DANIEL F. MERRIAM, Editor

GRAFPAC, GRAPHIC OUTPUT SUBROUTINES FOR THE GE 635 COMPUTER

By

F. JAMES ROHLF
The University of Kansas

COMPUTER CONTRIBUTION 36
State Geological Survey
The University of Kansas, Lawrence
1969

in cooperation with the
American Association of Petroleum Geologists
Tulsa, Oklahoma

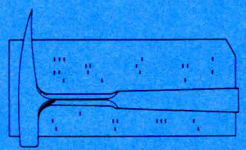
Editor's Remarks

Geologists are accustomed to graphic display. They make maps, cross sections, columnar columns, perspective drawings, charts, etc. to supplement their written descriptions. As with all subjects these illustrations immeasurably enhance the presentation and often are the "key" to acceptance or rejection of ideas. The old Chinese proverb is,if you remember, a picture is worth a thousand words - in many instances probably more, especially for those working in the earth sciences which is mainly a descriptive science.

In a recent poll of those receiving the COMPUTER CONTRIBUTION Series, 62 percent asked for additional programs for plotting and graphic display. This contribution by F.J. Rohlf entitled "GRAFPAC, graphic output subroutines for the GE 635 computer" should help remedy the deficiency of available programs in this area. The program package was developed for use in entomology but is readily adaptable in geology. The entomologists at KU have used the programs very successfully for several years. Now they have made them available to others.

The package is complete and thus the operational instructions are described in considerable detail. Geologists may fine the package difficult to use at first, but use should readily indicate its versatility. Provision is made for plotting scatter diagrams, making perspective plots and stereo pairs of perspective plots. As one reviewer commented, COMPUTER CONTRIBUTION 36 because of its universality will receive an enthusiastic response from the public. We hope so!

For a limited time the Geological Survey will make the package available on magnetic tape for $20.00 (US). An additional charge of $10.00 is made if punched cards are required. For an uptodate list of COMPUTER CONTRIBUTIONS write the Editor, COMPUTER CONTRIBUTIONS, Kansas Geological Survey, The University of Kansas, Lawrence, Kansas 66044, USA.



Several people have asked about the origin of our symbol for the "new geology" - that is the hammer and punched card. Mrs. Beth Clark Kolars of the Geological Survey staff designed it following my instructions to resemble the "regular" Survey symbol of a hammer on the geologic outline of the State of Kansas. No, the punches on the card do not signify anything!

---

*Active Member, °Associate Member, +Junior Member, American Association of Petroleum Geologists.

GRAFPAC, GRAPHIC OUTPUT SUBROUTINES FOR THE GE 635 COMPUTER

by

F. James Rohlf

INTRODUCTION

The graphic Output Subroutines described here are a set of subroutines to communicate graphic output information from a compiler-level language (FORTRAN) to a graphic output device. The form of many subroutines follows specifications set by the SHARE Standard Graphic Output Language Committee, and thus the description of many routines follows closely their report. The package of the Standard Graphic Output Subroutines is called "GRAFPAC".

The routines are designed so that FORTRAN programs written using GRAFPAC to produce plots on one device require minimal changes where other graphic output devices are available. Whereas GRAFPAC has been written to allow use of different plotting devices in the same program the addition of another graphic output device will require changes in several routines. The changes are not made in advance because they will differ slightly for different devices and because I wished to minimize the "overhead" involved if one had in fact only a single graphic output device.

The names of the routines that are a part of GRAFPAC have been chosen in an attempt to select ones that hopefully will conflict as little as possible with names of other subroutines. All names have a five character neumonic followed by a terminal "G", which stands for GRAFPAC.

Not all subroutines proposed by the SHARE committee have been implemented (because they corresponded to facilities not available at The University of Kansas) and some additional routines have been added. The subroutines proposed by the committee are marked with an "*" in the following list.

The graphic output subroutines may be placed into six broad groupings:

A. Display Environment
Included in these types of routines are specification of line width, line color, allowable surface to be used on the drawing board, etc. In addition, there are further capabilities that deal with the characteristics of outputting one logical plot at a time. These are features analogous to writing logical files of information onto a conventional type of output device. The routines and capabilities that are placed in this category are
   1. The AMODES array and MODESG*
   2. DEVONG*, DEVOFG*
   3. SUBJEG*, OBJECG*
   4. CSIZEG*
   5. ADVANG*
   6. FINISG*
   7. FORMSG*
   8. XINCHG, YINCHG
   9. XUNITG, YUNITG
   10. SUBJ3G

B. Point and Line Drawing Capabilities
The routines and capabilities in this category are
   1. POINTG*
   2. LINESG*
   3. SEGMTG*
   4. CIRARG*
   5. ARCITG
   6. HAXESG*, VAXESG*

C. Character Outputting Capabilities
Included here are facilities to draw characters of different size, as well as different fonts. Routines are provided to print text information as a part of an output display. Other routines will print identification information of use to the installation operations people. The routines and capabilities in this category are
   1. LEGNDG*
   2. IDENTG*

D. Utility Routines
Included here are a number of subroutines which are not a part of GRAFPAC but which are useful in programs setting up information to be plotted.
   1. ANGLEG
   2. ANGPLG
   3. CIRTRG
   4. MINMAX
   5. ROTAXG
   6. BOX
   7. TRANSG
   8. FPTBCD

E. Internal Routines
The routines are used to create the computer and device dependent plotting commands. The ordinary user of GRAFPAC need not be concerned with these. They are listed here only for completeness and because they will appear in the storage map.
   1. INCRPG

2. BLPLTG
3. ERRORG
4. ICHARG
5. PLTITG
6. SYMPTG
7. TABDVG
8. TRACBK
9. UNPAKG

F. Specialized Application Routines
1. FXYPLG

Acknowledgments.- The set of routines was designed following many suggestions outlined in "GRAFPAC, Specifications for Standard Graphic Output Subroutines" proposed by the SHARE Standard Graphic Output Language Committee in their report (prepared by P. Pickman) dated 8 August 1966.

Mr. Ron Bartcher did most of the original coding of the routines and his help is gratefully acknowledged. The routines INCRPG and BLPLTG which actually prepare the pen command codes and output them on to tape for the B/L plotter at The University of Kansas were based on routines prepared by Mr. P. F. Smith.

This work was supported, in part, by a grant (GM 11935) from the National Institute of Health to Dr. R.R. Sokal. Computer time was made available by the Computation Center at The University of Kansas and a Benson-Lehner plotter was made available through a gift from the Cities Service Oil Company to the Kansas Geological Survey.

EXAMPLES OF USE OF GRAFPAC ROUTINES

Bivariate Scatter Diagram.

A program to read in N data points and to construct a two-way scatter diagram using labels and scales provided on input parameter cards would consist of the following statements:

```
$       FORTRAN NDECK,NLSTOU
CBIVSCT
       SUBROUTINE BIVSCT
       DIMENSION X(500),Y(500),XLAB(10),YLAB(10),FMT(13)
       CALL MODESG
C      READ PARAMETERS
 1      READ(5,2)N,XMIN,XMAX,YMIN,YMAX,HEIGHT,WIDTH,IWX,IDX,IWY,IDY
 2      FORMAT(I5,6F5.2,4I2)
       IF (N.EQ.0) CALL FINISG
       READ(5,3)NX,(XLAB(I),I=1,NX)
 3     FORMAT(I2,10A6)
       READ(5,3) NY,(YLAB(I),I=1,NY)
C      SET UP SCALES  AND ALLOW FOR MARGINS OF 2,.5,1.5, AND.5 INCHES
       RX=(XMAX-XMIN)/WIDTH
        RY= (YMAX-YMIN)/HEIGHT
       CALL SUBJEG(XMIN-2.*RX,XMAX+.5*RX,YMIN-1.5*RY,YMAX+.5*RY)
       CALL OBJECG(0.,WIDTH+2.5,0.,HEIGHT+2.0)
C      DRAW AXES
        CALL VAXESG(XMIN,1,YMIN,YMAX,11,IWY,IDY,NY*6,YLAB)
        CALL HAXESG(YMIN,1,XMIN,XMAX,11,IWX,IDY,NX*6,XLAB)
C      READ DATA
       READ(5,4) FMT
 4     FORMAT(13A6)
       READ(5,FMT)(X(I),I=1,N)
       READ(5,FMT)(Y(I),I=1,N)
C      PLOT POINTS
       CALL POINTG(X,Y,N,2.)
       GO TO 1
       END
```

Using the data (projections of 75 samples onto the first two eigenvectors) from Wahlstedt and Davis (1968, p. 18) the input card deck would be as follows for the GE 635 system:

```
$       IDENT    XXXX,ROHLF          TEST BIVSCT
$       OPTION   FORTRAN,ERCNT/15/

 *****      PROGRAM DECKS GO HERE      *****

$       EXECUTE
$       LIMITS   05,20000,,2000
$       TAPE     08,A1D,,YYYYY,,,OUTPUT
$       FFILE    08,NSTDLB,NBUFFS/1,BUFSIZ/2000,LODENS,NOSRLS,FIXLNG/1999
   74 5.   65.  -20. 7.   10.  10.  5 1 5 1
 3  EIGENVECTOR 1
 3  EIGENVECTOR 2
(10F8.3)
   29171   24484   43932   48316   47676   50053   37535   43291   42565   52336
   46949   45582   49965   51331   47954   41559   46038   36256   36894   42565
   34246   40645   37632   41198   30586   30347   53065   53065   47042   56444
   47042   51808   52432   48046   61589   59200   55602   30415   25605   46588
   48959   46414   44030   36986   29524   17022   27265   25311   50198   33512
   39186   25492   27620   45667   45667   43027   40008   35324   28163   43659
   45668   52721   50198   54199   55199   46228   56817   31239   31239   08603
   08957   30704   40015   33149
  -17681  -11354   -4677   -7283   -4889    -549   -0078   -2286   -5065  -11281
   -7672   -8058  -10667  -10282   -8671   -4068   -1480    4710    2311   -5065
    6709    2116    5118   -5451  -12216   -3378   -8496   -8496   -2482  -10103
   -2482   -3653   -6086   -3485   -9846   -9276   -3745    1748   -2148   -9056
   -9669   -0059     524  -16807  -16316  -10866   -5506    1593   -0176    3913
   -3458  -12347   -4135   -2888   -2888    1527  -19787  -13454  -16689   -0885
   -2887   -4878   -0176   -4220   -5231  -10436   -8690  -14576  -14576   -4299
   -2932   -2003    4534    2523

$       ENDJOB
```

The printed output produced by GRAFPAC would be as follows:

```
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
START OF GRAFPAC PLOT
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG

GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
END OF GRAFPAC PLOT          3 PLOTS
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
```

The two messages indicate that the GRAFPAC routines have been initialized and that the plotting has been terminated, respectively. Note that three plots have been produced. The first plot writes "GRAFPAC", the second plot is the scatter diagram, and the last plot will always be "END OF GRAFPAC".

The output tape if plotted would produce the graph shown in Figure 1.

If it were desired to plot points sequentially numbered for identification, the following statements could be inserted into the above program:

```
      CALL POINTG(X,Y,N,1.)
       CHWDTG = .12
      CHIGHG = .12
      NDIGITS = 1
      DO 5 I=1,N
      IF(I.EQ.10.OR. I.EQ.100) NDIGITS = NDIGITS + 1
      IALPH =NCHAR(I*10**(6-NDIGITS))
    5 CALL LEGNDG(X(I),Y(I),IALPH,NDIGITS)
```

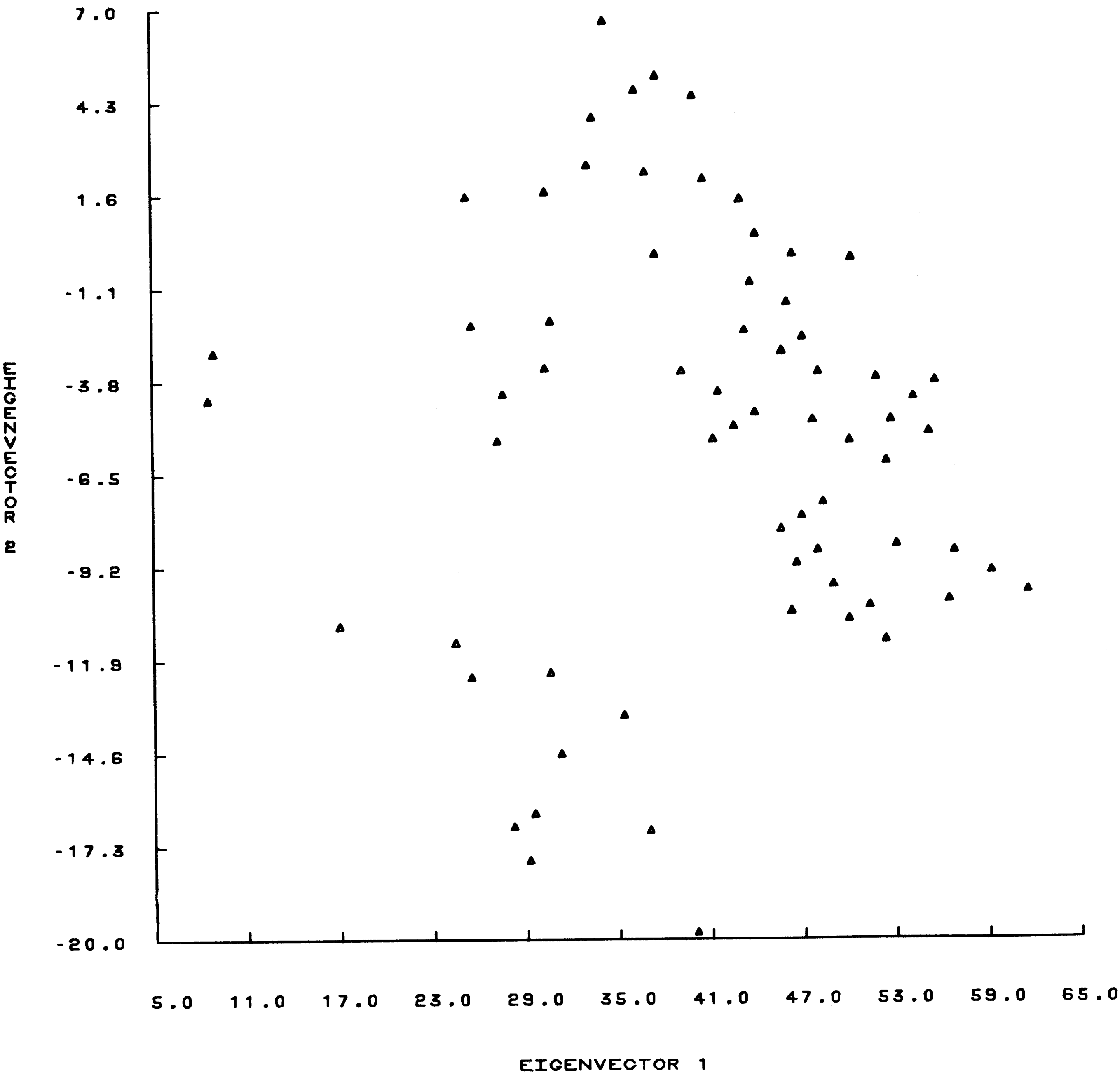


Figure 1.- Two-way scatter diagram produced by program BIVSCT given in text using projection of 74 samples of Pleistocene clays onto first two eigenvectors as given by Wahlstedt and Davis (1968).

Where NCHAR is a utility routine which converts an integer into a six character BCD word. The scaling is required so that the first "NDIGTS" of the BCD word will be the desired label. The results using the same data are shown in Figure 2.

Note that in this simple exemplary program no provision has been made to compensate for overprinting of points and their identifying labels nor have checks been made to test whether the observed data points fall within the limits specified by the input parameters. If a point (e.g., 99., 199.) outside the subject space limits were encountered the GRAFPAC routine PLTITG would detect this fact and the following error message would result:

```
ERROR IN ROUTINE PLTITG
PEN OUTSIDE SUBJECT SPACE LIMITS
N =    26 XLEFT =     -7.0000 XRIGHT =     68.0000 YLOWER =    -24.0500 YUPPER =      8.3500
XLAST =      99.0000 YLAST =    199.0000 TEXTUR =     1.0 ANTENS =     1.0 COLOR =     1.0
FONT =       1. CHWDTH =  0.040 CHIGHT =  0.040
ORIENT =   0.     PLOTCH =   2.000 FSTAT =      1.0
NUMDEV =        1 UNITS =    1.000 JPOINTR =   27 AINCR =0.00500
XCHAR =     0.2500 YCHAR =     3.6200 MAXCHG =  100000 DX =     0.              DY =  17.0000
ENDFLG =  0.   PT01SG =     0.0270
TYPDVC =  20.0 FILCD =      8. XMIN =       0.      XMAX =     12.5000 YMIN =      4.0000 YMAX =     16.0000
XOFF =       1.1667 XSCALE =      0.1667 YOFF =     12.9074 YSCALE =        0.3704 NPLOTS =     2
<*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*>
  TRACE OF CALLS IN REVERSE ORDER   ERROR NO. 01
     CALLING       ID     ABSOLUTE    ARGUMENT        ARGUMENT        ARGUMENT        ARGUMENT        ARGUMENT
     ROUTINE       #      LOCATION       #1              #2              #3              #4              #5
     TRACBK         3     032766      000000000001
     ERRORG        17     035511
     PLTITG        17     044441      474363316327    000000000006    472545204664
     POINTG        18     044047      016614000000    020616000000    000000000003
     BIVSCT        43     045706      016614000000    020616000000    000000000112    004400000000
     .SETU.        40     031262
TRACEBACK
<*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*><*>
```

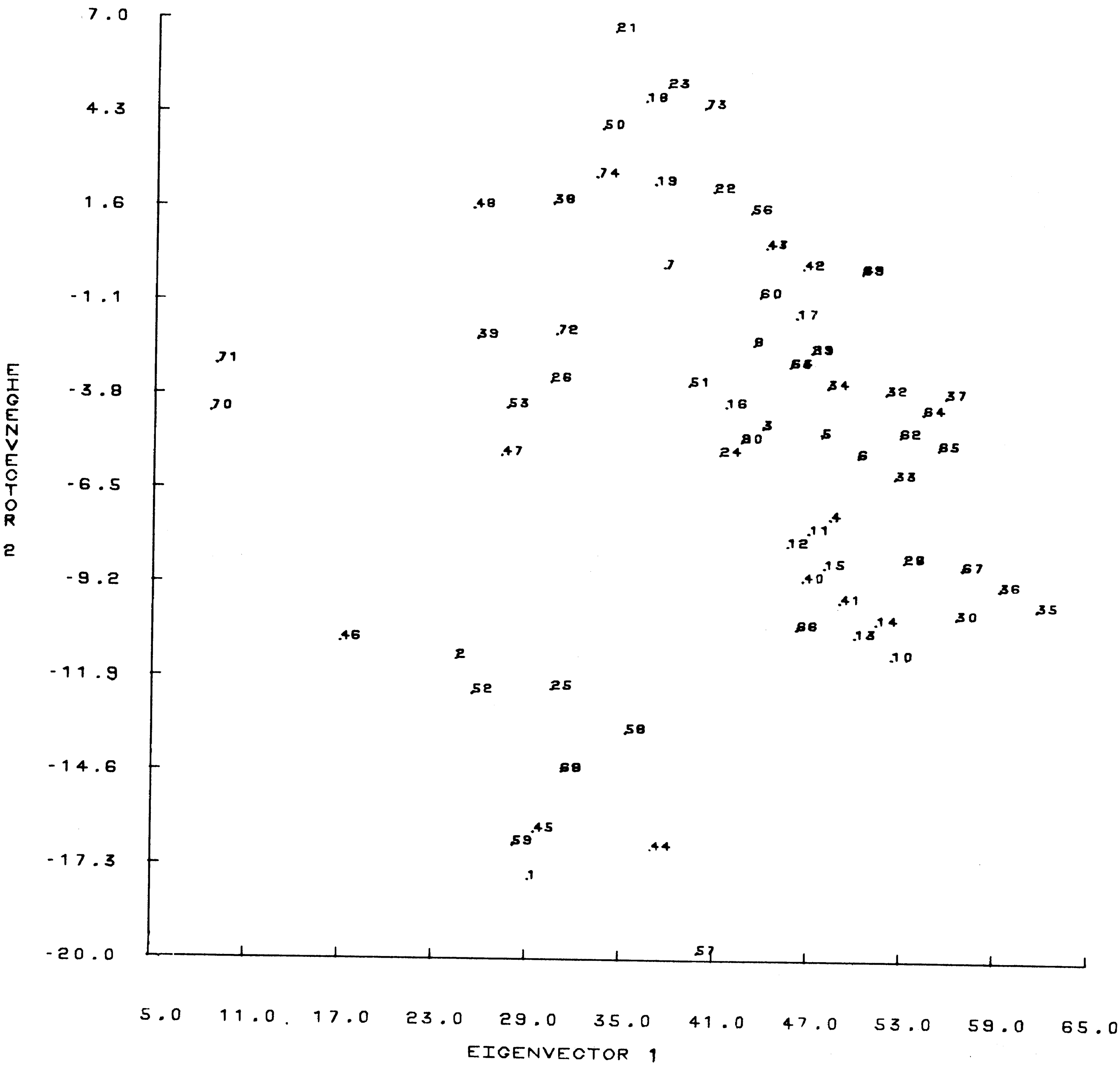


Figure 2.- Data as in Figure 1 but plotted as points and sequentially numbered for identification.

The coordinates of the offending point are given by "XLAST =" and "YLAST=". The trace of subroutine calls is provided as an aid in locating the error in complex programs. This is, of course, a system dependent routine but well worth the effort needed to implement it. All errors of this type are considered fatal.

A program can be constructed easily to prepare a perspective view of a trend surface such as shown in Figure 3.

First a function subprogram must be written which computes the height of the trend surface at each point X, Y. The division by SCALE at the end is to allow control over the vertical scaling of the plot.

Perspective Representation of Trend Surfaces.

```
$       FORTRAN NDECK,NLSTOU                                    FXYTR000
CFXYTRN                                                         FXYTR001
        FUNCTION FXYTRN(X,Y)                                    FXYTR002
        COMMON NP,SCALE,BZERO,B(39)                             FXYTR003
      DOUBLE PRECISION P(39),FXY
      P(1) = X                                                  FXYTR005
      P(2)=Y                                                    FXYTR006
      FXY    = BZERO+B(1)*X+B(2)*Y                              FXYTR007
      IF(NP.LE.1) GO TO 20                                      FXYTR008
      L = 3                                                     FXYTR009
      ISTART = 1                                                FXYTR010
      DO 10 I = 2,NP                                            FXYTR011
       IEND = ISTART + I - 1                                    FXYTR012
      DO 5 J = ISTART,IEND                                      FXYTR013
      P(L) = P(J)*X                                             FXYTR014
      FXY   =FXY   +P(L)*B(L)
 5    L = L+1                                                   FXYTR016
       P(L) = P(IEND)*Y                                         FXYTR017
      FXY   =FXY   +P(L)*B(L)
      L = L+1                                                   FXYTR019
 10   ISTART = IEND+1                                           FXYTR020
 20   FXYTRN= FXY   /SCALE
      RETURN                                                    FXYTR022
      END                                                       FXYTR023
```

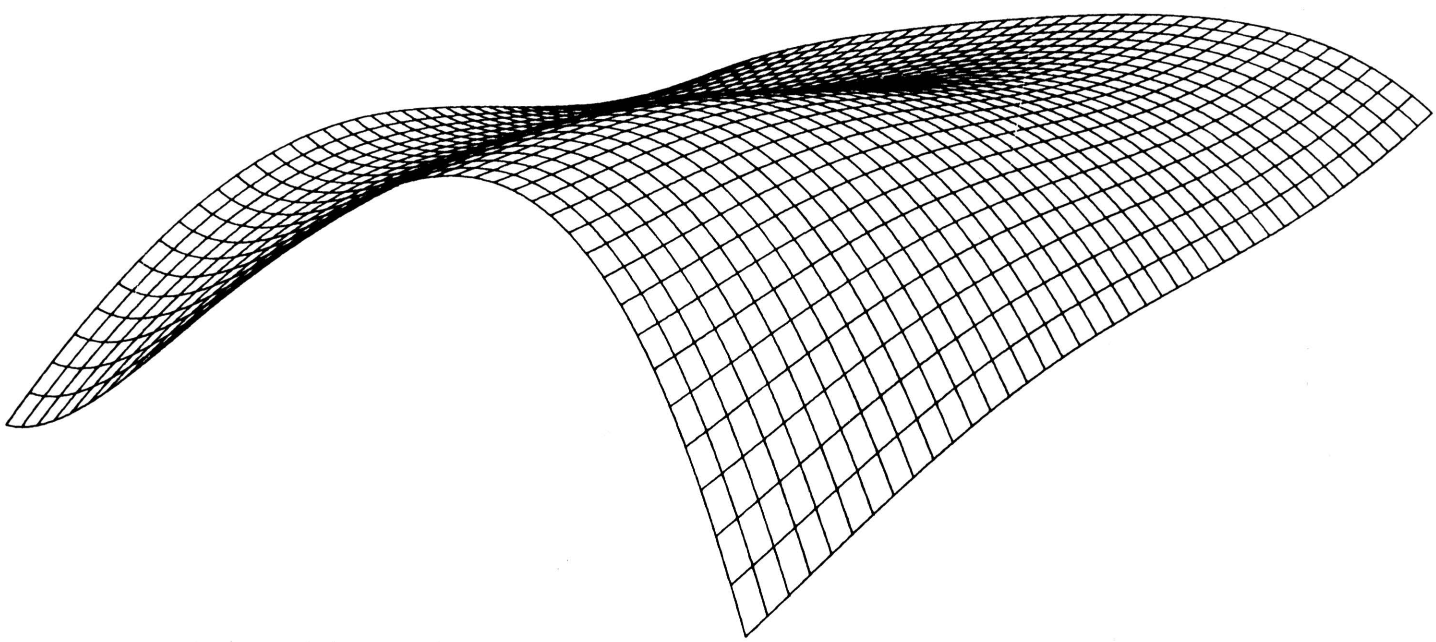


Figure 3.- Perspective plot of 5th-degree trend surface fitted to structure on top of Arbuckle Group (cambrian-Ordovician) as given by O'Leary, Lippert, and Spitz (1966). View is from upper front right of surface.

Then a main program must be prepared which reads in the X and Y dimensions of the plot, scale, coordinates of the viewing position, degree of the polynomial, number of grid lines in each direction, size, and the coefficients of the trend polynomial.

```
$        FORTRAN NDECK,NLSTOU
CTRNPLT
      SUBROUTINE TRNPLT
      REAL FMT(13)
      COMMON NP,SCALE,BZERO,B(39)
      REAL Z(50,50)
      EXTERNAL FXYTRN
      CALL MODESG
 1    READ(5,2) XMIN,XMAX,YMIN,YMAX,SCALE,CX,CY,CZ,NP,NX,NY,NDX,NDY,
     1 SIZE
 2    FORMAT(8F10.5/5I5,F5.2)
      IF(NX.LE.0) CALL FINISG
      NCOEFF = ((NP+1)*(NP+2))/2        - 1
      READ(5,3) FMT
 3    FORMAT(13A6)
      READ(5,FMT) BZERO,(B(I),I=1,NCOEFF)
      CALL ANGPLG(CX,CY,CZ,(XMIN+XMAX)/2.,(YMIN+YMAX)/2.,BZERO/SCALE,
     1 AANG,BANG,GANG,D)
      CALL FXYPLG(NX,NY,NDX,NDY,XMIN,XMAX,YMIN,YMAX,BZERO/SCALE
     1 , BZERO/SCALE,Z, FXYTRN,AANG,BANG,GANG,D,CX,CY,CZ,SIZE)
      GO TO 1
      END
```

A description of the arguments for ANGPLG (which computes the angles between the line connecting the viewing position and the centroid of the model and the X, Y, and Z coordinates axes) and FXYPLG (3-D surface program) are given along with descriptions of other GRAFPAC routines.

The input deck and $ control card sequences would be as for the previous example. The input data cards to construct the 5th-degree surface shown in Figure 3 would be as follows

```
0.         33.5       0.0        -27.0        50.        90.        -90.        -50.
   5   45   45    8    815.
(F10.5,5F13.5)
-279334892  0003001393  0001216366     -157745  0000487662  0000028134
0000006422     -040850  0000010596     -006898     -000111  0000001287
   -000526     -000037     -000605  0000000000     -000013  0000000008
0000000004  0000000004     -000009
```

The polynomial coefficients were taken from O'Leary, Lippert, and Spitz (1966). The X axis will range from 0.0 at the left to 33.5 at the right. The Y axis extends from 0.0 at the back of the model to -27.0 at the front (in order to agree with the conventions used by O'Leary, Lippert, and Spitz (1966), normally Y should increase towards the front of the model). Because the surface is known to range from an elevation of about -3500 feet to -2500 feet a vertical scaling factor of 50 is used to reduce the Z scale to about -70 to -50 (which is more commensurate with the length and width of the model). Relative to the dimensions a viewing position at the right front (X = 75, Y = 110, Z = -50) was used for Figure 3 and the pair of viewing positions at the left front was used for the stereopair in Figure 4 (left: X = -116.75, Y = 76.5, Z = -50; right: X = -106.75, Y = 86.5, Z = -50).

The printed output produced by FXYPLG is as follows for the first example:

```
GGGGGGGGGGGGGGGGGGGG   FXYPLG   GGGGGGGGGGGGGGGGGGG

 GRID ARRAY   45 BY   45 WITH  6 AND  5 LINE SEGMENTS BETWEEN NODES
 VIEWING DIRECTION =  121.0686  148.7551   92.9781 DISTANCE TO PROJECTION PLANE =  112.8704
 VIEWING POSITION =   75.0000  110.0000  -50.0000
 3-D SUBJECT SPACE LIMITS
X     0.        33.5000 Y     0.        27.0000 Z  -69.6551   -49.1578
 SIZE OF PLOT 15.0 BY  8.7 INCHES
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
```

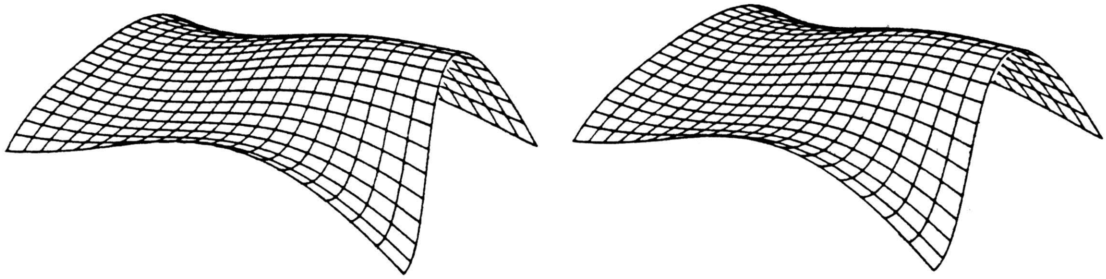


Figure 4.- Stereo pair of perspective plots of 5th-degree trend surface shown in Figure 3. View is from front and left.

For detailed studies of a surface such illustrations do not replace good contour maps, but they are useful to obtain a quick overall impression of the major features of a surface.

DESCRIPTIONS OF SUBROUTINES

Mode Specifications: MODESG

Routine Call:
CALL MODESG (d_1, d_2, . . .)

Routine Description:
A call to MODESG will initialize a block of labeled COMMON which has the label "/AMODES/". The block will contain numeric values corresponding to various general attributes of the output to be generated. Entries in the /AMODES/ block are as shown in Appendix A. The /AMODES/ block is divided into two logical segments: the first segment describes the drawing characteristics and the second describes the device (or devices, d_1, d_2, . . .) upon which it is to be made. The first argument in the /AMODES/ list is the number of words in the "drawing characteristics segment". Note that the block /AMODES/ has (N+1+11*NDEV) entries, where N is equal to the value of the variable in the first word, and NDEV is equal to the number of devices to be used by the "user" of GRAFPAC. At present, only a single device is provided for; hence, N = 26, NDEV = 1, and the total size of /AMODES/ is 38. A call to MODESG will set critical entries of /AMODES/ equal to the default values defined by the installation. Some initialization values are listed in Appendix A. A call to MODESG must precede any call to a GRAFPAC subroutine that specifies output so that entries 8-26 may be initialized. Initialization of all other entries is a user responsibility. Subsequent calls will reset any variables in the /AMODES/ array to values specified by MODESG. If a user wishes direct access to the parameters stored in the /AMODES/ block (this is not usually necessary), he may include the following statements in his program: COMMON /AMODES/ NNNNNG, XLEFTG, XRGHTG, YLWERG, YUPPRG, XLASTG, YLASTG, TEXTRG, INTENG, COLORG, FONTGG, CHWDTG, CHIGHG, ORIENG, PLTCHG, FSTATG, NUMDVG, UNITSG, JJJJJG, AINCRG, XCHARG, YCHARG, MAXCHG, DX, DY, ENDFLG, PTO1SG, DEV (11).

Variables:
d_1, d_2, . . . are a list of the device codes (real numbers) for the graphic output devices to be used by the program.

A device code of 20. has been assigned to the B/L plotter presently available at the KU Computation Center. A list of the possible device codes is:

Value of d	Type of Device
1.- 9.	Microfilm devices (SC-4020. SC-4060, IBM-2280, B/L 120, etc.).
10.-19.	Electromechanical, flatbed plotters such as those made by B/L and EAI.
20.-29.	Electromechanical incremental plotters such as CALCOMP and B/L Draftomatic.
30.-39.	Drafting machines such as the Gerber or Orthomat.
40.-49.	Online display devices.

Display Device: DEVONG, DEVOFG

CALL DEVONG (d) will turn device d "on".
CALL DEVOGF (d) will turn device d "off".
The call

CALL DEVONG (0)

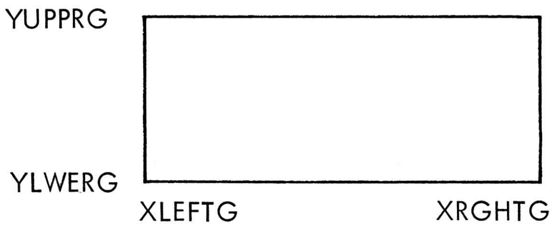
will turn all devices mentioned in previous calls to OBJECG "on". Similarly, the call

CALL DEVOFG (0)

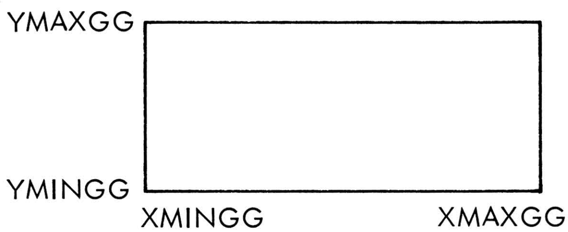
will turn all such devices off. The argument "d" is a FORTRAN real number corresponding to an available device. If only a single plotter is available, programmers need not include calls to these routines in their programs.

Display Mapping: SUBJEG, OBJECG

The user routine will issue two calls to specify the mapping between the "subject space" (the subject that the display represents) and the "object space" (the display itself). The first of these two calls is: CALL SUBJEG (XLEFTG, XRGHTG, XLWERG, YUPPRG) where the subject space has the following coordinates:



The second call is: CALL OBJECG (XMINGG, XMAXGG, YMINGG, YMAXGG, d) where the object space (the output graph) has the following relative coordinates:



The coordinates of the subject space will be represented as floating point quantities; it will be the user's responsibility to see that the units are consistent ones in terms of the subject space. The coordinates of the object space will be in inches, represented as positive floating point quantities. Thus they will be somewhat device-dependent (XMAXGG-YMINGG must be equal to or less than 28.0 inches on the B/L plotter). The variable d specifies the device for which this CALL OBJECG applies. A call to OBJECG will do three things:

a. specify the object space to be used to form the display,
b. make an appropriate entry in the /AMODES/ block for the device (providing sufficient space in the /AMODES/ block is a user responsibility; /AMODES/ must be dimensioned sufficiently large, 38 words at present.), and
c. turn device d "on" as though an explicit call to DEVONG had been executed.

In the situation of incremental (as opposed to absolute) plotting equipment, the initial placement of the "pen" on the drawing surface must be taken care of in some manner by OBJECG. The assumed initial placement of the "pen" will be at the point (XMINGG, YMINGG).

Example: To prepare an 8 x 10 inch graph of information which ranges from 15 to 33 mm. on the X axis and from 0 to 25 g. on the Y axis, the user would use the following pair of CALL statements:

CALL SUBJEG (15., 35., 0., 20.)

CALL OBJECG (0., 8., 0., 10.)

All scaling will be done automatically by the graphic output routine in terms of the mapping transformation specified in the above two calls. If the user desired to maintain geometric similarity, he would have to provide for it in the two calls; if the user chose not to maintain geometric similarity, he would have the option to do so.

Get Standard Character Size: CSIZEG

Routine Call:

CALL CSIZEG (d)

Routine Description:

The user routine will be able to interrogate the Graphics Output Package (GRAFPAC) to request the standard character size in terms of an object space dimension unit. Note that the user routine may first call CSIZEG and then compute values for the OBJECG call (so that the plot is made in proportion to the size of the standard characters). Or, alternatively, the OBJECG call may be changed in the user's program to correspond to any change in the display device being used.

Variables:

The standard character size dimensions for the plotting device "d" will be placed in the /AMODES/ block (words, CHIGHG, CHWDTG) by this subroutine:

CHIGHG: The character envelope height (allowing for vertical clearance) in object space units (inches).

CHWDTG: The character envelope width (allowing for horizontal spacing) in object space units (inches).

d: The variable "d" is the device number for which the character dimensions are requested.

Advance Frame: ADVANG

Routine Call: CALL ADVANG

Routine Description:
This completes plotting on the current drawing and positions new paper (or film) or signals the plotter operator to change paper. The plotter will be set to prepare another plot the same size as the present one (if another plot of a different size is to be made, one should call OBJECG). If desired, the call to ADVANG may be followed by a call to SUBJEG in order to change the dimensions of the subject space without changing the dimensions of the object space.

Plotting Finished: FINISG

Routine Call: CALL FINISG
Routine Description:
A call to this routine signifies the end of all plotting. It is a signal to GRAFPAC to take whatever action is required to wrap up the plotting output (for example, rewind the plotting tape).

Form Flash: FORMSG

Routine Call: CALL FORMSG (arg)
Routine Description:
This routine will control Form Flash (to display a fixed form) when such a feature is available on the display device (s) being used. This feature is not available at present.
Variable:
arg: The variable "arg" is a FORTRAN floating point number. If "arg" is 0., the Form Flash will be turned "off". If "arg" is 1., the Form Flash will be turned "on". The status of the "Form Flash" will be reset to "off": at the beginning of each logical frame. The Form Flash status word is word FSTATS OF /AMODES/.

Subject to Object Space Units: XINCHG and YINCHG

Routine Calls: XO = XINCHG (XS)
YO = YINCHG (YS)
Routine Description:
These function subprograms transform X and Y values in subject space units into object space units (in inches).
Variables:
XS X and Y coordinates in subject space units.
YS

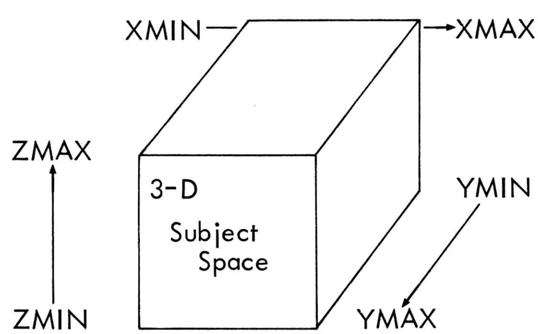
Object to Subject Space Units: XUNITG and YUNITG

Routine Calls: XS = XUNITG (XO)
YS = YUNITG (YO)
Routine Description:
These function subprograms transform X and Y values on object space units (inches) into subject space units.
Variables:
XO X and Y coordinates in object space units.
YO

Three-dimensional Display Mapping: SUBJ3G

Routine Call: CALL SUBJ3G (XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX, TRANSG)
Routine Description:
This subroutine is provided to facilitate the computation of the subject space limits in terms of a perspective view of a 3-dimensional space. The first six arguments the subject space limits for the X, Y, and Z coordinate axes and the last argument is the name of the subroutine used to compute the perspective transformation, e.g. TRANSG.

Variables:	Description
XMIN, XMAX YMIN, YMAX ZMIN, ZMAZ	These are the minimum and maximum values for the X, Y, and Z directions (FORTRAN REAL variables).
TRANSG	Name of the function used to compute the perspective transformation. This name also must be included in an EXTERNAL statement within the program. One can furnish their own routine or use the general routine TRANSG.



POINT AND LINE DRAWING CAPABILITIES

Plot Point: POINTG

Routine Call: CALL POINTG (X, Y, count, (plot character code))
Routine Description:
POINTG produces a special plot symbol at X, Y. The plot symbol used is that specified either in word PLTCHG of /AMODES/ or as the last optional argument. The following codes are used to specify a particular plot symbol:

1. •
2. Δ
3. ∇
4. □
5. X

The coordinates of the last (X,Y) will be placed in words XLASTG and YLASTG, respectively, of /AMODES/.

Variables:

X,Y: Specified in terms of the "subject space". The real variables X and Y indicate starting addresses of singly dimensioned real arrays.

count: A FORTRAN integer equal to the number of X,Y pairs specifying points to be plotted.

plot character code: A FORTRAN real number which is the plot symbol code. If this argument is not included in the call, then the code specified by word PLTCHG of /AMODES/ will be used.

Plot Contiguous Lines: LINESG

Routine Call: CALL LINESG (X,Y, count)

Routine Description:

LINESG produces connected line segments joining points specified by the X,Y array. The coordinates of the last (X,Y) will be placed in words XLASTG and YLASTG, respectively, of /AMODES/.

Variables:

X,Y: Specified in terms of the "subject space". The real variables X and Y specify starting addresses of singly dimensioned arrays.

count: A FORTRAN integer equal to the number of point pairs in the X,Y array. The value of count specifies, in a sense, the line drawing action to be taken. For example, the alternatives are:

"Count" Value	Action
0≤	Move the imaginary drawing pen to location X,Y. Draw no line.
1	Move the imaginary drawing pen from the last point plotted (specified in words XLASTG and YLASTG of /AMODES/) to the X,Y point specified in this call. Draw a line - i.e., move the "pen" with its point down.
≥2	Draw connected line segments, starting at the first point specified in X,Y array, and terminating at point number "count".

Plot Line Segments: SEGMTG

Routine Call: CALL SEGMTG (XI, YI, XT, YT, count)

Routine Description:

SEGMTG produces noncontiguous line segments. The initial X,Y coordinates of each segment are specified in the arrays XI, YI; the terminal X,Y coordinates are specified in the arrays XT, YT. The coordinates of the last (XT, YT) will be placed in words XLASTG and YLASTG, respectively, of /AMODES/.

Variables:

XI, YI: Specified in terms of the "subject space". The real variables XI, YI, XT specify starting addresses of singly dimensioned arrays. The XI array and the YI array, taken together, from the XI, YI array which specifies initial coordinates of each line segment.

XT, YT: Similarly, the XT and YT arrays taken together form the XT, YT array which specifies terminal coordinates of each line segment. A typical line segment would be drawn from (XI(i), YI(i)), to (XT(i), YT(i)).

count: A FORTRAN integer equal to the number of line segments to be drawn. If "count" is less than or equal to 0, no line segments will be drawn.

Plot Circular Arc: CIRARG

Routine Call: CALL CIRARG (XC, YC, radius, start, arc length))

Routine Description:

CIRARG will draw an arc of a circle with center at (XC, YC) and radius as specified. The arc will be drawn starting at "start" degrees and will be "arc length" degrees in extent. The coordinates of the last point plotted on the arc (X,Y) will be placed in words XLASTG and YLASTG, respectively, of /AMODES/. If the X and Y axes are not to the same scale, then an ellipse will result.

Variables:

XC, YC: Specified in terms of the "subject space". The variance XC and YC specify locations at which values for the X and Y coordinates of the arc center are stored.

radius: Specified in terms of the "subject space".

start: The number of degrees counterclockwise from the horizontal at which the arc is to start. If "start" is negative, the starting angle at which the curve begins will be measured clockwise from the horizontal.

arc length: The number of degrees over which the arc is to extend. If "arc length" is positive, the extent of the arc is counterclockwise; if negative, the extent of the arc is clockwise. If "arc length" is 0, a point will be drawn. See Figure 5 for a graphic description of the definitions of the arguments of subroutine CIRARG.

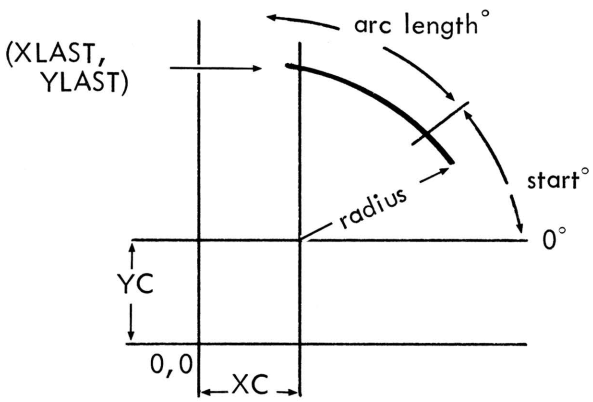


Figure 5.- Graphic display of subroutine CIRARG.

XCENTR, YCENTR: The coordinates of the center of the circle.

ERROR: This quantity specifies the maximum error which is tolerable when approximately a circular arc by a series of chords. ERROR is the maximum distance between the chord and the arc. ERROR should be specified in subject space units.

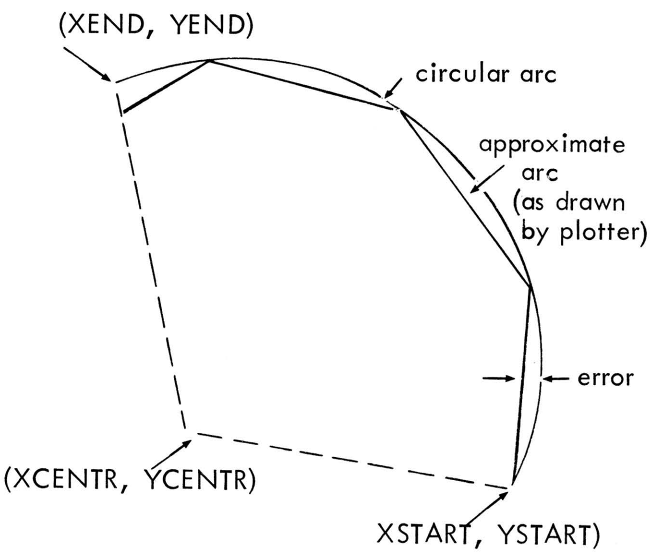


Figure 6.- Graphic display of plotting circular arc.

Plot Circular Arc: CALL ARCITG (XSTART, YSTART, XEND, YEND, XCENTR, YCENTR, ERROR)

This routine provides an alternative method for specifying how a circular arc is to be drawn. ARCITG will draw an arc of a circle with center at (XCENTR, YCENTR) counterclockwise from a point at (XSTART, YSTART) to (XEND), YEND). XEND and YEND will be placed in words XLASTG and YLASTG, respectively, of /AMODES/ block (Figure 6). If the X and Y axes are not to the same scale, then an ellipse will result.

Variables:

XSTART, YSTART: Specified in terms of "subject space". These two variables specify the coordinates of the start of the circular arc.

XEND, YEND: In a similar fashion these specify the coordinates of the end of the circular arc. Note: the arc is drawn in a counterclockwise direction.

Plot Axes: HAXESG and VAXESG

Routine Call: CALL HAXESG (Y, count, BEGINX, ENDX,NTICS, IWIDTH, IDECML, NCHAR, LABEL)
CALL VAXESG (X, count, BEGINY, ENDY, NTICS, IWIDTH, IDECML, NCHAR, LABEL)

Routine Description:

Because of their relationship, the routines will be described together.

HAXESG draws lines in the X direction (Horizontal lines).

VAXESG draws lines in the Y direction (Vertical lines).

The coordinates of the last point plotted (ENDX, Y) or (X, ENDY) will be placed in XLASTG and YLASTG, respectively, of /AMODES/.

Variables:

Y: The starting address of a singly dimensioned real array holding Y coordinates of the X axes in the "subject space".

X: The starting address of a singly dimensioned real array holding X coordinates of the Y axes in the "subject space".

count: A FORTRAN integer equal to the number of point pairs in the X and Y arrays (i.e., the number of axes to generate).

BEGINX, BEGINY: The X coordinate at which the X axis is to begin, specified in the "subject space". An analogous definition holds for BEGINY for a Y axis.

ENDX, ENDY: The X coordinate at which the X axis ends, specified in the "subject space". An analogous definition holds for ENDY for a Y axis.

NTICS: The absolute value of NTICS is number of tic marks to be placed along the axes. If NTICS is positive, the tic marks are placed in their normal position on the right or upper side of the axis. If negative they are placed on the left or lower side of the axis. If this argument is omitted, then no tic marks will be drawn.

IWIDTH, IDECML: If these arguments are present then the tic marks will be labeled with their numerical values using an Fw.d Format. IWIDTH = w and IDECML = d.

NCHAR, LABEL: If these optional arguments are present, then the alphanumeric label of NCHAR characters will be used to label the axes. NCHAR is an integer number and LABEL is an array containing the BCD characters packed 6 per word.

If the tic marks and axes are to be labelled then the call to SUBJEG must allow for this plotting to be done in the margin. If one wishes to construct a plot of size W by H inches, one must at least allow margins as follows (in inches): left margin = (IWIDTH + 4)* CHWDTG, right margin = IWIDTH/2 *CHWDTG, upper margin = CHIGHG/2., lower margin = 4.*CHIGHG. To compute the needed dimensions, compute the two ratios RX =(MAX-XMIN)/W, RY = (Y-MAX-YMIN)/H. The ample margins would be given by the following calls:

```
CALL SUBJEG (XMIN-2.*RX, X-
MAX + .5*RX, YMIN-1.5*RY,
YMAX + .5*RY)
CALL OBJECT (0., W + 2.5, 0.,
H + 2.)
```

An example of the output produced by these routines is given in Figures 1 and 2. The axes and tic marks will be drawn and labeled with an F5.2 format as a result of the following calls:

```
CALL HAXESG (YL, 1, XL, XR,
11, 5, 2, 1, 1HX)
CALL VAXESG (XL, 1, YL, YU,
11, 5, 2, 1, 1HY)
```

CHARACTER DISPLAY CAPABILITIES

There are two routines to generate text output. The first is intended to be used either to label graphic output or to be used by itself to generate text information. The second is intended to provide the programmer a way of communicating with the operator of the graphic output device. Using pen plotters, programmers frequently need to write output messages to the operator of the plotter.

Plot Legend: LEGNDG

Routine Call: CALL LEGNDG (X, Y, alpha, count)

Routine Description:
This routine will output characters (alphanumeric and special characters) starting at locations X,Y. XLASTG and YLASTG of /AMODES/ are unaffected, but the X and Y coordinates (in subject space units) of the lower right-hand corner of the last character to be plotted will be placed in XCHARG and YCHARG in the /AMODES/ block. The angle at which explanation information is to be written is specified in word "ORIENG" of the /AMODES/ block. The angle will be measured in degrees counterclockwise from the horizontal. If the number of characters exceeds MAXCHG (in the /AMODES/ block) or if the line of characters exceeds the limits of the subject space then a "carriage return" is performed and the outputting of characters continues on the next lower line. All characters in the GE 635 character set have been allowed for in character font number 1. Additional characters can be added to the system as font numbers, 2, 3, etc. if the coding for the additional character is added to the tables in routine LEGNDG. Users of other systems may wish to expand and rearrange the character set to correspond to the USASCI character set.

Variables:

X, Y: The location in the "subject space" at which the character string is to start.

alpha: The location of the beginning of the array where the (computer internal) character codes are stored.

count: A FORTRAN integer equal to the numbers of characters to be output.

Examples:
CALL LEGNDG (X, Y, 18HTHIS IS AN EXAMPLE, 18).
If MAXCHG is set to 1, then CALL LEGNDG (X, Y, LABEL, 4) would cause the contents of LABEL (e.g., "VERT") to be plotted as

V
E
R
T

The 64 characters presently allowed for in LEGNDG are shown below. They correspond to the GE character set for octal numbers 00 through 77.

0123456789[#@:>? ABCDEFGHI&.
](<\↑JKLMNOPQR-$*);'+/STUVWX
YZ←,%=" |

Plot Identification Information: IDENTG

Routine Call: CALL IDENTG (alpha, count, FLAG)

Routine Description:
This routine serves to identify plot batches or write instructions for the plotter operator on electro-mechanical plotters. In addition, using LEGNDG, it will write a line of Hollerith narrative specified by the calling routine. XLASTG and YLASTG of /AMODES/ are unaffected. In the present implementation, the narrative will be placed in the one-inch margin between plots.

Variables:
Alpha: The location of an array of BCD characters. The characters will be specified by the user program and will provide communication ability to the graphic output device (offline or online device) operator where communication ability exists.
count: The number of characters to be plotted.
FLAG: If this argument (a real number) is present and not equal to zero, then the plotter (if set for single rather than multiplot) will stop at the end of the message to allow change of pen points or other manual operation.

UTILITY ROUTINES

Compute Angle of Line Connecting Pair of Points: ANGLEG

Routine Call: CALL ANGLEG (X1, Y1, X2, Y2).

Routine Description:
The routine ANGLEG computes the angle (in radians) which a line connecting two points makes with the X axis (Figure 7).

Variables:
X1, Y2: Coordinates of the first point.
X2, Y2: Coordinates of the second point.
The function returns the angle of the segment as a real number in radians.

Angle Between Line and Coordinate Axes.

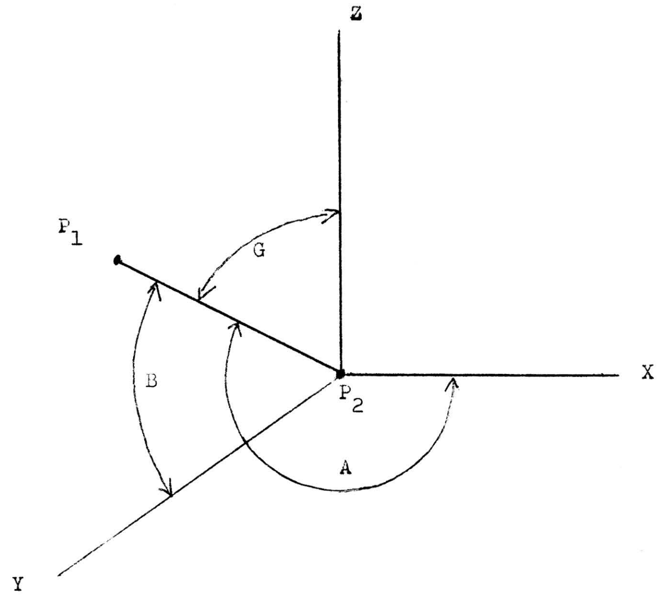


Figure 7.- Graphic display for routine ANGLEG.

Routine Call: CALL ANGPLG (CX, CY, CZ, CCX, CCY, CCZ, A, B, G, D)

Routine Description:
This routine computes the distance between two points and the angles which a line connecting the two points makes to the X, Y, and Z coordinate axes. This routine is used on conjunction with routine ITRANG.

Variables:	Description
CX, CY, CZ	Coordinates of first point (viewing position).
CCX, CCY, CCZ	Coordinates of second point (center of projection plane).
A, B, G	Angles which the line of sight from the first point to the second point makes with the X, Y, and Z axes, respectively.
D	Distance between the two points.

Line Segment Between Two Circles: CIRTRG

Routine Call: CALL CIRTRG (X1, Y1, X2, Y2, R1, R2)

Routine Description:
Given two circles with coordinates X1, Y1, and X2, Y2 with radii R1, and R2, respectively, this routine replaces the values for X1, X2, Y1, Y2 with the coordinates of the end points of a line connecting the two circles and intersecting the outside of the circle rather than connecting the centers of the circles.

Variables:
X1, Y1: Coordinates of the center of the first circle.
X2, Y2: Coordinates of the center of the se-

cond circle.
R1, R2: Radii of the two circles.

Find Minimum and Maximum of Array: MINMAX

Routine Call: CALL MINMAX (count, X, XMIN, XMAX)

Routine Description:

The subroutine searches the given array and returns smallest and largest values found in the array.

Variables:

count: An integer specifying the length of the array.

X: The starting address of a singly dimensioned real array holding the list of values to be searched.

XMIN, XMAX: The smallest and largest element found in the array X. Returned as real numbers.

Rotate Axes: ROTAXG

Routine Call: CALL ROTAXG (X, Y, count, angle)

Routine Description:

This subroutine performs a rigid rotation of the data points in the arrays X and Y relative to the X and Y coordinate axes. The X axis will be rotated towards the Y axis by the specified angle (in degrees).

Variables:

X, Y: The X and Y coordinates of the data points.

count: The length of the X and Y arrays specified above.

angle: The angle in degrees by which the axes are to be rotated. A positive angle indicates a rotation counterclockwise.

Coordinates of Corners of a Rectangle: BOX

Routine Call: CALL BOX (XL, XR, YL, YU, X, Y)

Routine Description:

Given the minimum and maximum X and Y dimensions of the rectangle this routine places in the arrays X and Y, the X and Y coordinates of the four corners of the rectangle. In addition, the first coordinate is repeated after the last corner so that a single call to the routine LINESG can draw the rectangle. The rectangle may be drawn at an angle by calling the routine ROTAXG before calling LINESG.

Variables:

XL, XR: Minimum and maximum dimensions of the rectangle along the X axis.

YL, YU: Lower and upper dimensions of the rectangle along the Y axis.

X, Y: Starting addresses of two arrays to hold the X and Y coordinates of the four corners of the rectangle. The length of the arrays must be at least 5.

Three-dimensional Perspective Transformation: ITRANG and TRANSG.

Routine Calls: CALL ITRANG (A, B, G, D, CX, CY, CZ)
CALL TRANSG (N, X, Y, Z, XT, YT)

Routine Description:

A call to the ITRANG subroutine is necessary to initialize parameters for a perspective projection of a series of points from a 3-dimensional space to a 2-dimensional space. The mathematical method used is described in a paper by Kubert, Szabo, and Giulieri (1968). The arguments for ITRANG give the angles of the X, Y, and Z axes, respectively, to the line of sight, the distance between the viewing position and the projection plane, and the X, Y, Z coordinates of the position. The arguments for routine TRANSG are the number of points to be tranformed, and the X, Y, Z coordinates of the points. This routine returns the X, Y coordinates of the projected points on the projection plane. If it is desired to have the projection plane at the centroid of the subject space and orthogonal to the line of sight then the subroutine ANGPLG will compute the angles A, B, G and the distance D needed for ITRANG given the coordinates of the viewing position and the coordinates of the centroid of the figure.

Variables:	Description
Initialization subroutine ITRANG	
A, B, G	Angles (in degrees) which the projection plane makes with the X, Y, Z axes, respectively.
D	Distance from the viewing position to the projection plane.
CX, CY, CZ	X, Y, and Z coordinates of the viewing position. (Real numbers, in subject space units).
Projection subroutine TRANSG	
n	Number of points to be projected.
X, Y, Z	Arrays giving the X, Y, and Z coordinates, respectively, of the n points in the 3-dimensional space.
XT, YT	X and Y coordinates of the n points on the projection plane.

These values are computed by the subroutine and may be plotted using the standard GRAFPAC subroutines.

Convert Floating Point Number of BCD: FPTBCD

Routine Call: CALL FPTBCD (X, w, d, array)
Routine Description:
This program converts a floating point number (FORTRAN REAL) to BCD using an Fw.d format so that the number can be plotted using routine LEGNDG.
Variables:

X	Floating point number to be converted.
w	Width of the desired BCD field (integer number).
d	Number of places to the right of the decimal place desired (integer number).
array	A FORTRAN array of sufficient length to hold BCD number. The array must be at least (w + 5)/6 words long.

For example, to plot the square root of Z at X, Y with an F5.2 format:

```
CALL FPTBCD (SQRT(Z), 5, 2, NUM)
CALL LEGNDG (X, Y, NUM, 5)
```

INTERNAL ROUTINES

A number of routines are needed to perform the actual plotting (placing pen commands on an output file). A list of these routines appears in the Introduction and the programs themselves are listed in Appendix B.

SPECIALIZED APPLICATION ROUTINES

1. Plot function of two variables (with hidden lines removed).

Routine Call: CALL FXYPLG (NX, NY, NDX, NDY, XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX, Z, FXY, A, B, G, D, CX, CY, CZ, SIZE)

Routine Description:
This subroutine produces a perspective plot of an arbitrary function of two variables (defined in the user supplied function subprogram FXY (X, Y)) as a surface over the range XMIN ≤ X ≤ XMAX and YMIN ≤ Y ≤ YMAX. The surface is depicted using a "fish net" grid. Lines on hidden surfaces are not plotted. Hidden points are determined by the method of Kubert, Szabo, and Giulieri (1968). NX grid lines will be placed along the X axis and NY along the Y axis. NDX and NDY indicate how many straight line segments are to be used to plot line segments between grid intersections in the X and Y directions respectively. The larger the number the smoother the graph will be (and the more computer time required). The perspective view is determined by the viewing position (at the point CX, CY, CZ) relative to the surface to be plotted, the viewing direction (at angles of A, B, and G degrees from the X, Y, and Z coordinate axes, respectively), and the distance, D, between the viewing position and the projection plane. In order to use this program, a short main program must be written which: (a) calls MODESG; (b) determines XMIN, XMAX, YMIN, YMAX, (determination of ZMIN, ZMAX is optional), and the viewing position parameters, etc; (c) calls FXYPLG for each graph to be plotted; and (d) calls FINISG after the last plot. In addition a subprogram must be prepared which computes the desired function. This function must be written as a real function with two arguments X and Y (in that order). If additional parameters are needed by this function they must be placed in COMMON. The main program must contain an EXTERNAL statement containing the name of this function.

Variables:	Description
NX, NY	The number (integer) of grid lines in the X and Y directions, respectively. These are also the dimensions of the array Z required by the program (see below).
NDX, NDY	The number (integer) of straight line segments (≤10) to be used to approximate curved line segments between the NX x NY grid intersections. Larger values will result in a smoother appearance but will require additional computer time. A value of 4 or 5 appears satisfactory for many applications.
YMIN, XMAX YMIN, YMAX ZMIN, ZMAX	Minimum and maximum value of the X, Y, and Z variables, respectively. If ZMIN=ZMAX the subroutines find the minimum and maximum values of the function. If ZMIN≠ ZMAX than any parts of the surface which extend outside of the interval ZMIN ≤ Z ≤ ZMAX will be truncated. These are FORTRAN REAL variables.
Z	An array of at least NX x NY elements used by the program to store the value of the function of each grid intersection and a special code indicating

	whether or not the point is visible.
FXY	The name of the user supplied function subprograms which computes the height of the surface given the X and Y coordinates. This name also must appear in an EXTERNAL statement.
A, B, G	Angles (in degrees) which the line of sight from the viewing position to the projection plane makes with X, Y, and Z coordinate axes, respectively. For a front view, angles of 90°, 0°, and 90° are used.
D	Distance from the viewing position to the projection plane (a FORTRAN REAL variable). The exact value of this variable is not critical. A value equal to the distance between the viewing position and the centroid of the surface is convenient.
CX, CY, CZ	The coordinates (FORTRAN REAL variables) of position from which the surface is to be viewed. The exact values depend upon the nature of the surface and the effect desired. As a start one might use the following: CX = XMIN + .3 (XMAX-XMIN) CY = YMIN + .4 (YMAX-YMIN) CZ = ZMIN + 1.5 (ZMAX-ZMIN)
SIZE	The maximum dimension (length or width, whichever is the largest) of the finished plot (a FORTRAN REAL variable, in inches).

An example of the output produced by this routine is given in Figure 3.

REFERENCES

Kubert, B., Szabo, J., and Giulieri, S., 1968, The perspective representation of functions of two variables: Jour. Assn. Computing Mach.,v. 15,p. 193-204.

O'Leary, M., Lippert, R.H., and Spitz, O.T., 1966, FORTRAN IV and MAP program for computation and plotting of trend surfaces for degrees 1 through 6. Kansas Geol. Survey, Computer Contr. 3, 48 p.

Wahlstedt, W.C., and Davis, J.C., 1968, FORTRAN IV program for computation and display of principal components: Kansas Geol. Survey, Computer Contr. 21, 27 pp.

APPENDIX A. - Description of Entries in /AMODES/ Block.

Position in block	Symbolic Designation	Description of Quantity	Initialization (Standard) Value for Present System
1	NNNNNG	-Number of words to follow describing the drawing (integer value).	26
2	XLEFTG	Dimensions of Subject Space. Set by SUBJEG call.	0.0
3	XRGHTG		0.0
4	YLWERG		0.0
5	YUPPRG		0.0
6	XLASTG	The last point plotted by a line plotting routine (Subject Space coordinates).	0.0
7	YLASTG		0.0
8	TEXTRG	-Texture specification for lines (solid versus dotted lines, etc.)	1.0
9	ANTENG	-Intensity specification for lines and points.	1.0
10	COLORG	- Color specification for lines, points and characters (not available at present).	1.0
11	FONTGG	- Font specification for characters	1.0
12	CHWDTG	- Character envelope width in terms of Object Space Dimension Units	0.16
13	CHIGHG	- Character envelope height in terms of Object Space Dimension Units.	0.16
14	ORIENG	- Character orientation in degrees	0.0
15	PLTCHG	- Plot character specification for POINTG Subroutine.	1.0
16	FSTATG	- Status of "Form" (not available at present).	0.0
17	NUMDVG	- The number of devices to be used to produce graphic output by this user.	1.0
18	UNITSG	- This word is used by the PLTITG routine to indicate if its arguments are in subject (non-negative value) or object space units (negative value).	1.0
19	JJJJJG	- A pointer word used to indicate which device is currently turned "on". Only one device may be turned on at a time (integer values only).	27
20	AINCRG	- Increment size of the currently turned on device, i.e., the smallest plottable distance (in inches).	0.005
21	XCHARG	X and Y coordinates (in subject space units) of lower right corner of the character envelope for the last character plotted.	0.0
22	YCHARG		0.0
23	MAXCHG	- Maximum number of characters that can be plotted on a line. If an attempt is made to exceed this, then a "carriage return"	99999

will be performed and the remaining characters will be drawn on the line below.	
- The coordinates (in object space units) of a point above the upper left corner of the graph being plotted. This information is used to enable proper spacing between a series of plots.	0.0
- This word is used by the subroutine FINISG to indicate that the subroutine DEVOFG is to turn off all devices and end all plotting.	0.0
- 0.01 inches in subject space units.	0.0
- Type of device: if positive, device is on; if negative, device is off.	20.0
- Logical file code of the file into which the plot commands are to be outputted.	8.0
Available plotting space in the X direction, in device-dependent units.	0.0 0.0
Available plotting space in the Y direction, in device-dependent units.	0.0 0.0
Conversion factors to go from subject space to object space.*	0.0 0.0 0.0 0.0
- Plot number of current plot for this device.	0

APPENDIX B.- Program Listings

These notes are intended to serve as an introduction and supplement to the comment statements included in the listings to aid in the conversion of GRAFPAC to other computer systems.

The difficulty in modifying GRAFPAC to a computer with a different word length and with a different number of bits per character will be the modification of routines LEGNDG, UNPAKG, and ICHARG. If this seems too formidable one might consider writing a simple routine which interfaces between GRAFPAC and a standard lettering routine which may be presently available.

The routine IARG (which allows a FORTRAN subroutine to have a variable number of arguments also may be a problem but it can be deleted if one always uses the complete set of arguments.

The routine TRACBK provides a trace of the subroutine calls and their arguments from the point at which the routine is called back to the main program. It is used by the GRAFPAC error message routine ERRORG. It is dependent on the GE error routine .FXEM. If an equivalent facility is not available this call may be deleted.

The routine BORTIT called by ERRORG does not exist. On the GE 635 the calling of a non-existant routine is a convenient way for a program to have an abnormal termination so that an optional dump may be obtained (by specifying "DUMP" on the $ EXECUTE card). On other systems one may wish to call EXIT or use the STOP statement.

If GRAFPAC is adapted to other plotters on will probably replace BLPLTG (and possibly INCRPG) by their equivalents.

The routine FPTBCD which converts a floating point number to BCD using an Fw.d FORMAT specification will have to be rewritten for most other systems because there is little standardization by which internal format conversion is done. On some systems it may be necessary to write the number onto tape and then reread it.

The other routines should cause little conversion problems.

The routines are listed below in the order in which they should appear on a library tape.

```
$      FORTRAN DECK,LSTOU                                                FXYPL001
CFXYPLG                    X-Y PLOT                                      FXYPL002
       SUBROUTINE FXYPLG(NX,NY,ND1,ND2,XMIN,XMAX,YMIN,YMAX,Z1,Z2,IZZ,    FXYPL003
     1 FXY,A,B,G,D,CX,CY,CZ,SIZE)                                        FXYPL004
      COMMON/AMODES/NNNNN,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,FXYPL005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,PLTCHG,FSTATS,NDEV    ,     FXYPL006
     2ERR(21)                                                            FXYPL007
      INTEGER IZZ(NX,NY)          ,       PREV                           FXYPL008
      INTEGER CHECKG                                                     FXYPL009
      EQUIVALENCE  (INTZ,Z)                                              FXYPL010
       COMMON/XYPGGG/IX,IY,X,Y,Z,DELX,DELY  ,ZMIN,ZMAX                   FXYPL011
      EXTERNAL TRANSG                                                    FXYPL012
      EXTERNAL FXY                                                       FXYPL013
C.... NOTE EQUIVALENCE OF INTZ AND Z AND THE FACT THAT THE VISIBILITY     FXYPL014
C     CODE IS STORED IN THE LAST TWO BITS OF EACH WORD IN IZZ(,)         FXYPL015
C     IZZ(,) CONTAINS FLOATING POINT NUMBERS                             FXYPL016
C                                                                        FXYPL017
      WRITE(6,400) NX,NY,ND1,ND2,A,B,G,D,CX,CY,CZ                        FXYPL018
 400  FORMAT(//20(1HG),3X,6HFXYPLG  ,3X,20(1HG)//                        FXYPL019
     111H GRID ARRAY ,I5,3H BY ,I5,5H WITH ,I3, 4H AND,I3,               FXYPL020
     228H LINE SEGMENTS BETWEEN NODES  /                                 FXYPL021
     3 20H VIEWING DIRECTION =,3F10.4     ,                              FXYPL022
     4 31H DISTANCE TO PROJECTION PLANE =,F10.4/                         FXYPL023
     5 19H VIEWING POSITION =  3F10.4  )                                 FXYPL024
C     INITIALIZE ROUTINES                                                FXYPL025
      CALL ICOMPG (IZZ,NX,NY,FXY)                                        FXYPL026
      NDX = ND1 + 1                                                      FXYPL027
      NDY = ND2 + 1                                                      FXYPL028
      IF(NDX.GT.10) NDX=10                                               FXYPL029
      IF(NDY.GT.10) NDY=10                                               FXYPL030
      FNX=NX                                                             FXYPL031
      FNY=NY                                                             FXYPL032
```

```
      FXS=(FNX-1.0)/(XMAX-XMIN)
      FXM=1.0-XMIN*FXS
      FYS=(FNY-1.0)/(YMAX-YMIN)
      FYM=1.0-YMIN*FYS
      DX=1./(FXS*FLOAT(NDX-1))
      DY= 1./(FYS*FLOAT(NDY-1))
      CALL ITRANG (A,B,G,D,CX,CY,CZ)
       IF(Z1.EQ.Z2)GO TO 200
       ZMIN=Z1
       ZMAX=Z2
       ASSIGN 250 TO INS1
       GO TO 201
C      SEARCH
 200   ZMIN=9.9E+10
       ZMAX=-9.9E+10
       ASSIGN 255 TO INS1
C      GENERATE FUNCTION ARRAY
 201  DELX=1./FXS
      DELY=1./FYS
      X=-FXM/FXS
      DO 121 IX=1,NX
      X=X+DELX
      Y=-FYM/FYS
       DO 120 IY=1,NY
      Y = Y + DELY
       Z=FXY(X,Y)
       GO TO INS1,(250,255)
250    IF(Z.LT.ZMIN)Z=ZMIN
       IF(Z.GT.ZMAX)Z=ZMAX
       GO TO 120
255    IF(Z.LT.ZMIN)ZMIN=Z
       IF(Z.GT.ZMAX)ZMAX=Z
120    IZZ(IX,IY)=INTZ
121    CONTINUE
      CALL ICHCKG(CX,CY,CZ,FXM,FXS,FYM,FYS,NX,NY,IZZ,DELX,DELY)
      CALL SUBJ3G(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,TRANSG)
      PROP = (XRIGHT-XLEFT)/(YUPPER-YLOWER)
      PROP1=SIZE
      PROP2=SIZE/PROP
      IF(PROP.GT.1.) GO TO 10
      PROP1=PROP*SIZE
      PROP2= SIZE
 10   WRITE(6,122) XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,PROP1,PROP2
 122   FORMAT( 25H 3-D SUBJECT SPACE LIMITS /2HX  2F10.4,2H Y 2F10.4,
     1 2H Z ,2F10.4/ 13H SIZE OF PLOT  F5.1, 3H BY ,F5.1,7H INCHES /
     2 50(1HG) //           )
      CALL OBJECG(0.,PROP1,0.,PROP2,20.)
C      CHECK VISIBILITY OF POINTS
       X=-FXM/FXS
       DO 80 IX=1,NX
       X=X+DELX
       Y=-FYM/FYS
       DO80IY=1,NY
       Y=Y+DELY
       INTZ=IZZ(IX,IY)
80     IZZ(IX,IY)=ILSTBG(INTZ,CHECKG(Z))
C     PLOT X CONSTANT LINES
       X=-FXM/FXS
C     SAVE DELTA VALUES
      DELYS = DELY
```

```
      DYS = DY                                                          FXYPL093
      IEVEN = -1                                                        FXYPL094
      DO 221 IX=1,NX                                                    FXYPL095
      DELY=-DELY                                                        FXYPL096
      DY=-DY                                                            FXYPL097
      I=NY                                                              FXYPL098
      IF(IEVEN.LT.0) GO TO 210                                          FXYPL099
      I = 1                                                             FXYPL100
  210  X=X+DELX                                                         FXYPL101
       Y=(FLOAT(I)-FYM)/FYS                                             FXYPL102
      INTZ=IZZ(IX,I)                                                    FXYPL103
      PREV=LASTBG(INTZ)                                                 FXYPL104
C     MOVE PEN POINT                                                    FXYPL105
      DO 220 I =2,NY                                                    FXYPL106
      IY=I                                                              FXYPL107
      IF(IEVEN .LT.0) IY=NY-I+1                                         FXYPL108
       Y=Y+DELY                                                         FXYPL109
  220 CALL COMPLG(PREV,     NDY,0.,DY)                                  FXYPL110
  221 IEVEN = -IEVEN                                                    FXYPL111
C     RESTORE DELTA VALUES                                              FXYPL112
C     PLOT Y CONSTANT LINES                                             FXYPL113
      DELY = DELYS                                                      FXYPL114
      DY = DYS                                                          FXYPL115
      Y=-FYM/FYS                                                        FXYPL116
      IEVEN = -1                                                        FXYPL117
      DO 331 IY=1,NY                                                    FXYPL118
      DELX=-DELX                                                        FXYPL119
      DX=-DX                                                            FXYPL120
      I = NX                                                            FXYPL121
      IF(IEVEN.LT.0) GO TO 310                                          FXYPL122
      I = 1                                                             FXYPL123
  310 Y=Y+DELY                                                          FXYPL124
      X=(FLOAT(I)-FXM)/FXS                                              FXYPL125
      INTZ=IZZ(I,IY)                                                    FXYPL126
      PREV=LASTBG(INTZ)                                                 FXYPL127
      DO 330 I =2,NX                                                    FXYPL128
      IX=I                                                              FXYPL129
      IF(IEVEN.LT.0) IX=NX-I+1                                          FXYPL130
       X=X+DELX                                                         FXYPL131
  330 CALL COMPLG(PREV,     NDX,DX,0.)                                  FXYPL132
  331 IEVEN = -IEVEN                                                    FXYPL133
      RETURN                                                            FXYPL134
      END                                                               FXYPL135
$      FORTRAN DECK,LSTOU                                               ICOMP001
CICOMPG          COMPLETE CURVE BETWEEN TWO POINTS                      ICOMP002
      SUBROUTINE ICOMPG(IZZ,NX,NY,FXY)                                  ICOMP003
       COMMON/XYPGGG/IX,IY,X,Y,Z,DELX,DELY   ,ZMIN,ZMAX                 ICOMP004
      INTEGER IZZ(NX,NY),P,PP,PREV                                      ICOMP005
       INTEGER   CHECKG                                                 ICOMP006
      EQUIVALENCE (INTZ,Z)                                              ICOMP007
      REAL             XA(10),YA(10),ZA(10),XT(10),YT(10)               ICOMP008
C     PRIMARY ENTRY IS USED ONLY TO SET UP  ADJUSTABLE DIM. ARRAY IZZ   ICOMP009
      RETURN                                                            ICOMP010
      ENTRY COMPLG(PREV,ND,DX,DY)                                       ICOMP011
      INTZ=IZZ(IX,IY)                                                   ICOMP012
       CALL TRANSG(1,X,Y,Z,XTN,YTN)                                     ICOMP013
      P = LASTBG(INTZ)                                                  ICOMP014
       XSAVE=X                                                          ICOMP015
       YSAVE=Y                                                          ICOMP016
       IF(DX.EQ.0.0)Y=Y-DELY-DY                                         ICOMP017
```

```
      IF(DY.EQ.0.0)X=X-DELX-DX
      IF(P*PREV)240,220,210
C     XTN AND YTN ARE TRANSFORMED COORDINATES OF POINT AT END OF
C     CURRENT LINE SEGMENT,
C     BOTH POINTS VISIBLE ----  PLOT
  210 DO 211 K=1,ND
      Y=Y+DY
      X=X+DX
C     (NOTE AT ANY ONE TIME EITHER DX OR DY IS ZERO)
      Z=FXY(X,Y)
      IF(Z.LT.ZMIN)Z=ZMIN
      IF(Z.GT.ZMAX) Z= ZMAX
      YA(K)=Y
      XA(K)=X
      ZA(K)=Z
  211 CONTINUE
      CALL TRANSG(ND,XA,YA,ZA,XT,YT)
      CALL LINESG(XT,YT,ND)
      GO TO 280
  220 IF(PREV)225,221,225
  221 IF(P)228,222,228
C     BOTH POINTS HIDDEN ----IGNORE
  222 GO TO  280
C     ONE POINT VISIBLE, THE OTHER HIDDEN
C     FIRST POINT VISIBLE
  225 DO 226 K=1,ND
      X=X+DX
      Y=Y+DY
      Z=FXY(X,Y)
      IF(Z.LT.ZMIN)Z=ZMIN
      IF(Z.GT.ZMAX) Z= ZMAX
       PP=CHECKG(Z)
      IF(PP.EQ.0) GO TO 227
      KS = K
      XA(K)=X
      YA(K)=Y
  226 ZA(K)=Z
  227 IF(KS.LE.1) GO TO 280
      CALL TRANSG(KS,XA,YA,ZA,XT,YT)
      CALL LINESG(XT,YT,KS)
      GO TO 280
C     SECOND POINT VISIBLE
C     FIND FIRST VISIBLE POINT
  228 DO 229 K=1,ND
      X=X+DX
      Y=Y+DY
       Z=FXY(X,Y)
      IF(Z.LT.ZMIN)Z=ZMIN
      IF(Z.GT.ZMAX) Z= ZMAX
      PP=CHECKG(Z)
      KK=K
  229 IF(PP.NE.0   ) GO TO 230
       GO TO 280
  230 IF(ND-KK.EQ.0) GO TO 280
      XA(KK)=X
      YA(KK)=Y
      ZA(KK)=Z
      KKP1=KK+1
      DO 231 K=KKP1,ND
      X=X+DX
      Y=Y+DY
```

```
      Z=FXY(X,Y)                                                        ICOMP079
       IF(Z.LT.ZMIN) Z=ZMIN                                             ICOMP080
      IF(Z.GT.ZMAX) Z= ZMAX                                             ICOMP081
      XA(K)=X                                                           ICOMP082
      YA(K)=Y                                                           ICOMP083
      ZA(K)=Z                                                           ICOMP084
  231 CONTINUE                                                          ICOMP085
      CALL TRANSG(ND-KK+1,XA(K),YA(K),ZA(K),XT,YT)                      ICOMP086
      CALL LINESG(XT,YT,ND-KK+1)                                        ICOMP087
      GO TO 280                                                         ICOMP088
C     SEGMENT BETWEEN THE TWO POINTS IS HIDDEN                          ICOMP089
  240 CONTINUE                                                          ICOMP090
      GO TO 210                                                         ICOMP091
  280 CONTINUE                                                          ICOMP092
      PREV = P                                                          ICOMP093
       X=XSAVE                                                          ICOMP094
       Y=YSAVE                                                          ICOMP095
      RETURN                                                            ICOMP096
      END                                                               ICOMP097
$      FORTRAN DECK,LSTOU                                               ICHC<001
CICHCKG                                                                 ICHC<002
      FUNCTION ICHCKG(CX,CY,CZ,FXM,FXS,FYM,FYS,NX,NY,ZZ,DDX,DDY)        ICHC<003
C     CHECK VISIBILITY OF POINT                                         ICHC<004
      LOGICAL FIRST,POS                                                 ICHC<005
      REAL  ZZ(NX,NY)                                                   ICHC<006
      COMMON/XYPGGG/D3,D4,X,Y,D5,D6,D7       ,ZMIN,ZMAX                 ICHC<007
      DELX=DDX                                                          ICHC<008
      DELY=DDY                                                          ICHC<009
      RETURN                                                            ICHCK010
       ENTRY CHECKG(Z)                                                  ICHC<011
      FIRST = .TRUE.                                                    ICHC<012
       DD=(X-CX)*(X-CX)+(Y-CY)*(Y-CY)    +(Z-CZ)*(Z-CZ)                 ICHC<013
      ICHCKG = 0                                                        ICHC<014
C     CHECK WITH CONSTANT X                                             ICHC<015
       XTILDA=-FXM/FXS                                                  ICHC<016
      DO 40 IX= 1,NX                                                    ICHC<017
       XTILDA=XTILDA+DELX                                               ICHC<018
      IF(ABS(X-XTILDA).LT.1.0E-6) GO TO 40                              ICHC<019
      SLPX = (XTILDA-CX)/(X-CX)                                         ICHC<020
      YTILDA = SLPX*(Y-CY)+CY                                           ICHC<021
       IY=YTILDA*FYS+FYM                                                ICHC<022
      IF(IY.LT.1 .OR. IY.GE.NY) GO TO 40                                ICHC<023
      ZTILDA = SLPX*(Z-CZ)+CZ                                           ICHC<024
       IF(DD.LT.(XTILDA-CX)*(XTILDA-CX)+(YTILDA-CY)*(YTILDA-CY)    +    ICHCK025
     1 (ZTILDA-CZ)*(ZTILDA-CZ)) GO TO 40                                ICHC<026
C     O.K.                                                              ICHC<027
       YY=(FLOAT(IY)-FYM)/FYS                                           ICHCK028
       YYP1=YY+DELY                                                     ICHC<029
      ZSTAR=((YTILDA-YY)/(YYP1-YY))*(ZZ(IX,IY+1)-ZZ(IX,IY))+ZZ(IX,IY)   ICHC<030
      DEL=ZTILDA-ZSTAR                                                  ICHC<031
       IF(ABS(DEL).LT.1.0E-6)GO TO 40                                   ICHC<032
      IF(.NOT.FIRST) GO TO 30                                           ICHC<033
      FIRST=.FALSE.                                                     ICHC<034
      POS=.TRUE.                                                        ICHCK035
      IF(DEL.LT.0.0) POS=.FALSE.                                        ICHC<036
      GO TO 40                                                          ICHCK037
   30 IF((POS.AND.DEL.LT.0.0).OR.(.NOT.POS.AND.DEL.GT.0.0))GO TO 160    ICHC<038
   40 CONTINUE                                                          ICHCK039
C     POINT VISIBLE SO FAR                                              ICHC<040
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ICHC<041
```

```
C     CHECK WITH CONSTANT Y                                                ICHC<042
        YTILDA=-FYM/FYS                                                    ICHC<043
      DO 140 IY=1,NY                                                       ICHC<044
        YTILDA=YTILDA+DELY                                                 ICHC<045
      IF(ABS(Y-YTILDA).LT.1.0E-6) GO TO 140                                ICHC<046
      SLPY=(YTILDA-CY)/(Y-CY)                                              ICHC<047
      XTILDA=SLPY*(X-CX)+CX                                                ICHC<048
       IX=XTILDA*FXS+FXM                                                   ICHC<049
      IF(IX.LT.1.OR.IX.GE.NX) GO TO 140                                    ICHC<050
      ZTILDA = SLPY*(Z-CZ)+CZ                                              ICHC<051
       IF(DD.LT.(XTILDA-CX)*(XTILDA-CX)+(YTILDA-CY)*(YTILDA-CY)    +       ICHC<052
     1  (ZTILDA-CZ)*(ZTILDA-CZ)   ) GO TO 140                              ICHC<053
       XX=(FLOAT(IX)-FXM)/FXS                                              ICHC<054
       XXP1=XX+DELX                                                        ICHC<055
      ZSTAR=((XTILDA-XX)/(XXP1-XX))*(ZZ(IX+1,IY)-ZZ(IX,IY))+ZZ(IX,IY)      ICHC<056
      DEL=ZTILDA-ZSTAR                                                     ICHC<057
       IF(ABS(DEL).LT.1.0E-6)GOTO140                                       ICHC<058
      IF(.NOT.FIRST) GO TO 130                                             ICHC<059
      FIRST=.FALSE.                                                        ICHC<060
      POS=.TRUE.                                                           ICHC<061
      IF(DEL.LT.0.0) POS=.FALSE.                                           ICHC<062
      GO TO 140                                                            ICHC<063
  130 IF((POS.AND.DEL.LT.0.0).OR.(.NOT.POS.AND.DEL.GT.0.0))GO TO 160       ICHC<064
  140 CONTINUE                                                             ICHC<065
C     POINT IS VISIBLE                                                     ICHC<066
      ICHCKG=-1                                                            ICHC<067
      IF(POS)ICHCKG=1                                                      ICHC<068
      RETURN                                                               ICHC<069
C     HIDDEN POINT                                                         ICHC<070
  160 RETURN                                                               ICHC<071
      END                                                                  ICHC<072
$       GMAP      DECK,LSTOU,NXEC                                          LAST3001
        SYMDEF    LASTBG                                                   LAST3002
        LBL       LASTBG01,FUNCTION LASTBG(ARG)                            LAST3003
        TTL       SUBR TO RETURN LAST 2 BITS                               LAST3004
LASTBG  LDA       2,1*                                                     LAST3005
        LDQ       0,DL                                                     LAST3006
        LRL       2                                                        LAST3007
        QRS       34                                                       LAST3008
        TRA       0,1                                                      LAST3009
        ERLK                                                               LAST3010
        END                                                                LAST3011
$       GMAP      DECK,LSTOU,NXEC                                          ILST3001
        SYMDEF    ILSTBG                                                   ILST3002
        LBL       ILSTBG01,FUNCTION ILSTBG(ARG,IND)                        ILST3003
        TTL       SUBR TO STUFF IND INTO LAST TWO BITS OF ARG              ILST3004
ILSTBG  LDA       2,1*                                                     ILST3005
        ARL       2                                                        ILST3006
        LDQ       3,1*                                                     ILST3007
        QLS       34                                                       ILST3008
        LRL       34                                                       ILST3009
        TRA       0,1                                                      ILST3010
        ERLK                                                               ILST3011
        END                                                                ILST3012
$       FORTRAN DECK,LSTOU                                                 IDENT001
CIDENTG      IDENTIFY PLOTS                                                IDENT002
      SUBROUTINE IDENTG (ALPHA, N, FLAG)                                   IDENT003
      COMMON /AMODES/ DEV(38)                                              IDENT004
      REAL DUMMY (3)                                                       IDENT005
      EQUIVALENCE   (XCHARG,DEV(21)),(YCHARG,DEV(22)),(XLEFTG,DEV(2 ) )    IDENT006
```

```
     1 ,(YUPPRG,DEV(5)),(NPLOTG,DEV(38)),(XLASTG,DEV(6)),(YLASTG,DEV(7))IDENT007
     2 ,(CHIGHG,DEV(13))    ,  (J,DEV(19))                              IDENT008
      DATA LSTPLT, X, Y / -1, 0.0,0.0/                                  IDENT009
      NARG = IARG (DUMMY)                                               IDENT010
      IF (LSTPLT . NE. NPLOTG) GO TO 10                                 IDENT011
    5 CALL LEGNDG (X, Y, ALPHA, N)                                      IDENT012
      X = X CHARG                                                       IDENT013
       Y = YCHARG                                                       IDENT014
C     IF FLAG IS SET, CREATE NEW PLOT NUMBER SO PLOTTER CAN BE STOPPED  IDENT015
      IF (NARG. EQ.3.AND.FLAG.NE. 0.0) GO TO 20                         IDENT016
      RETURN                                                            IDENT017
C     FIRST TIME THROUGH THIS ROUTINE FOR CURRENT PLOT.                 IDENT018
   10 X = XLEFTG                                                        IDENT019
       Y=YUPPRG+(1.-CHIGHG)/DEV(J+10)                                   IDENT020
      LSTPLT = NPLOTG                                                   IDENT021
      GO TO 5                                                           IDENT022
C     STOP                                                              IDENT023
 20    CALL PLTITG (XLASTG,YLASTG,-1)                                   IDENT024
       WRITE(6,25) NPLOTG                                               IDENT025
 25    FORMAT( 50(1HG)/                                                 IDENT026
     1 52H ASK OPERATOR TO STOP PLOTTER AT END OF PLOT NUMBER     ,I4   IDENT027
     2 ,21H FOR MANUAL OPERATION        5X/50(1HG)   )                  IDENT028
      NPLOTG = NPLOTG + 1                                               IDENT029
      LSTPLT = NPLOTG                                                   IDENT030
      RETURN                                                            IDENT031
      END                                                               IDENT032
$      FORTRAN DECK,LSTOU                                               VAXES001
CVAXESG                 VERTICAL AXES                                   VAXES002
       SUBROUTINE VAXESG(X,IC,YI,YT,NTICS,IW,ID,NCHAR,ALPH)             VAXES003
C     PLOT SEVERAL VERTICAL AXES                                        VAXES004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    VAXES005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,        VAXES006
     2 ERR(21)                                                          VAXES007
      REAL DEV(1)                                                       VAXES008
      LOGICAL FLAG                                                      VAXES009
      DIMENSION X(1)     ,DUMMY(9),ALPH(10)                             VAXES010
      EQUIVALENCE (DEV,N)   ,(J,ERR(2)),(MAXCHG,ERR(6))                 VAXES011
      DATA NAME/6HVAXESG/                                               VAXES012
C     SET PARAMETERS                                                    VAXES013
      FLAG = .FALSE.                                                    VAXES014
      NARG = IARG (DUMMY)                                               VAXES015
      IF (NARG.LE.4.OR.NTICS.EQ.0) GO TO 1                              VAXES016
      IF (NTICS.LT.0) FLAG = .TRUE.                                     VAXES017
      NTICS = IABS (NTICS)                                              VAXES018
      TIC = 0.1/DEV (J+8)                                               VAXES019
      IF (FLAG) TIC = -TIC                                              VAXES020
      DEL = (YT-YI)/FLOAT (NTICS-1)                                     VAXES021
      IF (NARG.LE.5 .OR. IW.EQ.0) GO TO 1                               VAXES022
      WD = IW+2                                                         VAXES023
      IF (FLAG) WD = -2                                                 VAXES024
      XSPACE = WD * CHWDTH/DEV(J+8)                                     VAXES025
      YDOWN = 0.5* CHIGHT/DEV(J+10)                                     VAXES026
 1    DO 10 I = 1,IC                                                    VAXES027
      CALL SEGMTG(X(I),YI,X(I),YT,1)                                    VAXES028
      IF (NARG.LE.4) GO TO 10                                           VAXES029
C     DRAW TIC MARKS                                                    VAXES030
       YL=YT                                                            VAXES031
      DO 5 K=1,NTICS                                                    VAXES032
      CALL SEGMTG (X(I), YL ,X(I) + TIC,YL,1)                           VAXES033
      IF (NARG.LE.5 ,OR.IW.EQ.0) GO TO 5                                VAXES034
      CALL FPT BCD (YL, IW, ID, DUMMY)                                  VAXES035
```

```
      CALL LEGNDG(X(I)-XSPACE, YL-YDOWN, DUMMY,IW)                      VAXES036
 5      YL=YL-DEL                                                        VAXES037
   10 CONTINUE                                                           VAXES038
C     LABEL AXIS                                                         VAXES039
      IF (NARG.LE.7.OR.NCHAR.LE.0) RETURN                                VAXES040
      FLNGTH = FLOAT (NCHAR) *CHIGHT/DEV(J+10)                           VAXES041
      MAX CHG=1                                                          VAXES042
      TOP = (YI+YT)/2.+0.5*FLNGTH                                        VAXES043
      WD = WD + 3.                                                       VAXES044
      IF (FLAG) WD = -IW-2                                               VAXES045
      CALL LEGNDG (X(1) -WD*CHWDTH/DEV(J+8),                             VAXES046
     1TOP, ALPH, NCHAR)                                                  VAXES047
      MAXCHG = 999                                                       VAXES048
      RETURN                                                             VAXES049
      END                                                                VAXES050
$      FORTRAN DECK,LSTOU                                                HAXES001
CHAXESG             HORIZONTAL AXES                                      HAXES002
       SUBROUTINE HAXESG(Y,IC,XI,XT,NTICS,IW,ID,NCHAR,ALPH)              HAXES003
C.... PLOT SEVERAL HORIZONTIAL AXES, CORDINATE=Y(I), I=1,IC               HAXES004
C.... ABSCISSA = XI TO XT                                                HAXES005
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,     HAXES006
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,         HAXES007
     2 ERR(21)                                                           HAXES008
      REAL DEV(1)                                                        HAXES009
      DIMENSION Y(1)    ,DUMMY(9),ALPH(10)                               HAXES010
      LOGICAL FLAG                                                       HAXES011
      EQUIVALENCE (DEV,N)     ,(J,ERR(2) )                               HAXES012
      DATA NAME/6HHAXESG/                                                HAXES013
C     SET PARAMETERS                                                     HAXES014
      FLAG = .FALSE.                                                     HAXES015
      NARG = IARG (DUMMY)                                                HAXES016
      IF (NARG.LE.4.OR.NTICS.EQ.0) GO TO 1                               HAXES017
      IF (NTICS.LT.0) FLAG = .TRUE.                                      HAXES018
      NTICS = IABS (NTICS)                                               HAXES019
      TIC = 0.1/DEV (J+10)                                               HAXES020
      IF (FLAG) TIC = -TIC                                               HAXES021
      DEL = (XT-XI)/FLOAT (NTICS-1)                                      HAXES022
      IF (NARG.LE.5.OR. IW.EQ.0) GO TO 1                                 HAXES023
      HFID = (FLOAT (IW)/2.)* CHWDTH/DEV (J+8)                           HAXES024
       DOWN = 2. * CHIGHT/DEV(J+10)                                      HAXES025
      IF (FLAG) DOWN = -DOWN                                             HAXES026
 1    DO 10 I=1,IC                                                       HAXES027
      CALL SEGMTG(XI,Y(I),XT,Y(I),1)                                     HAXES028
      IF (NARG.LE.4. OR.NTICS.EQ.0) GO TO 10                             HAXES029
C     DRAW TIC MARKS                                                     HAXES030
       XL=XT                                                             HAXES031
      DO 5 K=1,NTICS                                                     HAXES032
      CALL SEGMTG (XL, Y(I), XL, Y(I)+TIC,1)                             HAXES033
      IF (NARG.LE.5. OR. IW.EQ.0) GO TO 5                                HAXES034
      CALL FPT BCD (XL,IW, ID, DUMMY)                                    HAXES035
      CALL LEGNDG (XL-HFID, Y(I)-DOWN, DUMMY, IW)                        HAXES036
 5     XL=XL-DEL                                                         HAXES037
   10 CONTINUE                                                           HAXES038
C     DRAW LABEL FOR AXIS                                                HAXES039
      IF (NARG.LE.7.OR.NCHAR.LE.0 ) RETURN                               HAXES040
      FLNGTH = FLOAT (NCHAR) * CHWDTH/DEV(J+8)                           HAXES041
      START = (XI + XT)/2.-0.5*FLNGTH                                    HAXES042
      DOWN = DOWN * 2.0                                                  HAXES043
      CALL LEGNDG (START, Y(1)-DOWN, ALPH, NCHAR)                        HAXES044
      RETURN                                                             HAXES045
      END                                                                HAXES046
```

```
$       FORTRAN DECK,LSTOU                                          1/31/68     MODES001
CMODESG       BLOCK DATA SUBROUTINE MODESG                                      MODE   2
      SUBROUTINE MODESG(Q)                                                      MODE   3
C     INITIALIZE THINGS                                                         MODE   4
C      THIS PROGRAM MUST BE CALLED FIRST AND ONLY ONCE                          MODE   5
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,            MODE   6
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)         MODE   7
      REAL DEV(10)                                                              MODE   8
      REAL DUMY(37)                                                             MODE   9
       REAL D(10)                                                               MODE  10
      EQUIVALENCE (DEV,N)   ,(N,DUMY)                                           MODE  11
      EQUIVALENCE (J,ERR(2)) ,(NCHR,ERR(6))                                     MODE  12
       EQUIVALENCE (NPLOTG,ERR(21))                                             MODE  13
      WRITE(6,6)                                                                MODE  14
    6 FORMAT(///1X,80(1HG)//23H START OF GRAFPAC PLOT  //1X,80(1HG))            MODE  15
       DO 5 I=1,37                                                              MODE  16
5      DUMY(I)=0.                                                               MODE  17
       NPLOTG = 0                                                               MODE  18
       N=26                                                                     MODE  19
       NDEV=1                                                                   MODE  20
       ERR(1)=1.                                                                MODE  21
       NCHR = 9999                                                              MODE  22
       ERR(3)=0.005                                                             MODE  23
       J=0                                                                      MODE  24
       ERR(11)=20.                                                              MODE  25
       L=IARG(D)                                                                MODE  26
       IF(L.NE.0)GO TO 1                                                        MODE  27
       L=1                                                                      MODE  28
       D(1)=20.                                                                 MODE  29
1      CONTINUE                                                                 MODE  30
       CALL SMODEG                                                              MODE  31
       DO 2 I=1,L                                                               MODE  32
       CALL SUBJEG(0.,.5,0.,3.)                                                 MODE  33
       CALL OBJECG(0.,.5,0.,3.,D(I))                                            MODE  34
       ORIENT=90.                                                               MODE  35
       C ALL LEGNDG(.25, .5,7HGRAFPAC,7)                                        MODE  36
       ORIENT=0.                                                                MODE  37
2      CONTINUE                                                                 MODE  38
       RETURN                                                                   MODE  39
      END                                                                       MODE  40
$      FORTRAN DECK,LSTOU                                          12/16/67     SMODE001
CSMODEG       SUBROUTINE SMODEG                                                 SMODE002
      SUBROUTINE SMODEG                                                         SMODE003
C     SET STANDARD VALUES                                                       SMODE004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,            SMODE005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,PLTCHG,FSTATS,NDEV,ERR(21)         SMODE006
      REAL DEV(1)                                                               SMODE007
      EQUIVALENCE (DEV,N)                                                       SMODE008
      DATA NAME/6HSMODEG/                                                       SMODE009
      TEXTUR=1.                                                                 SMODE010
      ANTENS=1.                                                                 SMODE011
      COLOR=1.                                                                  SMODE012
      FONT=1.                                                                   SMODE013
      CHIGHT=.16                                                                SMODE014
      CHWDTH=.16                                                                SMODE015
      ORIENT=0.                                                                 SMODE016
      PLTCHG=1.                                                                 SMODE017
      FSTATS=1.                                                                 SMODE018
      RETURN                                                                    SMODE019
      END                                                                       SMODE020
```

```
$       FORTRAN DECK,LSTOU                                        12/15/67    CSIZE001
CCSIZEG       SUBROUTINE CSIZEG                                               CSIZE002
      SUBROUTINE CSIZEG(D)                                                    CSIZE003
C     SET CHAR. SIZE                                                          CSIZE004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,          CSIZE005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)       CSIZE006
      REAL DEV(1)                                                             CSIZE007
      EQUIVALENCE (DEV,N)                                                     CSIZE008
      DATA NAME/6HCSIZEG/                                                     CSIZE009
      NUM =  IARG(D)                                                          CSIZE010
      IF (NUM.EQ. 0) GO TO 2                                                  CSIZE011
      IF(D.NE.20.0)CALL ERRORG(NAME     ,4,20HDEVICE NOT AVAILABLE)           CSIZE012
 2    CHIGHT=.16                                                              CSIZE013
      CHWDTH=.16                                                              CSIZE014
      RETURN                                                                  CSIZE015
      END                                                                     CSIZE016
$       FORTRAN DECK,LSTOU                                         2/27/68    SUBJ3001
CSUBJEG          3-D SUBJECT SPACE                                            SUBJ3002
      SUBROUTINE SUBJ3G(XL,XR,YB,YF,ZB,ZT,TRANSG)                             SUBJ3003
      COMMON /AMOD3G/XC(8),YC(8),ZC(8),XTC(8),YTC(8)                          SUBJ3004
C.... COMPUTE MAXIMA                                                           SUBJ3005
      DO 10 I = 1,4                                                           SUBJ3006
      XC(I) = XL                                                              SUBJ3007
      XC(I+4)=XR                                                              SUBJ3008
      YC(2*I-1) = YB                                                          SUBJ3009
      YC(2*I) = YF                                                            SUBJ3010
      IND  = ((I-1)/2)*2 + I                                                  SUBJ3011
      ZC(IND)= ZB                                                             SUBJ3012
 10   ZC(IND+2) = ZT                                                          SUBJ3013
C.... TRANSFORM                                                                SUBJ3014
      CALL TRANSG(8,XC,YC,ZC,XTC,YTC)                                         SUBJ3015
C.... FIND MAXIMA                                                              SUBJ3016
      CALL MINMAX(8,XTC,XMIN,XMAX)                                            SUBJ3017
      CALL MINMAX(8,YTC,YMIN,YMAX)                                            SUBJ3018
      CALL SUBJEG(XMIN,XMAX,YMIN,YMAX)                                        SUBJ3019
      RETURN                                                                  SUBJ3020
      END                                                                     SUBJ3021
$       FORTRAN DECK,LSTOU                                        12/16/67    SUBJE001
CSUBJEG       SUBROUTINE SUBJEG                                               SUBJE002
      SUBROUTINE SUBJEG(XL,XR,YL,YU)                                          SUBJE003
C     TO SET SUBJECT SPACE LIMITS                                             SUBJE004
C.... FIRST CALL TO SUBJEG MUST BE FOLLOWED BY A CALL TO OBJECG.               SUBJE005
C.... THERFAFTER CAN CALL ADVANG (POSSIBLY FOWWLOWED BY A CALL TO              SUBJE006
C.... THEREAFTER CAN CALL ADVANG (POSSIBLY FOLLOWED BY A CALL TO               SUBJE007
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,          SUBJE008
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)       SUBJE009
      REAL DEV(1)                                                             SUBJE010
      EQUIVALENCE (J,ERR(2))                                                  SUBJE011
      EQUIVALENCE (DEV,N)                                                     SUBJE012
      DATA NAME/6HSUBJEG/                                                     SUBJE013
      XLEFT=XL                                                                SUBJE014
      XRIGHT=XR                                                               SUBJE015
      YLOWER=YL                                                               SUBJE016
      YUPPER=YU                                                               SUBJE017
      IF(J .NE. 0) CALL DEVONG(ABS(DEV(J+1)))                                 SUBJE018
      RETURN                                                                  SUBJE019
      END                                                                     SUBJE020
```

```
$       FORTRAN DECK,LSTOU                                              2/27/68     OBJEC001
COBJECG       SUBROUTINE OBJECG                                                     OBJEC002
      SUBROUTINE OBJECG(XN,XX,YN,YX,DD)                                             OBJEC003
C.... SET OBJECT SPACE LIMITS                                                       OBJEC004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,                OBJEC005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)             OBJEC006
      REAL DEV(1),ARG(5)                                                            OBJEC007
      EQUIVALENCE (DEV,N)                                                           OBJEC008
      EQUIVALENCE (J,ERR(2))                                                        OBJEC009
      DATA NAME/6HOBJECG/                                                           OBJEC010
C.... CHECK FOR OPTIONAL ARGUMENT                                                   OBJEC011
      D=DD                                                                          OBJEC012
      IF(IARG(ARG).EQ.4) D=20.                                                      OBJEC013
      IF(D        .NE.20.0)CALL ERRORG(NAME  ,4,20HDEVICE NOT AVAILABLE)            OBJEC014
      IF(J.EQ.0 )GO TO 1                                                            OBJEC015
1     CONTINUE                                                                      OBJEC016
      CALL TABDVG(D)                                                                OBJEC017
      DEV(J+3)=0.                                                                   OBJEC018
      DEV(J+4)=XX-XN                                                                OBJEC019
      DEV(J+5)=0.                                                                   OBJEC020
      DEV(J+6)=YX-YN                                                                OBJEC021
      IF(DEV(J+6).GT.28.)CALL ERRORG(6HOBJECG,6,32HHEIGHT OF OBJECT SPACOBJEC022
     1E TOO LARGE    )                                                              OBJEC023
      CALL ADVANG                                                                   OBJEC024
      RETURN                                                                        OBJEC025
      END                                                                           OBJEC026
$       FORTRAN DECK,LSTOU                                             12/15/67     DEVON001
CDEVONG       SUBROUTINE DEVONG                                                     DEVON002
      SUBROUTINE DEVONG(D)                                                          DEVON003
C.... TO TURN DEVICE ON                                                             DEVON004
C.... MUST CALL SUBJEG THEN CALL OBJECG BEFORE CALLING DEVONG                       DEVON005
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,                DEVON006
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)             DEVON007
      REAL DEV(1)                                                                   DEVON008
      EQUIVALENCE (DEV,N)                                                           DEVON009
      EQUIVALENCE (DX,ERR(7)),(DY,ERR(8))                                           DEVON010
      EQUIVALENCE (J,ERR(2))                                                        DEVON011
      DATA NAME/6HDEVONG/                                                           DEVON012
C.... SET POINTER IN TABLE                                                          DEVON013
      CALL TABDVG(D)                                                                DEVON014
C.... COMPUTE TRANSFORMATIONS FROM SUBJECT TO OBJECT SPACE                          DEVON015
      DEV(J+8)=(DEV(J+4)-DEV(J+3))/(XRIGHT-XLEFT)                                   DEVON016
      DEV(J+7)=-DEV(J+8)*XLEFT+DX                                                   DEVON017
      DEV(J+10)=(DEV(J+6)-DEV(J+5))/(YUPPER-YLOWER)                                 DEVON018
      DEV(J+9)=-DEV(J+10)*YLOWER+DY                                                 DEVON019
      IF(ABS(D)   .NE.20.0)CALL ERRORG(NAME    ,3,14HNO SUCH DEVICE)                DEVON020
      DEV(J+2)=8.                                                                   DEVON021
      CALL SMODEG                                                                   DEVON022
      ERR(3)=0.005                                                                  DEVON023
C.... .01 INCHES IN SUBJECT SPACE UNITS (USED IN CIRARG)                            DEVON024
      ERR(10)=AMIN1(.01/DEV(J+8), .01/DEV(J+10))                                    DEVON025
      IF(DEV(J+6).GT.29.0)CALL ERRORG(NAME,3,14HYMAX TOO LARGE)                     DEVON026
      CALL PLTITG(XLEFT,YLOWER,3)                                                   DEVON027
      RETURN                                                                        DEVON028
      END                                                                           DEVON029
$       FORTRAN DECK,LSTOU                                             12/16/67     TABDV001
CTABDVG       SUBROUTINE TABDVG                                                     TABDV002
      SUBROUTINE TABDVG(D)                                                          TABDV003
C     SET STARTING POINT IN TABLE FOR DEVICE D                                      TABDV004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,                TABDV005
```

```
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) TABDV006
      EQUIVALENCE (DEV,N)                                               TABDV007
      EQUIVALENCE (J,ERR(2))                                            TABDV008
      REAL DEV(1)                                                       TABDV009
      DATA NAME/6HTABDVG/                                               TABDV010
      J = N + 1                                                         TABDV011
      DO 5 I=1,NDEV                                                     TABDV012
      IF(ABS(DEV(J+1)).EQ.ABS(D))GO TO 3                                TABDV013
5     J=J+10                                                            TABDV014
      J = N + 1                                                         TABDV015
      DO 2 I=1,NDEV                                                     TABDV016
      IF((ABS(DEV(J+1)).EQ.0.0).OR.(ABS(DEV(J+1)).GT.50.))GO TO 3       TABDV017
2     J=J+10                                                            TABDV018
      CALL ERRORG(NAME    ,4,22HNO SPACE LEFT IN TABLE)                 TABDV019
3     CONTINUE                                                          TABDV020
      DEV(J+1)=D                                                        TABDV021
      RETURN                                                            TABDV022
      END                                                               TABDV023
$      FORTRAN DECK,LSTOU                                      5/13/68  ADVAN001
CADVANG      SUBROUTINE ADVANG                                          ADVAN002
      SUBROUTINE ADVANG(DIST)                                           ADVAN003
C.... TO ADVANCE TO NEXT PLOT  BUT NO CHANGE IN SUBJECT OR OBJECT SPACE ADVAN004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    ADVAN005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) ADVAN006
      REAL DEV(1)                                                       ADVAN007
      EQUIVALENCE (DEV,N)                                               ADVAN008
      EQUIVALENCE (J,ERR(2))                                            ADVAN009
      EQUIVALENCE (DX,ERR(7)),(DY,ERR(8))                               ADVAN010
      EQUIVALENCE (NPLOTG,ERR(21))                                      ADVAN011
      DATA NAME/6HADVANG/                                               ADVAN012
      DATA DVX/0.0/                                                     ADVAN013
C.... ARG. DIST IS NOT USED FOR BLPLOTTER                               ADVAN014
      IF(J.EQ.0)                                                        ADVAN015
     1CALL ERRORG(NAME    ,4,19HNO DEVICES ARE OPEN)                    ADVAN016
2     IF(DEV(J+1).NE.20.)CALL ERRORG(NAME    ,3,14HNO SUCH DEVICE)      ADVAN017
      D = ABS(DEV(J+1))                                                 ADVAN018
C.... ADVANCE GRAPH IN Y DIRECTION IF POSSIBLE ELSE GO IN X DIRECTION   ADVAN019
      DVY = DY + DEV(J+6) - DEV(J+5) + 1.0                              ADVAN020
C.... CHECK IF PLACEMENT OF NEW PLOT ABOVE WOULD EXCEED MAXIMUM         ADVAN021
      IF (DVY .GT. 28.0) GO TO 6                                        ADVAN022
      DVX = AMAX1(DX+(DEV(J+4)-DEV(J+3)+1.0)   , DVX)                   ADVAN023
8     DEV(J+6)=DEV(J+6)+DY                                              ADVAN024
      DEV(J+5)=DEV(J+5)+DY                                              ADVAN025
      DEV(J+4)=DEV(J+4)+DX                                              ADVAN026
      DEV(J+3)=DEV(J+3)+DX                                              ADVAN027
      IF(NPLOTG.EQ. 0) GO TO 10                                         ADVAN028
C.... FINISH OLD PLOT SET UP TRANSFORMATIONS FOR NEW ONE                ADVAN029
 9    CALL DEVOFG                                                       ADVAN030
 10   NPLOTG = NPLOTG + 1                                               ADVAN031
      CALL DEVONG (D)                                                   ADVAN032
      DY = DVY                                                          ADVAN033
      RETURN                                                            ADVAN034
C.... PLACE NEXT PLOT ALONG SIDE AT BOTTOM                              ADVAN035
 6    DVY = DEV(J+6) - DEV(J+5) + 1.0                                   ADVAN036
      DY = 0.0                                                          ADVAN037
      DX = DVX                                                          ADVAN038
      DEV(J+6)=DEV(J+6)-DEV(J+5)                                        ADVAN039
      DEV(J+5)=0.                                                       ADVAN040
      DEV(J+4)=DEV(J+4)+DX                                              ADVAN041
      DEV(J+3)=DEV(J+3)+DX                                              ADVAN042
```

```
      DVX = DX + DEV(J+4) - DEV(J+3) + 1.0                              ADVAN043
      GO TO 9                                                           ADVAN044
      END                                                               ADVAN045
$      FORTRAN DECK,LSTOU                                   12/15/67    DEVOF001
CDEVOFG       SUBROUTINE DEVOFG                                         DEVOF002
      SUBROUTINE DEVOFG                                                 DEVOF003
C.... TURN DEVICE OFF                                                   DEVOF004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    DEVOF005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) DEVOF006
      REAL DEV(1)                                                       DEVOF007
      EQUIVALENCE (DEV,N)                                               DEVOF008
      EQUIVALENCE (J,ERR(2))                                            DEVOF009
      DATA NAME/6HDEVOFG/                                               DEVOF010
      IF(J.EQ.0)RETURN                                                  DEVOF011
      IF(DEV(J+1).LT.0.0)GO TO 1                                        DEVOF012
C.... MOVE PEN TO (0,0)POINT OF NEXT PLOT                               DEVOF013
      ERR(1) = -1.0                                                     DEVOF014
      CALL PLTITG(ERR(7),ERR(8),3      )                                DEVOF015
C.... END CURRENT PLOT, PREPARE FOR NEXT                                DEVOF016
      CALL PLTITG (ERR(7),ERR(8),-1)                                    DEVOF017
      ERR(1) = 1.0                                                      DEVOF018
      DEV(J+1)=-DEV(J+1)                                                DEVOF019
1     CONTINUE                                                          DEVOF020
C.... IF ERR(9).EQ.1 TURN DEVICE OFF COMPLETELY                          DEVOF021
      IF(ERR(9).NE.1.0)GO TO 2                                          DEVOF022
       K=DEV(J+2)                                                       DEVOF023
       REWIND K                                                         DEVOF024
2      ERR(9)=0.                                                        DEVOF025
      RETURN                                                            DEVOF026
      END                                                               DEVOF027
$      FORTRAN DECK,LSTOU                                   12/15/67    CIRAR001
CCIRARG       SUBROUTINE CIRARG                                         CIRAR002
      SUBROUTINE CIRARG(XC,YC,RAD,ST,ARCLEN)                            CIRAR003
C.... PLOT CIRULAR ARC,CENTER AT(XC,YC), RADIUS=RAD STARTING ANGLE=ST,  CIRAR004
C.... ARC LENGTHS=ARCLEN                                                CIRAR005
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    CIRAR006
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,PLTCHG,FSTATS,NDEV,ERR(21) CIRAR007
      REAL DEV(1)                                                       CIRAR008
      EQUIVALENCE (DEV,N)                                               CIRAR009
      DATA NAME/6HCIRARG/                                               CIRAR010
      RN(X)=X*3.14159/180.                                              CIRAR011
      AST=ST                                                            CIRAR012
      IF(AST.LT.0.)AST=AST+360.                                         CIRAR013
      BST=AST+ARCLEN                                                    CIRAR014
      XS=XC+RAD*COS(RN(AST))                                            CIRAR015
      YS=YC+RAD*SIN(RN(AST))                                            CIRAR016
      XD=XC+RAD*COS(RN(BST))                                            CIRAR017
      YD=YC+RAD*SIN(RN(BST))                                            CIRAR018
      IF(ARCLEN.EQ.0.)GO TO 2                                           CIRAR019
      IF(RAD.LE.ERR(10))GO TO 1                                         CIRAR020
      CALL ARCITG(XS,YS,XD,YD,XC,YC,ERR(10))                            CIRAR021
      RETURN                                                            CIRAR022
1     CONTINUE                                                          CIRAR023
C.... IF RADIUS = ZERO PLOT A POINT                                     CIRAR024
      FJJ=PLTCHG                                                        CIRAR025
      CALL POINTG(XC,YC,1,1.)                                           CIRAR026
3     PLTCHG=FJJ                                                        CIRAR027
      RETURN                                                            CIRAR028
C.... IF ARC LENGTHS=ZERO PLOT POINT ON CIRCLE                          CIRAR029
2     FJJ=PLTCHG                                                        CIRAR030
```

```
      CALL POINTG(XS,YS,1,1.)                                           CIRAR031
      GO TO 3                                                           CIRAR032
      END                                                               CIRAR033
$      FORTRAN DECK,LSTOU                                     12/15/67  ARCIT001
CARCITG        SUBROUTINE ARCITG                                        ARCIT002
      SUBROUTINE ARCITG (X1,Y1,X2,Y2,XC,YC,DEVI)                        ARCIT003
C     PLOT ARC OF A CIRCLE WITH CENTER AT (XC,YC)  FROM (X1,Y1)         ARCIT004
C     TO (X2,Y2)  WITH DEVIATION DEVI                                   ARCIT005
C     DEVIATION WILL BE MAX, DISTANCE BETWEEN TRUE ARC AND SECANT LINE  ARCIT006
      COMMON/AMODES/NNN,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,  ARCIT007
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) ARCIT008
      REAL DEV(1)                                                       ARCIT009
      EQUIVALENCE (DEV,NNN)                                             ARCIT010
      DATA NAME/6HARCITG/                                               ARCIT011
      R=SQRT((X1-XC)**2+(Y1-YC)**2)                                     ARCIT012
      THET1=ANGLEG(X1,Y1,XC,YC)                                         ARCIT013
      THET2=ANGLEG(X2,Y2,XC,YC)                                         ARCIT014
      THETA=THET2-THET1                                                 ARCIT015
      DEVA=AMIN1(DEVI,R/50.)                                            ARCIT016
      IF(THETA.LE.0.)THETA=THETA+6.2832                                 ARCIT017
      RR=R-DEVA                                                         ARCIT018
      THINC=2.*ATAN( SQRT(R*R-RR*RR)/RR)                                ARCIT019
      N=THETA/THINC                                                     ARCIT020
      IF(ABS(X1-XLAST)+ABS(Y1-YLAST).NE.0.)CALL PLTITG(X1,Y1,3)         ARCIT021
      DO 55 K=1,N                                                       ARCIT022
      THET1=THET1+THINC                                                 ARCIT023
      X3=R*COS(THET1)+XC                                                ARCIT024
      Y3=R*SIN(THET1)+YC                                                ARCIT025
55    CALL PLTITG(X3,Y3,2)                                              ARCIT026
      CALL PLTITG(X2,Y2,2)                                              ARCIT027
      RETURN                                                            ARCIT028
      END                                                               ARCIT029
$      FORTRAN DECK,LSTOU                                     12/15/67  ANGLE001
CANGLEG        SUBROUTINE ANGLEG                                        ANGLE002
      FUNCTION ANGLEG(X1,Y1,XC,YC)                                      ANGLE003
C     COMPUTE ANGLE OF LINE IN RADIANS BETWEEN PAIR OF POINTS           ANGLE004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    ANGLE005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) ANGLE006
      REAL DEV(1)                                                       ANGLE007
      EQUIVALENCE (DEV,N)                                               ANGLE008
      DATA NAME/6HANGLEG/                                               ANGLE009
      XD1=X1-XC                                                         ANGLE010
      YD1=Y1-YC                                                         ANGLE011
      IF(XD1) 300,200,300                                               ANGLE012
  200 THET1=1.5708                                                      ANGLE013
      GO TO 110                                                         ANGLE014
  300 CONTINUE                                                          ANGLE015
      T=YD1/XD1                                                         ANGLE016
      T1=ABS(T)                                                         ANGLE017
      THET1=ATAN(T1)                                                    ANGLE018
      IF(T) 100,120,110                                                 ANGLE019
  100 THET1=3.1416-THET1                                                ANGLE020
  110 CONTINUE                                                          ANGLE021
      IF(YD1) 10,20,20                                                  ANGLE022
  120 IF (XD1) 10,20,20                                                 ANGLE023
   10 THET1=THET1+3.1416                                                ANGLE024
   20 CONTINUE                                                          ANGLE025
      ANGLEG=THET1                                                      ANGLE026
      RETURN                                                            ANGLE027
      END                                                               ANGLE028
```

```
$      FORTRAN DECK,LSTOU                                          5/13/68    FINIS001
CFINISG        SUBROUTINE FINISG                                              FINIS002
      SUBROUTINE FINISG                                                       FINIS003
C.... TO TERMINATE ALL PLOTTING                                               FINIS004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,          FINIS005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)       FINIS006
      REAL DEV(1)                                                             FINIS007
      EQUIVALENCE (DEV,N)                                                     FINIS008
      EQUIVALENCE (J,ERR(2))                                                  FINIS009
      EQUIVALENCE (NPLOTG,ERR(21))                                            FINIS010
      DATA NAME/6HFINISG/                                                     FINIS011
      IF(J.EQ.0)GO TO 1                                                       FINIS012
C.... WRITE END MESSAGE ON PLOTTER FOR OPERATOR                                FINIS013
      CALL SMODEG                                                             FINIS014
      CALL SUBJEG(0.,1.,0.,3.)                                                FINIS015
      CALL OBJECG(0.,1.,0.,3.,ABS(DEV(J+1) ) )                                FINIS016
      ORIENT = 90.                                                            FINIS017
      CALL LEGNDG( .5,.5  ,14HEND OF GRAFPAC , 14)                            FINIS018
       ERR(9)=1.                                                              FINIS019
      CALL DEVOFG                                                             FINIS020
1     CONTINUE                                                                FINIS021
      WRITE(6,2)NPLOTG                                                        FINIS022
 2    FORMAT(///1X,80(1HG)//20H END OF GRAFPAC PLOT,I10,6H PLOTS//            FINIS023
     11X,80(1HG))                                                             FINIS024
      STOP                                                                    FINIS025
      END                                                                     FINIS026
$      FORTRAN DECK,LSTOU                                         12/16/67    POINT000
CPOINTG        SUBROUTINE POINTG                                              POINT001
      SUBROUTINE POINTG(X,Y,IC, POINT)                                        POINT002
C     PLOT PLOTCH AT SEVERAL POINTS CENTERED AT (X(I),Y(I))                   POINT003
      COMMON/AMODES/NNN,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,        POINT004
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,PLTCHG,FSTATS,NDEV,ERR(20)       POINT005
      REAL DEV(1)                                                             POINT006
      DIMENSION X(1),Y(1),PP(4)                                               POINT007
C     IPLTCH=1     DOT                                                        POINT008
C     IPLTCH=2     UP TRI.                                                    POINT009
C     IPLTCH=3     DOWN TRI.                                                  POINT010
C     IPLTCH=4     SQUARE                                                     POINT011
C     IPLTCH=5     X                                                          POINT012
      INTEGER NN(5),MM(2,5),IXX(20),IYY(20)                                   POINT013
      INTEGER IP(6)   ,JXX(5),JYY(5)                                          POINT014
      EQUIVALENCE (DEV,NNN)                                                   POINT015
      DATA NAME/6HPOINTG/                                                     POINT016
C     IP = NUMBER OF POINTS IN EACH SYMBOL                                    POINT017
C     NN = NUMBER OF ORIGIN POINTS IN EACH SYMBOL                             POINT018
C     MM(2,.) = POINTER TO FIRST COORDINATE OF EACH SYMBOL                    POINT019
C     IXX AND IYY = COORDINATES OF SYMBOLS                                    POINT020
      DATA IP/3,4,4,5,2,2/                                                    POINT021
      DATA NN/1,1,1,1,2/                                                      POINT022
      DATA MM/1,1,2,4,3,8,4,12,5,17/                                          POINT023
      DATA IXX/0,1,0,-8,0,8,-8,0,-8,8,0,-8,8,8,-8,-8,-8,8,-8,8/               POINT024
      DATA IYY/0,0,1,-8,8,-8,-8,-8,8,8,-8,-8,-8,8,8,-8,-8,8,8,-8/             POINT025
C.... CHECK IF OPTIONAL ARGUMENT IS PRESENT                                    POINT026
      IF (IARG(PP).EQ.4) PLTCHG=POINT                                         POINT027
      IPLTCH=PLTCHG                                                           POINT028
      ORNT=ORIENT                                                             POINT029
      ORIENT=0.                                                               POINT030
      CHH=CHIGHT                                                              POINT031
      CHW=CHWDTH                                                              POINT032
      CHIGHT=8.*ERR(3)                                                        POINT033
```

```
      CHWDTH=CHIGHT                                                     POINT034
      IF(IPLTCH.LE.0.OR.IPLTCH.GT.5)CALL ERRORG(NAME    ,3,14HINVALID IPPOINT035
     1LTCH)                                                             POINT036
      J=MM(1,IPLTCH)                                                    POINT040
      K=MM(2,IPLTCH)                                                    POINT041
      NNI=NN(IPLTCH)                                                    POINT042
C     LOOP FOR EACH POINT                                               POINT037
      DO 1 I=1,IC                                                       POINT038
      CALL PLTITG(X(I),Y(I),3)                                          POINT039
      M=0                                                               POINT043
      N=J                                                               POINT044
C     LOOP FOR EACH SYMBOL PART                                         POINT045
      DO 2 L=1,NNI                                                      POINT046
      M=M+IP(N)                                                         POINT047
2     N=N+1                                                             POINT048
      N=K                                                               POINT049
      DO 3 L=1,M                                                        POINT050
      JXX(L)=IXX(N)                                                     POINT051
      JYY(L)=IYY(N)                                                     POINT052
3     N=N+1                                                             POINT053
1     CALL SYMPTG(NNI,IP(J),JXX,JYY)                                    POINT054
      CALL PLTITG(X(IC),Y(IC),3)                                        POINT055
      CHIGHT=CHH                                                        POINT056
      CHWDTH=CHW                                                        POINT057
      ORIENT=ORNT                                                       POINT058
      RETURN                                                            POINT059
      END                                                               POINT060
$      FORTRAN DECK,LSTOU                                      12/15/67 LEGND001
CLEGNDG       SUBROUTINE LEGNDG                                         LEGND002
      SUBROUTINE LEGNDG(X,Y,IALPHA,IC)                                  LEGND003
C     TO PLOT IC BCD CHAR. IN ARRAY IALPHA WITH (X,Y) BEING LOWER LEFT  LEGND004
C     CORNER OF CHAR. BEX OF FIRST CHAR.  THE CHAR. WILL BE PACKED 6    LEGND005
C     TO A WORD IN ARRAY IALPHA.(E.G., CALL LEGNDG(X,Y,4HHERE,4)  ).    LEGND006
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    LEGND007
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) LEGND008
      REAL DEV(1)                                                       LEGND009
      DIMENSION IALPHA(1),ISYMPT(64)  ,ISYMBL(124),IP(5),IXX(20),IYY(20)LEGND010
      EQUIVALENCE (DEV,N)                                               LEGND011
      EQUIVALENCE (NUMCHR,ERR(6))                                       LEGND012
      DATA NAME/6HLEGNDG/                                               LEGND013
C     LIST OF POINTERS TO THE START OF EACH SYMBOL IN THE TABLE.        LEGND014
      DATA ISYMPT/1,3,4,6,8,9,11,14,15,18,21,22,25,29,32,33,35,36,38,   LEGND015
     141,43,45,47,49,51,53,55,58,60,61,62,63,64,66,68,70,71,73,74,      LEGND016
     176,78,81,83,84,88,90,91,94,95,97,98,101,103,105,106,108,110,      LEGND017
     1112,113,115,117,120,122,124/                                      LEGND018
C     SYMBOL TABLE FORMAT-                                              LEGND019
C        FIRST TWO OCTAL CHARACTERS = NUMBER, N, OF UNCONNECTED LINE    LEGND020
C        SEGMENTS IN THE SYMBOL.  THE NEXT TWO OCTAL CHARACTERS = NUMBERLEGND021
C      ,M , OF POINTS IN THE FIRST LINE SEGMENT.  THE NEXT M PAIRS OF   LEGND022
C        OCTAL NUMBERS ARE THE COORDINATES OF THE NODES ON THE SYMBOL.  LEGND023
C        IF N IS GREATER THAN UNITY THEN THE NEXT TWO OCTAL CHARACTERS  LEGND024
C        WILL BE THE NUMBER OF POINTS IN THE SECOND LINE SEGMENT, ETC.  LEGND025
      DATA (ISYMBL(I),I= 1,65)/001113122637,057666251310O,              LEGND026
     10010326474100,0011216275766,0655424131161,0011112215162,          LEGND027
     20635424672700,0010451571373,0011112215162,0635424276700,          LEGND028
     20011413245463,0625121121627,0576600000000,0010331672700,          LEGND029
     20012024151627,0576665546362,0512112132454,0011412215162,          LEGND030
     20665727161524,0546500000000,0010451212757,0040222460266,          LEGND031
     20420263230225,0650000000000,0020234541633,0344554537375,          LEGND032
     20573715133151,0730000000000,                                      LEGND033
```

```
     10020532334342,0320535364645,0350000000000,
     2                            0010332643600,0011032345465,
     20665727160000,0000000000000,0020311477102,0632300000000,
     20011314546566,0571711516263,0540000000000,0011072613113,
     20153767760000,0010711175775,0735111000000,0020471111777,
     20021444000000,0020311177702,0144400000000,0011153735131,
     20131537577500,0030211170214,0740277710000,0030211710277,
     20170247410000,0020762512112,0132466057557,0372671000000,
     20010541425251,0410000000000,0010431616737,0010441323647,
     20010352245600,0010226620000,0020241470325,0476500000000/
      DATA (ISYMBL(I),I=66,124)/0020512215162,0670257770000,
     20030211170277,0130245710000,0010317117100,0010511174477,
     20710000000000,0010411177177,0011131131537,0577573513100,
     20010711175766,0655414000000,0021131131537,0577573513102,
     20537100000000,0020711175766,0655414023461,0010234540000,
     20031412215162,0635424151627,0576502474102,0313700000000,
     20030234540255,0330235530000,0010441525647,0020641525343,
     20425205353646,0453500000000,0010246470000,0020224640246,
     20420000000000,0010222660000,0011412216172,0736424151627,
     20677600000000,0020217770247,0410000000000,0010617122161,
     20727700000000,0010317417700,0010517114471,0770000000000,
     20020211770217,0710000000000,0020317447702,0444100000000,
     20010417771171,0020332143602,0147400000000,0010641525343,
     20425200000000,0020631672726,0353705526261,0515200000000,
     20020223530255,0250000000000,0020236370257,0560000000000,
     20010242470000/
      IF(FONT.NE.1.)CALL ERRORG(NAME    ,2,12HNO SUCH FONT)
      XSAVE = XLAST
      YSAVE = YLAST
      XT=XUNITG(X)
      YT=YUNITG(Y)
      ERR(1) = -1.0
      CALL PLTITG(XT,YT,3)
      K=0
      IS=1
      ID=64
      I=1
      J=0
2     J=J+1
      I6=0
      IAL=IALPHA(J)
1     ISC=ICHARG(IAL,I6)
      IF(ISC.GT.ID)CALL ERRORG(NAME    ,5,26HNO SUCH CHAR. IN THIS FONT)
      ISC=IS+ISYMPT(ISC)-1
C.... UNPACK SYMBOL
      CALL UNPAKG(ISYMBL(ISC),NN,IP,IXX,IYY)
C.... PLOT SYMBOL
      CALL SYMPTG(NN,IP,IXX,IYY)
      K=K+1
C.... CHECK IF TOO MANY CHARACTERS ON A LINE
      IF(K.LT.NUMCHR)IF(XUNITG(XRIGHT)-XUNITG(XLAST)-CHWDTH)4,3,3
C.... IF YES, DO A CARRIAGE RETURN
4     K=0
      YT=YT-CHIGHT
      ERR(1)=-1.
      CALL PLTITG(XT,YT,3)
      ERR(1)=1.
3     CONTINUE
      I6=I6+1
      I=I+1
```

```
      IF(I.GT.IC)GO TO 6                                                LEGND094
      IF(I6-6)1,2,2                                                     LEGND095
6     CONTINUE                                                          LEGND096
      ERR(4)=XLAST                                                      LEGND097
      ERR(5)=YLAST                                                      LEGND098
C.... RESTORE POINTER TO LAST NON-LETTERING POINT PLOTTED                LEGND099
      XLAST = XSAVE                                                     LEGND100
      YLAST = YSAVE                                                     LEGND101
      RETURN                                                            LEGND102
      END                                                               LEGND103
$       GMAP      DECK,LSTOU,NXEC                          12/15/67     ICHAR001
        LBL       ICHARG                                                ICHAR002
        TTL       SUBROUTINE ICHARG                                     ICHAR003
        SYMDEF    ICHARG                                                ICHAR004
*                 UNPACK WORD AND RETURN ITH CHARACTER RIGHT ADJUSTED   ICHAR005
ICHARG  LDQ       2,1                                                   ICHAR006
        STQ       BB                                                    ICHAR007
        LDQ       3,1*                                                  ICHAR008
        ORQ       =0000100                                              ICHAR009
        STCQ      BB,07                                                 ICHAR010
        LDQ       0,DL                                                  ICHAR011
        LDQ       BB,CI                                                 ICHAR012
        ADQ       1,DL                                                  ICHAR013
        TRA       0,1                                                   ICHAR014
BB      TALLY     0,0,0                                                 ICHAR015
        END                                                             ICHAR016
$       GMAP      DECK,LSTOU,NXEC                          12/16/67     UNPAK001
        LBL       UNPAKG                                                UNPAK002
        TTL       SUBROUTINE UNPAKG                                     UNPAK003
        SYMDEF    UNPAKG                                                UNPAK004
*                 UNPACK SYMBOL TABLE FOR LEGNDG                        UNPAK005
UNPAKG  STX1      .E.L..                                                UNPAK006
        SREG      REG                                                   UNPAK007
        LDX3      5,1                                                   UNPAK008
        LDX7      6,1                                                   UNPAK009
        LDX6      2,1                                                   UNPAK010
        LDQ       0,6                                                   UNPAK011
        ALS       30                                                    UNPAK012
        LLS       6                                                     UNPAK013
        EAX2      0,AL                                                  UNPAK014
        SXL2      3,1*                                                  UNPAK015
        TZE       RET                                                   UNPAK016
        LDX4      5,DU                                                  UNPAK017
        LDX1      4,1                                                   UNPAK018
NXTSET  ALS       30                                                    UNPAK019
        LLS       6                                                     UNPAK020
        EAX5      0,AL                                                  UNPAK021
        SXL5      0,1                                                   UNPAK022
        ADX1      1,DU                                                  UNPAK023
        SBX4      1,DU                                                  UNPAK024
        TNZ       *+4                                                   UNPAK025
        ADX6      1,DU                                                  UNPAK026
        LDQ       0,6                                                   UNPAK027
        LDX4      6,DU                                                  UNPAK028
NXTPR   ALS       33                                                    UNPAK029
        LLS       3                                                     UNPAK030
        STA       0,3                                                   UNPAK031
        ALS       33                                                    UNPAK032
        LLS       3                                                     UNPAK033
        STA       0,7                                                   UNPAK034
```

```
         ADX3      1,DU
         ADX7      1,DU
         SBX4      1,DU
         TNZ       *+4
         ADX6      1,DU
         LDQ       0,6
         LDX4      6,DU
         SBX5      1,DU
         TNZ       NXTPR
         SBX2      1,DU
         TNZ       NXTSET
RET      LREG      REG
         TRA       0,1
         ERLK
REG      8BSS      8
         END
$        FORTRAN DECK,LSTOU                                      3/11/68
CSYMPTG          SUBROUTINE SYMPTG
      SUBROUTINE SYMPTG(NN,IP,IXX,IYY)
C     TO PLOT SYMBOL OF NN PARTS EACH HAVING IP(I) CONNECTED SEGMENTS
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)
      EQUIVALENCE  (JJJJJG,ERR(2) )
      REAL DEV(1)
      DIMENSION IP(1),IXX(1),IYY(1)
      EQUIVALENCE (DEV,N)
C.... COMPUTE SIZE AND ORIENTATION
      EQUIVALENCE (AX,IX),(AY,IY)
      DATA NAME/6HSYMPTG/
      RN=ORIENT*3.14159/180.
      CA=COS(-RN)
      SA=SIN(-RN)
      M=0
      XLT=XLAST
      YLT=YLAST
      XT=XUNITG(XLAST)
      YT=YUNITG(YLAST)
      IF(NN.EQ.0)GO TO 4
      DO 1 I=1,NN
1     M=M+IP(I)
      DO 2 I=1,M
      XM=(FLOAT(IXX(I))/8.)*CHWDTH
      YM =(FLOAT(IYY(I))/8.)*CHIGHT
      AX=XT+(CA*XM+SA*YM)
      AY=YT+(-SA*XM+CA*YM)
      IXX(I)=IX
2     IYY(I)=IY
C.... PLOT IN OBJECT SPACE UNITS
      ERR(1)=-1.
      J=1
      DO 3 I=1,NN
      IF(IP(I).GT.1)GO TO 5
      CALL PLTITG(IXX(J),IYY(J),3)
      CALL PLTITG(IXX(J),IYY(J),2)
      CALL PLTITG(IXX(J),IYY(J),3)
      GO TO 3
5     CONTINUE
      CALL LINESG(IXX(J),IYY(J),IP(I))
3     J=J+IP(I)
```

```
      ERR(1)=1.                                                         SYMP  44
C.... FIND POSITION OF NEXT LETTER                                      SYMP  45
 4    XLAST = XLT + (CHWDTH/DEV(JJJJJG+8))*COS(RN)                      SYMP  46
      YLAST = YLT + (CHIGHT/DEV(JJJJJG+10))*SIN(RN)                     SYMP  47
      RETURN                                                            SYMP  48
      END                                                               SYMP  49
$      FORTRAN DECK,LSTOU                                    12/15/67   LINES001
CLINESG       SUBROUTINE LINESG                                         LINES002
      SUBROUTINE LINESG(X,Y,IC)                                         LINES003
C.... PLOT CONNECTED SEGMENTS FROM (X(1),Y(1)) TO (X(IC),Y(IC))         LINES004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    LINES005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) LINES006
      REAL DEV(1)                                                       LINES007
      DIMENSION X(1),Y(1)                                               LINES008
      EQUIVALENCE (DEV,N)                                               LINES009
      DATA NAME/6HLINESG/                                               LINES010
      IF(IC-1)1,2,1                                                     LINES011
2     CALL SEGMTG(XLAST,YLAST,X,Y,1)                                    LINES012
      RETURN                                                            LINES013
1     CALL PLTITG(X,Y,3)                                                LINES014
      IF(IC.LE.0)RETURN                                                 LINES015
      DO 3 I=2,IC                                                       LINES016
3     CALL PLTITG(X(I),Y(I),2)                                          LINES017
      RETURN                                                            LINES018
      END                                                               LINES019
$      FORTRAN DECK,LSTOU                                    12/16/67   SEGMT001
CSEGMTG       SUBROUTINE SEGMTG                                         SEGMT002
      SUBROUTINE SEGMTG(XI,YI,XT,YT,IC)                                 SEGMT003
C     PLOT UNCONNECTED SEGMENTS                                         SEGMT004
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,    SEGMT005
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) SEGMT006
      REAL DEV(1)                                                       SEGMT007
      DIMENSION XI(1),YI(1),XT(1),YT(1)                                 SEGMT008
      EQUIVALENCE (DEV,N)                                               SEGMT009
      DATA NAME/6HSEGMTG/                                               SEGMT010
      J=0                                                               SEGMT011
4     J=J+1                                                             SEGMT012
      IF(J.GT.IC)RETURN                                                 SEGMT013
C     COMPUTE NEAREST POINT THEN GO THERE AND PLOT TO OTHER POINT       SEGMT014
      A=XLAST-XI(J)                                                     SEGMT015
      AB=A*A                                                            SEGMT016
      A=YLAST-YI(J)                                                     SEGMT017
      AB=A*A+AB                                                         SEGMT018
       IF(AB.EQ.0.) GO TO 3                                             SEGMT019
      A=XLAST-XT(J)                                                     SEGMT020
      AC=A*A                                                            SEGMT021
      A=YLAST-YT(J)                                                     SEGMT022
      AC=A*A+AC                                                         SEGMT023
      IF(AC.EQ.0.)GO TO 2                                               SEGMT024
      IF(AC.GT.AB)GO TO 3                                               SEGMT025
2     CALL PLTITG(XT(J),YT(J),3)                                        SEGMT026
      CALL PLTITG(XI(J),YI(J),2)                                        SEGMT027
      GO TO 4                                                           SEGMT028
3     CALL PLTITG(XI(J),YI(J),3)                                        SEGMT029
1     CALL PLTITG(XT(J),YT(J),2)                                        SEGMT030
      GO TO 4                                                           SEGMT031
      END                                                               SEGMT032
$      FORTRAN DECK,LSTOU                                    12/16/67   PLTIT001
CPLTITG       SUBROUTINE PLTITG                                         PLTIT002
      SUBROUTINE PLTITG (X,Y,N)                                         PLTIT003
```

```
C       TO MOVE PEN TO (X,Y)                                            PLTIT004
        COMMON/AMODES/NNN,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,  PLTIT005
       1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) PLTIT006
        REAL DEV(1)                                                     PLTIT007
        EQUIVALENCE (DEV,NNN)                                           PLTIT008
        EQUIVALENCE (J,ERR(2))                                          PLTIT009
        DATA NAME/6HPLTITG/                                             PLTIT010
        XX=X                                                            PLTIT011
        YY=Y                                                            PLTIT012
C       CHECK FOR X AND Y IN OBJECT SPACE CO-ORD.                       PLTIT013
        IF(ERR(1).LT.0.)GO TO 3                                         PLTIT014
C       X AND Y ARE IN SUBJECT SPACE UNITS                              PLTIT015
        IF(J.EQ.0)                                                      PLTIT016
       1CALL ERRORG(NAME   ,3,15HNO DEVICE IS ON)                       PLTIT017
 2      IF(DEV(J+1).NE.20.0)CALL ERRORG(NAME,6,31HNO PLOT ROUTINE FOR THISPLTIT018
       1 DEVICE)                                                        PLTIT019
C                                                                       PLTIT020
        XX=XUNITG(XX)                                                   PLTIT021
        YY=YUNITG(YY)                                                   PLTIT022
        XLAST=X                                                         PLTIT023
        YLAST=Y                                                         PLTIT024
        IF((X.LT.XLEFT).OR.(X.GT.XRIGHT).OR.(Y.LT.YLOWER).OR.(Y.GT.YUPPER)PLTIT025
       1 ) CALL ERRORG(NAME,6,32HPEN OUTSIDE SUBJECT SPACE LIMITS     )  PLTIT026
        GO TO 5                                                         PLTIT027
  3     XLAST=XINCHG(XX)                                                PLTIT028
        YLAST=YINCHG(YY)                                                PLTIT029
  5     CALL INCRPG (XX,YY,N)                                           PLTIT030
        RETURN                                                          PLTIT031
        END                                                             PLTIT032
$        FORTRAN DECK,LSTOU                                  12/16/67   XINCH001
CXINCHG          SUBROUTINE XINCHG                                      XINCH002
        FUNCTION XINCHG(X)                                              XINCH003
C       TO CONVERT FROM X IN OBJECT SPACE TO SUBJECT SPACE              XINCH004
        COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,  XINCH005
       1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) XINCH006
        REAL DEV(1)                                                     XINCH007
        EQUIVALENCE (DEV,N)                                             XINCH008
        EQUIVALENCE (J,ERR(2))                                          XINCH009
        DATA NAME/6HXINCHG/                                             XINCH010
        IF(J.EQ.0)CALL ERRORG(NAME,3,14HNO DEVICE OPEN)                 XINCH011
        XINCHG=(X-DEV(J+7))/DEV(J+8)                                    XINCH012
        RETURN                                                          XINCH013
        END                                                             XINCH014
$        FORTRAN DECK,LSTOU                                  12/16/67   XUNIT001
CXUNITG          SUBROUTINE XUNITG                                      XUNIT002
        FUNCTION XUNITG(X)                                              XUNIT003
C       TO CONVERT FROM X IN SUBJECT SPACE TO OBJECT SPACE              XUNIT004
        COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,  XUNIT005
       1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21) XUNIT006
        REAL DEV(1)                                                     XUNIT007
        EQUIVALENCE (DEV,N)                                             XUNIT008
        EQUIVALENCE (J,ERR(2))                                          XUNIT009
        DATA NAME/6HXUNITG/                                             XUNIT010
        IF(J.EQ.0)CALL ERRORG(NAME,3,14HNO DEVICE OPEN)                 XUNIT011
        XUNITG=X*DEV(J+8)+DEV(J+7)                                      XUNIT012
        RETURN                                                          XUNIT013
        END                                                             XUNIT014
$        FORTRAN DECK,LSTOU                                  12/16/67   YINCH001
CYINCHG          SUBROUTINE YINCHG                                      YINCH002
        FUNCTION YINCHG(Y)                                              YINCH003
```

```
C       TO CONVERT FROM Y IN OBJECT SPACE TO SUBJECT SPACE                  YINCH004
        COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,      YINCH005
       1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)  YINCH006
        REAL DEV(1)                                                          YINCH007
        EQUIVALENCE (DEV,N)                                                  YINCH008
        EQUIVALENCE (J,ERR(2))                                               YINCH009
        DATA NAME/6HYINCHG/                                                  YINCH010
        IF(J.EQ.0)CALL ERRORG(NAME,3,14HNO DEVICE OPEN)                      YINCH011
        YINCHG=(Y-DEV(J+9))/DEV(J+10)                                        YINCH012
        RETURN                                                               YINCH013
        END                                                                  YINCH014
$        FORTRAN DECK,LSTOU                                     12/16/67     YUNIT001
CYUNITG          SUBROUTINE YUNITG                                           YUNIT002
        FUNCTION YUNITG(Y)                                                   YUNIT003
C       TO CONVERT FROM Y IN SUBJECT SPACE TO OBJECT SPACE                   YUNIT004
        COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,      YUNIT005
       1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)  YUNIT006
        REAL DEV(1)                                                          YUNIT007
        EQUIVALENCE (DEV,N)                                                  YUNIT008
        EQUIVALENCE (J,ERR(2))                                               YUNIT009
        DATA NAME/6HYUNITG/                                                  YUNIT010
        IF(J.EQ.0)CALL ERRORG(NAME,3,14HNO DEVICE OPEN)                      YUNIT011
        YUNITG=Y*DEV(J+10)+DEV(J+9)                                          YUNIT012
        RETURN                                                               YUNIT013
        END                                                                  YUNIT014
$        FORTRAN DECK,LSTOU                                     12/15/67     INCRPG01
CINCRPG        SUBROUTINE INCRPG                                             INCRPG02
        SUBROUTINE INCRPG(X,Y,N)                                             INCRPG03
C                                                                            INCRPG04
C       X AND Y ARE GIVEN IN INCHES                                          INCRPG05
C                                                                            INCRPG06
C          ROUTINE TO CREATE TAPES COMPATIBLE WITH                           INCRPG07
C          INCREMENTAL PLOTTER.                                              INCRPG08
C                                                                            INCRPG09
C                 X = X COORDINATE TO WHICH PEN IS TO BE MOVED.              INCRPG10
C                 Y = Y COORDINATE TO WHICH PEN IS TO BE MOVED.              INCRPG11
C                 N = MODE INDICATOR.  THE CODE IS                           INCRPG12
C                        1 = NO CHANGE IN PEN POSITION.                      INCRPG13
C                        2 = PLOT WITH PEN DOWN.                             INCRPG14
C                        3 = PLOT WITH PEN UP.                               INCRPG15
C                        0 = INITIALIZE PLOT.                                INCRPG16
C                       -1 = END OF PLOT.  NO REINITIALIZATION IS PERFORMED. INCRPG17
C                       -2 = END PLOT AND REINITIALIZE.                      INCRPG18
C                                                                            INCRPG19
C                                                                            INCRPG20
        INTEGER OLDX,OLDY,XDIFF,YDIFF,PCOUNT,SCOUNT,AXNUM,ANGNUM             INCRPG21
C                                                                            INCRPG22
C         THE NUMBER IN THE FOLLOWING VARIABLES INDICATE THE ANGLE OF THE    INCRPG23
C         WHICH THEY REPRESENT IN DEGREES                                    INCRPG24
C                                                                            INCRPG25
        DATA NAME/ 6HINCRPG /                                                INCRPG26
        DATA IFIRST/0/                                                       INCRPG27
        DATA STP,SZX,SZY/.0025,200.,200./                                    INCRPG28
        DATA IPENSW,IPEN,I0,I45,I90,I135,I180,I225,I270,I315/0,3,1,2,3,4,    INCRPG29
       15,6,7,8/                                                             INCRPG30
C                                                                            INCRPG31
        IF(IFIRST .NE. 0 ) GO TO 55                                          INCRPG32
C.... INITIALIZE FIRST TIME ROUTINE IS CALLED                                INCRPG33
        CALL BLPLTG(0,0   )                                                  INCRPG34
        IFIRST = 1                                                           INCRPG35
```

```
      OLDX = 0                                                          INCRPG36
      OLDY = 0                                                          INCRPG37
      CALL BLPLTG(0,1)                                                  INCRPG38
      IF(N)455,453,60                                                   INCRPG39
 55   IF(N)455,460,60                                                   INCRPG40
C                                                                       INCRPG41
C        DETERMINE NECESSARY PEN INSTRUCTIONS                           INCRPG42
C                                                                       INCRPG43
   60 JPEN=N                                                            INCRPG44
      IF(JPEN-1)160,160,63                                              INCRPG45
   63 IPEN=JPEN                                                         INCRPG46
      IF(IPEN-3)70,110,160                                              INCRPG47
70    IF(IPENSW.NE.0)CALL BLPLTG(0,1)                                   INCRPG48
      IPENSW=0                                                          INCRPG49
      GO TO 160                                                         INCRPG50
110   IF(IPENSW.EQ.0)CALL BLPLTG(9,1)                                   INCRPG51
      IPENSW=1                                                          INCRPG52
160   JX=(X+STP)*SZX                                                    INCRPG53
      IF(JX.LT. 0) CALL ERRORG(NAME, 4, 21HPEN OFF PAPER AT LEFT   )    INCRPG54
      JY=(Y+STP)*SZY                                                    INCRPG55
      IF(JY.LT.0) CALL ERRORG(NAME,4,23HPEN OFF PAPER AT BOTTOM)        INCRPG56
      IF(JY.GT.5800)CALL ERRORG(NAME,4,20HPEN OFF PAPER AT TOP)         INCRPG57
      XDIFF=JX-OLDX                                                     INCRPG58
      YDIFF=JY-OLDY                                                     INCRPG59
C                                                                       INCRPG60
C        DETERMINE ANGLE OF LINE REQUIRED TO REACH NEW COORDINATES      INCRPG61
C     DETERMINE PRIMARY ANGLE AND SECONDARY ANGLE AND HOW MANY          INCRPG62
C     INCREMENTS ALONG PRIMARY ANGLE TO BE FOLLOWED BY HOW MANY         INCRPG63
C     INCREMENTS ALONG SECONDARY ANGLE AND TOTAL NUMBER OF INCREMENTS    INCRPG64
C                                                                       INCRPG65
      IF(XDIFF)175,250,220                                              INCRPG66
  175 IXONLY=I180                                                       INCRPG67
      XDIFF=-XDIFF                                                      INCRPG68
      IF(YDIFF)185,280,200                                              INCRPG69
  185 IANGLE=I225                                                       INCRPG70
  190 IYONLY=I270                                                       INCRPG71
      YDIFF=-YDIFF                                                      INCRPG72
      GO TO 300                                                         INCRPG73
  220 IXONLY=I0                                                         INCRPG74
      IF(YDIFF)235,280,240                                              INCRPG75
  235 IANGLE=I315                                                       INCRPG76
      GO TO 190                                                         INCRPG77
  240 IANGLE=I45                                                        INCRPG78
      GO TO 210                                                         INCRPG79
  250 CONTINUE                                                          INCRPG80
      SCOUNT=0                                                          INCRPG81
      IF(YDIFF)265,453,270                                              INCRPG82
  265 IPRIME=I270                                                       INCRPG83
      PCOUNT=-YDIFF                                                     INCRPG84
      YDIFF=PCOUNT                                                      INCRPG85
      GO TO 290                                                         INCRPG86
  270 IPRIME=I90                                                        INCRPG87
      PCOUNT=YDIFF                                                      INCRPG88
      GO TO 290                                                         INCRPG89
  280 IPRIME=IXONLY                                                     INCRPG90
      PCOUNT=XDIFF                                                      INCRPG91
      SCOUNT=0                                                          INCRPG92
  290 RATIO=PCOUNT                                                      INCRPG93
      GO TO 370                                                         INCRPG94
  200 IANGLE=I135                                                       INCRPG95
```

```
  210 IYONLY=I90
  300 IF(XDIFF-YDIFF)320,310,330
  310 IPRIME=IANGLE
      SCOUNT=0
      PCOUNT=XDIFF
      RATIO=PCOUNT
      GO TO 370
  320 NOTANG=IYONLY
      ANGNUM=XDIFF
      AXNUM=YDIFF-XDIFF
      GO TO 340
  330 NOTANG=IXONLY
      ANGNUM=YDIFF
      AXNUM=XDIFF-YDIFF
  340 IF(ANGNUM-AXNUM)350,345,360
  345 RATIO=1.
      GO TO 355
  350 ANU=AXNUM
      DEN=ANGNUM+1
      RATIO=ANU/DEN
  355 CONTINUE
      IPRIME=NOTANG
      ISEC=IANGLE
      PCOUNT=AXNUM
      SCOUNT=ANGNUM
      GO TO 370
  360 ANU=ANGNUM
      DEN=AXNUM+1
      RATIO=ANU/DEN
      IPRIME=IANGLE
      ISEC=NOTANG
      PCOUNT=ANGNUM
      SCOUNT=AXNUM
  370 PRCNT=RATIO
C
C     ANGLES AND COUNTS HAVE BEEN DETERMINED
C
  380 CALLBLPLTG  (IPRIME,1)
      PCOUNT=PCOUNT-1
      PRCNT=PRCNT-1.
  400 IF(PCOUNT)440,440,410
  410 IF(PRCNT-1.) 420,380,380
  420 PRCNT=PRCNT+RATIO
      IF(SCOUNT)380,380,430
  430 CALL BLPLTG(ISEC,1)
      SCOUNT=SCOUNT-1
      GO TO 400
  440 IF(SCOUNT)450,450,430
C
C     FINISHED
C
  450 OLDX=JX
      OLDY=JY
  453 RETURN
  455 CALL BLPLTG(0,0)
      IF(N.EQ.(-1))RETURN
C
C       INITIALIZE REMAINING PARAMETERS AND RAISE PEN
C
460   CONTINUE
```

```
      OLDX=0                                                            INCRPG56
      OLDY=0                                                            INCRPG57
      CALL BLPLTG(0,0)                                                  INCRPG58
      IF(IPENSW.EQ.0)CALL BLPLTG(9,1)                                   INCRPG59
      IPENSW=1                                                          INCRPG60
      RETURN                                                            INCRPG61
      END                                                               INCRPG62
$       FORTRAN DECK,LSTOU                                              BLPLTG01
CBLPLTG          SUBROUTINE BLPLTG                                      BLPLTG02
      SUBROUTINE BLPLTG(IW,N)                                           BLPLTG03
C                                                                       BLPLTG04
C        ROUTINE BLPLTG  - USED BY ROUTINE INCRPG TO WRITE PLOTTER INSTRUBLPLTG05
C        ON TAPE.                                                       BLPLTG06
C                                                                       BLPLTG07
C             IW = CODE NUMBER FOR INSTRUCTION TO BE USED               BLPLTG08
C             N  = REINITIALIZATION INDICATOR                           BLPLTG09
C                  0 = WRITE INSTRUCTION ARRAY, END PLOT, AND CREATE NEXTBLPLTG10
C                      PLOT IDENTIFICATION NUMBER                       BLPLTG11
C                  ANY POSITIVE NUMBER INDICATES NORMAL INSTRUCTION CREATBLPLTG12
C                  ANY NEGATIVE NUMBER IS AN INVALID VALUE - NO ACTION TABLPLTG13
C                                                                       BLPLTG14
C        SUBROUTINES REQUIRED - NONE.                                   BLPLTG15
C                                                                       BLPLTG16
C        SIZE OF INSTRUCTION ARRAY MAY BE CHANGED BY CHANGING VALUES OF  BLPLTG17
C        THE DIMENSION OF KWDS, THE VALUE OF NWDS, AND THE SIZE OF FIXLNGBLPLTG18
C        BUFSIZ ON THE FFILE CONTROL CARD.                              BLPLTG19
C                                                                       BLPLTG20
      COMMON /AMODES/ DEV(38)                                           BLPLTG21
      EQUIVALENCE (JJJJJG,DEV(19)        )                              BLPLTG22
      DIMENSION LWDS(10)                                                BLPLTG23
      DIMENSION KWDS(1994)                                              BLPLTG24
      DIMENSION IC(9),IA(2)                                             BLPLTG25
      EQUIVALENCE (NUM1,IN1),(NUM2,IN2),(IPNO,ID)                       BLPLTG26
      EQUIVALENCE(IA(2),IC(1))                                          BLPLTG27
      EQUIVALENCE (LWDS(1),IPDN),(LWDS(2),I0),(LWDS(3),I45),            BLPLTG28
     1(LWDS(4),I90),(LWDS(5),I135),(LWDS(6),I180),(LWDS(7),I225),       BLPLTG29
     2(LWDS(8),I270),(LWDS(9),I315),(LWDS(10),IPUP)                     BLPLTG30
C                                                                       BLPLTG31
C        VALUES FOR CREATION OF PLOT IDENTIFICATION NUMBERS             BLPLTG32
C                                                                       BLPLTG33
      DATA IC/0100410050000,0100410060000,0100410070000,0100510040000,  BLPLTG34
     10100510050000,0100510060000,0100510070000,0100610040000,0100610050BLPLTG35
     2000/                       ,MASK/0000077770000/,IA(1)/0100410040000/ BLPLTG36
C                                                                       BLPLTG37
      DATA NWDS/1994/,IPNO,JUP,JDOWN,I/1,18,75,0/
C                                                                       BLPLTG39
C        PEN INSTRUCTIONS - 'H' AND '8' ARE FILLER CHARACTERS TO SIMULATEBLPLTG40
C        DENSITY TAPE READ BY PLOTTER.                                  BLPLTG41
C                                                                       BLPLTG42
      DATA IPUP/6HH68685/,I0/6HH78685/,I45/6HH78786/,I90/6HH68786/,     BLPLTG43
     1I135/6HH58786/,I180/6HH58686/,I225/6HH58586/,I270/6HH68586/,      BLPLTG44
     2I315/6HH78586/,IPDN/6HH68687/,IP1/6HH48383/,IP2/6HH38383/,        BLPLTG45
     3IP3/6HH38382/,IHD2/6HH38381/,IHD3/6HH18383/,IHD4/6HH38384/,       BLPLTG46
     4JEND1/6HH48683/,JEND2/6HH48000/,NOP/6HH48484/                     BLPLTG47
C                                                                       BLPLTG48
      IFILE = DEV(JJJJJG+2)                                             BLPLTG49
      IF(N)270,300,30                                                   BLPLTG50
   30 JW=IW+1                                                           BLPLTG51
      I=I+1                                                             BLPLTG52
C                                                                       BLPLTG53
```

```
C         CREATE INSTRUCTION, ADDING FILLER AFTER ANY PEN INSTRUCTION        BLPLTG54
C                                                                             BLPLTG55
      KWDS(I)=LWDS(JW)                                                        BLPLTG56
      IF(JW.NE.1)IF(JW-10)240,60,60                                           BLPLTG57
C     PEN DOWN                                                                BLPLTG58
C     WAS PREVIOUS COMMAND A PEN UP                                           BLPLTG59
      IF(I.LT.20) GO TO 51                                                    BLPLTG60
      IF(KWDS(I-19).NE.IPUP) GO TO 51                                         BLPLTG61
      I=I-20                                                                  BLPLTG62
      RETURN                                                                  BLPLTG63
 51   LIM = JDOWN                                                             BLPLTG64
      GO TO 225                                                               BLPLTG65
C     PEN UP                                                                  BLPLTG66
 60   CONTINUE                                                                BLPLTG67
 61   LIM=JUP                                                                 BLPLTG68
  225 IF(I+LIM-NWDS)230,280,280                                               BLPLTG69
  230 DO 235 LL=1,LIM                                                         BLPLTG70
      I=I+1                                                                   BLPLTG71
  235 KWDS(I)=0                                                               BLPLTG72
      RETURN                                                                  BLPLTG73
240   IF(I.LT.NWDS)RETURN                                                     BLPLTG74
C                                                                             BLPLTG75
C         WRITE INSTRUCTION ARRAY IF IT IS FULL.                              BLPLTG76
C                                                                             BLPLTG77
  280 WRITE(IFILE)IP1,IP2,IP3,(KWDS(J),J=1,I),JEND1,JEND2                     BLPLTG78
      I=0                                                                     BLPLTG79
  290 IF(N)270,320,270                                                        BLPLTG80
  300 IF(I)320,320,280                                                        BLPLTG81
C                                                                             BLPLTG82
C         CREATE PLOT IDENTIFICATION NUMBERS                                  BLPLTG83
C                                                                             BLPLTG84
  320 I1= ID/100                                                              BLPLTG85
      I2=(ID-I1*100)/10                                                       BLPLTG86
      I3=ID-I1*100-I2*10                                                      BLPLTG87
      NUM1=IC(I1)                                                             BLPLTG88
      II=IC(I2)/16777216                                                      BLPLTG89
      NUM1=OR(NUM1,II)                                                        BLPLTG90
      NUM2=AND(MASK,IC(I2))                                                   BLPLTG91
      NUM2=NUM2*4096                                                          BLPLTG92
      II=IC(I3)/4096                                                          BLPLTG93
      NUM2=OR(NUM2,II)                                                        BLPLTG94
      WRITE(IFILE)IP1,IP2,IHD2,IN1,IN2,IHD3,IP2,IHD4,(NOP,KQ=9,1999)          BLPLTG95
      IPNO=IPNO+1                                                             BLPLTG96
      IF(IPNO-999)270,270,330                                                 BLPLTG97
  330 IPNO=1                                                                  BLPLTG98
270   CONTINUE                                                                BLPLTG99
      RETURN                                                                  BLPLTG00
      END                                                                     BLPLTG01
$      FORTRAN DECK,LSTOU                                                     ERRORG01
CERRORG                                                                       ERRORG02
      SUBROUTINE ERRORG(IR,NN,IA)                                             ERRORG03
C     ERROR MESSAGES                                                          ERRORG04
      COMMON/AMODES/N,XLEFT,XRIGHT,YLOWER,YUPPER,XLAST,YLAST,TEXTUR,          ERRORG05
     1ANTENS,COLOR,FONT,CHWDTH,CHIGHT,ORIENT,IPLTCH,FSTATS,NDEV,ERR(21)       ERRORG06
      REAL DEV(1)                                                             ERRORG07
      EQUIVALENCE (DEV,N)                                                     ERRORG08
      EQUIVALENCE (J,ERR(2))                                                  ERRORG09
      DIMENSION IA(1)                                                         ERRORG10
      DATA  IFIRST /1/                                                        ERRORG11
      DATA NAME/6HERRORG/                                                     ERRORG12
```

```
      WRITE(6,1)IR,(IA(I),I=1,NN)                                        ERRORG13
1     FORMAT(18H0ERROR IN ROUTINE ,A6///(1X,21A6))                       ERRORG14
       NWDS=N+1+11*NDEV                                                   ERRORG15
       WRITE(6,5) (DEV(I),I=1,NWDS)                                       ERRORG16
 5    FORMAT(1X3HN =I5,1X7HXLEFT =F11.4,1X8HXRIGHT =F11.4,1X8HYLOWER =F  ERRORG17
     111.4,1X8HYUPPER =F11.4/1X7HXLAST = F11.4,1X7HYLAST =F11.4,1X8HTEXTERRORG18
     2UR = F7.1,1X8HANTENS =F7.1,1X7HCOLOR = F7.1/1X6HFONT = F7.0,1X8HCHERRORG19
     3WDTH = F7.3,1X8HCHIGHT =F7.3/1X8HORIENT =F8.3,1X8HPLOTCH =F8.3,1X7ERRORG20
     4HFSTAT =F8.1/1X8HNUMDEV =I8,8H UNITS =F9.3,1X9HJPOINTR =I5,1X7HAINERRORG21
     5CR =F7.5/1X7HXCHAR =F10.4,1X7HYCHAR =F10.4,1X8HMAXCHG =I8,1X4HDX =ERRORG22
     6F9.4,7X4HDY =F9.4/9H ENDFLG =F5.1,1X8HPT01SG =F9.4/(1X8HTYPDVC =F6ERRORG23
     7.1,1X7HFILCD =F6.0,1X6HXMIN =F11.4,1X6HXMAX = F11.4,1X6HYMIN =F11.ERRORG24
     84,7H YMAX =F11.4/1X6HXOFF =F11.4,1X8HXSCALE =F11.4,1X6HYOFF =F11.4ERRORG25
     9,1X8HYSCALE =F14.4,1X8HNPLOTS =I5))                                ERRORG26
      IF(IFIRST .NE. 1) GO TO 10                                         ERRORG27
      IFIRST = 0                                                         ERRORG28
      CALL TRACBK                                                        ERRORG29
      ERR(1) = 1.0                                                       ERRORG30
      CALL FINISG                                                        ERRORG31
 10   CALL BORTIT                                                        ERRORG32
      STOP                                                               ERRORG33
      END                                                                ERRORG34
$      FORTRAN DECK,LSTOU                                      1/26/68   ROTAX001
CROTAXG                     ROTATE AXES                                  ROTAX002
      SUBROUTINE ROTAXG(X,Y,N,ANG)                                       ROTAX003
C.... ROTATE X AXES TOWARD Y AXES BY ANG DEGREES (RIGHT HAND RULE)       ROTAX004
      REALX(N),Y(N),DUM(2)                                               ROTAX005
      COMPLEX   R, H                                                     ROTAX006
      EQUIVALENCE(R,DUM(1)),(RX,DUM(1)),(RY,DUM(2))                      ROTAX007
      IF ( (N.LE. 0) .OR. (ANG .EQ. 0.0) ) RETURN                        ROTAX008
      ANGR=ANG*0.017453                                                  ROTAX009
      H=CEXP(CMPLX(0.,ANGR))                                             ROTAX010
      DO 10 I=1,N                                                        ROTAX011
      RX=X(I)                                                            ROTAX012
      RY=Y(I)                                                            ROTAX013
      R=H*R                                                              ROTAX014
      X(I)=RX                                                            ROTAX015
   10 Y(I)=RY                                                            ROTAX016
      RETURN                                                             ROTAX017
      END                                                                ROTAX018
$      FORTRAN DECK,LSTOU                                      3/ 4/68   FPTBC001
CFPTBCD                                                                  FPTBC002
      SUBROUTINE FPTBCD(F,IW,ID,IARRAY)                                  FPTBC003
C     CONVERT BIN. FLOATING PT. NUMBER F USING F FORMAT OF WIDTH IW AND  FPTBC004
C     ID DECIMAL PLACES AND RETURNS THE IW CHARS PACKED IN IARRAY SO     FPTBC005
C     THAT LEGNDG MAY THEN BE CALLED                                     FPTBC006
      INTEGER FMT(5),IARRAY(1)                                           FPTBC007
      DATA FMT/5H(1X,F,1H ,1H.,1H ,1H)/                                  FPTBC008
      FMT(2)=NCHAR(IW)                                                   FPTBC009
      FMT(4)=NCHAR(ID)                                                   FPTBC010
      J=(IW+5)/6                                                         FPTBC011
      CALL RREAD1                                                        FPTBC012
      WRITE(40,FMT)F                                                     FPTBC013
      CALL RREAD2                                                        FPTBC014
      READ(40,1)(IARRAY(I),I=1,J)                                        FPTBC015
1     FORMAT(1X,20A6)                                                    FPTBC016
      RETURN                                                             FPTBC017
      END                                                                FPTBC018
$      GMAP     DECK,LSTOU,NXEC                                1/26/68   IARG 001
       LBL      IARG                                                     IARG 002
```

```
        SYMDEF   IARG                                                     IARG 003
*                THIS IS A SPECIAL PURPOSE ROUTINE WHICH MAY BE DELETED IFIARG 004
*                DESIRED.  IT ENABLES A FORTRAN ROUTINE TO HAVE A VARIABLEIARG 005
*                NUMBER OF ARGUMENTS.  IT RETURNS THE NUMBER OF           IARG 006
*                ARGUMENTS AS ITS VALUE AND PLACES THE ARGUMENTS IN AN    IARG 007
*                ARRAY.                                                   IARG 008
IARG    STX1     RET                                                      IARG 009
        LDX0     2,1                                                      IARG 010
        STX0     TLY                                                      IARG 011
        LDX1     1,1                                                      IARG 012
        LDX1     0,1                                                      IARG 013
        ADX1     1,DU                                                     IARG 014
        LDX0     0,DU                                                     IARG 015
LOOP    ADX1     1,DU                                                     IARG 016
        LDA      0,1                                                      IARG 017
        ANA      =0777000,DL                                              IARG 018
        TNZ      OUT                                                      IARG 019
        LDA      0,1*                                                     IARG 020
        STA      TLY,ID                                                   IARG 021
        ADX0     1,DU                                                     IARG 022
        TRA      LOOP                                                     IARG 023
OUT     EAQ      0,0                                                      IARG 024
        QRL      18                                                       IARG 025
RET     TRA      **                                                       IARG 026
TLY     TALLY    **                                                       IARG 027
        ERLK                                                              IARG 028
        END                                                               IARG 029
$       FORTRAN DECK,LSTOU                                                TRANS001
CTRANSG                   PERSPECTIVE TRANSFORMATION                      TRANS002
C        Z IS UP                                                          TRANS003
C                 AUG 26, 1968              F.J. ROHLF                    TRANS004
      SUBROUTINE TRANSG(N,X,Y,Z,XP,YP)                                    TRANS005
      REAL  X(N),Y(N),Z(N),XP(N),YP(N),K                                  TRANS006
C        ARITH STAT FN FOLLOWS                                            TRANS007
         ACOS(C)=ATAN(SQRT(1.-C*C)/C)/.017453                             TRANS008
C                                                                         TRANS009
      DO 5 I=1,N                                                          TRANS010
C     COMPUTE COMMON RATIO                                                TRANS011
      K=D/((X(I)-CX)*CA+(Y(I)-CY)*CB+(Z(I)-CZ)*CG)                        TRANS012
C     COMPUTE COORDINATES OF IMAGE                                        TRANS013
      XI=CX+K*(X(I)-CX)                                                   TRANS014
      ETA=CY+K*(Y(I)-CY)                                                  TRANS015
      ZETA=CZ+K*(Z(I)-CZ)                                                 TRANS016
       GO TO INS1,(2,10)                                                  TRANS017
C     COMPUTE COORDINATES OF IMAGE RELATIVE TO PLANE                      TRANS018
 2     XP(I)=-((XI-QX)*CB-(ETA-QY)*CA)/SG                                 TRANS019
      YP(I)=(ZETA-QZ)/SG                                                  TRANS020
      GO TO 5                                                             TRANS021
 10    XP(I)=(-(XI-QX)*CG+(ZETA-QZ)*CA)/SB                                TRANS022
      YP(I)=(ETA-QY)/SB                                                   TRANS023
 5    CONTINUE                                                            TRANS024
      RETURN                                                              TRANS025
C     ENTRY POINT TO INITIALIZE ROUTINE.  ANGLES A,B, AND G DETERMINE     TRANS026
C     LINE OF SIGHT FROM POINT (CX,CY,CZ) TO THE PROJECTION PLANE AT A    TRANS027
C     DISTANCE  D FROM THE VIEWING POSITION)                              TRANS028
      ENTRY ITRANG(A,B,G,D,CX,CY,CZ)                                      TRANS029
      ALPHA=A*.017453                                                     TRANS030
      BETA=B*.017453                                                      TRANS031
      GAMMA=G*.017453                                                     TRANS032
      SG=SIN(GAMMA)                                                       TRANS033
```

```
      ASSIGN 2 TO INS1                                                  TRANS034
      IF(SG.NE.0.0) GO TO 20                                            TRANS035
      ASSIGN 10 TO INS1                                                 TRANS036
      SB=SIN(BETA)                                                      TRANS037
 20   CA=COS(ALPHA)                                                     TRANS038
      CB=COS(BETA)                                                      TRANS039
      CG=COS(GAMMA)                                                     TRANS040
C     COMPUTE COORDINATES OF INTERSECTION BETWEEN LINE OF SIGHT AND     TRANS041
C     PROJECTION PLANE                                                  TRANS042
      QX=CX+D*CA                                                        TRANS043
      QY=CY+D*CB                                                        TRANS044
      QZ=CZ+D*CG                                                        TRANS045
      RETURN                                                            TRANS046
C                                                                       TRANS047
C     COMPUTE ANGLES BETWEEN X,Y,Z AXES AND THE LINE FROM C TO CC       TRANS048
      ENTRY ANGPLG(CX,CY,CZ,CCX,CCY,CCZ,A,B,G,D)                        TRANS049
      D=SQRT((CX-CCX)*(CX-CCX)+(CY-CCY)*(CY-CCY)+                       TRANS050
     1 (CZ-CCZ)*(CZ-CCZ) )                                              TRANS051
      A = ACOS((CCX-CX)/D)                                              TRANS052
      IF(A.LT.0.) A=A+180.                                              TRANS053
      B = ACOS((CCY-CY)/D)                                              TRANS054
      IF(B.LT.0.) B=B+180.                                              TRANS055
      G = ACOS((CCZ-CZ)/D)                                              TRANS056
      IF(G.LT.0.) G=G+180.                                              TRANS057
      RETURN                                                            TRANS058
      END                                                               TRANS059
$     FORTRAN DECK,LSTOU                                    12/15/67    BOX   001
CBOX             SUBROUTINE TO COMPUTE CORNERS OF BOX                   BOX   002
      SUBROUTINE BOX(XL,XR,YL,YU,X,Y)                                   BOX   003
      DIMENSION X(5),Y(5)                                               BOX   004
      DO 10 I = 1,2                                                     BOX   005
      IND = 1 + (I-1)*3                                                 BOX   006
      X(IND) = XL                                                       BOX   007
      IND = 2 + (I-1)                                                   BOX   008
      X(IND) = XR                                                       BOX   009
      Y(I) = YL                                                         BOX   010
 10   Y(I+2) = YU                                                       BOX   011
      X(5) = XL                                                         BOX   012
      Y(5) = YL                                                         BOX   013
      RETURN                                                            BOX   014
      END                                                               BOX   015
$     FORTRAN DECK,LSTOU                                    12/15/67    MINMA001
CMINMAX          SUBROUTINE MINMAX                                      MINMA002
      SUBROUTINE MINMAX(N,X,XMIN,XMAX)                                  MINMA003
      DIMENSION X(N)                                                    MINMA004
      XMIN=X(1)                                                         MINMA005
      XMAX=XMIN                                                         MINMA006
      DO 10 I = 1,N                                                     MINMA007
      IF(X(I).GT.XMAX) XMAX=X(I)                                        MINMA008
      IF(X(I).LT.XMIN) XMIN= X(I)                                       MINMA009
 10   CONTINUE                                                          MINMA010
      RETURN                                                            MINMA011
      END                                                               MINMA012
$     FORTRAN DECK,LSTOU                                                CIRTRG01
CCIRTRG        SUBROUTINE CIRTRG                                        CIRTRM02
      SUBROUTINE CIRTRG(XS,YS,XF,YF,RS,RF)                              CIRTRM03
      D = XF - XS                                                       CIRTRM04
      IF(D.EQ. 0.0) D = 1.0 E - 6                                       CIRTRM05
      T = (YF-YS)/D                                                     CIRTRM06
      IF(D.LT.0.0) CALL SWITCH(XS,YS,XF,YF,RS,RF)                       CIRTRM08
```

```
 20     THETA1 = ATAN(T)                                                    CIRTRM11
        XS=XS+RS* COS(THETA1)                                               CIRTRM12
        YS=YS+RS* SIN(THETA1)                                               CIRTRM13
        THETA2=3.14159+THETA1                                               CIRTRM14
        XF=XF+RF* COS(THETA2)                                               CIRTRM15
        YF=YF+RF* SIN(THETA2)                                               CIRTRM16
        IF(D.LT.0.0) CALL SWITCH(XS,YS,XF,YF,RS,RF)                         CIRTRM17
        RETURN                                                              CIRTRM18
        END                                                                 CIRTRM19
$        FORTRAN DECK,LSTOU                                                 SWITCH01
CSWITCH         SUBROUTINE SWITCH  FOR MOD3DP                               SWITCH02
        SUBROUTINE SWITCH (I,J,L,M,IR,LR)                                   SWITCH03
        IS = I                                                              SWITCH04
        JS = J                                                              SWITCH05
        I = L                                                               SWITCH06
        J = M                                                               SWITCH07
        L = IS                                                              SWITCH08
        M = JS                                                              SWITCH09
        IS=IR                                                               SWITCH10
        IR=LR                                                               SWITCH11
        LR=IS                                                               SWITCH12
        RETURN                                                              SWITCH13
        END                                                                 SWITCH14
$       GMAP     DECK,LSTOU,NXEC                                            TRCB    1
        LBL      TRCBK                                                      TRCB    2
TRACBK  SAVE                                                                TRCB    3
        CALL  .FXEM.(X)                                                     TRCB    4
        RETURN   TRACBK                                                     TRCB    5
X       ZERO     0,1                                                        TRCB    6
        ZERO     MES1,2                                                     TRCB    7
        ZERO     MES2,1                                                     TRCB    8
MES1    BCI      2,TRACEBACK                                                TRCB    9
MES2    BCI      1,                                                         TRCB   10
        END                                                                 TRCB   11
```

KANSAS GEOLOGICAL SURVEY COMPUTER PROGRAM
THE UNIVERSITY OF KANSAS, LAWRENCE

PROGRAM ABSTRACT

Title (If subroutine state in title):

GRAFPAC (system of FORTRAN IV graphic output subroutines)

Date: 18 February 1969

Author, organization: F. James Rohlf, Department of Entomology
The University of Kansas

Direct inquiries to: After July, 1969

Name: F. James Rohlf

Address: Department of Biology
State University of New York
Stony Brook, New York

Purpose/description: GRAFPAC consists of a system of FORTRAN callable subroutines to facilitate graphic output. Routines also are provided for 3-D perspective drawings of surfaces with hidden lines deleted.

Mathematical method:

Restrictions, range:

Computer manufacturer: GE

Model: 625 or 635

Programming language: FORTRAN, GEMAP

Memory required: K

Approximate running time: Depends upon application

Special peripheral equipment required: Tape drive and B/L plotter

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)

Uses only standard FORTRAN features but several routines assume 36 bit BCD character words. Routines have been used successfully by several persons over the past two years.

Routines specific to B/L plotters can easily be replaced for use with other plotters.

User must supply his own mainline program!

Computer Contribution

1. Mathematical simulation of marine sedimentation with IBM 7090/7094 computers, by J.W. Harbaugh, 1966 . . . . (out of print)
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 $1.50
24. Mathematical model and FORTRAN IV program for computer simulation of deltaic sedimentation, by G.F. Bonham-Carter and A.J. Sutherland, 1968 . . . . . $1.00
25. FORTRAN IV CDC 6400 computer program for analysis of subsurface fold geometry, by E.H.T. Whitten, 1968 . . . . $1.00
26. FORTRAN IV computer program for simulation of transgression and regression with continuous-time Markov models, by W.C. Krumbein, 1968 . . . . . . $1.00
27. Stepwise regression and nonpolynomial models in trend analysis, by A.T.Miesch and J.J. Connor, 1968 . . . . $1.00
28. KWIKR8 a FORTRAN IV program for multiple regression and geologic trend analysis, by J.E. Esler, P.F. Smith, and J.C. Davis, 1968 . . . . $1.00
29. FORTRAN IV program for harmonic trend analysis using double Fourier series and regularly gridded data for the GE 625 computer, by J.W. Harbaugh and M.J. Sackin, 1968 . . $1.00
30. Sampling a geological population (workshop on experiment in sampling), by J.C. Griffiths and C.W. Ondrick, 1968 . . . . $1.00
31. Multivariate procedures and FORTRAN IV program for evaluation and improvement of classifications, by Ferruh Demirmen, 1969 . . . . $1.00
32. FORTRAN IV programs for canonical correlation and canonical trend-surface analysis, by P.J. Lee, 1969. . . . . $1.00
33. FORTRAN IV program for construction of Pi diagrams with the Univac 1108 computer, by Jeffrey Warner, 1969 . . . . $1.00
34. FORTRAN IV program for nonlinear estimation, by R.B. McCammon, 1969 . . . . $0.75
35. FORTRAN IV computer program for fitting observed count data to discrete distribution models of binomial, Poisson and negative binomial, by C.W. Ondrick and J.C. Griffiths, 1969 . . . . $0.75
36. GRAFPAC, graphic output subroutines for the GE 635 computer, by F.J. Rohlf, 1969 . . $1.00