UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

PCCONTUR VERSION 1.0

a microcomputer general purpose contouring program

by

Richard H. Godson[1], Robert E. Bracken[1], and Michael W. Webring[1]

Open-File Report 88-593A-D

1988

88 -	A	Documentation (Paper Copy)
88 -	B	Executable PCCONTUR and test data Disk
88 -	C	Source code PCCONTUR Disk
88 -	D	Plot system source code and library Disk

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Use of trade names in this report is for descriptive purposes only and does not imply endorsement by the USGS. Although this program has been extensively

[1] U.S. Geological Survey, Denver, Colorado 80225

Table of Contents

Introduction

The basic contouring subroutines of this program were developed by Gerald Ian Evenden (Evenden, 1975) of the U.S. Geological Survey in the mid-70's on a DEC-10 computer. The authors of this report have since converted the program first to a Honeywell Multics 68/80 computer (Godson and Webring, 1982), then to a DEC VAX computer, and recently to IBM PC and compatible microcomputers. During these conversions several new features have been added to the original program. Source, executable code, test data, and plot system are provided with this report on three 5-1/4 inch 360K diskettes.

Software

The source code for the contour program was compiled with Microsoft Fortran v. 4.1. The program uses a device independent plotting system (Evenden, 1975) that has been modified to run on PC's. Most of the accompanying plot library subroutines were also compiled using the above-mentioned compiler. The graphics subroutines that write to a monitor were compiled using Microsoft Quick C v. 1.0. Two subroutines were written in assembler language, one controls the input/output to a serial port and the other controls the monitor mode, and they were compiled using Mircrosoft Assember v 4.0. The executable code was produced using Microsoft linker v. 5.01.2 with the Quick C graphics library and the large Fortran and C memory modules with in-line floating-point instructions (LLIBFOR7.LIB and LLIBC7.LIB). See Appendix B for instructions on compiling and linking the program. Therefore, the executable code will work only on machines that have a math coprocessor (8087 or 80287). For those users not possessing a math coprocessor, the source code can be compiled and linked using one of the different libraries supplied by Microsoft with their compilers or with a different vendor's software.

Hardware

The program has been tested using MS DOS v.3.- on an IBM XT, AT and several IBM-compatible computers running at clock speeds ranging from 4.77 mhz to 12 mhz with either an 8087 or 80287 coprocessor. Minimum memory size is unknown as it has not been tested with machines having less than 512K bytes of memory. The program has worked with several types of CGA and EGA monitors and also on a Matsubishi multiscan with a graphics board that supports the VGA mode of 640 by 480 with 16 colors. The only hard copy device supported is a Hewlett-Packard (HP) plotter or compatible that uses the Hewlett-Packard Graphics Language (HP-GL).

Description

PCCONTUR is a microcomputer program for contouring two dimensional data that are in a standard grid format described in appendix A. The program does not have gridding algorithms included so random data must be transformed to a grid format using a external program such as MINC (Webring, 1981). The standard format allows data to be ordered in several different ways: different x- and y-grid spacings, varying distances between grid positions in either the x- or y- direction and data arranged to form quadrilaterals whose interior angles do not exceed 180 degrees. The program recognizes grid values in excess of 1.0E37 as uncontourable, and thus irregularly shaped data sets may be contoured.

The program was written to handle large data sets commonly encountered in geophysical data processing. Input data are segmented for contouring into row tiers, the number of rows in a tier is approximately defined by the formula: 4 *(16000-number of columns)/(5* number of columns + 4). The present implementation allows contouring of an infinite number of rows containing up to approximately 4500 columns. The number of columns can be increased by changing two variables in the program and by using the [HUGE] attribute (for Microsoft Fortran) in appropriate places.

Linear interpolation is used to obtain the coordinates of each contour line as it passes through the side of grid cells. This method produces a very accurate representation of the gridded data but in some cases produces angular contour lines; the amount of angularity is dependent on the grid interval and the scale at which the contours are to be plotted. Smooth contours can be produced by using a B-spline option in the program.

There are several options available with the program, e.g., labelling of axes, printing of titles, placement of text anywhere on the plot, drawing of border lines, drawing of lines or randomly spaced data from an external file, plotting of latitude and longitude tick marks, various means of eliminating contours in areas of high gradient, variable line widths, variable colors for contour lines, hachuring of lows or plotting of L and H symbols, variable spacing of contour labels and hachure marks, elimination of very small closures or short line segments, smoothing of contours made from a coarse grid, writing contour line coordinates to a disk file instead of drawing them on an output device, and plotting of random data points without contouring.

Program Usage

Instructions to the program are obtained from an ASCII command file created by the user prior to running the program. The command file has a namelist format used by many mainframe Fortran compilers, but it is not an ANSI Fortran 77 standard and therefore not included in Microsoft Fortran v 4.1. A subroutine is included in the contouring program to simulate a Fortran namelist read; the only known restriction is the nonability to handle repeat assignments that use an asterisk (e.g. jcdash = 5*100). The file therefore starts and ends with an ampersand or dollar sign with any number of variable assignments separated by blanks or commas (see examples below). There are presently 87 parameters that can be used in the command file; however, a contour plot can be produced by including only one variable: the input grid filename. Many of the parameters have default values and usually only a few input variables are necessary.

Program execution

It is necessary to create two files prior to running the program; 1) a command file as explained above and 2) a standard grid file containing the data to be contoured (see Appendix A). Four additional optional files to be described later can also be used as input to the program.

To execute the program type PCCONTUR. The program will print some header information and ask the question: enter command filename: . Enter the name of the command file and the program will then print on the monitor all of the namelist variables and their assignments. The user can then break out of the program to correct mistakes or hit return and the program will then continue until a plot is made. If the plot is made on a monitor, the image will stay until any key is hit, at which time the program will terminate.

Since there are so many combinations of input parameters to be chosen, the following subheadings will consist of explanations and examples of command files that will produce plots ranging from simple to complex. Following this section will be an alphabetical description of all 87 input variables. A sample grid file and a command file are provided on one of the accompanying diskettes and the user is encouraged to try different variables with this sample grid to get a 'feel' for the program. Appendix D contains examples of some plots generated from the sample grid file.

Plot with border, contour lines and labels

If an EGA monitor is being used and the coordinates of the grid are in kilometers, a simple plot can be generated either with only one variable: IFILE in a command file, or by entering the return key to the question: enter command filename. In the later case the program will prompt for the input grid filename. The program will assign an appropriate contour interval up to a maximum of 20 levels and an appropriate format for labelling, by reading the data set and determining the minimum and maximum values.

Example: & ifile = 'test.grd'&

Contour levels

A constant contour interval is specified by using DCVAL. The upper and lower limits for contouring using the variable can be set using CMAX and CMIN.

Specific contour levels to be contoured are controlled by the array variable ACVAL.

NSEC is a variable that specifies the number of contour intervals between primaries. It is only used with DCVAL.

IDASHS determine the line type and thickness of primary and secondary contour lines.

Example: & ifile = 'test.grd' dcval = 5 nsec = 2 idashs = -9 &

This example will produce primary contour lines with double thickness at every other contour interval and secondary dashed contour lines.

Plot devices and scaling

IPLOTR is a number that determines the type of plot device. The default number is 9 for EGA mode on a color monitor.

Scaling is performed using the variable MSCALE or alternatively XSCALE and YSCALE.

The origin of the plot area on a device is determined from the variables PLLX and PLLY.

Example: $ ifile = 'test.grd', iplotr=5, pllx=2, mscale=250000&

This command file will produce a plot on an HP plotter at a scale of 1:250,000 with the x origin shifted to the right 2 inches.

Plot using colors or different pens

Specify the individual contour levels by using the variable NCVAL and the array variable ACVAL instead of using DCVAL, and use the array variable JCDASH to assign colors or pen numbers. Each of these arrays is dimensioned 200. A maximum of 16 colors can be specified for a monitor plot and a maximum of six pens for a pen plotter. To assign a color/pen using JCDASH, first determine the type of line desired (see IDASHS) and then add 100 to that number for color/pen 1, 200 for color/pen 2, etc., up to 1600 maximum.

```
Example: $ ifile = 'test.grd' iplotr = 10 ncval = 5
           acval = -20, -10, 0, 10, 20
          jcdash = 100, 201, 305, 408, 516 $
```

The above example will plot on a VGA monitor contour value -20 with a solid line and color 1, contour value -10 with a dashed line and color 2, contour value 0 with another type of dashed line and color 3, contour value 10 with a thick solid line and color 4 and contour value 20 with a thicker solid line and color 5.

Plot with hachures in closed lows

Include the variable MXHACH in the command file. Other optional variables that control the type of hachuring are HACHSP, HACHLN, HACHGP, HACHLM and HACHVB.

An alternative to hachuring is provided by plotting an H or L symbol at approximate high and low places in the data. The variables to use for this option are LOWHI, SIZEHI and DGRAD.

Labelling of axis

Set NCHARX = -1 and NCHARY = -1 and the program will do automatic formatting. If one wants to control tick intervals and formats use the variables ADELX, ADELY, LINTX, LINTY, FMTX, FMTY, NCHARX, NCHARY, SIZEX and SIZEY.

Smooth contours

Add the variable NSIG and set it to non zero. An optional variable that can be used with this option is SIGMA. Smoothing is performed by using B-splines.

Plotting latitude/longitude tick marks

The program has projection routines included, so if the user knows the parameters used to generate the grid file coordinates, geodetic coordinates can be properly placed on a plot. The variables needed are BASLAT, CM, IPROJ, LATM, LATX, LONGM, LONGX, TINT, ITPOST, SIZEP, PHI1 and PHI2. West longitudes are negative.

```
Example:  $ ifile = 'test.grd', dcval 5,
           mxhach = 1 nsig = 1
             latm = 38,45, latx = 39
            longm = -110,30, longx = -111
           baslat = 38,30, cm = -111
            iproj = 2, tint = 15, itpost = 1 &
```

This example in addition to contouring with hachured lows and smoothing will put latitude/longitude tick marks every 15 minutes within the boundary specified and label each on the borders of the plot.

Plotting of text

Two lines of title, each 56 characters long, can be plotted below the 56 characters of identification obtained from the header record of the grid file (TITLE1) by using the variables TITLE2, TITLE3, and SIZEL. This title information is centered below the plot.

In addition to the title, text can be placed anywhere on the plot by providing the necessary information in an external ASCII file called text. dat. See variable TXFILE for the format of this file.

Plotting of line data

Line data or a series of line data segments may be plotted inside the area covered by the gridded data set. The variables required for this option are LXFILE, LXFMT, LXPROJ, LXUNIT, and LXTYPE. A border can be drawn around the data area by using the variable NEAT.

Plotting of random data points

Random data (e.g., the input data that were used to produce the grid file) can be plotted by using the variables IFILE2, ISPOST, IFMTV, FMTV, NCHARV, VMIN, VMAX, SZPOST, SZLAB, NID, ICH, and CHID.

```
Example:  &parms ifile = 'test.grd', dcval = 5,
          ispost = 2, ifmtv = '(3F10.2)',
          fmtv = '(F5.0)',ncharv=4,ifile2 = "test.dat' &
```

The above example will plot 4 characters of the z value from a file called test.dat that has data in a x, y, z format of 3F10.2, along with the contours of the grid file. If no contours are desired, leave out IFILE and DCVAL.

Contour labelling

The following parameters control various aspects of labelling of contours: NSEC, FMTC, NCHAR, SIZE and DELB. If FMTC is not used, then automatic formatting of labelling is performed.

Contour elimination

Short contour lines can be eliminated by specifying in map inches the length less than which no contour lines will be drawn. The variable for this is CONLIM, and by definition the value is scale dependent.

Contour gradients

Parts of contour lines can be eliminated in areas where the contours become very close to one another. Variables to control this are GRADI, ACDEL, and ACGRAD.

Capture of contour line coordinates to a disk file

Instead of plotting contour lines on a plot device, an option has been provided to write the ground coordinates where the contours intersect the grid cells to a disk file by using the variable ICAPT. This option has proved useful for importing contour line data into GSMAP and GSDRAW (Selner, and Taylor, 1988) and for producing Digital Line Graph (DLG) files for import into Geographical Information Systems (GIS) such as ARC/INFO and GRASS.

If this option is used, two variables GRADI and CONLIM must be computed differently than described in the next section. CONLIM will be units of the grid coordinates rather than in map inches and GRADI = GRADI/(factor*MSCALE /1,000,000) where factor = 25.4 for a grid with kilometers as units, factor = 25,400 for a grid with meter coordinates and factor = 1 for ground coordinates in inches.

Command File Variables

ACDEL - real*4. An array of length NCVAL containing the maximum allowable gradient in Z-data units per contour line. The contour line will not be plotted across grid cells with higher gradients. The numbers in this array must be in the same order as the contour levels which are specified in ACVAL. If any element of ACDEL is set to a negative number, the corresponding ACVAL is ignored and a set of contour lines are plotted at exact multiples of ABS(ACDEL) from the lowest portion of the map to the highest (or from CMIN to CMAX if they are not both 0 or from ACMIN to ACMAX if they are specified). In this case the maximum gradient per contour line for each contour level is equal to ABS(ACDEL) (i.e., negative ACDEL determines both the contour interval and the maximum gradient in Z-data units per contour line whereas positive ACDEL determines only the maximum gradient in Z-data units per contour line for the corresponding ACVAL element). If more than one ACDEL is negative, the contour levels produced from the ACDEL with the larger absolute value take precedence over all levels derived from negative ACDEL. Default is 0 which means the gradient will be infinity (no contour dropping in high gradient grid cells) or if NCVAL is 0, gradient information will be taken from DCVAL. The ACDEL array may be specified in the second column of the ACFILE.

ACFILE- character*56. The name of an ASCII free format file containing parallel columns of the following arrays: column 1 ACVAL, column 2 ACDEL, column 3 ACGRAD, column 4 JCDASH, column 5 ACSIZE, column 6 ACMIN, column 7 ACMAX. When ACFILE is used, the following parms need not be specified in the command file: ACVAL, ACDEL, ACGRAD, JCDASH, ACSIZE, ACMIN, ACMAX, NCVAL, DCVAL, NSEC, IDASHS, GRADI, SIZE, CMIN, CMAX. Each line in the ACFILE is like a miniature self-contained contouring package. That is, each line is independent of the other lines and will make its own special set of contours according to the its values of ACVAL, ACDEL, ACGRAD, JCDASH, ACSIZE, ACMIN, and ACMAX. The lines may be arranged in any order because PCCONTUR will automatically determine priorities in cases of conflicts (e.g., where the same contour level is specified on more than one ACFILE line). On any given line the last few columns may be left blank. In this case, the values of the blank columns will be set to the default values. Blank columns may not be followed by non-blank columns. A single isolated period (.) will be taken as a ditto mark meaning that the value of its column will be replaced by the value of the corresponding column in the line immediately above. Periods in the first line will be the same as blank columns. Any non-numeric character on a given line will signify the beginning of a comment and everything after it on that line will be ignored (e.g., C). The default value for ACFILE is blanks which means no ACFILE will be used and therefore all applicable parameters must be specified in the command file. Following is an example printout of a file specified by ACFILE:

```
C ---------------------------------------------------------------
C ACVAL    ACDEL    ACGRAD    JCDASH    ACSIZE    ACMIN    ACMAX
C
C    DESCRETE CONTOUR LEVELS
C
1000        1000       50        16       12    C  THESE COLS
      0       .         .         .        .    C  DO NOT
   1000       .         .         .        .    C  APPLY
C
C    GENERAL CONTOUR LEVELS
C
      0   -2500        50         8      .10      -12500    20000
      0    -500         .         8    C  THESE COLS WILL DEFAULT
      0    -100         .         0      .08       -2000     3000
      0     -20         .         0     0.00        -300      500
      0      -4         .         3        .         -80       80
C
C    END of ACFILE
C ---------------------------------------------------------------
```

ACGRAD- real*4. An array of length NCVAL containing the maximum allowable gradient in contour lines per map inch. The contour line will not be plotted across grid cells with higher gradients (the gradient in Z-data units per map inch = (gradient in z-data units per contour line) * [gradient in contour lines per map inch] = ACDEL * ACGRAD). The numbers in this array must be in the same order as the gradients specified in ACDEL. Default is 0 which means the gradient information will be taken from GRADI. The ACGRAD array may be specified in the third column of the ACFILE.

ACMIN- real*4. An array of length NCVAL containing the minimum contour level when the contour levels are being determined by a negative ACDEL. Default is 0 which means that if ACMAX is zero then all possible contour levels will be plotted. The ACMIN array may be specified in the sixth column of the ACFILE.

ACMAX- real*4. An array of length NCVAL containing the maximum contour level when the contour levels are being determined by a negative ACDEL. Default is 0 which means that if ACMIN is zero then all possible contour levels will be plotted. The ACMAX array may be specified in the seventh column of the ACFILE.

ACSIZE- real*4. An array of length NCVAL containing the label size in map inches for each contour level specified by ACVAL. Default is equal to SIZE (see documentation for SIZE). To eliminate labels set ACSIZE to 0. The ACSIZE array may be specified in the fifth column of the ACFILE.

ACVAL- real*4. An array of length NCVAL containing specific contour levels to be plotted. They may be arranged in any order. The ACVAL array may be specified in the first column of the ACFILE.

ADELX- real*4. The interval in data units along the x axis where a tick will be made. Default is zero which means that only the minimum and maximum values will be posted.

ADELY- real*4. The interval in data units along the y axis where a tick will be made. Default is zero which means only the minimum and maximum values will be posted.

BASLAT- real*4. A three unit array containing the base latitude in degrees, minutes and seconds that is used in the projection routines for plotting latitude/longitude tick marks. Default is 999,0,0.

CHID- character*4. A list of character strings of no more than four characters each that are used to match the first characters of identification on a post file, which is explained under ISPOST. A maximum of 20 strings can be specified. Default is blanks.

For example, let NID = 3, ICH = 3,1,2,11 and CHID = 'abc', 'tc01', 'HUR'. Station abc will be plotted with symbol 3, station tc01 will be plotted with symbol 1 and station HUR will be plotted with symbol 2. The NID + 1 number for ICH, in this case 11, will be used to plot all those stations that did not match the specified chid character strings. This number can be omitted and the default number 2 will be used.

CM- real*4. A three unit array containing the central meridian in degrees, minutes, and seconds that is used in the projection routines for plotting latitude/longitude tick marks. West longitudes are negative. Default is 999,0,0.

CMAX- real*4. The upper limit of the contour values when DCVAL is used. If CMAX is greater than the maximum value of the data, then the maximum data value is used to determine the last contour value. Default is zero.

CMIN- real*4. The lower limit of the contour values when DCVAL is used. If CMIN is less than the minimum value of the data, then the minimum data value is used to determine the first contour value. Default is zero. If both CMIN and CMAX are zero, then the range of contouring is determined from the data.

CONLIM- real*4. Contour minimum length limitation. The minimum length (in map inches) allowed for any contour, whether or not it is closed. Shorter contour lines will not be plotted. Default is 0 which means that no checking for contour length will be performed. Simultaneously setting CONLIM, MXHACH, and DELB to 0 will decrease CPU time and disk space requirements.

DCVAL- real*4. The contour interval. This value must be greater than zero if ACVAL is not used. Default is zero.

DELB- real*4. The desired distance in map inches between contour line labels. Default is 7.0 inches.

DGRAD- real*4. Used in conjunction with LOWHI to determine whether or not a 'low' or 'hi' has enough curvature to be labeled. This is useful for maps which have high amplitude short wavelength anomalies combined with quiet areas. For these maps, LOWHI can be set to a small number of grid cells to include all of the short wavelength anomalies and DGRAD can be made some non-zero number to eliminate cluttering of labels in the quiet areas. DGRAD is the vertical difference in Z data units between the center and the edge of a 'low' or 'hi'. The edge is at the radius determined by LOWHI. Default is zero which means anything that qualifies as a 'low' or 'hi' within the radius specified by LOWHI will be labeled.

FMTC- character*16. The format to be used for labelling the contour values (either F or E format). The parameter must be enclosed in quotes, e.g., '(F5.0)'. Default is blanks, which will trigger automatic formatting. When using the F-format the number of digits will be automatically adjusted for each contour level in order to eliminate leading blanks, i.e., for contour level = 500, if FMTC = '(F10.0)', it will be adjusted to '(F4.0)' and when the level reaches 1000 it will be re-adjusted to '(F5.0)'. The number of digits to the right of the decimal point will not be adjusted.

FMTV- character*16. The format to be used when labelling the z values (either E or F) of a random data file. The parameter must be enclosed in quotes, e.g., '(F6.1)'. Default is '(F7.2)'.

FMTX- character*16. The format (either F or E) to be used when labelling the x axis This parameter must be enclosed in quotes, e.g., '(F9.2)'. Default is blanks.

FMTY- character*16. The format (either F or E) to be used when labelling the y axis This parameter must be enclosed in quotes, e.g., '(F8.3)'. Default is blanks.

GRADI- real*4. The maximum gradient in contour lines per inch before the secondary contours in a grid cell are dropped. Default is 30. If GRADI is negative then when DCVAL is being used the primaries will be eliminated at the same time the secondaries are eliminated. (This yields a sort of spotty map). If GRADI is negative and ACVAL is being used, then ABS(GRADI) = (The gradient in Z-units per map inch where all contours are cut). If ACDEL and ACGRAD, or ACFILE are being used, GRADI will be ignored.

HACHGP- real*4. Hachure gap ratio. HACHGP allows hachures to be made on contour lines which intersect the edge of the map or no data areas and which partially enclose a local low. HACHGP = (Direct distance between the end points of the contour line) / (The total length of the contour line from end point to end point). Contour lines which do not completely enclose a low but which have a gap smaller than HACHGP will be hachured. Default is 0 which means only completely closed contours surrounding a low will be hachured.

HACHLN- real*4. Hachure tick length. The length (in map inches) of each hachure tick mark. Default is .05 inches. Setting HACHLN to 0 will turn off the hachuring option.

HACHLM- real*4. The minimum circumference (in map inches) of a low closure that will be subject to the limitation imposed by MXHACH. For example: a low is surrounded by 2 closed contours. The first (lowest) is 2 inches around and the second is 3 inches around. MXHACH is set to 1. If HACHLM is 1.5 inches, then only the first contour will be hachured. But, if HACHLM is 2.5, inches then both the first and second will be hachured (because, the first contour circumference is less than the minimum specified by HACHLM and is therefore not counted when finding the number of closures surrounding the low). The purpose of HACHLM is to disable the functioning of MXHACH unless low closures are large enough to contain a significant number of hachure ticks. Default is -1 which means that HACHLM is found by multiplying the hachure tick spacing (HACHSP) by a factor of 7.

HACHSP- real*4. Hachure tick spacing. The distance (in map inches) along a contour line between hachure ticks on low closures. Default is .15 which means hachures will be placed every .15 inches on contour lines enclosing a local low. Setting HACHSP to 0 will turn off the hachuring option.

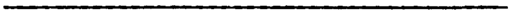
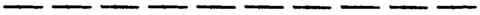
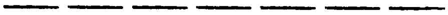
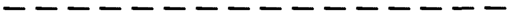
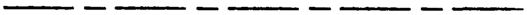
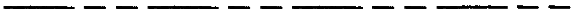
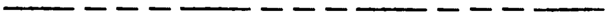
HACHVB- real*4. *******> WARNING ***> If used improperly this parameter may produce deceptive results on your map. Do not use HACHVB unless you are absolutely certain about what you are doing. <****************** On occasion a low closure may surround a high. In this case the high prevents the low closure from being hachured. Hachures may be forced onto the surrounding low closure by setting HACHVB to the circumference of the enclosed high. Any high with this circumference or less will not prevent the surrounding low from being hachured.

IBOUND- integer*4. A variable that determines whether state boundaries are to be plotted. This variable is not presently implemented in PCCONTUR but is documentated for future use. Default is 999, which means no state boundaries will be plotted. If plotting of boundaries are desired, then set IBOUND to a number from 0 to 6 inclusive. The number selected will determine the type of line desired for the boundaries (see IDASHS). If boundaries are desired, then the variables BASLAT, CM, IPROJ, LATM, LATX, LONGM, LONGX, and UNIT must also be specified.

ICAPT- integer*4. This parameter determines whether contour line coordinates will be written to a disk file rather than plotting the contour lines on a plot device. The coordinates will be in units of the grid interval of the input grid file (e.g. kilometers). Two output ASCII files are created with this option. One called contour.val contains the values of the individual contour levels created and the other called contour.cor contains a series of segments with header lines containing the contour value and the number of points followed by lines of x and y coordinates. Any nonzero number will trigger this option.

ICH- integer*4. A list of symbol numbers to be used. These numbers refer to the desired characters as shown in Appendix C. A maximum of 20 can be specified. Default numbers are 4, which is an X.

IDASHS- integer*4. Determines line thicknesses and also which lines are to be dashed.
idashs = ± (N + ITH * 8) where:
N = 0 = all contours plotted as solid lines
N = -1 to -6 = secondary contour lines are plotted as dashed lines
N = 1 to 6 = primary contour lines are plotted as dashed lines
ITH = the additional thickness of the primary contours; e.g., ITH=1 produces primaries twice the normal width of a line, ITH=2 produces primaries three times the normal width of a line, etc.

N value	Plotted line
0	
1	
2	
3	
4	
5	
6	

IFILE- character*56. The file name of the input gridded data set that is to be contoured. It must be enclosed in quotes e.g., 'test.grd'. Default is blanks, which means that the program will prompt for the name.

IFILE2- character*56. The file name of the random data points that were used to produce the gridded data set. It must be enclosed in quotes. Default is blanks.

IFMTV- character*56. Format of the input random data file IFILE2, used when ISPOST is equal to equal to 1 or 2. It must be enclosed in quotes, e.g., '(2F10.3, 3F6.1)'. Default is blanks which means the input random data will be in binary form.

IPLOTR- integer*4. A number which determines which device is to be used.

5 = Hewlett-Packard 7400/7500 series plotters or compatibles that use HP-GL instructions.
8 = CGA video mode (640 x 200)
9 = EGA video mode (640 x 350)
10 = VGA video mode (640 x 480)

IPROJ- integer*4. A number referring to the type of projection to be used
1 = American polyconic
2 = ellipsoidal transverse mercator
3 = mercator
4 = lambert (see variables PHI1 and PHI2)
5 = albers equal area for the conterminous U.S. (standard parallels of 29.5^{o} and 45.5^{o}
6 = albers equal area for Alaska (standard parallels of 55^{o} and 65^{o})
7 = albers equal area for Hawaii (standard parallels of 8^{o} and 18^{o})
9 = spherical transverse mercator used for the Decade of North America (DNAG) maps
Default is 999 for no projection

ISPOST- integer*4. A variable that determines whether random data positions will be plotted and, if so, what type of input data is to be read and posted. Default is zero which means no plotting of positions will be done.

1 = plot in vector form. The input records will contain the following information: x-position, y-position, magnitude, inclination and declination. The maximum number of records is 200. The inclinations are plotted next to a circle, which marks the xy position. The declination is indicated by the direction of an arrow and the magnitude by the length of the arrow.

2 = position plot with the records in an x, y and z format, where z is the value of the field at position x, y.

Negative values are used to indicate post files, which are file outputs from the gravity reduction program bouguer. Post files are binary records containing the following information: 8 characters of station identification, x-coordinate, y-coordinate, and six z-values (free-air anomaly, complete bouguer anomaly one, elevation in feet or meters (depending on input units), inner terrain correction, outer terrain correction and observed gravity minus 980000 milligals.

-1 to -6 will print one of the z values next to the station symbol

-11 to -16 will print the station identification next to the station symbol

-21 to -26 will print the station identification and one of the z-values next to the station symbol

To plot every ITH point from an XYZ file, set ISPOST to 2I. To plot every ITH point from a post file, set ISPOST = (-31I)+(the value of ISPOST for plotting every point). This will only work with post files if the IDS are blank. If the IDS are not blank, then every station will be plotted and their IDS will be plotted instead of the station symbol. (This may be changed in the future).

These additions to ISPOST are helpful for plotting flightline positions. To add flight lines to your map set:

ISPOST= (The appropriate number as described above)

IFILE2= 'name of input xyz or post file'

PCCONTUR will ask for the input file format immediately following the .CMD file name input. Use a blank line for binary xyz or post files.)

ICH = 7 for EW lines, 108 for NS lines, 47 for NESW lines, and 92 for NWSE lines.

SZPOST= size of symbol in map inches.

JCDASH- integer*4. An array of length NCVAL containing the line type designation numbers (see IDASHS) for each individual contour level when ACVAL is being used. The numbers in JCDASH must be in the same order as the contour levels which are specified in ACVAL. Default is 999 which means line type information will be taken from IDASHS. The JCDASH array may be specified in the fourth column of the ACFILE.

When drawing to a color monitor, up to sixteen colors can be specified by adding 100, 200, ... 1600 to the value of JCDASH. When using a pen plotter, up to six pens can be specified by adding 100, 200, ... 600 to the value of JCDASH.

LATM- real*4. A three unit array containing the minimum latitude in degrees, minutes and seconds to be used in plotting latitude/longitude tick marks. Default is 0,0,0.

LATX- real*4. A three unit array containing the maximum latitude in degrees, minutes and seconds to be used in plotting latitude/longitude tick marks. Default is 0,0,0.

LINTX- integer*4. The interval of x axis ticks to be labelled. Default is 1.

LINTY- integer*4. The interval of y axis ticks to be labelled. Default is 1.

LONGM- integer*4. A three unit array containing the right (east) longitude in degrees, minutes and seconds to be used in plotting latitude/longitude tick marks. West longitudes are negative. Default is 0,0,0.

LONGX- real*4. a three unit array containing the left (west) longitude in degrees, minutes and seconds to be used in plotting latitude/longitude tick marks. West longitudes are negative. Default is 0,0,0.

LOWHI- integer *4. A parameter to plot the symbol "L" in low closures and the symbol "H" in high closures on contour plots. To implement this option set "lowhi" equal to a number of grid intervals to use as a search radius. Usually a value of 4 is adequate. Default is 0.

This option will not work if LOWHI is greater than the number of columns in a grid or the grid intervals are not equal in either the x- or y-direction. If LOWHI is made a negative number, then the actual value of the low or high will be plotted rather than an 'L' or 'H'.

LXFILE- character*56. The name of an external file containing ordered pairs of a line segment or segments to plot. Each line segment must terminate with an end-of-file or a number greater than or equal to 1.0E+37 in either x or y or both fields. Default is blanks which means no external line plotting will be done.

LXFMT- character*16. The format of the input external line file. If non-blank, the data is read in list format. If list format is used, the input data may be in xy or xyz form. Default is blanks which means binary xy file.

LXPROJ- integer*4. The number of the projection of LXFILE. If the file has been projected, it must have the same projection number (IPROJ), central meridian (CM), and base latitude (BASLAT) as the grid file. If this parameter is used then IPROJ must also be specified and set equal to LXPROJ. Default is 999 which means unprojected data.

LXTYPE- integer*4. The number of the desired line type (see idashs). Default is 0 which means a solid unthickened line will be used. If LXTYPE is set to 999, the first point of each line segment in LXFILE will be taken as the value for LXTYPE. In this case both x and y must be equal to each other and they must be equal to a valid line type number.

LXUNIT- integer*4. The number representing the type of units in the LXFILE. 0 - inches (projected), seconds (unprojected); 1 - meters, minutes; and default is 2 - kilometers, degrees. Note: If LXPROJ=999 then LXUNIT is seconds, minutes, or degrees. Otherwise it is inches, meters, or kilometers. If the grid is not projected, setting unit = 0, 1, or 2 will specify seconds, minutes, and degrees for the grid so that the grid units (use UNIT) may be matched to the LXFILE units (use LXUNIT).

MSCALE- integer*4. Scaling factor that can be used in lieu of XSCALE and YSCALE. It is given in map scale units such as 62500, 250000, etc. Default is zero.

MXHACH- integer*4. The number of contour closures surrounding a low which will be hachured. The purpose of MXHACH is to prevent large numbers of hachured closures from being stacked up around a single low. Default is 0 which means that the hachuring option is turned off. Setting MXHACH to a negative number will allow all possible low closures to be hachured. Simultaneously setting MXHACH, CONLIM, and

DELB to 0 will decrease CPU time and disk space requirements.

NCHAR- integer*4. The maximum number of characters to be used when labelling the contours. For example, with NCHAR equal to 4 and FMTC equal to "(F5.0)", four digits would be plotted without a trailing period. Contour labelling is performed for all specified contour levels (ACVAL) and for all primary incremental levels (NSEC). Default is zero which means no labelling. NCHAR will be adjusted with FMTC, i.e., when FMTC is adjusted from '(F10.0)' to '(F4.0)', if NCHAR = 9, it will be adjusted to 3. In this example the trailing decimal point will be eliminated for all contour levels.

NCHARX- integer*4. the maximum number of characters to be used from format FMTX when labelling the x axis. Default is zero for no labelling. If set to -1, automatic labelling of the x axis will be done.

NCHARY- integer*4. the maximum number of characters to be used from format FMTY when labelling the y axis. Default is zero for no labelling. If set to -1, automatic labelling of the y axis will be done.

NCHARV- integer*4. the maximum number of characters to be used from the format FMTV when plotting the z value. Default is zero for no plotting of z value.

NCVAL- integer*4. The number of values specified under ACVAL above. Default is zero which means that a constant contour interval will be obtained from the variable DCVAL.

NEAT- integer*4. A value that determines whether a rectangular border line will be drawn around the contour plot. The default is zero, meaning a border line will be drawn. Any other number will suppress the border.

NID- integer*4. The number of different station symbols to be used; maximum is 19. Default is zero which means all stations will be plotted with one symbol.

NSEC- integer*4. The interval of contour levels which are considered primary. This parameter is used when DCVAL is specified. Default is five, which means every fifth level is considered primary. Only primary contours are labelled.

NSIG- integer*4. Setting NSIG to any number except 0 invokes the B-spline option. Default is 0 which means no spline will be performed.

PHI1- real*4. A lower parallel of latitude in degrees to be used in a Lambert projection (IPROJ=4). Default is 33 degrees for the standard conterminous U.S. parallel.

PHI2- real*4. An upper parallel of latitude in degrees to be used in a Lambert projection (IPROJ=4). Default is 45 degrees for the standard conterminous U.S. parallel.

PLLX- real*4. PLLX is the distance in map inches along the +x direction of the y-axis from the left edge of the plot. Setting PLLX to 0 causes it to default to 2.

PLLY- real*4. PLLY is the distance in map inches along the +y direction of the x-axis from the bottom edge of the plot. Setting PLLY to 0 causes it to default to 2.

SIGMA- real*4. Adjusts the smoothness of the splined curve. SIGMA is the maximum allowable average error (in thousandths of an inch) between the spline contour line curve and a perfectly smooth curve. The minimum grid cell area that will be splined is: 8 * [.001*SIGMA / (PI-2)] ** 2 square map inches. Decreasing sigma will increase the number of points in each contour line, resulting in higher resolution but also increasing CPU time. Note: A given value for SIGMA will result in uniform smoothness for all map scales and grid cell sizes.

Default is 7.3 mils which is sufficient for raster plots with pixel sizes of .01 inches or larger. Smaller rasters or pen plots may require a smaller SIGMA. Following is a chart showing the total number of points produced in each grid cell for a given SIGMA and length of grid cell diagonal (in map inches):

(sigma) / (diagonal length)	(no. of points per cell)
0.00 - 1.03	10
1.03 - 1.27	10
1.27 - 1.60	9
1.60 - 2.09	8
2.09 - 2.85	7
2.85 - 4.09	6
4.09 - 6.38	5
6.38 - 11.27	4
11.27 - 24.92	3
24.92 - 90.85	2
90.85 - infin	1

SIZE- real*4. The height in inches of labelled contour values. Default is .08.

SIZEHI- real*4. Size of the LOWHI symbols in map inches. Default is .08.

SIZEL- real*4. The height in inches of the first title line. TITLE2 and TITLE3 will be plotted at .75*SIZEL. Default is zero, which means no title lines will be plotted.

SIZEP- real#4. The height in inches of the tick mark labels. Default is set at .20 inches.

SIZEX- real*4. The height in inches of the labelled x axis values. Default is .08.

SIZEY- real*4. The height in inches of the labelled y axis value. Default is .08.

SZLAB- real*4. Size of the symbol labels (symbol sizes are specified by SZPOST) in map inches. Default is .08.

SZPOST- real*4. The station symbol size in inches. Default is .08.

TINT- real*4. The interval of tick marks in minutes. Default is 15.

TITLE2- character *56. Identification to be plotted below the 56 characters obtained from the header record of the input gridded data file. This parameter must be enclosed in quotes, e.g., 'scale 1:250,000'. Default is blanks.

TITLE3- character*56. Identification to be plotted below TITLE2. This parameter must also be enclosed in quotes and the default is blanks.

TXFILE- character*56. The name of an ASCII file containing the information necessary to plot text on the plot. Each line of this file will plot one block of text. List format is used for these lines and the order and meaning of the variables is exactly the same as explained under subroutine vchar in Appendix C, with the exception that the number of characters to plot is not included. The order is therefore x coordinate, y coordinate, text to be plotted, enclosed in quotes (80 characters maximum), icode (determine whether coordinates are in plot or data units, data units can only be used inside the data area), size of the characters, rotation of the characters, x offset in plot units for the first character and y offset in plot units for the first character.

The first five parameters are required, the last three are optional. Default is blanks for the file name.

UNIT- real*4. The units of the grid interval in the input gridded data set to be contoured. 0 = inches, 1 = meters, and 2 = kilometers. Default is 0. This parameter must be specified if MSCALE is used.

VMAX- real*4. Used when ISPOST=1. It is the maximum vector length in inches. Default is 1.

VMIN- real*4. Used when ISPOST=1. It is the minimum vector length in inches. Default is zero.
The vectors are logarithmically scaled from VMIN to VMAX.

XSCALE- real*4. The x axis data units per inch. For example, if the grid interval is in kilometers, then a value such as 6.35 would be used to produce a map at a scale of 1:250,000. Default is zero.

XXX- real*4. An array dimensioned two that contains the left and right coordinates of a random data input file (IFILE2). This is only used if one wants to plot data without providing a matching grid file. When this variable is specified along with the variable YYY, the contouring option is bypassed and all other options (lat/lon tick marks, axis labelling, etc.) not directly related to contours are still functional.

YSCALE- real*4. The y axis data units per inch. Default is zero. If both XSCALE and YSCALE are zero then they will both be set to the same value that will allow plotting on the chosen plot device or 10 by 8 inches, whichever is smaller. If one scale factor is zero and the other not, then the nonzero scale factor will be assigned to the zero scale factor.

YYY- real*4. An array dimensioned two that contains the bottom and top coordinates of a random data input file (IFILE2). This is only used if one wants to plot data without providing a matching grid file. When this variable is specified along with the variable XXX, the contouring option is bypassed and all other options (lat/lon tick marks, axis labelling, etc.) not directly related to contours are still functional.

References

Evenden, Gerald Ian, 1975, A general purpose contour system: U.S. Geological Survey Open-File Report 75-317, 107 p.

Godson, R. H., and Webring, M. W., 1982, CONTOUR: A modification of G. I. Evenden's general purpose contouring program: U.S. Geological Survey Open-File Report 82-797, 67 p.

Selner, G. I. and Taylor, R. B., 1988, GSDRAW and GSMAP Version 5.0: Prototype programs, Level 5, for the IBM PC and compatible microcomputers to assist compilation and publication of geologic maps and illustrations: U.S. Geological Survey Open-File Report 88-295A, documentation and tutorial, 130 p., 88-295B executable program disks (2).

Webring, Michael, 1981, MINC: A gridding program based on minimum curvature: U.S. Geological Survey Open-File Report 81-1230, 41 p.

Appendix A

Standard Grid

The grid can be any one of many types: (a) a rectangular grid with equal spacing in the x- and y-directions, (b) a rectangular grid with constant but unequal x- and y-intervals, (c) a rectangular grid with varying distances between grid positions in either the x- or y-direction or both and d) a quadrilateral grid which consists of connecting quadrilaterals whose interior angles do not exceed 180 degrees.

The file of the gridded data consists of two basic parts: (1) a header record and optionally, a following record that contains the x-coordinates for each column and (2) a series of data records, each containing the column values for one row.

The following diagram shows the relation of the grid elements in the usual case where dx and dy are positive.

```
NROW      X     X     X      X     X      X   last row stored

   3      X     X     X      X     X      X
                      dy
   2      X     X     X  dx  X     X      X

   1      X     X     X      X     X      X   first row stored

          1     2     3      4     5      NCOL
      (xo,yo)
```

A. Header record (23 4 byte words long)

id: 56 ASCII characters of identification (14 words).

pgm: 8 ASCII characters of creation program identification (2 words).

ncol: number of columns of data (integer, 1 word).

nrow: number of rows of data (integer, 1 word).

nz: number of words per data element (integer, 1 word). For single precision use 1, double precision or complex use 2, double precision complex use 4. For quadrilateral grids this value is 3.

xo: position of first column of data (real, 1 word).

dx: equal spacing interval of columns (real, 1 word).

If equal to zero, then coordinates for each column are in the following data record; otherwise the following record consists of data for row one.

yo: position of first row of data (real, 1 word).

dy: equal spacing interval of rows (real, 1 word). If equal to zero, the coordinate for each row is the first word of each data record row.

B. Column coordinate record, present only if dx of header record is equal to zero. Record consists of ncol real words specifying the coordinates of each data column in monotonic order. If nz=3, then this record is present but the values are meaningless.

C. Data record. Each data record contains one row of real data items. The record length is ncol times (nz plus 1) words. For quadrilateral data the sequence of data values is x, y, and z. The first word contains the row coordinate if dy of the header record is zero, else the value is a dummy. Again, the row coordinates should be in monotonic sequence, if specified.

In general, i/o for this standard file can be stated in Fortran as:

```
 dimension g(iz,ix,iy),id(14),pgm(2),x(ix),y(iy)
 read or write (..) id,pgm,ncol,nrow,nz,xo,dx,yo,dy
 if (dx.eq.0.) read or write (..) (x(i),i=1,ncol)
 if (dy.ne.0.) go to 15
 do 10 j+1,nrow
```

```
10    read or write (..)y(j),((g(k,i,j),k=1,nz),i=1,ncol)
      go to 25
15    do 20 j=1,nrow
20    read or write (..) dum,((g(k,i,j),k=1,nz),i=1,ncol)
25    continue
```

In the usual case where dx and dy are constant and nz=1, the code simplifies to:

```
      dimension g(ix,iy),id(14),pgm(2)
      Read or write (..) id,pgm,ncol,nrow,nz,xo,px,yo,dy
      DO 10 J=1, nrow
    10 read or write(..) dum,(g(i,j),i=1,ncol)
```

Appendix B

Compilation and Linkage Instructions

The following steps outline the method used to provide the executable code for program PCCONTUR using Microsoft Corp. languages and associated tools. These steps are presented for those who will modify the code and need to make new executable modules or those who wish to make an executable module that emulates coprocessor instructions rather than requiring a coprocessor. The steps shown are those that were used and work; they are not presented as the definitive way to create an executable module. The languages used were Fortran v. 4.01 or 4.1, Quick C and Assembler v.4.0 or 5.0. The linker was v. 5.01.20.

1. Create a large memory Fortran library module (LLIBFOR7.LIB) without C graphics compatibility and with in-line floating-point instructions.
2. Create a large memory C Library (LLIBC7.LIB) without built-in graphics.
3. Compile PCCONTUR.FOR with /c /Gt8000
4. Compile HACHERY7.FOR with /c /Gt4000
5. Compile LXPLOT.FOR,HEADER.FOR and NAMEL.FOR with just /c.
6. If it is necessary to compile TLINE.FOR in the plot library use /c and /Gt28000.
7. To compile C programs in the plot library use QCL /c /FPi87 /AL
8. Link by using the following:

   ```
   /SE:129 /NOI /NOD /NOE PCCONTUR  HACHERY7
   LXPLOT HEADER NAMEL,,,PLOT GRAPHICS LLIBC7 LLIBFOR7
   ```

Appendix C

Plot System

The program uses an internally written plot system (Evenden, 1975) that has been modified to run under MS-DOS. All of the subroutines are written in Fortran except for a few C language routines that handle graphics for the video monitor and two assembly language routines that control the monitor mode and control input/output for a serial port. All of the subroutines exist in a library called PLOT.LIB, which is resident on one of the accompanying 5-1/4 inch disks, along with the source code for the individual subroutines.

A brief description of the plot system subroutines called by PCCONTUR follows. These are simplified explanations, especially VCHAR, and the reader is referred to the above reference for a more detailed explanation.

ENDPT(ie)- terminates a plot

ie: an integer *4 array that contains plot diagnostics. If ie(1) = 0 or 1, no diagnostics are returned.

LINE(x,y,n,icon,ipn)-draws a line

x: real*4 array containing x coordinates in data units.

y: real*4 array containing y coordinates in data units.

n: an integer *4 variable that contains the number of x and y points.

icon: an integer *4 variable that determines whether a line continues from a previous one. 0 means start a new line; #0 means continue from previous line.

ipn: an integer *4 variable that determines the type of line to be drawn (solid or dashed). Up to sixteen colors can be specified by adding 100, 200, ... 1600 to the value of IPN. The line type for each IPN is:

IPN	Plotted line
0	————————————————
1	— — — — — — — — — — —
2	—— —— —— —— —— —— —— ——
3	- - - - - - - - - - - - - - - - -
4	—— - —— - —— - —— - -
5	—— - - —— - - —— - - —— - -
6	—— - - - —— - - - —— - - - ——

NEATL - draws a border around the data area.

PLTSET - (iplotr,xbd,ybd,isl)-initializes plot system.

iplotr: an integer *4 variable that contains a number that determines the plot device to be used.
5=Hewlett-Packard 7500/7600 series plotters or compatibles.
8=CGA video mode; 9=EGA video mode; 10=VGA video mode.

xbd: a real*4 variable in which the maximum x size of the plotter selected is returned.

ybd: a real*4 variable in which the maximum y size of the plotter selected is returned.

isl: an integer *4 array of length one where isl(1)=1 for inches to be used for the plot units or =2 for centimeters to be used as the plot units.

SCALE - (dxp,dyp,xp,yp,nopts,ier) - defines the data range and the plot size (see figure 1).

dxp: a real *4 array of length two that contains the left and right limits respectively of the data in data units.

dyp: a real *4 array of length two that contains the bottom and top limits, respectively, of the data in data units.

xp: a real *4 array of length nopts where:
xp (1) = size of the data area along x axis in plot units.
xp(2) = 0 for linear representation of the x coordinates or 1 for a logarithmic representation.
xp(3) = offset along the x axis in plot units where the data area starts.
xp(4) = size of the plot area along the x axis in plot units.

Only xp(1) is required. xp(2) and xp(3) default to zero and if NOPTS is <=3, then xp(4) is = xp(1) + xp(3).

yp: a real *4 array of length nopts where:
yp(1) = size of the data area along the y axis in plot units.
yp(2) = 0 for linear representation of the y coordinates or 1 for a logarithmic representation.
yp(3) = offset along the y axis in plot units where the data area starts.
yp(4) = size of the plot area along the y axis in plot units.

Only yp(1) is required. yp(2) and yp(3) default to zero and if nopts is <=3 then yp(4) is = yp(1) + yp(3).

nopts: an integer *4 variable containing the number of parameters in the xp and yp arrays.

ier: an integer *4 variable that contains a number returned by subroutine scale. 0 means no error; any other number indicates an error was detected and user action is needed.

VCHAR(x,y,ich, n, icode,size,theta,xoff,yoff)-writes characters on a plot.

x: a real *4 variable that contains the x coordinate(s) of the center of a character.

y: a real *4 variable that contains the y coordinate(s) of the center of a character.

ich: integer array containing a number of a symbol from the plot system character set (see figure 2) or the character or character string to be ploted.

n: an integer *4 variable containing the number of points or the number of characters to plot (see icode below).

icode: an integer *4 variable that determines the characteristics of the variables x,y and ich.

icode=1 means x and y are real *4 arrays of length n and that a character in ich will be plotted at each x,y location.

icode=2 means that x and y are single locations and that n contains the number of characters to plot at that location.

icode=3 has the same meaning as icode=2 except that the x any y locations are in plot inches from the lower left origin instead of in data units as in icode=1 or 2. Up to sixteen colors can be specified by adding 100, 200, ... 1600 to the value of ICODE.

theta: a real *4 variable containing the rotation in radians of the character string before plotting. Counterclockwise is positive.

xoff: a real *4 variable containing the offset in plot units along the x axis of the first character to be plotted.

yoff: a real *4 variable containing the offset in plot units along the y axis of the first character to be plotted.

XAXIS(dxp,dyp,xp,del,ip,size,fmt,nfmt)-draws and labels the x axis.

YAXIS(dyp,dxp,yp,del,ip,size,fmt,nfmt)-draws and labels the y axis.

dxp,dyp,xp and yp have the same meanings as described under subroutine scale.

del: a real *4 variable containing the interval in data units of the tick marks.

ip: an interger *4 variable that determines the interval between each labelling of the tick marks.

size: a real *4 variable containing the size in plot units of the characters used to label the axes.

fmt: an integer *4 array containing the format for labelling the axes.

nfmt: an integer *4 variable containing the number of characters to use from FMT.

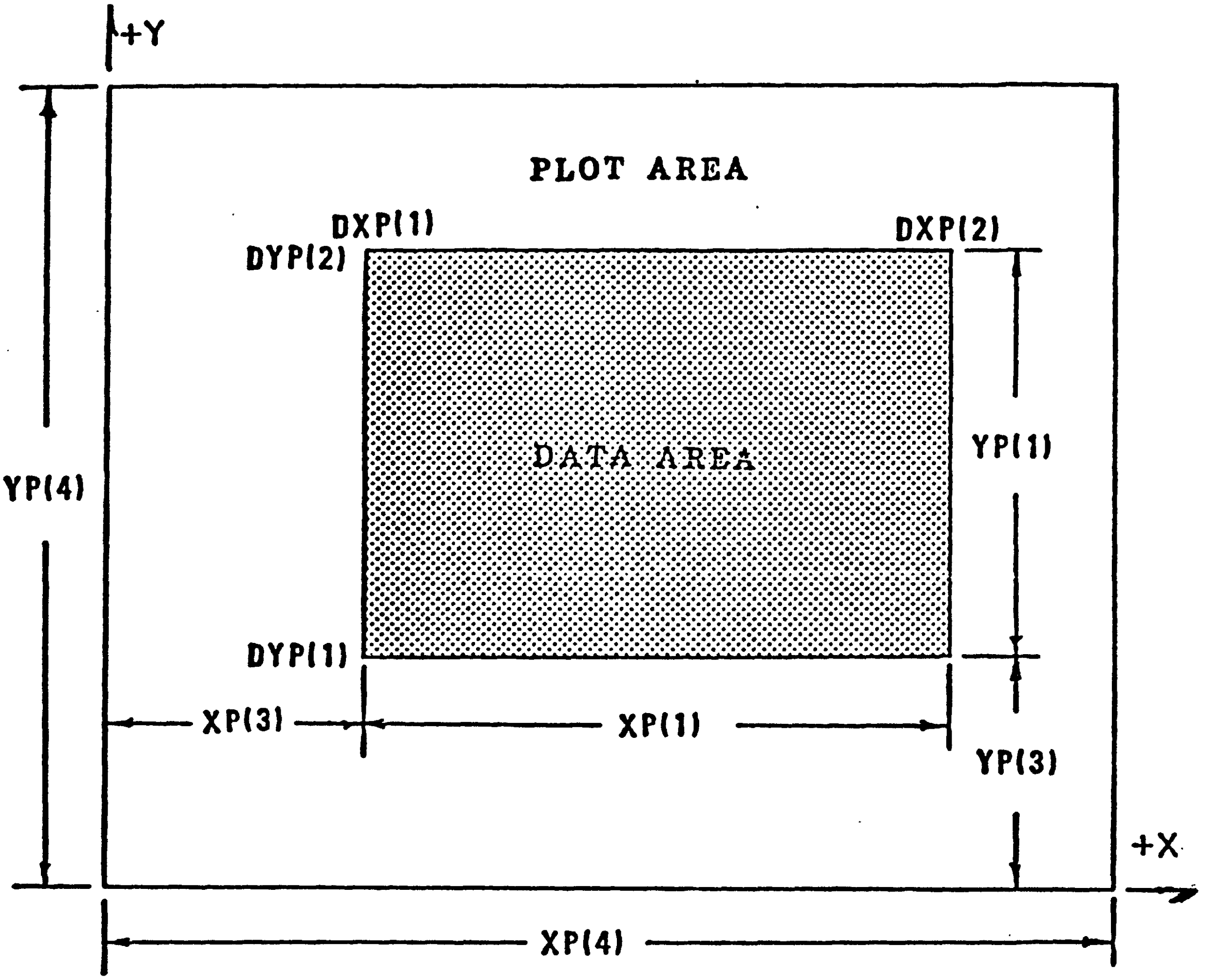
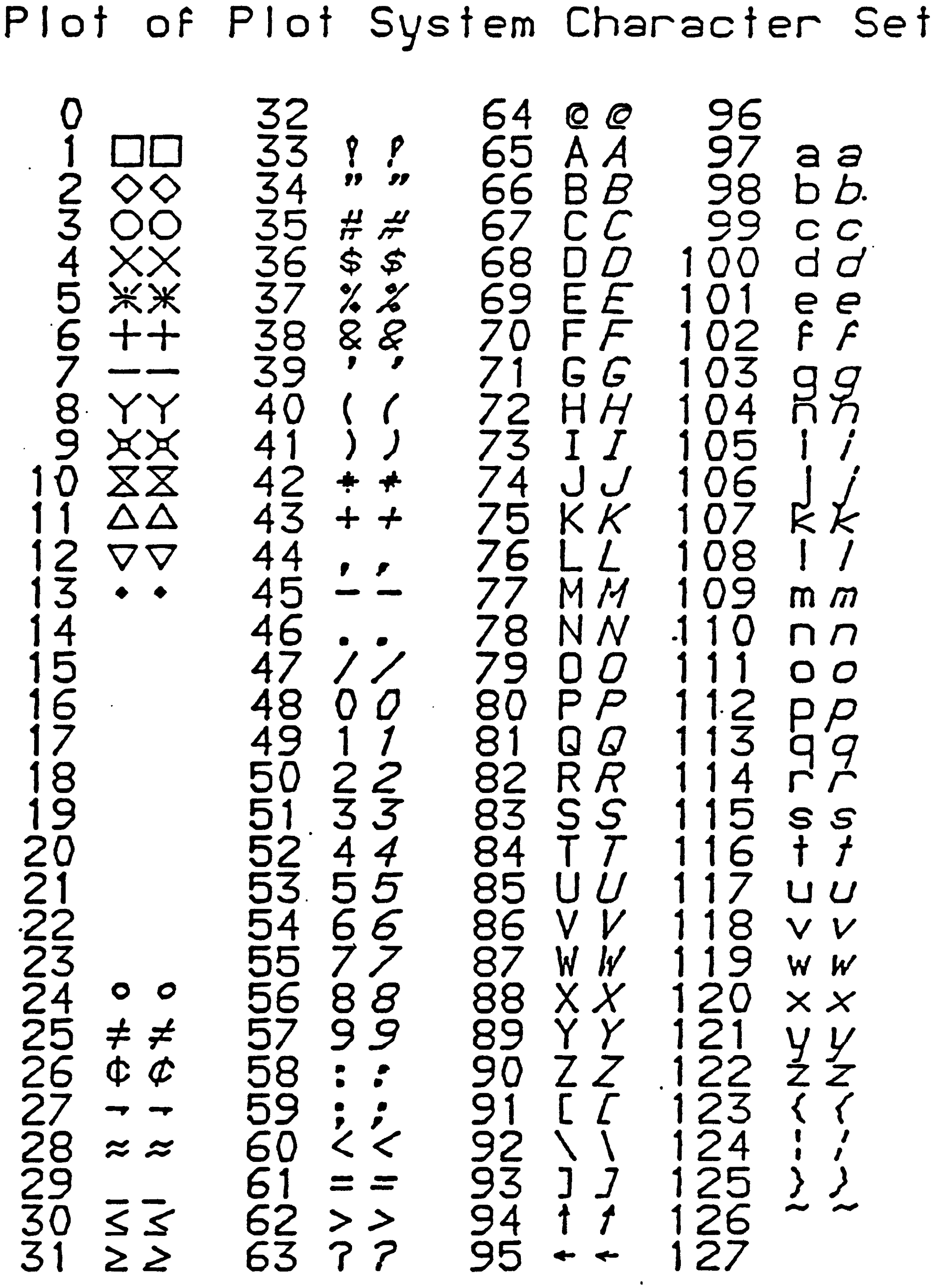


Figure 1. Relationship of plot and data areas

DXP and DYP are in data units.
XP and YP are in plot units.



Plot of Plot System Character Set

0			32			64	@	@	96		
1	□	□	33	!	!	65	A	A	97	a	a
2	◇	◇	34	"	"	66	B	B	98	b	b
3	○	○	35	#	#	67	C	C	99	c	c
4	✕	✕	36	$	$	68	D	D	100	d	d
5	✳	✳	37	%	%	69	E	E	101	e	e
6	+	+	38	&	&	70	F	F	102	f	f
7	—	—	39	'	'	71	G	G	103	g	g
8	Y	Y	40	(	(	72	H	H	104	h	h
9	⋈	⋈	41	)	)	73	I	I	105	i	i
10	⧗	⧗	42	*	*	74	J	J	106	j	j
11	△	△	43	+	+	75	K	K	107	k	k
12	▽	▽	44	,	,	76	L	L	108	l	l
13	◆	◆	45	-	-	77	M	M	109	m	m
14			46	.	.	78	N	N	110	n	n
15			47	/	/	79	O	O	111	o	o
16			48	0	0	80	P	P	112	p	p
17			49	1	1	81	Q	Q	113	q	q
18			50	2	2	82	R	R	114	r	r
19			51	3	3	83	S	S	115	s	s
20			52	4	4	84	T	T	116	t	t
21			53	5	5	85	U	U	117	u	u
22			54	6	6	86	V	V	118	v	v
23			55	7	7	87	W	W	119	w	w
24	∘	∘	56	8	8	88	X	X	120	x	x
25	≠	≠	57	9	9	89	Y	Y	121	y	y
26	¢	¢	58	:	:	90	Z	Z	122	z	z
27	¬	¬	59	;	;	91	[	[	123	{	{
28	≈	≈	60	<	<	92	\	\	124	¦	¦
29	_	_	61	=	=	93	]	]	125	}	}
30	≤	≤	62	>	>	94	↑	↑	126	~	~
31	≥	≥	63	?	?	95	←	←	127		

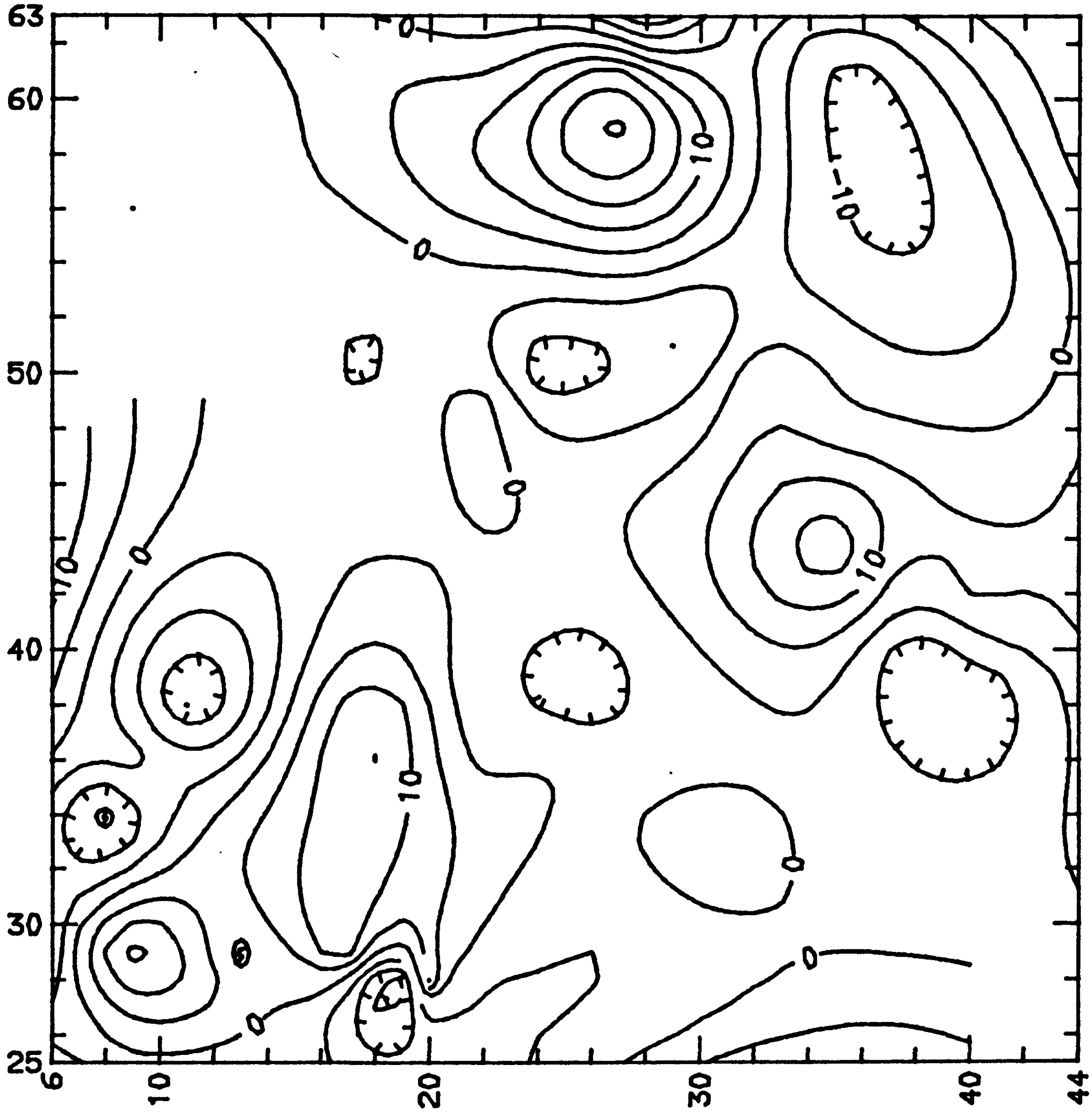
Figure 2.

Basic Plot System Errors

Plot system error no.	Routine	Basic Plot System Errors Meaning
1	SCALE	NOPTS <1 OR >4
2	SCALE	Can't initialize plotting device
3	SCALE	xp(1) or YP(1)=0 (data area size=0)
4	SCALE	XP(3) or YP(3) <0
5	SCALE	XP(4) or YP(4) <=0
6	SCALE	Plot dimension requested larger than plotters drawing dimension
7	SCALE	D{X/Y}P(1)=D{X/Y}P(2)
8	SCALE	Data values for logarithmic display <=0
9	SCALE	{X/Y}P(4)<{X/Y}P(1)+{XY}P(3)
10	SCALE	PLTSET called when the plot system was in the scaled state--tried to use the plot system when it was not in the scaled state.
11	LINE	no. of points passed <=0
12	CHAR	size of chars <=0
13	CHAR	no. of chars <=0
14	CHAR	ICODE <0 or >7
15		reserved
16	LINE	IPN <0 or >7
17-50		reserved
51	PLTSET	Plotter requested not available - Stop 175
52-54		reserved
55	LINE	Bad dash-line control - stop 755
56	LINE	Pathology in vector determination

Appendix D

Examples of Plots and Command Files

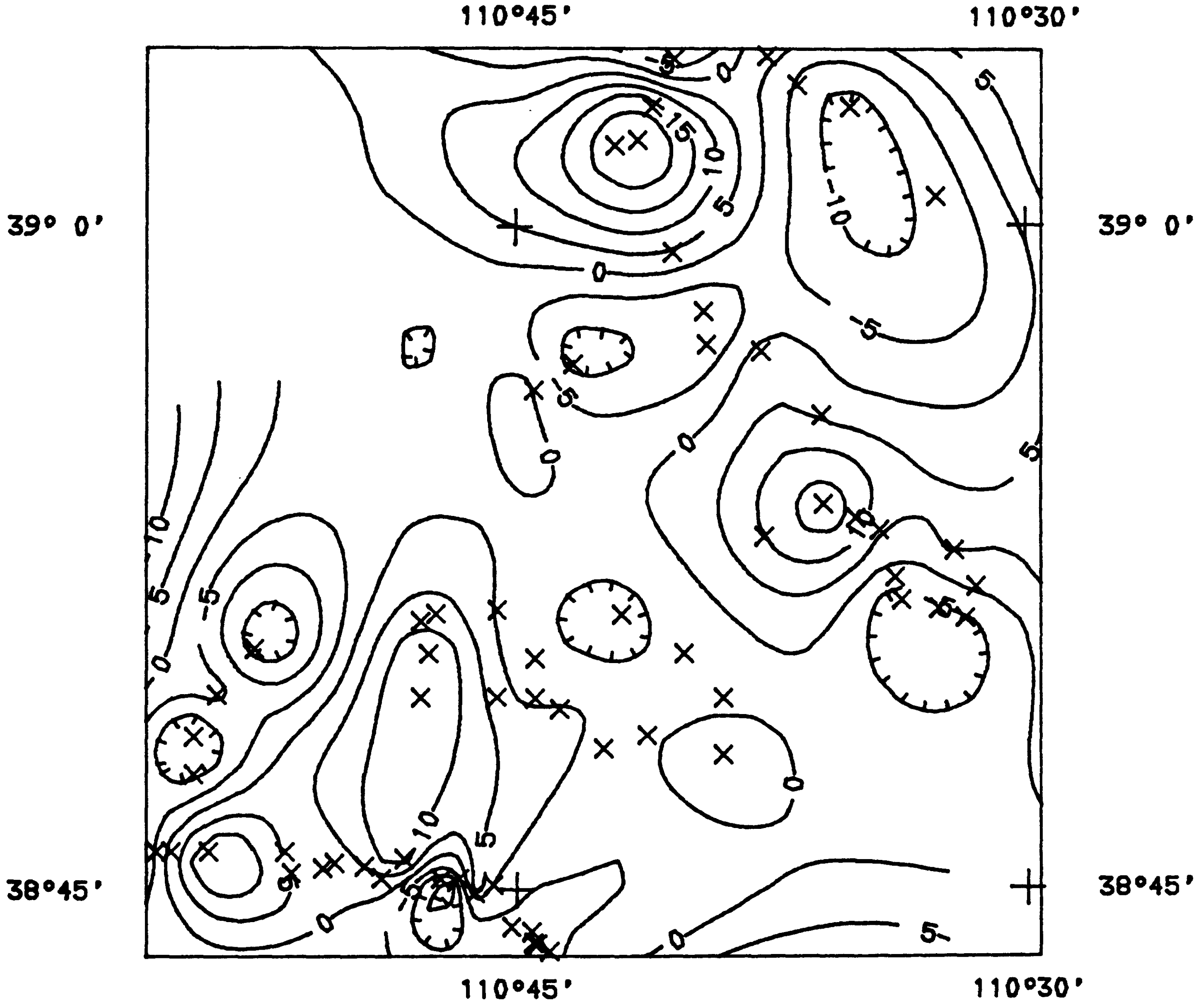


RESIDUAL GRAVITY DATA FROM FIFTH ORDER SURFACE

Command File:

```
$ ifile='test.grd'  iplotr=5
dcval=5, nsec=2, nsig=1,mxhach=1
ncharx=1,nchary=-1
sizel=.12
&
```

Plot Example 1



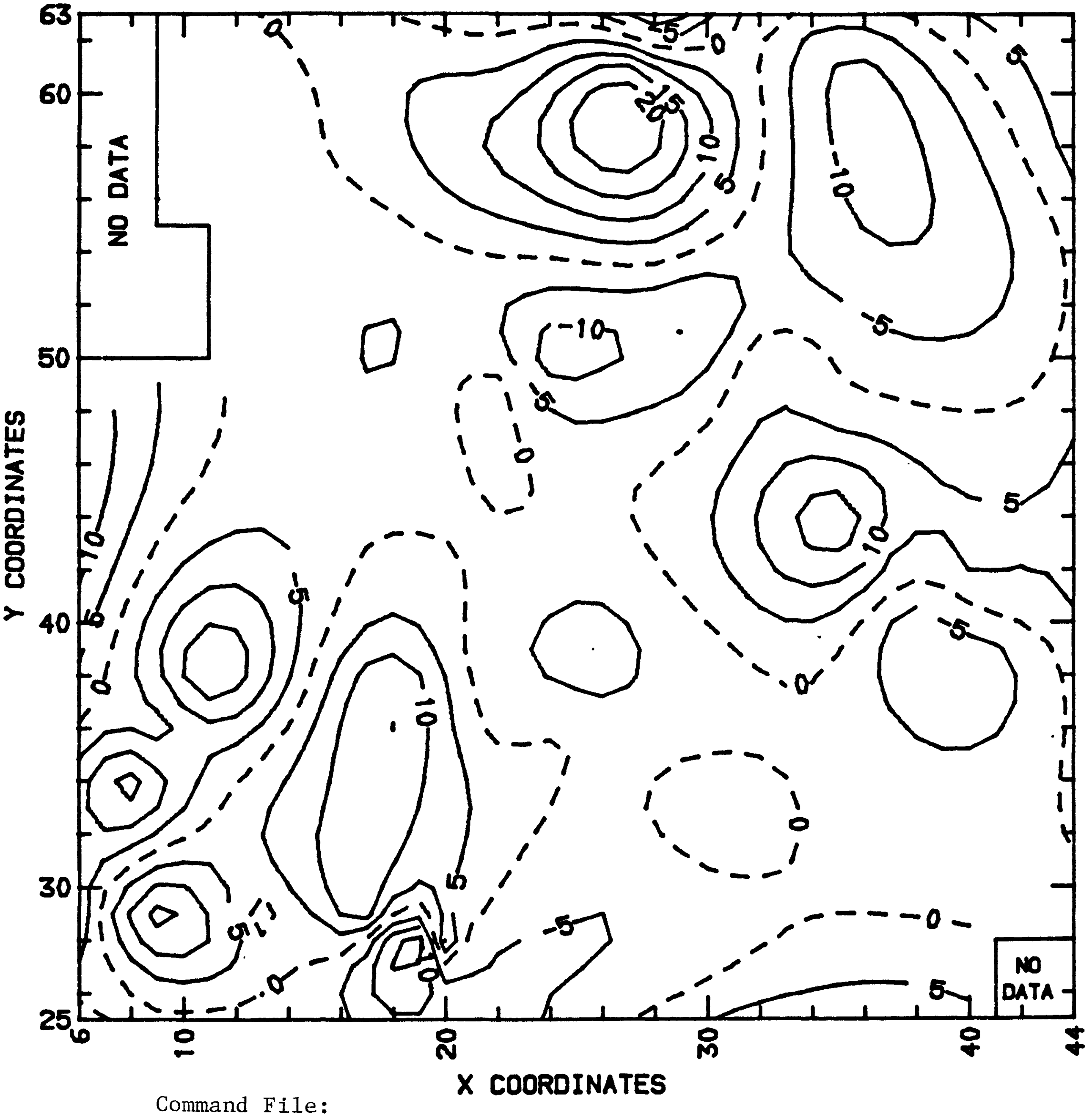
RESIDUAL GRAVITY DATA FROM FIFTH ORDER SURFACE

SHOWING STATION LOCATIONS IN ADDITION TO CONTOURS

Command File:

```
&parms
mxhach=1, ifile='test.grd',
acval=-20,-15,-10,-5,0,5,10,15,20
ncval=9, conlim=.5, nsig=1
ifile2='test.dat',ispost=2, ifmtv-'(3f10.2)'
iplotr=5
cm=-111,baslat=38,30
latm=38,45,latx=39, longm=-110,30,longx=-111
tint=15, itpost=1, iproj=2
title2='SHOWING STATION LOCATIONS IN ADDITION TO CONTOURS'
size1=.12
&
```

Plot Example 2



```
Command File:
& ifile='test.grd',iplotr=5
ncharx=-1 nchary=-1
acval=-15,-10,-5,0,5,10,15,20
jcdash=0,0,0,1,0,0,0,0
ncval=8
txfile='txfile.dat'
lxfile='lxfile.dat',lxfmt='no',lxproj=2 iproj=2
&
```

```
lxfile.dat:
7 50 1
11 50 1
11 55 1
9 55 1
0 63 1
41 25 2
41 28 2
44 28 2
```

```
txfile.dat:
3.6 .1 'X COORDINATES' 3 .1
.8 3 'Y COORDINATES' 3 .1 1.57
7.5 54.5 'NO DATA' 2 .1 1.57
42 27 'NO' 2 .08
41.5 26 'DATA' 2 .08
```

Plot Example 3