON IFST,N1500,LINEK,NSET
      COMMON POINTS(4,250)
      COMMON B(100)
  107 FORMAT(13H DATA PLOTTED)
  521 FORMAT(12H SCALING  SF,F6.2,8H, FACTOR,F6.2)
  523 FORMAT(9H LIMITS U,F5.2,3H TO,F7.2,3H, V,F5.2,3H TO,F7.2,
     1      5H.   X,F5.1,3H, Y,F5.1,7H INCHES)
      NSET=100
      READ (KDSR) DUMMY
      REWIND KDSR
      IFST=1
      DO 50 I=1,JX
      READ (KDSR) (B(K),K=1,JY)
C
C     SUBDIVIDE INTO TRIANGULAR AREAS, DETERMINE STRAIGHT LINE SEGMENTS.
C
   50 CALL DRAFT
      REWIND KDSR
      WIDTH=FLOAT(JX-1)*GRID+5.0
      WRITE (3,107)
      WRITE(3,521) SF,FACT
      WRITE (3,523) UINL,UFNL,VINL,VFNL,UINCH,VINCH
C
C     FETCH DBMCI5  (ORIGINALLY SUBROUTINE SIFT)
C
      NXTPH =-993864509
      NXTPH2=-906674112
      CALL CHAIN(NXTPH)
      END
```

```
// EXEC FORTRAN
C                                                                       DBM5    0
C     DBMCI5   THIS VERSION USES CALCOMP SUBROUTINE PLOTS (ENTRY POINTS DBM5    1
C              FACTOR AND PLOT).   SUBROUTINE SIFT IS CALLED.          DBM5    2
C                                                                       DBM5    3
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT                               DBM5    4
      COMMON N,NY,INY,NP,JX,JY                                          DBM5    5
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN                        DBM5    6
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL                                    DBM5    7
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY        DBM5    8
      COMMON SS,ISYM,NPLACE,SIZE                                        DBM5    9
      COMMON IFST,N1500,LINEK,NSET                                      DBM5   10
      COMMON P(4,250)                                                   DBM5   11
      COMMON BUFFER(1000)                                               DBM5   12
  108 FORMAT(' CONTOURING COMPLETED')                                   DBM5   13
      CALL PLOTS(BUFFER(1),4000)                                        DBM5   14
      CALL FACTOR(FACT)                                                 DBM5   15
      CALL SIFT                                                         DBM5   16
      CALL PLOT (WIDTH,0.0,-3)                                          DBM5   17
      WRITE (3,108)                                                     DBM5   18
      CALL EXIT                                                         DBM5   19
C                                                                       DBM5   20
C     IT IS SIMPLER TO ISSUE A CONTROL CARD TO CAUSE EXECUTION OF       DBM5   21
C     PHASE1 (DBMCI1) AT THIS TIME, RATHER THAN TO ISSUE A FETCH BY     DBM5   22
C     CALLING CHAIN AND KEEPING TRACK OF A RESTART SITUATION.           DBM5   23
C                                                                       DBM5   24
C     FETCH PHASE1 TO START AGAIN.     DBMCI1                           DBM5   25
C     NXTPH =-993864509                                                 DBM5   26
C     NXTPH2=-906936256                                                 DBM5   27
C     CALL CHAIN(NXTPH)                                                 DBM5   28
      END                                                               DBM5   29
```

```
// EXEC FORTRAN
C                                                                       SIFT    0
      SUBROUTINE SIFT                                                   SIFT    1
C                                                                       SIFT    2
C     ARRANGE LINE SEGMENTS IN OPTIMUM ORDER AND PLOT.                  SIFT    3
C                                                                       SIFT    4
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT                               SIFT    5
      COMMON N,NY,INY,NP,JX,JY                                          SIFT    6
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN                        SIFT    7
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL                                    SIFT    8
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY        SIFT    9
      COMMON SS,ISYM,NPLACE,SIZE                                        SIFT   10
```

```
      COMMON IFST,N1500,LINEK,NSET
      COMMON P(4,250)
      COMMON BUFFER(1000)
      N=LINEK
      IF (LINEK) 85,5,40
5     READ (IDSR) P
      N1500=N1500-1
      N=NSET
40    CALL PLOT(P(1,N),P(2,N),3)
      XNOW=P(3,N)
      YNOW=P(4,N)
      JSTORE=1
      N=N-1
      CALL PLOT(XNOW,YNOW,2)
      GO TO 51
45    CALL PLOT(XNOW,YNOW,2)
      N=N-1
      DO 50 J=JSTORE,N
      DO 49 I=1,4
49    P(I,J)=P(I,J+1)
50    CONTINUE
51    SMIN2=(XNOW-P(1,1))**2+(YNOW-P(2,1))**2
      IF (N-1) 60,60,55
55    DO 20 I=1,3,2
      I=I
      DO 15 J=1,N
      J=J
      DIST2=(XNOW-P(I,J))**2+(YNOW-P(I+1,J))**2
      IF (DIST2-SMIN2) 10,10,15
10    SMIN2=DIST2
      JSTORE=J
      ISTORE=I
      IF (SMIN2) 21,21,15
15    CONTINUE
20    CONTINUE
21    IF (ISTORE-1) 30,30,35
30    IF (SMIN2) 32,32,31
31    CALL PLOT(P(1,JSTORE),P(2,JSTORE),3)
32    XNOW=P(3,JSTORE)
      YNOW=P(4,JSTORE)
      GO TO 45
35    IF (SMIN2) 42,42,41
41    CALL PLOT(P(3,JSTORE),P(4,JSTORE),3)
42    XNOW=P(1,JSTORE)
      YNOW=P(2,JSTORE)
      GO TO 45
60    DIST2=(XNOW-P(3,1))**2+(YNOW-P(4,1))**2
      IF (DIST2-SMIN2) 68,65,65
65    IF (SMIN2) 67,67,66
66    CALL PLOT(P(1,1),P(2,1),3)
67    CALL PLOT(P(3,1),P(4,1),2)
      GO TO 80
68    IF (DIST2) 70,70,69
69    CALL PLOT(P(3,1),P(4,1),3)
70    CALL PLOT(P(1,1),P(2,1),2)
80    IF (N1500) 85,81,5
81    CONTINUE
82    REWIND IDSR
85    RETURN
      END
```

```
// EXEC FORTRAN
C                                                                       DBMD
C     DBMCID - REPLACES DBMCI1 WHEN CONTOURING WITHOUT A PRECEDING TRENDDBMD
C                                                                       DBMD
C     THIS VERSION USES 3 CALCOMP SUBROUTINES - PLOTS (WITH ENTRY POINTSDBMD
C          FACTOR AND PLOT), SYMBOL, AND NUMBER.                        DBMD
C     USE SUBROUTINES GETDTA, CHECK1, AND CHAIN.                        DBMD
C                                                                       DBMD
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT                               DBMD
      COMMON N,NY,INY,NP,JX,JY                                          DBMD
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN                        DBMD
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL                                    DBMD
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY        DBMD
      COMMON SS,ISYM,NPLACE,SIZE                                        DBMD
      COMMON BUFFER(150)                                                DBMD
      DIMENSION TITLE(18),B(100)                                        DBMD
 100  FORMAT(18A4)                                                      DBMD
 104  FORMAT(8F6.2,3I2)                                                 DBMD
 105  FORMAT(2I2)                                                       DBMD
 106  FORMAT(1H /1H ,18A4)                                              DBMD
 107  FORMAT(13H DATA PLOTTED)                                          DBMD
 110  FORMAT(51H **********DATA PLOTTED BUT NOT CONTOURED**********)    DBMD
 111  FORMAT(43H **********DATA CANNOT BE PLOTTED**********)            DBMD
 112  FORMAT(3I2,I4)                                                    DBMD
 502  FORMAT(1H ,'JOB COMPLETED')                                       DBMD
 521  FORMAT(12H SCALING  SF,F6.2,8H, FACTOR,F6.2)                      DBMD
 523  FORMAT(9H LIMITS U,F5.2,3H TO,F7.2,3H, V,F5.2,3H TO,F7.2,         DBMD
     1       5H.   X,F5.1,3H, Y,F5.1,7H INCHES)                         DBMD
C                                                                       DBMD
C     SUBROUTINE FACTOR WILL BE CALLED WITH ARGUMENT FACT.              DBMD
C     IF FACT=0. IT WILL BE SET EQUAL TO 1.   SET FACT=2. FOR 0.005 INC.DBMD
C     UINL IS THE VALUE OF U AT THE ORIGIN.  UFNL IS THE HIGHEST VALUE. DBMD
C     VINL IS THE VALUE OF V AT THE ORIGIN.  VFNL IS THE HIGHEST VALUE. DBMD
C     THE PLOT IS NOT TO EXCEED XMAXP INCHES ON X AXIS.                 DBMD
C     THE PLOT IS NOT TO EXCEED YMAXP INCHES ON Y AXIS.                 DBMD
C     SIZE IS HEIGHT OF LETTERS. ISYM SELECTS SYMBOL.                   DBMD
C     NPLACE IS NUMBER OF DECIMALS FOR WRITING DATA.                    DBMD
C                                                                       DBMD
C     IF ISYM IS ZERO OMIT PLOT OF DATA POINTS.                         DBMD
C     IF NPLACE IS NEGATIVE DO NOT WRITE DATA VALUES.                   DBMD
C                                                                       DBMD
C     READ DATA SET REFERENCE NUMBERS OF 3 WORK FILES.                  DBMD
C     N IS NUMBER OF STATIONS.                                          DBMD
C                                                                       DBMD
      READ (1,112) IDSR,JDSR,KDSR,N                                     DBMD
C                                                                       DBMD
 1    READ (1,104) FACT,ZUINL,ZUFNL,ZVINL,ZVFNL,XMAXP,YMAXP,SIZE,ISYM,  DBMD
     1   NPLACE                                                         DBMD
C                                                                       DBMD
C     A BLANK CARD CALLS EXIT                                           DBMD
C                                                                       DBMD
      REWIND IDSR                                                       DBMD
      REWIND JDSR                                                       DBMD
      REWIND KDSR                                                       DBMD
      IF (XMAXP) 99,99,29                                               DBMD
 29   IF (FACT) 30,30,333                                               DBMD
 30   FACT=1.0                                                          DBMD
 333  CALL PLOTS(BUFFER(1),600)                                         DBMD
      CALL FACTOR(FACT)                                                 DBMD
 7    READ (1,100) TITLE                                                DBMD
```

```
 15    WRITE(3,106) TITLE
C
C      PLOT TITLE TO IDENTIFY THIS OUTPUT.
C
       CALL SYMBOL(-1.0,0.5, 0.14,TITLE(1),90.0,72)
       UINL=ZUINL
       UFNL=ZUFNL
       VINL=ZVINL
       VFNL=ZVFNL
 16    CALL GETDTA
       IF (IWARN-1) 40,301,300
 300   WRITE(3,110)
       GO TO 20
 301   WRITE(3,111)
 35    CALL PLOT(5.0,0.0,-3)
       GO TO 50
 40    WRITE(3,107)
       GO TO 20
 20    CALL PLOT(WIDTH,0.0,-3)
 50    WRITE (3,521) SF,FACT
       WRITE (3,523) UINL,UFNL,VINL,VFNL,UINCH,VINCH
       GO TO 1
 99    WRITE (3,502)
       CALL EXIT
       END
```

```
// EXEC FORTRAN
C
      SUBROUTINE GETDTA
C
C     PROCESS DATA FILE.
C
      COMMON KTAPE,IDSR,JDSR,KDSR,SF,FACT
      COMMON N,NY,INY,NP,JX,JY
      COMMON WIDTH,UINCH,VINCH,GRID,CONTIN,IWARN
      COMMON ZUINL,ZUFNL,ZVINL,ZVFNL
      COMMON UINL,UFNL,UINC,VINL,VFNL,VINC,XMAXP,YMAXP,BIGY,TINY
      COMMON SS,ISYM,NPLACE,SIZE
      COMMON BUFFER(150)
      DIMENSION FMT(20)
 100  FORMAT(3F7.3,I2)
 511  FORMAT(20A4)
 512  FORMAT(///////////////////////////////////////////////////////////
     */////////////////////////////////////////////////////////////////)
 513  FORMAT(18H **********STATION,I4,2H U,E13.5,7H   UINL,E13.5,
     1   10H**********)
 514  FORMAT(18H **********STATION,I4,2H U,E13.5,7H   UFNL,E13.5,
     1   10H**********)
 515  FORMAT(18H **********STATION,I4,2H V,E13.5,7H   VINL,E13.5,
     1   10H**********)
 516  FORMAT(18H **********STATION,I4,2H V,E13.5,7H   VFNL,E13.5,
     1   10H**********)
C
C     MULTIPLY U AND V BY SF TO CONVERT TO INCHES.
C     IF SF IS ZERO SET SF = 1.0
C     GRID IS LENGTH OF SIDE OF GRID SQUARE IN INCHES.
C     CONTIN IS THE CONTOUR INTERVAL.
C
C     CHECK THAT LIMITS OF AREA DEFINED ARE WITHIN MAXIMUM DIMENSIONS
C     PERMITTED.
C
C     'RAW DATA'
      NAME1=-641604032
      NAME2=-993926207
C
C     IF ISYM IS ZERO DO NOT PLOT DATA.
C     IF NPLACE IS NEGATIVE DO NOT WRITE DATA VALUES.
C     IF CONTIN NOT POSITIVE, DO NOT CONTOUR.
C
      READ(1,100) SF,GRID,CONTIN
 20   INY=1
 21   CALL CHECK1
      IF (IWARN) 2,2,99
C
C     PICK LARGEST AND SMALLEST VALUES OF DATA FUNCTION
C     RETURN IF ANY STATION IS OUTSIDE THE LIMITS OF THE AREA DEFINED.
C
C
C     FMT IS FORMAT FOR STATION DATA.  U,V,DF
C
 2    READ(1,511) FMT
      CALL FORMAT(FMT)
      WRITE(3,512)
C
C     THE PRECEDING WRITE IS A DUMMY REQUIRED BY FORMAT.
C
```

```
      DO 320 K=1,N
      READ (1,512) UD,VD,DF
      WRITE (JDSR) UD,VD,DF
      IF (K-1) 350,350,351
 350  BIGY=DF
      TINY=DF
C
      GO TO 383
 351  IF (DF-BIGY) 381,383,380
 380  BIGY=DF
      GO TO 383
 381  IF (DF-TINY) 382,383,383
 382  TINY=DF
 383  IF (UD-UINL) 313,316,314
C
 313  WRITE (3,513) K,UD,UINL
      GO TO 999
 314  IF (UFNL-UD) 315,316,316
 315  WRITE (3,514) K,UD,UFNL
      GO TO 999
 316  IF (VD-VINL) 317,320,318
 317  WRITE (3,515) K,VD,VINL
      GO TO 999
 318  IF (VFNL-VD) 319,320,320
 319  WRITE (3,516) K,VD,VFNL
 320  CONTINUE
      REWIND JDSR
 3    CALL SYMBOL(-0.5,0.5,0.14,NAME1,90.0,8)
C
C     OUTLINE THE AREA AND PLOT VALUES AT STATIONS.
C
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(UINCH,0.0,2)
      CALL PLOT(UINCH,VINCH,2)
      CALL PLOT(0.0,VINCH,2)
      CALL PLOT(0.0,0.0,2)
C
C     IF ISYM IS ZERO DO NOT PLOT DATA.
C
      IF (ISYM) 29,8,29
 29   DO 30 K=1,N
      READ (JDSR) UD,VD,DF
C
C     USE RELATIVE BASE SO THAT ORIGIN WILL BE U=0.0, V=0.0
C
      U=(UD-UINL)*SF
      V=(VD-VINL)*SF
      CALL SYMBOL(U,V,SIZE,ISYM,0.0,-1)
C
C     IF NPLACE IS NEGATIVE DO NOT WRITE DATA VALUES.
C
      IF (NPLACE) 30,28,28
 28   U=U+SIZE*1.5
C
C     ROUND PRIOR TO PLOTTING.
C
      DFTP=DF +0.5/10.0**NPLACE
      CALL NUMBER(U,V,SIZE,DFTP ,0.0,NPLACE)
 30   CONTINUE
      REWIND JDSR
```

```
 8     WIDTH=UINCH+4.0                                                  GETD 119
       IF (CONTIN) 99,99,10                                             GETD 120
C                                                                       GETD 121
C      CONTOUR THE VALUES PLOTTED.                                      GETD 122
C                                                                       GETD 123
C      NEXT PHASE IS DBMCI2 (ORIGINALLY SUBROUTINE SGRID)               GETD 124
C                                                                       GETD 125
 10    CALL PLOT(0.0,0.0,-3)                                            GETD 126
       NXTPH=-993864509                                                 GETD 127
       NXTPH2=-906870720                                                GETD 128
       CALL CHAIN(NXTPH)                                                GETD 129
 99    RETURN                                                           GETD 130
 999   IWARN=1                                                          GETD 131
       REWIND JDSR                                                      GETD 132
       RETURN                                                           GETD 133
       END                                                              GETD 134
```

```
// EXEC ASSEMBLY
*                                                                       CHAI   0
*        CHAIN  -  SUBROUTINE FETCH AVAILABLE FOR FORTRAN               CHAI   1
*                                                                       CHAI   2
*        DIMENSION IPHASE(2)                                            CHAI   3
*          .  .  .  .  .  .                                             CHAI   4
*        CALL CHAIN(IPHASE)                                             CHAI   5
*                                                                       CHAI   6
*        IPHASE IS 8 BYTE NAME OF NEXT PHASE TO BE FETCHED              CHAI   7
*        FROM CORE IMAGE LIBRARY.                                       CHAI   8
*        PASS CONTROL TO IPHASE.                                        CHAI   9
*                                                                       CHAI  10
CHAIN    START 0                                                        CHAI  11
       L        1,0(0,1)                                                CHAI  12
       FETCH    (1)                                                     CHAI  13
         END                                                            CHAI  14
```

```
// EXEC ASSEMBLY
*                                                                       FILE   0
*        FILE  -  MANIPULATES TAPE FILES                                FILE   1
*                                                                       FILE   2
*        CALL FILE(NTAPE,NFILES)                                        FILE   3
```

```
*        NTAPE IS DATA SET REFERENCE NUMBER OF TAPE
*        NFILES POSITIVE - FSF NFILES
*        NFILES ZERO - WRITE TAPE MARK + DO NOT REWIND
*        NFILES NEGATIVE - BACKWARD SPACE NFILES.  DO NOT USE TO REWIND
*                           NFILES -1 MEANS GO TO START OF CURRENT FILE
*            ALWAYS LEAVE TAPE POSITIONED IMMEDIATELY BEYOND TM
*
     PRINT NOGEN
FILE START 0
 USING FILE,15    TEMPORARY REGISTER
 STM 14,3,12(13)   SAVE REGISTERS
 ST 13,SAVE   SAVE REGISTER 13 FOR PIOCS
 LR 3,15   FOR A BASE REGISTER
 USING FILE,3
 DROP 15
 L 2,0(0,1)   ADDRESS OF NTAPE
 L 2,0(0,2)  NTAPE IN GR 2
 LA 0,3   SUBTRACT 3 FROM THE DATA SET REFERENCE NUMBER TO GET LOG NO.
 SR 2,0   DSR-3 IN R2   THIS MUST BE POSITIVE OR THIS IS IN ERROR
 BC 13,ERR   IF IT IS NOT POSITIVE IS AN ILLEGAL DEVICE
 STC 2,FILEBLK+7   ASSIGN IT TO SYS-NNN
 L 2,4(0,1)   SET THE ADDRESS OF NFILES IN R2
 L 2,0(0,2)  NFILES IN GPR2
 LTR 2,2   CHECK FOR BSF OR FSF OR WTM
    BC 9,WTM  IF NFILES = 0,WRITE A TAPE MARK
 BC 2,BEGIN   IF POSITIVE THEN IT IS INDEED FSF
 NI FILECCW,X'2F'  BSF  SET CCW COMMAND TO 00101111
 LPR 2,2   SET COUNT POSITIVE
BEGIN OI FILEBLK+2,X'11'
 EXCP FILEBLK  EXECUTE THE CHANNEL PROGRAM
 WAIT FILEBLK   WAIT FOR THE FILE TO BE SKIPPED
 TM FILEBLK+2,X'20'
 BO FIN
 BCT 2,BEGIN   FOR ANOTHER FILE
 TM FILECCW,X'10' TEST FOR BACKSPACE--IF SO,FORWARD SPACE ONE AT END
 BO RETURN   IF IT WAS FSF,RETURN
 OI FILECCW,X'3F'   RESET COMMAND TO FSF FROM BSF
 EXCP FILEBLK    FORWARD SPACE ONE FILE
 WAIT FILEBLK
RETURN L 13,SAVE
 LM 14,3,12(13)  RESTORE REGISTERS
 MVI 12(13),X'FF'
 BCR 15,14
WTM MVI FILECCW,X'1F'  COMMAND FOR WTM
    EXCP FILEBLK
    WAIT FILEBLK
  MVI FILECCW,X'3F'     SO THAT ROUTINE MAY BE USED AGAIN
   B RETURN   RETURN TO PROGRAM
FILEBLK CCB SYS000,FILECCW,X'1100'
FILECCW CCW X'3F',0,X'00',1  COUNT MUST BE NONZERO
SAVE DS F
ERR EXCP PRINT
 BC 15,RETURN
PRINT CCB SYSLST,PRTCCW
PRTCCW CCW X'0B',0,X'60',1
       CCW X'01',ERRM,X'20',16
ERRM DC C'ILLEGAL FILE NO.'
FIN MVI FILECCW,X'3F'
 B RETURN
 END
```

```
// EXEC ASSEMBLY
*
*               VARIABLE FORMAT FOR /360 DOS
*
*               DIMENSION FMT(20)
* 99            FORMAT(. . . . . . . . . . . .. . . . . . . .)
* 100           FORMAT(20A4)
*               READ (1,100) FMT
*               CALL FORMAT(FMT)
*               READ (1,99)
*               . . .
*               . . .
*               END
*
*               IN THIS CASE, 99 FORMAT WILL BE REPLACED
*               BY A FORMAT CODE GENERATED FROM THE INFORMATION
*               IN ARRAY FMT.
*
*
*               THE DUMMY READ STATEMENT, WHICH IMMEDIATELY FOLLOWS
*               THE CALL FORMAT, WILL BE IGNORED AND BYPASSED.
*               IT SERVES TO GIVE THE ADDRESS OF THE FORMAT TO BE
*               CHANGED
*               ALL STANDARD FORMAT CODES ARE INTERPRETED
*               BOTH 026 AND 029 CODES ARE VALID WHERE APPLICABLE
                PRINT NOGEN
FORMAT          START    0
                USING    *,15
                STM      14,12,12(13)      SAVE THE REGISTERS
                L        12,0(0,1)       A(FMT) IN 12
*
*               GET ADDRESS OF FORMAT IN GPR 3
*               KICK UP REGISTER 14 FOR THE RETURN
*
                CLI      0(14),X'07'     TEST FOR A BCR AS A RESULT OF
                BNE      TRYBAL     THE CNOP 0,4
                LA       14,2(0,14)      ADVANCE BEYOND 0700
TRYBAL          CLI      4(14),X'45'     IS IT A BAL
                BNE      ERROR        IN ERROR IF IT IS NOT
                MVC      LOADFAD+1(3),13(14)     FORMAT ADDRESS
                OI       LOADFAD+1,X'30'    LOAD GPR 3
LOADFAD         L        3,0    A(FORMAT) IN GPR 3
                LA       10,4   GO BACK 4 BYTES
                LNR      10,10   -4
                AR       10,3     PICK UP ADDRESS OF INSTR AFTER FORMAT
                LA       14,20(0,14)     FOR ADVANCING PAST WRITE
                MVC      LOADFL+2(2),2(10)   NEXT INSTR
                XI       LOADFL+2,X'B0'  CHANGE TO REGISTER 10
LOADFL          LA       4,0   ADDRESS OF INSTR AFTER FORMAT IN GPR4
*
*               FIND A BLANK OR A (       (OR A %)
*
FIND            CLI      0(12),C'('
                BE       FOUND
                CLI      0(12),C'%'      = (
                BE       FOUND
                CLI      0(12),C' '   A BLANK IS ALLOWABLE
                BNE      ERROR    BUT NOTHING ELSE IS
                LA       12,1(0,12)   TRY NEXT BYTE
                B        FIND
```

```
FOUND     LA    9,1      LEFT PARN. COUNT                              FRMT  59
LOOP      MVI   NSW,X'01'      COUNT = 1                               FRMT  60
TRANSLAT  SR    2,2                                                    FRMT  61
          TRT   1(256,12),TABLE     GET NEXT CODE                      FRMT  62
          BC    8,ERROR     ERROR IF ALL FUNCTION BYTES ZERO           FRMT  63
          LR    12,1     ADDRESS OF ARGUMENT BYTE IN GPR 12            FRMT  64
          CLI   DOREORF,X'01' IF IN MIDDLE OF D OR E OR F FORM         FRMT  65
          BE    REQUIRED     IF SO,CHECK IT OUT                        FRMT  66
ITSOK     LH    2,ATABLE(2)     USE FUNCTION BYTE TO GET ADDRESS       FRMT  67
ATABLE    BC    15,FORMAT(2)     COMPUTED GO TO                        FRMT  68
          DC    AL2(IROUT-FORMAT)   I                                  FRMT  69
          DC    AL2(FROUT-FORMAT)   F                                  FRMT  70
          DC    AL2(RTPARN-FORMAT)  )                                  FRMT  71
          DC    AL2(LTPARN-FORMAT)  (                                  FRMT  72
          DC    AL2(HROUT-FORMAT)   H                                  FRMT  73
          DC    AL2(XROUT-FORMAT)   X                                  FRMT  74
          DC    AL2(QUOTE-FORMAT)   '                                  FRMT  75
          DC    AL2(EDROUT-FORMAT)  E,D                                FRMT  76
          DC    AL2(AROUT-FORMAT)   A                                  FRMT  77
          DC    AL2(PROUT-FORMAT)   P                                  FRMT  78
          DC    AL2(SLASH-FORMAT)   /                                  FRMT  79
          DC    AL2(TROUT-FORMAT)   T                                  FRMT  80
          DC    AL2(MINUS-FORMAT)   -                                  FRMT  81
          DC    AL2(DECPT-FORMAT)   .                                  FRMT  82
          DC    AL2(NUMBER-FORMAT)  0 THROUGH 9                        FRMT  83
*                                                                      FRMT  84
*         FORMAT ROUTINES                                              FRMT  85
*                                                                      FRMT  86
LTPARN    MVI   1(3),LPCODE                                            FRMT  87
          LA    9,1(0,9)   INCREASE LEFT PARN. COUNT BY ONE            FRMT  88
PUTT      MVC   2(1,3),NSW   MOVE IN A COUNT IF THERE                  FRMT  89
          LA    3,2(0,3)   INCREASE GPR 3 BY 2                         FRMT  90
          B     LOOP     GET NEXT CODE OF FORMAT                       FRMT  91
RTPARN    MVI   1(3),RPCODE                                            FRMT  92
          LA    3,1(0,3)         INCREMENT GPR 3                       FRMT  93
          BCT   9,LOOP      IF ZERO,NEXT INSTR,DONE,RETURN             FRMT  94
          MVI   0(3),EOFORMAT   END OF FORMAT CHARACTER                FRMT  95
          CR  3,4                                                      FRMT  96
          BNL   ERROR                                                  FRMT  97
          LM    2,12,28(13)    RETURN LOAD REGISTERS                   FRMT  98
          MVI   12(13),X'FF'      SIGNAL ALL DONE                      FRMT  99
          L     15,=V(IJTACOM)   FORTRAN REQUIRES THIS                 FRMT 100
          BR    14     OF COURSE SKIPPING WRITE                        FRMT 101
TROUT     MVI   1(3),TCODE                                             FRMT 102
          B     REPS     TO FILL IN THE NUMBER                         FRMT 103
MINUS     OI    NSW,X'80'     SET NSW MINUS FOR - P CODES              FRMT 104
          B     TRANSLAT                                               FRMT 105
SLASH     MVI   1(3),SLCODE                                            FRMT 106
          LA    3,1(0,3)   INCREMENT GR 3                              FRMT 107
          B     LOOP                                                   FRMT 108
PROUT     MVI   1(3),PCODE                                             FRMT 109
          B     PUTT   PUT IN NUMBER AND INCEMENT GPR 3                FRMT 110
XROUT     MVI   1(3),XCODE                                             FRMT 111
          B     PUTT   PUT IN NUMBER AND INCREMENT GPR 3               FRMT 112
IROUT     BAL   11,DUPCHK      CHECK FOR DUPLICATION                   FRMT 113
          MVI   1(3),ICODE                                             FRMT 114
          B     REPS     FOR LENGTH INSERTION                          FRMT 115
AROUT     BAL   11,DUPCHK     CHECK DUPLICATION FACTOR                 FRMT 116
          MVI   1(3),ACODE                                             FRMT 117
          B     REPS                                                   FRMT 118
FROUT     MVI   DEC,FCODE                                              FRMT 119
```

```
         B       COMONRTN                                               FRMT 120
EDROUT   MVI     DEC,EDCODE                                             FRMT 121
COMONRTN BAL     11,DUPCHK                                              FRMT 122
         MVC     1(1,3),DEC   MOVE D-E OR F CODE IN                     FRMT 123
         MVI     DOREORF,X'01'    REQUIRE A .                           FRMT 124
         B       REPS                                                   FRMT 125
DECPT    CLI     DOREORF,X'01'                                          FRMT 126
         BNE     ERROR                                                  FRMT 127
         MVI     DOREORF,X'00'                                          FRMT 128
         BCTR    3,0        DECREMENT BY ONE                            FRMT 129
         B       REPS        PUT IN NO. OF SIGNIFICANT FIGURES          FRMT 130
HMOVE    MVC     3(1,3),1(12)   MOVE N CHARACTERS                       FRMT 131
HROUT    MVI     1(3),HCODE                                             FRMT 132
         MVC     2(1,3),NSW                                             FRMT 133
         IC      10,NSW                                                 FRMT 134
         BCTR    10,0   REDUCE COUNT BY ONE                             FRMT 135
         EX     10,HMOVE   10 GIVES THE LENGTH                          FRMT 136
         LA      3,3(10,3)                                              FRMT 137
         LA      12,1(10,12)                                            FRMT 138
         B       LOOP                                                   FRMT 139
QUOTE    SR      10,10                                                  FRMT 140
         MVI     1(3),HCODE                                             FRMT 141
TEST1    CLI     1(12),C''''    TEST FOR A QUOTE                        FRMT 142
         BE      TEST2      LOOK FOR A SECOND QUOTE                     FRMT 143
         CLI     1(12),C'@'      ACCEPT 026 QUOTE                       FRMT 144
         BE      TEST2                                                  FRMT 145
ADDQ     LA      10,1(0,10)     INCREASE COUNT BY ONE                   FRMT 146
         IC     8,1(0,12)   STORE CHARACTER BY USING                    FRMT 147
         STC     8,2(10,3)    INDIRECT ADDRESS PROCEDURE                FRMT 148
         LA      12,1(0,12)    SIMPLY INCREMENT GPR 12                  FRMT 149
         B       TEST1     GET NEXT CHARACER                            FRMT 150
TEST2    LA      12,1(0,12)                                             FRMT 151
         CLI     1(12),C''''                                            FRMT 152
         BE      ADDQ    TWO QUOTES TOGETHER COUNT AS 1                 FRMT 153
         CLI     1(12),C'@'                                             FRMT 154
         BE      ADDQ                                                   FRMT 155
         STC     10,2(0,3)    STORE COUNT IN COUNT FIELD                FRMT 156
         LA      3,2(10,3)     ADJUST REGISTER 3                        FRMT 157
         B       LOOP      GET NEXT FORMAT CODE                         FRMT 158
NUMBER   LA      6,TRANSLAT       FOR RETURN                            FRMT 159
NUMBERIN SR      10,10    CLEAR GPR 10                                  FRMT 160
TUNDERM  TM      0(12),X'F0'                                            FRMT 161
         BC      14,DONE     IF NOT A NUMBER WITH HEX F                 FRMT 162
         SLL     10,8       SHIFT GPR 10 8 TO THE LEFT                  FRMT 163
         IC      10,0(0,12)       ADD INTEGER CHARACTER                 FRMT 164
         LTR     10,10   NEGATIVE IF 4 DIGITS ARE IN 10                 FRMT 165
         BM      ERROR   IF NEGATIVE, THERE IS AN ERROR                 FRMT 166
         LA      12,1(0,12)     TRY NEXT CHARACTER                      FRMT 167
         B       TUNDERM                                                FRMT 168
DONE     BCTR    12,0    CUT IT BACK                                    FRMT 169
         ST      10,DEC                                                 FRMT 170
         NI      DEC+3,X'CF'    SET SIGN PLUS IN EBCDIC CODE            FRMT 171
         PACK    DEC1(8),DEC(4)      PACK THE ZONED STUFF               FRMT 172
         CVB     10,DEC1     CONVERT TO BINARY IN GPR10                 FRMT 173
         STC     10,DEC      STORE ONLY ONE BYTE                        FRMT 174
         NI      NSW,X'80'      SAVE ONLY SIGN OF NSW                   FRMT 175
         OC      NSW(1),DEC   ALLOW FOR - ON P CODES                    FRMT 176
         BR      6   TO TRANSLAT OR REPTURN                             FRMT 177
REPS     LA      12,1(0,12)    PUT IN STUFF AFTER LETTER                FRMT 178
*        E.G. GET THE 4 OF A 3A4 FORMAT                                 FRMT 179
         TM      0(12),X'F0'                                            FRMT 180
```

```
                  BC        14,ERROR
                  BAL       6,NUMBERIN
REPTURN           MVC       2(1,3),NSW     FILL IN THE COUNT
                  LA        3,2(0,3)
                  B         LOOP
DUPCHK            CLI       NSW,X'01'    CHECK IF COUNT IS 1
                  BCR       8,11       RETURN IF IT IS 1
                  MVI       1(3),DUPCODE
                  MVC       2(1,3),NSW     MOVE IN THE COUNT
                  LA        3,2(0,3)
                  BR        11       RETURN
*
*                 AT THIS POINT, THE JOB WILL BE CANCELLED SO NO MORE
*                 REGISTERS NEED BE SAVED
*                 USE FORTRAN IOCS TO WRITE AN ERROR MESSAGE
*                 THEN CANCEL THE JOB
*
ERROR             LR        4,15       FOR INDEXING OF FORMAT ADDRESS
                  CNOP        0,4     ADJUST FOR FORTAN I/O
                  L         15,=V(IJTACOM)      =V(IBCOM)
                  BAL      14,X'14'(0,15)     USE FIOCS
                  DC        XL1'0'      UNIT IS INTEGER
                  DC        AL3(3)      WRITE ON SYSLST
                  DC        XL1'84'         FORMAT
                  DC        AL3(X'004000'+ERRADDR-FORMAT)
                  BAL       14,X'28'(0,15)       FORMAT LIST IS DONE
                  CANCEL
ERRADDR           DC        A(ERRFORMT)
ERRFORMT          DC        X'001818'      H,24 CHARACTERS
                  DC         C' ERROR IN FORMAT'
                  DC        C' ROUTINE'
                  DC         X'34'
*
*
*                 FOR DOS FORTRAN, THE CODES ARE
*
LPCODE            EQU       X'04'
DUPCODE           EQU       X'08'
RPCODE            EQU       X'0C'
PCODE             EQU       X'10'
XCODE             EQU       X'14'
HCODE             EQU       X'18'
TCODE             EQU       X'1C'
ACODE             EQU       X'20'
ICODE             EQU       X'24'
FCODE             EQU       X'28'
EDCODE            EQU       X'2C'
SLCODE            EQU       X'30'
EOFORMAT          EQU       X'34'
*
*
REQUIRED          STC       2,DEC
                  CLC       DEC(1),TABLE+C'.'   CHECK FOR DECIMAL POINT
                  BE        ITSOK
                  B         ERROR
*                 TRANSLATE AND TEST TABLE
*
TABLE             DC        75X'00'
                  DC        X'1E'    .
                  DC        X'08'    ) = )
```

```
         DC    X'0A'     (                                FRMT 241
         DC    15X'00'                                    FRMT 242
         DC    X'08'     )                                FRMT 243
         DC    2X'00'                                     FRMT 244
         DC    X'1C'     -                                FRMT 245
         DC    X'18'     /                                FRMT 246
         DC    10X'00'                                    FRMT 247
         DC    X'0A'     ( = (                            FRMT 248
         DC    15X'00'                                    FRMT 249
         DC    X'10'     ' = '                            FRMT 250
         DC    X'10'     '                                FRMT 251
         DC    67X'00'                                    FRMT 252
         DC    X'14'     A                                FRMT 253
         DC    2X'00'                                     FRMT 254
         DC    X'12'     D                                FRMT 255
         DC    X'12'     E                                FRMT 256
         DC    X'06'     F                                FRMT 257
         DC    X'00'                                      FRMT 258
         DC    X'0C'     H                                FRMT 259
         DC    X'04'     I                                FRMT 260
         DC    13X'00'                                    FRMT 261
         DC    X'16'     P                                FRMT 262
         DC    11X'00'                                    FRMT 263
         DC    X'1A'     T                                FRMT 264
         DC    3X'00'                                     FRMT 265
         DC    X'0E'     X                                FRMT 266
         DC    8X'00'                                     FRMT 267
         DC    10X'20'     0 THROUGH 9                    FRMT 268
         DC    6X'00'                                     FRMT 269
DEC1     DS    D                                          FRMT 270
DEC      DS    F                                          FRMT 271
DOREORF  DC    X'00'      SET TO 0 ORIGINAL               FRMT 272
NSW      DS    CL1                                        FRMT 273
         END   FORMAT                                     FRMT 274
```

```
// EXEC ASSEMBLY
*
*          DOUBLE PRECISION MATRIX INVERSION
*     OPTIMIZED FOR SYSTEM /360 BY USING 16 GENERAL PURPOSE REGISTERS
*     AND 4 FLOATING POINT REGISTERS.
*
*        CALL DINVRT(X,N,M,S1,S2,DET)
*
*        X(M,M) IS DOUBLE PRECISION.  INVERT THE SUBMATRIX X(N,N).
*        S1(N),S2(N) ARE SINGLE PRECISION WORK AREAS.
*        DET IS DETERMINANT.
*
DINVRT START 0
          USING      *,15
          STM        14,12,12(13)        SAVE REGISTERS
          ST         13,SALVE            SAVE REGISTER 13
          LR         14,1                1 IN 14
          STD        0,ST0
          STD        2,ST2
          STD        4,ST4
          STD        6,ST6
          L          5,8(0,1)
          L          5,0(0,5)  M IN GPR 5
          LR         7,5  M IN GPR 7
          L          9,4(0,1)
          L          9,0(0,9)  N IN GPR9
          BCTR       9,0   DECREMENT BY ONE
          SLL        9,3  8*(N-1)  DOUBLE PRECISION
          MR         6,9  4*M*(N-1) IN GPR7
          LR         6,5  M IN GPR6
          SLL        6,3                 4*M IN GPR6
          LA         8,8                 8 IN GPR8
          L          5,0(0,1)
          LR         3,5                 A(K,K) INITIALLY A(1,1)
          LD         6,ONEPT
          LDR        2,6                 DETERMINANT INITIALLY 1.
          AR         7,5                 8*M*(N-1)+A(MATRIX) IN GPR7
          L          12,12(0,1)
          L          13,16(0,1)
          SR         1,1                 4*K=0
*         INITIALIZATION IS DONE-START INVERSION
*         FIND THE BIGGEST ELEMENT
SEARCH    SDR        0,0   BIGGEST VALUE SET TO 0.
          LR         4,3                 START SECOND SUBSCRIPT HERE
*         GET SAME AS THE PRESENT KA VALUE
JBIG      LR         2,1                 ALSO SET FIRST SUBSCRIPT
*         TO FIRST SUBSCRIPT FROM K
IBIG      LD         4,0(2,4)            A(I,J) I FP 4
*         IN PRECEDING INSTRUCTION I,J NOT K
          LPDR       4,4                 TAKE ABSOLUTE VALUE
          CDR        4,0                 IS IS BIGGER THAN PREVIOUS
          BC         12,ENDBIG           NO CONTINUE ON
          LDR        0,4                 YES--SAVE THE NEW ONE
          LR         10,2                SAVE IP*8
          LR         11,4                SAVE JP*8*M+A(MATRIX)
ENDBIG    BXLE       2,8,IBIG           INCREASE I
          BXLE       4,6,JBIG            INCREASE J
*         NOW BIGGEST ELEMENT IS IN FPR0,ADDRESS
*         IS SUM OF REGISTERS 10 AND 11
*         INTERCHANGE COLUMNS
*         A(I,JP) AND A(I,K) UNLESS JP=K
```

```
         CR     3,11             IF JP=K,
         BC     8,COLDON         BRANCH
         SR     2,2              I=0
STCOL    LD     0,0(2,11)        A(I,J)
         LD     4,0(2,3)         A(I,K)
         LCDR   4,4
         STD    4,0(2,11)
         STD    0,0(2,3)
         BXLE   2,8,STCOL
*        INTERCHANGE ROWS
*        A(IP,J),A(K,J) UNLESS IP K
COLDON   CR     1,10             IN THAT CASE
         BC     8,ENDROW         BRANCH
         LR     4,5
STROW    LD     0,0(10,4)        A(IP,J)
         LD     4,0(1,4)         A(K,J)
         LCDR   4,4              -A(K,J)
         STD    4,0(10,4)
         STD    0,0(1,4)
         BXLE   4,6,STROW
*        DIVIDE COLUMN BY MINUS PIVOT
*        A(I,K)=A(I,K)/-A(K,K) EXCEPT FOR I=K
ENDROW   LD     0,0(1,3)
         LCDR   0,0
         SR     2,2
LOOP     CR     1,2
         BC     8,NOTNOW
         LD     4,0(2,3)
         DDR    4,0
         STD    4,0(2,3)
NOTNOW   BXLE   2,8,LOOP
*        REDUCE MATRIX
*        A(I,J)=A(I,J)+A(K,J)*A(I,K) UNLESS
*        EITHER I OR J SHOULD EQUAL K
         LR     4,5              J ADDRESS OF MATRIX A
JRED     SR     2,2              I=0
IRED     CR     2,1              DOES I=5
         BC     8,ENDRED         YES--BRANCH
         CR     3,4  DOES J=K
         BC     8,ENDRED         YES--BRANCH
         LD     4,0(2,3)
         MD     4,0(1,4)
         AD     4,0(2,4)
         STD    4,0(2,4)  NEW A(I,J)
ENDRED   BXLE   2,8,IRED
         BXLE   4,6,JRED
*        DIVIDE ROW BY PIVOT
*        A(K,J)=A(,K,J) J NOT EQUAL TO K
         LCDR   0,0              A(K8K) IN FPRO
         LR     4,5              ADDRESS OF A
DIVROW   CR     3,4              DOES J=K
         BC     8,DIVEND         IF YES,BRANCH
         LD     4,0(1,4)
         DDR    4,0
         STD    4,0(1,4)
DIVEND   BXLE   4,6,DIVROW
*        MULTIPLY DETERMINANT BY PIVOT
         SRL    1,1   SHIFT LOGICAL RIGHT 1 TO DIVIDE BY 2
         ST     10,0(1,12)  SAVE IP(K)
         ST     11,0(1,13)  SAVE JP(K)
         SLL    1,1              SHIFT BACK AGAIN
```

```
         MDR       2,0
         LDR       4,6  1.00 IN FPR4
         DDR       4,0   1.00/A(K,K) IN PR4
         STD       4,0(1,3)
         LA        3,0(6,3)
         BXLE      1,8,SEARCH
*        K NOW HAS 8*N
         SR        3,6
         SR        1,8
SKIPB    SR        3,6
         SR        1,8
         BC        4,RETURN  IF K NEGATIVE,DONE
         SRL       1,1
         L         11,0(1,13)
         SR        11,5
         SR        10,10
         DR        10,6
         SLL       11,3
         LR        0,11
         L         11,0(1,12)
         SLL       1,1
         SRL       11,3
         MR        10,6
         AR        11,5
         LR        10,0
         SR        2,2
         CR        11,3
         BC        8,SKIPA
ENDA     LD        0,0(2,3)
         LD        4,0(2,11)
         LCDR      4,4
         STD       4,0(2,3)
         STD       0,0(2,11)
         BXLE      2,8,ENDA
SKIPA    LR        4,5
         CR        10,1
         BC        8,SKIPB
ENDB     LD        0,0(1,4)
         LD        4,0(10,4)
         LCDR      4,4
         STD       4,0(1,4)
         STD       0,0(10,4)
         BXLE      4,6,ENDB
         BC        15,SKIPB
RETURN   L         1,20(0,14)
         STD       2,0(0,1)
         L         13,SALVE
         LM        14,12,12(13)
         LD        0,ST0
         LD        2,ST2
         LD        4,ST4
         LD        6,ST6
         MVI       12(13),X'FF'
         BCR       15,14   RETURN TO CALLING PROGRAM
SALVE    DS        F
ONEPT    DC        D'1.'
ST0      DS        D
ST2      DS        D
ST4      DS        D
ST6      DS        D
 END DINVRT
```

```
// EXEC FORTRAN
C                                                                           FSRT   0
      SUBROUTINE FSORTA(IARRAY,N)                                           FSRT   1
C                                                                           FSRT   2
C     SORT INTO ASCENDING ORDER THE FIRST N ELEMENTS OF THE REAL ARRAY.     FSRT   3
C                                                                           FSRT   4
      DIMENSION IARRAY(1)                                                   FSRT   5
      REAL IARRAY                                                           FSRT   6
      REAL IG                                                               FSRT   7
C     SORTS SING. PREC. F.P. NUMBERS IN ASCENDING ORDER                     FSRT   8
      M=N                                                                   FSRT   9
10    M=M/2                                                                 FSRT  10
      IF(M) 99,99,1                                                         FSRT  11
1     K=N-M                                                                 FSRT  12
      J=1                                                                   FSRT  13
9     I=J                                                                   FSRT  14
    7 IM=I+M                                                                FSRT  15
      IF(IARRAY(I)-IARRAY(IM)) 8,8,4                                        FSRT  16
4     IG=IARRAY(I)                                                          FSRT  17
      IARRAY(I)=IARRAY(IM)                                                  FSRT  18
      IARRAY(IM)=IG                                                         FSRT  19
      I=I-M                                                                 FSRT  20
      IF(I-1) 8,7,7                                                         FSRT  21
8     J=J+1                                                                 FSRT  22
      IF(J-K) 9,9,10                                                        FSRT  23
99    RETURN                                                                FSRT  24
      END                                                                   FSRT  25
```

```
// EXEC FORTRAN
C                                                                           ISRT   0
      SUBROUTINE ISORTA(IARRAY,N)                                           ISRT   1
      DIMENSION IARRAY(1)                                                   ISRT   2
C     SORT THE FIRST N ELEMENTS OF THE INTEGER ARRAY IARRAY IN ASCENDINGISRT   3
      M=N                                                                   ISRT   4
10    M=M/2                                                                 ISRT   5
      IF(M) 99,99,1                                                         ISRT   6
1     K=N-M                                                                 ISRT   7
      J=1                                                                   ISRT   8
9     I=J                                                                   ISRT   9
    7 IM=I+M                                                                ISRT  10
      IF(IARRAY(I)-IARRAY(IM)) 8,8,4                                        ISRT  11
4     IG=IARRAY(I)                                                          ISRT  12
      IARRAY(I)=IARRAY(IM)                                                  ISRT  13
      IARRAY(IM)=IG                                                         ISRT  14
      I=I-M                                                                 ISRT  15
      IF(I-1) 8,7,7                                                         ISRT  16
8     J=J+1                                                                 ISRT  17
      IF(J-K) 9,9,10                                                        ISRT  18
99    RETURN                                                                ISRT  19
      END                                                                   ISRT  20
```

```
// EXEC TREND
  10    9    8    7 100    1              8   14    1                              DATA    0
PC112  100 RANDOM POINTS. KANSAS CONTR.12, 1967, P 45-46                           DATA    1
THE COORDINATES U AND V WERE FIXED BY A RANDOM NUMBER ROUTINE                      DATA    2
BUT THE VALUES OF Y ARE INTERPOLATED FROM A GIVEN PATTERN.                         DATA    3
(T21,A2,T1,F3.3,F7.3,F7.2)                                                         DATA    4
(' ',14F8.2)                                                                       DATA    5
(' ',2X,A2,2F7.3,3F10.2)                                                           DATA    6
(' ',2X,'ID',5X,'U',6X,'V',8X,'Y',8X,'PRED',6X,'RESID')                            DATA    7
99                                                                                 DATA    8
99                                                                                 DATA    9
 1                                                                                 DATA   10
99                                                                                 DATA   11
-1        10000  0500 00000 10000  0500                                            DATA   12
99                                                                                 DATA   13
99                                                                                 DATA   14
 0        10000  0500 00000 10000  0500                                            DATA   15
                100                                                                DATA   16
                900                                                                DATA   17
 57      51     270    0                                                           DATA   18
 41     274     400    1                                                           DATA   19
 85     339     470    2                                                           DATA   20
 61     343     470    3                                                           DATA   21
 34     385     490    4                                                           DATA   22
 28     497     540    5                                                           DATA   23
 12     923     280    6                                                           DATA   24
169      59     290    7                                                           DATA   25
125     178     350    8                                                           DATA   26
182     270     440    9                                                           DATA   27
116     375     530   10                                                           DATA   28
159     492     670   11                                                           DATA   29
150     553     680   12                                                           DATA   30
132     691     580   13                                                           DATA   31
171     745     500   14                                                           DATA   32
132     830     410   15                                                           DATA   33
136     891     380   16                                                           DATA   34
177     893     330   17                                                           DATA   35
165     949     280   18                                                           DATA   36
235     962     220   19                                                           DATA   37
278     905     270   20                                                           DATA   38
218     640     630   21                                                           DATA   39
241     633     640   22                                                           DATA   40
209     351     580   23                                                           DATA   41
293     189     490   24                                                           DATA   42
274     169     450   25                                                           DATA   43
232      89     360   26                                                           DATA   44
350      30     360   27                                                           DATA   45
353     144     480   28                                                           DATA   46
339     229     550   29                                                           DATA   47
339     431     790   30                                                           DATA   48
394     571     680   31                                                           DATA   49
325     602     670   32                                                           DATA   50
320     649     580   33                                                           DATA   51
365     718     470   34                                                           DATA   52
385     723     450   35                                                           DATA   53
301     745     450   36                                                           DATA   54
378     755     430   37                                                           DATA   55
364     805     350   38                                                           DATA   56
331     848     330   39                                                           DATA   57
373     899     330   40                                                           DATA   58
```

352	998	330	41	DATA 59
417	965	410	42	DATA 60
419	933	400	43	DATA 61
458	794	420	44	DATA 62
403	691	480	45	DATA 63
407	390	770	46	DATA 64
488	388	800	47	DATA 65
446	276	650	48	DATA 66
442	275	640	49	DATA 67
423	157	520	50	DATA 68
518	33	390	51	DATA 69
593	264	640	52	DATA 70
501	473	700	53	DATA 71
552	477	680	54	DATA 72
523	552	600	55	DATA 73
536	678	510	56	DATA 74
553	786	490	57	DATA 75
512	971	500	58	DATA 76
651	993	470	59	DATA 77
674	937	460	60	DATA 78
660	926	470	61	DATA 79
666	758	570	62	DATA 80
690	731	590	63	DATA 81
641	726	580	64	DATA 82
605	433	650	65	DATA 83
614	282	650	66	DATA 84
623	160	540	67	DATA 85
696	111	490	68	DATA 86
627	81	460	69	DATA 87
694	36	410	70	DATA 88
719	84	460	71	DATA 89
736	215	560	72	DATA 90
784	287	510	73	DATA 91
723	311	500	74	DATA 92
716	421	510	75	DATA 93
794	489	480	76	DATA 94
746	852	420	77	DATA 95
757	902	390	78	DATA 96
763	976	410	79	DATA 97
830	990	340	80	DATA 98
806	949	300	81	DATA 99
842	921	270	82	DATA 100
853	835	350	83	DATA 101
897	771	430	84	DATA 102
876	663	520	85	DATA 103
865	578	610	86	DATA 104
806	512	440	87	DATA 105
826	461	450	88	DATA 106
833	52	420	89	DATA 107
977	213	340	90	DATA 108
959	406	330	91	DATA 109
914	457	400	92	DATA 110
906	531	500	93	DATA 111
961	536	460	94	DATA 112
936	541	490	95	DATA 113
955	612	490	96	DATA 114
925	642	560	97	DATA 115
964	764	480	98	DATA 116
951	831	340	99	DATA 117

```
// EXEC DBMCI1
 7 8 9                                                                        PLOT   0
        0.0   1.0   0.0   1.0  10.1  10.1   0.07 3 110                        PLOT   1
 0 1                                                                          PLOT   2
  7.500  0.750  1.000 1                                                       PLOT   3
// EXEC DBMCI1
 7 8 9                                                                        PLOT   4
        0.0   1.0   0.0   1.0  10.1  10.1   0.07 3-110                        PLOT   5
 0 1                                                                          PLOT   6
  7.500  0.750  1.000 1                                                       PLOT   7
// EXEC DBMCI1
 7 8 9                                                                        PLOT   8
        0.0   1.0   0.0   1.0  10.1  10.1   0.07 0  10                        PLOT   9
 0 1                                                                          PLOT  10
  7.500  0.750  1.000 1                                                       PLOT  11
// EXEC DBMCI1
 7 8 9                                                                        PLOT  12
        0.0   1.0   0.0   1.0  10.1  10.1   0.07 3 110                        PLOT  13
 3 2                                                                          PLOT  14
  7.500         1.000                                                         PLOT  15
// EXEC DBMCI1
 7 8 9                                                                        PLOT  16
        0.0   1.0   0.0   1.0  10.1  10.1   0.07 3 110                        PLOT  17
 4 3                                                                          PLOT  18
  7.500  0.750  0.100                                                         PLOT  19


// EXEC DBMCID
 7 8 9 100                                                                    CONT   0
        0.0   1.0   0.0   1.0  10.1  10.1   0.07 3-1                          CONT   1
PC112    CONTOUR MAP OF 100 RANDOM POINTS                                     CONT   2
  7.500  0.750  1.000                                                         CONT   3
(F3.3,F7.3,F7.2)                                                              CONT   4
 57      51     270     0                                                     CONT   5
 41     274     400     1                                                     CONT   6
 85     339     470     2                                                     CONT   7
 61     343     470     3                                                     CONT   8
 34     385     490     4                                                     CONT   9
 28     497     540     5                                                     CONT  10
 12     923     280     6                                                     CONT  11
169      59     290     7                                                     CONT  12
125     178     350     8                                                     CONT  13
182     270     440     9                                                     CONT  14
116     375     530    10                                                     CONT  15
159     492     670    11                                                     CONT  16
150     553     680    12                                                     CONT  17
132     691     580    13                                                     CONT  18
171     745     500    14                                                     CONT  19
132     830     410    15                                                     CONT  20
136     891     380    16                                                     CONT  21
 .       .       .      .                                                       .    .
 .       .       .      .                                                       .    .
 .       .       .      .                                                       .    .
906     531     500    93                                                     CONT  98
961     536     460    94                                                     CONT  99
936     541     490    95                                                     CONT 100
955     612     490    96                                                     CONT 101
925     642     560    97                                                     CONT 102
964     764     480    98                                                     CONT 103
951     831     340    99                                                     CONT 104
```

KANSAS GEOLOGICAL SURVEY COMPUTER PROGRAM
THE UNIVERSITY OF KANSAS, LAWRENCE

PROGRAM ABSTRACT

Title (If subroutine state in title):

Computer Programs for Automatic Contouring

Date: 15 March 1968

Author, organization: Donald B. McIntyre, David D. Pollard, and Roger Smith

Pomona College

Direct inquiries to:

Name: Donald B. McIntyre

Address: Department of Geology
Pomona College
Claremont, California 91711

Purpose/description: Plot contoured maps of irregularly spaced data, trend surfaces, and residuals.

Mathematical method: Trend surfaces are least-square polynomials. Contouring is based on a square grid, the points of which are determined by second-degree surfaces fitted to the nearest stations.

Restrictions, range: Limit on trend surface is eighth-degree with cross products (45 coefficients).

Computer manufacturer: IBM

Model: System 360/40

Programming language: FORTRAN and Assembly

Memory required: 32K bytes K

Approximate running time: Eighth-degree trend is fitted to 100 points and contoured in about 8 minutes.

Special peripheral equipment required: Three tape drives (2400). One disc drive. Calcomp 565 plotter on-line.

Remarks (special compilers or operating systems, required word lengths, number of successful runs, other machine versions, additional information useful for operation or modification of program) The program as listed runs under Disc Operating System Version 2, Release 11, with 6K Supervisor. The program was written originally for an IBM 7094 with off-line plotting on Calcomp digital plotters, and EAI and Benson-Lehner analog plotters. It has been used for production work for several years.

COMPUTER CONTRIBUTIONS

Kansas Geological Survey
University of Kansas
Lawrence, Kansas

Computer Contribution

1. Mathematical simulation of marine sedimentation with IBM 7090/7094 computers, by J.W. Harbaugh, 1966 . . . . $1.00
2. A generalized two-dimensional regression procedure, by J.R. Dempsey, 1966 . . . . $0.50
3. FORTRAN IV and MAP program for computation and plotting of trend surfaces for degrees 1 through 6, by Mont O'Leary, R.H. Lippert, and O.T. Spitz, 1966 . . . . $0.75
4. FORTRAN II program for multivariate discriminant analysis using an IBM 1620 computer, by J.C. Davis and R.J. Sampson, 1966 . . . . $0.50
5. FORTRAN IV program using double Fourier series for surface fitting of irregularly spaced data, by W.R. James, 1966 . . . . $0.75
6. FORTRAN IV program for estimation of cladistic relationships using the IBM 7040, by R.L. Bartcher, 1966 . . . . $1.00
7. Computer applications in the earth sciences: Colloquium on classification procedures, edited by D.F. Merriam, 1966 . . . . $1.00
8. Prediction of the performance of a solution gas drive reservoir by Muskat's Equation, by Apolonio Baca, 1967. . . . . $1.00
9. FORTRAN IV program for mathematical simulation of marine sedimentation with IBM 7040 or 7094 computers, by J.W. Harbaugh and W.J. Wahlstedt, 1967 . . . . $1.00
10. Three-dimensional response surface program in FORTRAN II for the IBM 1620 computer, by R.J. Sampson and J.C. Davis, 1967 . . . . $0.75
11. FORTRAN IV program for vector trend analyses of directional data, by W.T. Fox, 1967 . . . . $1.00
12. Computer applications in the earth sciences: Colloquium on trend analysis, edited by D.F. Merriam and N.C. Cocke, 1967 . . . . $1.00
13. FORTRAN IV computer programs for Markov chain experiments in geology, by W.C. Krumbein, 1967 . . . . $1.00
14. FORTRAN IV programs to determine surface roughness in topography for the CDC 3400 computer, by R.D. Hobson, 1967 . . . . $1.00
15. FORTRAN II program for progressive linear fit of surfaces on a quadratic base using an IBM 1620 computer, by A.J. Cole, C. Jordan, and D.F. Merriam, 1967 . . . . $1.00
16. FORTRAN IV program for the GE 625 to compute the power spectrum of geological surfaces, by J.E. Esler and F.W. Preston, 1967 . . . . $0.75
17. FORTRAN IV program for Q-mode cluster analysis of nonquantitative data using IBM 7090/7094 computers, by G.F. Bonham-Carter, 1967 . . . . $1.00
18. Computer applications in the earth sciences: Colloquium on time-series analysis, D.F. Merriam, editor, 1967 . . . . $1.00
19. FORTRAN II time-trend package for the IBM 1620 computer, by J.C. Davis and R.J. Sampson, 1967 . . . . $1.00
20. Computer programs for multivariate analysis in geology, D.F. Merriam, editor, 1968 . . . . . $1.00
21. FORTRAN IV program for computation and display of principal components, by W.J. Wahlstedt and J.C. Davis, 1968 . . . . $1.00
22. Computer applications in the earth sciences: Colloquium on simulation, D.F. Merriam and N.C. Cocke, editors, 1968 . . . . $1.00
23. Computer programs for automatic contouring, by D.B. McIntyre, D.D. Pollard, and R. Smith, 1968 . . . . $1.50