N73 24213

NASA TECHNICAL MEMORANDUM

NASA TM X-64744

COMPUTER USER'S MANUAL FOR A GENERALIZED CURVE FIT AND PLOTTING PROGRAM

By Ronald A. Schlagheck,
B. D. Beadle II, B. D. Dolerhie, Jr.,
and J. W. Owen
Astronautics Laboratory

CASE FILE COPY

January 1, 1973

NASA

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

MSFC - Form 3190 (Rev June 1971)

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.
NASA TM X- 64744

2. GOVERNMENT ACCESSION NO.

3. RECIPIENT'S CATALOG NO.

4. TITLE AND SUBTITLE
Computer User's Manual for a Generalized Curve Fit and Plotting Program

5. REPORT DATE
January 1, 1973

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S) Ronald A. Schlagheck, B. D. Beadle II, B. D. Dolerhie, Jr., J. W. Owen

8. PERFORMING ORGANIZATION REPORT #

9. PERFORMING ORGANIZATION NAME AND ADDRESS
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.

13. TYPE OF REPORT & PERIOD COVERED
Technical Memorandum

12. SPONSORING AGENCY NAME AND ADDRESS
National Aeronautics and Space Administration
Washington, D. C. 20546

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES
Prepared by Astronautics Laboratory,
Science and Engineering

16. ABSTRACT

A Fortran-coded program has been developed for generating plotted output graphs on 8-1/2 by 11-inch paper. The program is designed to be used by engineers, sceintists, and non-programming personnel on any IBM 1130 system that includes a 1627 plotter. The program has been written to provide a fast and efficient method of displaying plotted data without having to generate any additional Fortran instructions. Various output options are available to the program user for displaying data in four different types of formatted plots. These options include discrete, linear, continuous, and histogram graphical outputs. The manual contains information about the use and operation of this program. Example cases illustrate the sample input and output for five selected plots. A mathematical description of the least-squares "goodness of fit" test is presented. A program listing is also included.

17. KEY WORDS
regression analysis
computer software
probability and statistics
plotting and graphing

18. DISTRIBUTION STATEMENT
Unclassified - Unlimited
Ronald A. Schlagheck

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	65	NTIS

For sale by National Technical Information Service, Springfield, Virginia 22151

Page Intentionally Left Blank

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

COMPUTER USER'S MANUAL FOR A GENERALIZED CURVE FIT AND PLOTTING PROGRAM

SECTION I. INTRODUCTION

The use of graphically displayed data is often quite beneficial to engineers, scientists, and other decision makers. Such graphs can be in the form of reporting charts, presentation viewgraphs, and technical illustrations. A program has been developed for the IBM 1130[1] computing system (with a 1627 plotter) that enables a non-computer oriented individual to easily generate a plot of a set or sets of numerical data on 8.5×11 in. standard[2] paper. This program is entitled the Generalized Curfit and Plotting (GECAP) program. This computer program has various output options in which it can display the data in different forms. These options include discrete, linear, continuous, and histogram types of output plots. The continuous option also performs a least-squares curve fit (Regression Analysis) on the input data. A statistical analysis of variance test determines the best-degree polynomial fit, and a printout of the polynomial coefficients is given.

Graph titles, coordinate labels, and symbol identification are included on each plot. The user of the program needs only to supply two instruction cards and the necessary data to be plotted.

The rest of this manual contains information about the use and operation of GECAP. Five example plots and associated input data are illustrated. These examples demonstrate various combinations of the input options available to the program. A brief summary of the program description is included along with Fortran listings of the coded subroutines. Modification of the program can be easily undertaken by experienced programmers since the logic

1. This program can be readily converted over to the other computing systems with minor modification.

2. Any type of paper may be used; however, this size is most common in written reports and viewgraphs.

operations within the program listing have been well illustrated. A comprehensive list of various error messages is also contained within the program logic.

SECTION II. CONTROL CARDS, INPUT DATA AND OPERATING INSTRUCTIONS

A. Control Cards

The following is a description of the control cards necessary for the program.

- Control Card 1 may be either a cold start card or a job card A cold start card is mandatory each time a different disc is changed out of the computer. A job card has the following form:

// JOB

- Control Card 2 is an execution card and has the following form:

// XEQ GECAP 1

- Control Card 3 initiates the segmentation of the program in order to minimize the core requirements for GECAP. This card has the following form:

*LOCALGECAP, LABEL, HIST, ERRO

B. Input Data

The following is a description of the data cards necessary for the program.

- Case Card

The format for this card is (A4, I2, lX, A2, I2, 3X, 5A6, 3X, I2, 3X, I2)

Columns 1-4: Punch the letters CASE.

Columns 5-6: Punch the number of the case. (Punch 1 if the first case, 2 if second case, etc.)

Columns 8-9: Punch the letters OF.

Columns 10-11: Punch the total number of cases to be plotted per graph.

Columns 15-44: These spaces are reserved for the name or label the user may wish to place on each individual data set.[3] (Optional, leave blank if not needed)

Columns 48-49: These columns are reserved for the exponent used to label the x axis.[4] (Optional, leave blank if not needed)

Columns 53-54: These columns are reserved for the exponent used to label the y axis.[4] (Optional, leave blank if not needed)

Columns 55-80: These columns should always be left blank.

An example of this card is given as follows:

Columns

1-4	6	8-9	11	15-44	48-49	53-54
CASE	1	OF	4	TEST RUN 24D	-1	-1

- Program Instruction Card

This card contains information about how the grid should be drawn and how the data should be plotted. The format for this card is (6F10.4, 4A1)

---

3. This name will be printed below the x axis and will be preceded by a data symbol corresponding to the data mark symbols on the curve of interest.

4. These exponents will cause the last labeled 'tic' mark on each axis to be followed by a '$\times 10^{exp}$', where exp is the input value. ($-9 \le exp \le 99$). Example: 20×10^{-1}. See Example Case 4 for use of this input variable.

Columns 1-10: XINC - This variable is the numerical increment between 'TIC' marks for the x axis.

Columns 11-20: YINC - Same as XINC except for the y axis.

Columns 21-30: XMAX - Upper limit for the x axis.

Columns 31-40: YMAX - Upper limit for the y axis.

Columns 41-50: XSTRT - Value assigned to the x origin.

Columns 51-60: YSTRT - Value assigned to the y origin.

NOTE: A restriction of the program requires that none of the six variables defined above contain more than two decimal places. If more decimal places are used, an error message will be printed. Execution may be continued but caution is advised, as invalid results may be generated.

Columns 61: TYPE (1) - Label type (integer or real) to be placed on the x axis. A 'D' punched will plate integer numbers on the axis, and either an 'L' or 'C' punched will place real numbers on the axis.

Column 62: TYPE (2) - Same as TYPE (1) but for the y axis.

Column 63: TYPE (3) - Type of plot to be generated (discrete, continuous, linear, or histogram). The options are specified as follows:

Punch	Option
D	This punch will cause the discrete option to be executed. Only the input data is plotted and the points are not connected in any manner.
C	This punch exercises the continuous plot option. After the individual points are plotted, the program calculates a best-fit function using the least-squares method and then plots the calculated function. (See Appendix A for details.)

Punch	Option
L	This punch causes the program to plot each individual point and at the same time connect each point to the previous one with a straight line.
H	This punch causes a histogram to be generated from the input.

Column 64: NODAT - Punch any symbol in this column and no data points will be marked on the grid.

- An example of this card is as follows:

Columns

1-10	11-20	21-30	31-40	41-50	51-60	61	62	63	64
1.0	2.0	9.0	90.0	1.0	60.0	D	C	C	*

- Title Card

This data card contains the labels for the grid. The format for this data card is (6A4, A2, 6A4, A2, 7A4).

TITLX - This is a string of 26 characters to be placed on the x axis.

TITLY - This is a string of 26 characters to be placed on the y axis.

TITLE - This is a string of 28 characters to be placed above the grid. An example of the title card is as follows:

Columns

1-26	27-52	53-80
PRESSURE CO_2 MM-HG	EFFICIENCY-PCT.	MOLECULAR SIEVE

- Deck of x and y coordinates

These data cards contain the x and y values of the data to be plotted and are punched according to the format 2F10.4.

When using the continuous or histogram option, the data points must be entered in the data set in order of increasing x value; that is, x_1 must be less than x_2; x_2 must be less than x_3, ... x_{n-1} must be less than x_n. If this restriction is violated, invalid results will be obtained.

- Data Card

Columns

1-10	11-20
1.53	100.5

- End Card

An END card must be placed at the end of each data set. This card has the following form:

Columns

21-23
END

Multiple curves can be plotted on a single graph. Data for each new set can be stacked behind the previous case by adding a new CASE card and END card. Figure 1 illustrates a generalized GECAP deck setup. Section II shows a variety of different example plots which demonstrate the use of the different options available to the GECAP user.

C. Operating Instructions

The following is a list of operating instructions for using GECAP on the IBM 1130.

1. Place disc in position in the disc storage unit.

2. Turn file ON/OFF switch to the ON position.

3. Wait for the FILE READY light to appear on the console typewriter.

4. Press the NPRO button on card reader.

5. Load the GECAP deck.

GENERALIZED GECAP DECK SET-UP

```
000000000111111111122222222223333333333444444444455555555556666666666777777
123456789012345678901234567890123456789012345678901234567890123456789012345
// JOB    OR    //***COLD START CARD***//
// XEQ GECAP     1
*LOCALGECAP,LABEL,HIST,ERRO
CASE CARD 1
PROGRAM INSTRUCTION CARD
TITLE CARD
XVALUE(1) YVALUE(1)          (THIS CARD BEGINS THE DATA SET FOR CASE 1)
XVALUE(2) YVALUE(2)
  -
  -
  -
XVALUE(N) YVALUE(N)
                    END
CASE CARD 2
XVALUE(1) YVALUE(1)          (THIS CARD BEGINS THE DATA SET FOR CASE 2)
XVALUE(2) YVALUE(2)
  -
  -
  -
XVALUE(N) YVALUE(N)
                    END
  -
  -

  -
  -
CASE CARD M
XVALUE(1) YVALUE(1)          (THIS CARD BEGINS THE DATA SET FOR CASE M)
XVALUE(2) YVALUE(2)
  -
  -
  -
XVALUE(N) YVALUE(N)
                    END


--NOTE--MAXIMUM VALUE OF N IS 100.
--NOTE--MAXIMUM VALUE OF M (NUMBER OF CASES) IS UNLIMITED.
        THE NUMBER OF DATA MARK SYMBOLS, HOWEVER, IS
        LIMITED TO SIX.  THEREFORE, IF DATA MARKS ARE USED,
        ONLY THE DATA POINTS FOR THE FIRST SIX CASES WILL
        BE LABELED WITH THESE MARKS.
```

Figure 1. Generalized GECAP deck set-up.

6. Check to see that all 15 sense-switches above the console typewriter are in the OFF position.

7. Press the START button on card reader.

8. If a Cold Start Card is being used, press the PROGRAM STOP button, press the RESET button, and then press PROGRAM LOAD button on the console typewriter.

9. If a JOB card is being used, press the green START button on the console typewriter.

10. Wait for user message (1) to appear on the console typewriter. Press the START button on the console typewriter after this message has been written on the typewriter.

11. Let the program run. (Error messages may appear on the console typewriter indicating input errors).

SECTION III. EXAMPLE CASES

Five example cases are shown in Figures 2 through 16 in an effort to illustrate the options and combinations of options available in GECAP. Each example includes the following information:

1. A listing of the input deck.

2. A listing of the printed output.

3. A copy of the output plot.

Example Case 1 illustrates the linear option with data points marked. The x-axis is labeled with real numbers and the y-axis is labeled with integer numbers. Each of the plotted coordinates are connected with a straight line. Note that the output listing validates the value of the input variables.

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
// JOB    OR    //***COLD START CARD***//
// XEQ GECAP     1
*LOCALGECAP,LABEL,HIST,ERRO
CASE 1 OF 1
 1.        5.          10.           100.         0.          0.         CDL
TEST CASE 1 X-COORDINATE   TEST CASE 1 Y-COORDINATE   LINEAR OPTION
 0.         0.
 1.         10.
 2.         30.
 3.         5.
 4.         13.87
 5.         81.
 6.         100.
 7.         54.
 8.         49.
 9.         49.
 10.        0.0
                  END
```

Figure 2. Listing of input deck for Example Case 1.

Example Case 2 illustrates the discrete option. Both axes are labeled with integer numbers.

Example Case 3 illustrates the histogram option. Note that the maximum x value for the case is specified as 20 on the Program Instruction Card while the maximum x value in the input data is only 19. When the H option is being used, the value for XMAX on the Program Instruction Card should always be at least one x increment larger than the maximum value of the input data.

Example Case 4 illustrates the continuous option with no data points marked. Also illustrated here is the use of the exponential option to scale the axis values. Note that the x and y values of the input data are not scaled down but are within the specified limits of the program instruction card.

Example Case 5 illustrates the continuous option with three cases plotted on the same graph. The plot includes symobls identifying each of the three corresponding curves. Notice that in the printed output, a user's message appears which indicates that the polynomial calculated may not be accurate. After inspection of the curve on the output plot, it is seen that the curve is acceptable for most practical purposes. A printout of the polynomial coefficients is given for each curve along with the square of the errors from each data point.

```
X-AXIS INCREMENT =        1.00       Y-AXIS INCREMENT =        5.00
X-AXIS LIMIT =           10.00       Y-AXIS LIMIT =          100.00
X-AXIS ORIGIN =           0.00       Y-AXIS ORIGIN =           0.00
NCASES          =           1
TYPE( 1)   =          C
TYPE( 2)   =          D
TYPE( 3)   =          L


                   ***INPUT DATA***
TEST CASE 1 X-COORDINATE        TEST CASE 1 Y-COORDINATE


                       CASE 1 OF 1
            0.0000                        0.0000
            1.0000                       10.0000
            2.0000                       30.0000
            3.0000                        5.0000
            4.0000                       13.8700
            5.0000                       81.0000
            6.0000                      100.0000
            7.0000                       54.0000
            8.0000                       49.0000
            9.0000                       49.0000
           10.0000                        0.0000
```

Figure 3. Listing of printed output for Example Case 1.

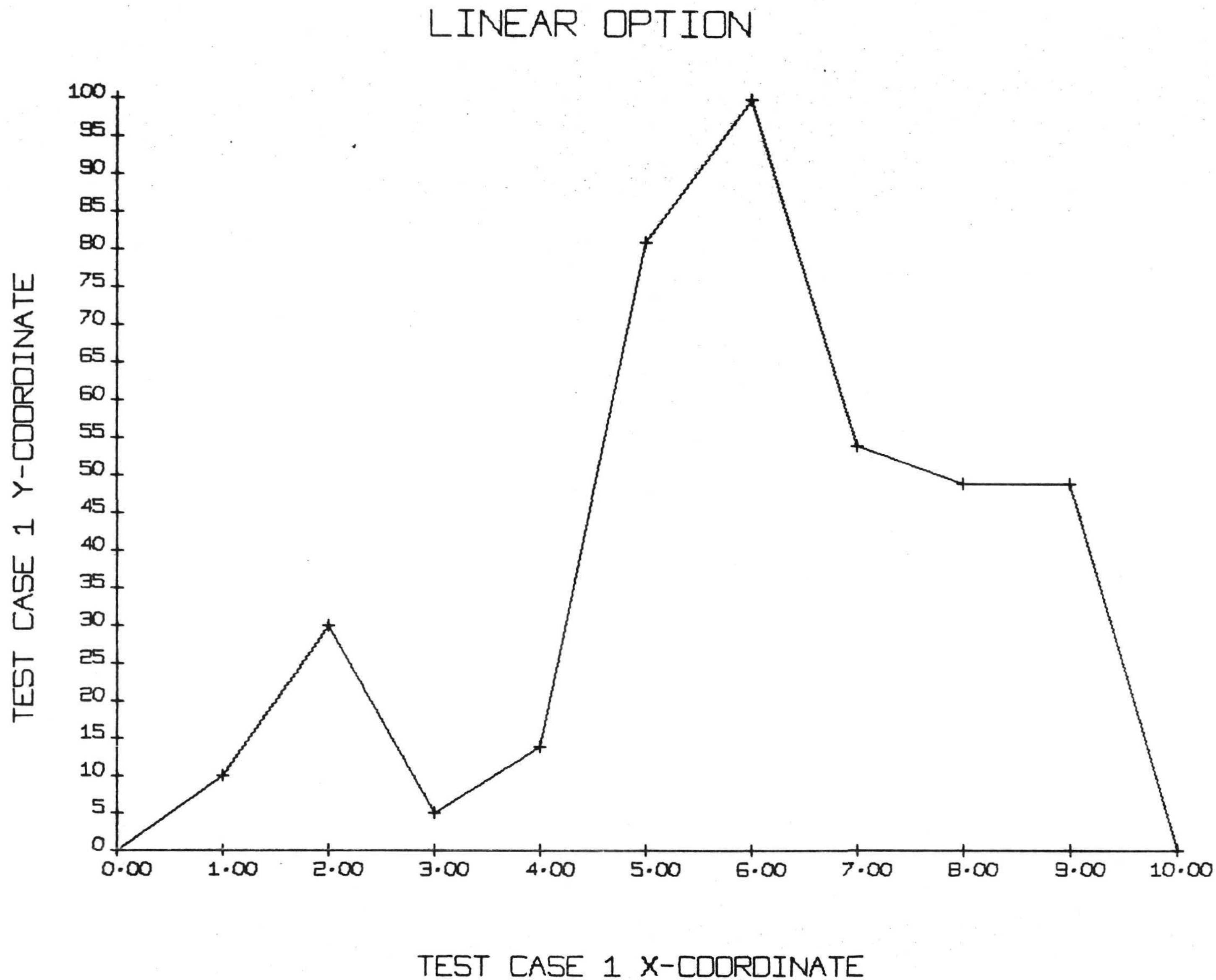

Figure 4. Plotted output for Example Case 1.

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
// JOB    OR     //***COLD START CARD***//
// XEQ GECAP      1
*LOCALGECAP,LABEL,HIST,ERRO
CASE 1 OF 1
 2.         10.         20.         200.        0.          50.       DDD
TEST CASE 2 X-COORDINATE  TEST CASE 2 Y-COORDINATE  DESCRETE OPTION
 2.         110.
 4.         120.
 6.         115.56
 8.         170.2
 9.5         120.
 10.        125.5
 12.        130.
 14.        105.
 16.        190.
 18.        200.
 20.        160.
                    END
```

Figure 5. Listing of input deck for Example Case 2.

```
X-AXIS INCREMENT =        2.00     Y-AXIS INCREMENT =        10.00
X-AXIS LIMIT =           20.00     Y-AXIS LIMIT =           200.00
X-AXIS ORIGIN =           0.00     Y-AXIS ORIGIN =           50.00
NCASES           =           1
TYPE( 1)   =           D
TYPE( 2)   =           D
TYPE( 3)   =           D


                 ***INPUT DATA***

TEST CASE 2 X-COORDINATE        TEST CASE 2 Y-COORDINATE


                   CASE 1 OF 1
            2.0000                    110.0000
            4.0000                    120.0000
            6.0000                    115.5600
            8.0000                    170.2000
            9.5000                    120.0000
           10.0000                    125.5000
           12.0000                    130.0000
           14.0000                    105.0000
           16.0000                    190.0000
           18.0000                    200.0000
           20.0000                    160.0000
```

Figure 6. Listing of printed output for Example Case 2.

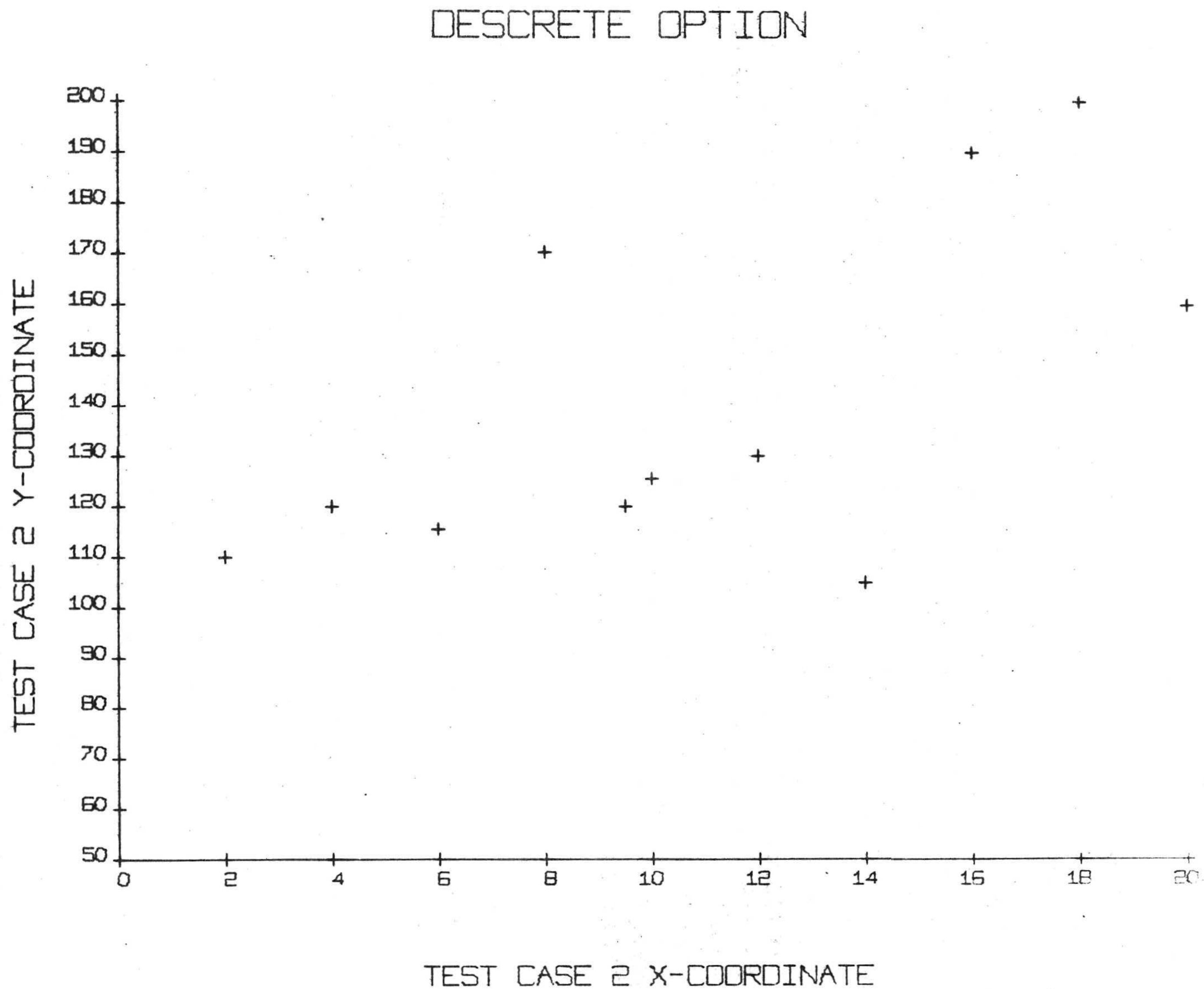

Figure 7. Plotted output for Example Case 2.

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
// JOB    OR    //***COLD START CARD***//
// XEQ GECAP      1
*LOCALGECAP,LABEL,HIST,ERRO
CASE 1 OF 1
 1.          20.        20.         120.        1.          20.       DCH
TEST CASE 3--CELL MEAN    TEST CASE 3--FREQUENCY    HISTOGRAM OPTION
 1.           25.
 2.           39.
 3.           50.
4.           55.
 5.          58.
 6.           60.
 7.          75.
8.            95.
 9.          103.98
 10.         115.24
 11.         116.5
12.          101.
 13.         92.
 14.         84.
15.          60.
16.          42.25
 17.         36.
 18.         30.
 19.         22.04
                        END
```

Figure 8. Listing of input deck for Example Case 3.

```
X-AXIS INCREMENT =        1.00     Y-AXIS INCREMENT =        20.00
X-AXIS LIMIT =           20.00     Y-AXIS LIMIT =           120.00
X-AXIS ORIGIN =           1.00     Y-AXIS ORIGIN =           20.00
NCASES          =            1
TYPE( 1)   =           D
TYPE( 2)   =           C
TYPE( 3)   =           H


                    ***INPUT DATA***
TEST CASE 3--CELL MEAN              TEST CASE 3--FREQUENCY


                            CASE 1 OF 1
                1.0000                         25.0000
                2.0000                         39.0000
                3.0000                         50.0000
                4.0000                         55.0000
                5.0000                         58.0000
                6.0000                         60.0000
                7.0000                         75.0000
                8.0000                         95.0000
                9.0000                        103.9800
               10.0000                        115.2400
               11.0000                        116.5000
               12.0000                        101.0000
               13.0000                         92.0000
               14.0000                         84.0000
               15.0000                         60.0000
               16.0000                         42.2500
               17.0000                         36.0000
               18.0000                         30.0000
               19.0000                         22.0400
```

Figure 9. Listing of printed output for Example Case 3.

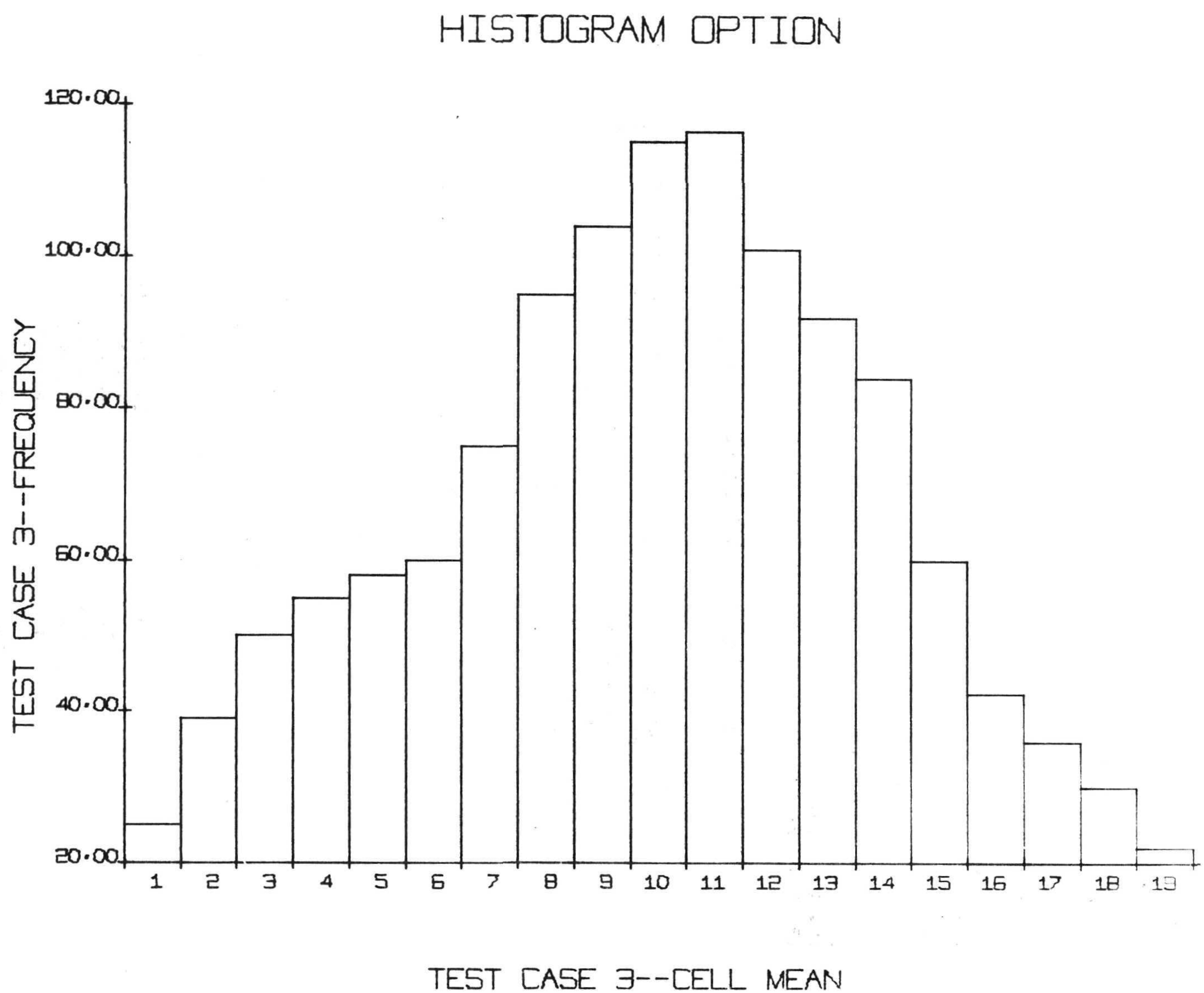

Figure 10. Plotted output for Example Case 3.

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
// JOB    OR    //***COLD START CARD***//
// XEQ GECAP      1
*LOCALGECAP,LABEL,HIST,ERRO
CASE 1 OF 1                                            -1     -1
 5.         5.          40.         10.          0.          -10.      CCC*
TEST CASE 4 X-COORDINATE  TEST CASE 4 Y-COORDINATE  CONT. OPTION, NO DATA MARKS
 0.         10.
 1.05       9.95
 1.92       9.82
 2.79       9.61
 3.67       9.34
 4.54       8.99
 5.41       8.57
 6.28       8.09
 7.16       7.55
 8.03       6.95
 8.90       6.29
 9.77       5.59
 10.65      4.85
 11.52      4.07
 12.39      3.26
 13.26      2.42
 14.14       1.56
 15.01       0.70
 15.71      0.0
 16.76      -1.05
 17.63      -1.91
 18.50      -2.76
 19.38      -3.58
 20.25      -4.38
 21.12      -5.15
 21.99      -5.88
 22.86      -6.56
 23.73      -7.19
 24.60      -7.77
 25.47      -8.29
 26.34      -8.75
 27.21      -9.14
 28.08      -9.46
 28.85      -9.70
 29.72      -9.88
 30.59      -9.98
 31.46      -10.00
                   END
```

Figure 11. Listing of input deck for Example Case 4.

```
X-AXIS INCREMENT =       5.00     Y-AXIS INCREMENT =      5.00
X-AXIS LIMIT =          40.00     Y-AXIS LIMIT =         10.00
X-AXIS ORIGIN =          0.00     Y-AXIS ORIGIN =       -10.00
NCASES         =           1
TYPE( 1)  =       C
TYPE( 2)  =       C
TYPE( 3)  =       C


             ***INPUT DATA***

TEST CASE 4 X-COORDINATE       TEST CASE 4 Y-COORDINATE


                   CASE 1 OF 1
        0.0000                    10.0000
        1.0500                     9.9500
        1.9200                     9.8200
        2.7900                     9.6100
        3.6700                     9.3400
        4.5400                     8.9900
        5.4100                     8.5700
        6.2800                     8.0900
        7.1600                     7.5500
        8.0300                     6.9500
        8.9000                     6.2900
        9.7700                     5.5900
       10.6500                     4.8500
       11.5200                     4.0700
       12.3900                     3.2600
       13.2600                     2.4200
       14.1400                     1.5600
       15.0100                     0.7000
       15.7100                     0.0000
       16.7600                    -1.0500
       17.6300                    -1.9100
       18.5000                    -2.7600
       19.3800                    -3.5800
       20.2500                    -4.3800
       21.1200                    -5.1500
       21.9900                    -5.8800
       22.8600                    -6.5600
       23.7300                    -7.1900
       24.6000                    -7.7700
       25.4700                    -8.2900
       26.3400                    -8.7500
       27.2100                    -9.1400
       28.0800                    -9.4600
       28.8500                    -9.7000
       29.7200                    -9.8800
       30.5900                    -9.9800
       31.4600                   -10.0000
```

Figure 12. Listing of printed output for Example Case 4.

```
*CO-EFFICIENTS FOR POLYNOMIAL OF DEGREE  7*
    10.000709097832
     0.003862024605
    -0.051973071545
     0.000391693861
     0.000002719892
     0.000002066409
    -0.000000070663
     0.000000000697
```

```
*CALCULATED VALUES AND SUM OF ERRORS SQUARED FOR 7 DEGREE POLYNOMIAL*
FOR POINT  1  YCALC =      10.000709      DIFF**2 =      0.000000502803
FOR POINT  2  YCALC =       9.947923      DIFF**2 =      0.000004313210
FOR POINT  3  YCALC =       9.819390      DIFF**2 =      0.000000371601
FOR POINT  4  YCALC =       9.615908      DIFF**2 =      0.000034915439
FOR POINT  5  YCALC =       9.335927      DIFF**2 =      0.000016588005
FOR POINT  6  YCALC =       8.988198      DIFF**2 =      0.000003247140
FOR POINT  7  YCALC =       8.572699      DIFF**2 =      0.000007289361
FOR POINT  8  YCALC =       8.092588      DIFF**2 =      0.000006702353
FOR POINT  9  YCALC =       7.544891      DIFF**2 =      0.000026091918
FOR POINT 10  YCALC =       6.946120      DIFF**2 =      0.000015053024
FOR POINT 11  YCALC =       6.294849      DIFF**2 =      0.000023513143
FOR POINT 12  YCALC =       5.595945      DIFF**2 =      0.000035346223
FOR POINT 13  YCALC =       4.845912      DIFF**2 =      0.000016704003
FOR POINT 14  YCALC =       4.067449      DIFF**2 =      0.000006506254
FOR POINT 15  YCALC =       3.258165      DIFF**2 =      0.000003365522
FOR POINT 16  YCALC =       2.424215      DIFF**2 =      0.000017769932
FOR POINT 17  YCALC =       1.562073      DIFF**2 =      0.000004300971
FOR POINT 18  YCALC =       0.697891      DIFF**2 =      0.000004445361
FOR POINT 19  YCALC =      -0.001492      DIFF**2 =      0.000002226602
FOR POINT 20  YCALC =      -1.049323      DIFF**2 =      0.000000457832
FOR POINT 21  YCALC =      -1.909124      DIFF**2 =      0.000000767067
FOR POINT 22  YCALC =      -2.754392      DIFF**2 =      0.000031449202
FOR POINT 23  YCALC =      -3.588137      DIFF**2 =      0.000066218449
FOR POINT 24  YCALC =      -4.385229      DIFF**2 =      0.000027344544
FOR POINT 25  YCALC =      -5.149389      DIFF**2 =      0.000000372938
FOR POINT 26  YCALC =      -5.875004      DIFF**2 =      0.000024950876
FOR POINT 27  YCALC =      -6.556761      DIFF**2 =      0.000010490893
FOR POINT 28  YCALC =      -7.189656      DIFF**2 =      0.000000118064
FOR POINT 29  YCALC =      -7.769007      DIFF**2 =      0.000000985816
FOR POINT 30  YCALC =      -8.290437      DIFF**2 =      0.000000191150
FOR POINT 31  YCALC =      -8.749856      DIFF**2 =      0.000000020600
FOR POINT 32  YCALC =      -9.143423      DIFF**2 =      0.000011718138
FOR POINT 33  YCALC =      -9.467486      DIFF**2 =      0.000056053369
FOR POINT 34  YCALC =      -9.693488      DIFF**2 =      0.000042395776
FOR POINT 35  YCALC =      -9.876957      DIFF**2 =      0.000009259824
FOR POINT 36  YCALC =      -9.980715      DIFF**2 =      0.000000511452
FOR POINT 37  YCALC =     -10.000982      DIFF**2 =      0.000000964853

SUM OF SQUARES OF ERRORS FOR CURVE-FIT =      0.000513523724
```

Figure 12. (Concluded)

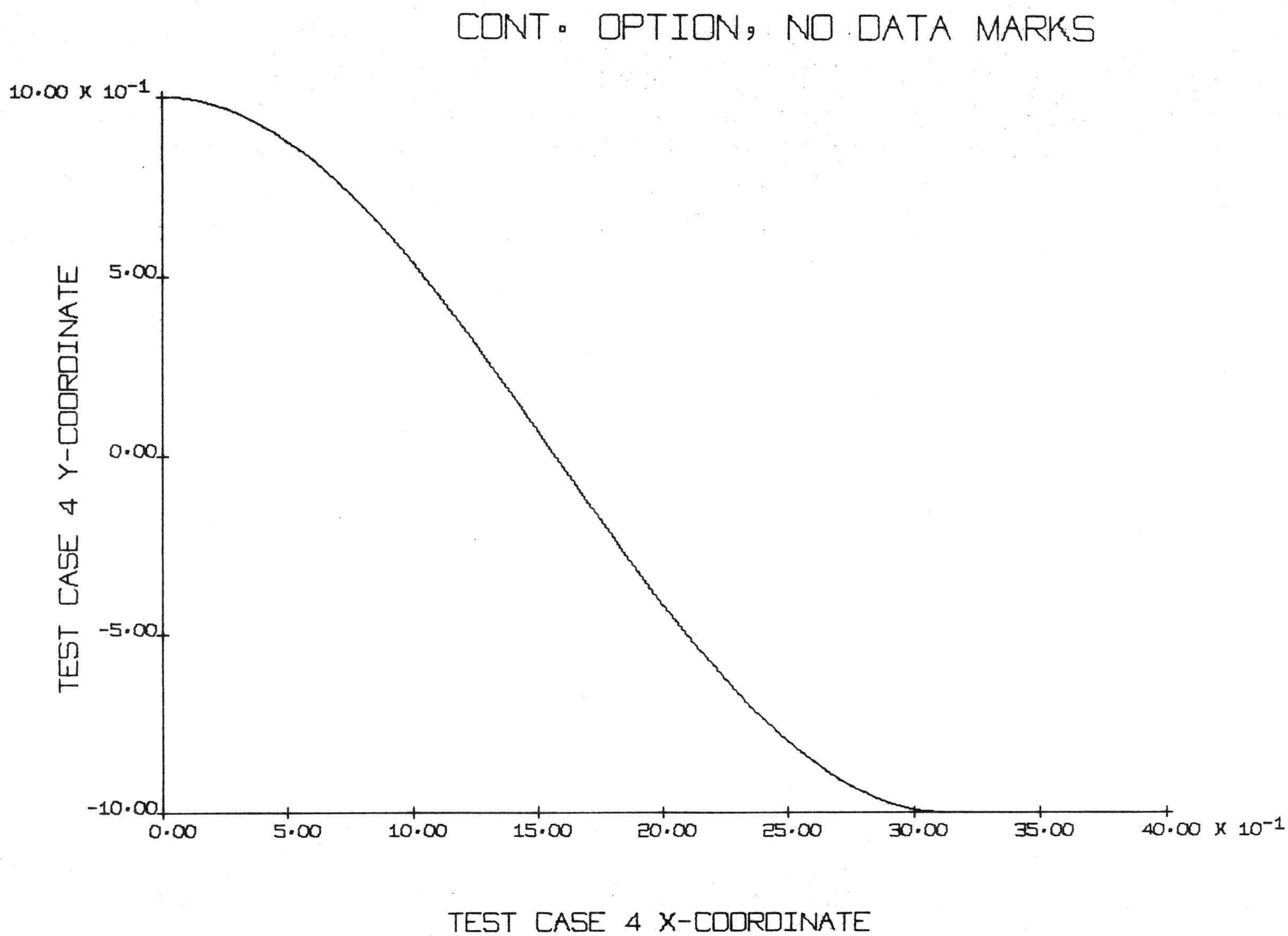

Figure 13. Plotted output for Eaxmple Case 4.

```
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
// JOB    OR    //***COLD START CARD***//
// XEQ GECAP     1
*LOCALGECAP,LABEL,HIST,ERRO
CASE 1 OF 3    EXAMPLE CURVE 1
 .1         .10001     .8         1.1        0.0        0.0      CCC
TEST CASE 5 X-COORDINATE  TEST CASE 5 Y-COORDINATE  MULTIPLE CONTINUOUS OPTION
    0.04000    0.00164
    0.06000    0.00883
    0.08000    0.02758
    0.10000    0.06336
    0.12000    0.11892
    0.14000    0.19338
    0.16000    0.28254
    0.18000    0.38022
    0.20000    0.47972
    0.22000    0.57508
    0.24000    0.66188
    0.26000    0.73748
    0.28000    0.80085
    0.30000    0.85216
    0.32000    0.89247
    0.34000    0.92331
    0.36000    0.94630
    0.38000    0.96305
    0.40000    0.97501
    0.42000    0.98337
    0.44000    0.98913
    0.46000    0.99301
    0.48000    0.99560
                        END
CASE 2 OF 3    EXAMPLE CURVE 2
    0.12000    0.01022
    0.14000    0.02879
    0.16000    0.05832
    0.18000    0.10040
    0.20000    0.15527
    0.22000    0.22173
    0.24000    0.29730
    0.26000    0.37858
    0.28000    0.46193
    0.30000    0.54363
    0.32000    0.62076
    0.34000    0.69098
    0.36000    0.75305
    0.38000    0.80622
    0.40000    0.85065
    0.42000    0.88680
    0.44000    0.91556
    0.46000    0.93806
    0.48000    0.95526
    0.50000    0.96817
    0.52000    0.97769
    0.54000    0.98463
    0.56000    0.98959
    0.58000    0.99311
    0.60000    0.99555
                        END
```

Figure 14. Listing of input deck for Example Case 5.

CASE 3 OF 3 EXAMPLE CURVE 3

0.22000	0.01197
0.24000	0.03007
0.26000	0.05566
0.28000	0.08968
0.30000	0.13234
0.32000	0.18334
0.34000	0.24188
0.36000	0.30617
0.38000	0.37426
0.40000	0.44415
0.42000	0.51360
0.44000	0.58064
0.46000	0.64370
0.48000	0.70159
0.50000	0.75377
0.52000	0.79961
0.54000	0.83909
0.56000	0.87246
0.58000	0.90018
0.60000	0.92280
0.62000	0.94100
0.64000	0.95543
0.66000	0.96671
0.68000	0.97540
0.70000	0.98202
0.72000	0.98698
0.74000	0.99067
0.76000	0.99338
0.78000	0.99535
0.80000	0.99677

END

Figure 14. (Concluded).

X-AXIS INCREMENT =	0.10	Y-AXIS INCREMENT =	0.10
X-AXIS LIMIT =	0.80	Y-AXIS LIMIT =	1.10
X-AXIS ORIGIN =	0.00	Y-AXIS ORIGIN =	0.00
NCASES =	3		
TYPE(1) =	C		
TYPE(2) =	C		
TYPE(3) =	C		

INPUT DATA

TEST CASE 5 X-COORDINATE	TEST CASE 5 Y-COORDINATE
CASE 1 OF 3	
0.0400	0.0016
0.0600	0.0088
0.0800	0.0275
0.1000	0.0633
0.1200	0.1189
0.1400	0.1933
0.1600	0.2825
0.1800	0.3802
0.2000	0.4797
0.2200	0.5750
0.2400	0.6618
0.2600	0.7374
0.2800	0.8008
0.3000	0.8521
0.3200	0.8924
0.3400	0.9233
0.3600	0.9463
0.3800	0.9630
0.4000	0.9750
0.4200	0.9833
0.4400	0.9891
0.4600	0.9930
0.4800	0.9956

Figure 15. Listing of printed output for Example Case 5.

```
*CO-EFFICIENTS FOR POLYNOMIAL OF DEGREE  9*
     -0.039928562066
      2.680569725111
    -65.717747807502
    718.222608039447
  -2499.070896148681
   1838.338735103607
   6513.944360733032
 -10910.395404815673
  -2377.100768089294
   9138.639202117919
```

```
*CALCULATED VALUES AND SUM OF ERRORS SQUARED FOR 9 DEGREE POLYNOMIAL*
FOR POINT  1  YCALC =        0.001927      DIFF**2 =     0.000000082703
FOR POINT  2  YCALC =        0.008772      DIFF**2 =     0.000000003316
FOR POINT  3  YCALC =        0.026791      DIFF**2 =     0.000000621963
FOR POINT  4  YCALC =        0.063058      DIFF**2 =     0.000000091133
FOR POINT  5  YCALC =        0.119515      DIFF**2 =     0.000000354491
FOR POINT  6  YCALC =        0.194297      DIFF**2 =     0.000000841481
FOR POINT  7  YCALC =        0.283006      DIFF**2 =     0.000000217405
FOR POINT  8  YCALC =        0.379886      DIFF**2 =     0.000000111237
FOR POINT  9  YCALC =        0.478844      DIFF**2 =     0.000000766828
FOR POINT 10  YCALC =        0.574275      DIFF**2 =     0.000000647926
FOR POINT 11  YCALC =        0.661666      DIFF**2 =     0.000000045699
FOR POINT 12  YCALC =        0.737961      DIFF**2 =     0.000000231697
FOR POINT 13  YCALC =        0.801681      DIFF**2 =     0.000000691816
FOR POINT 14  YCALC =        0.852820      DIFF**2 =     0.000000436204
FOR POINT 15  YCALC =        0.892541      DIFF**2 =     0.000000005145
FOR POINT 16  YCALC =        0.922735      DIFF**2 =     0.000000330597
FOR POINT 17  YCALC =        0.945499      DIFF**2 =     0.000000640990
FOR POINT 18  YCALC =        0.962653      DIFF**2 =     0.000000157419
FOR POINT 19  YCALC =        0.975387      DIFF**2 =     0.000000142702
FOR POINT 20  YCALC =        0.984216      DIFF**2 =     0.000000716604
FOR POINT 21  YCALC =        0.989394      DIFF**2 =     0.000000069805
FOR POINT 22  YCALC =        0.992002      DIFF**2 =     0.000001015472
FOR POINT 23  YCALC =        0.995954      DIFF**2 =     0.000000125563

SUM OF SQUARES OF ERRORS FOR CURVE-FIT =     0.000008348206
```

Figure 15. (Continued).

INPUT DATA

TEST CASE 5 X-COORDINATE	TEST CASE 5 Y-COORDINATE
CASE 2 OF 3	
0.1200	0.0102
0.1400	0.0287
0.1600	0.0583
0.1800	0.1004
0.2000	0.1552
0.2200	0.2217
0.2400	0.2973
0.2600	0.3785
0.2800	0.4619
0.3000	0.5436
0.3200	0.6207
0.3400	0.6909
0.3600	0.7530
0.3800	0.8062
0.4000	0.8506
0.4200	0.8868
0.4400	0.9155
0.4600	0.9380
0.4800	0.9552
0.5000	0.9681
0.5200	0.9776
0.5400	0.9846
0.5600	0.9895
0.5800	0.9931
0.6000	0.9955

USERS MESSAGE THE POLYNOMINAL CALCULATED MAY OR MAY NOT BE THE BEST FITTED CURVE

Figure 15. (Continued).

```
*CO-EFFICIENTS FOR POLYNOMIAL OF DEGREE  9*
      0.733166191959
    -17.548953562974
    157.079668283462
   -618.872270822525
    296.712045907974
   9378.129894256591
 -44079.530563354492
  90706.581665039062
 -90827.897857666015
  36035.444580078125
```

```
*CALCULATED VALUES AND SUM OF ERRORS SQUARED FOR 9 DEGREE POLYNOMIAL*
FOR POINT  1   YCALC =          0.011873       DIFF**2 =       0.000002733528
FOR POINT  2   YCALC =          0.026310       DIFF**2 =       0.000006146741
FOR POINT  3   YCALC =          0.056916       DIFF**2 =       0.000001969616
FOR POINT  4   YCALC =          0.101147       DIFF**2 =       0.000000558885
FOR POINT  5   YCALC =          0.157209       DIFF**2 =       0.000003763071
FOR POINT  6   YCALC =          0.223401       DIFF**2 =       0.000002793994
FOR POINT  7   YCALC =          0.297697       DIFF**2 =       0.000000157887
FOR POINT  8   YCALC =          0.377576       DIFF**2 =       0.000001006416
FOR POINT  9   YCALC =          0.460065       DIFF**2 =       0.000003477294
FOR POINT 10   YCALC =          0.541940       DIFF**2 =       0.000002853150
FOR POINT 11   YCALC =          0.620046       DIFF**2 =       0.000000509448
FOR POINT 12   YCALC =          0.691631       DIFF**2 =       0.000000425034
FOR POINT 13   YCALC =          0.754671       DIFF**2 =       0.000002630334
FOR POINT 14   YCALC =          0.808083       DIFF**2 =       0.000003472011
FOR POINT 15   YCALC =          0.851795       DIFF**2 =       0.000001312294
FOR POINT 16   YCALC =          0.886659       DIFF**2 =       0.000000019819
FOR POINT 17   YCALC =          0.914176       DIFF**2 =       0.000001914500
FOR POINT 18   YCALC =          0.936084       DIFF**2 =       0.000003901386
FOR POINT 19   YCALC =          0.953881       DIFF**2 =       0.000001901406
FOR POINT 20   YCALC =          0.968385       DIFF**2 =       0.000000046550
FOR POINT 21   YCALC =          0.979562       DIFF**2 =       0.000003508122
FOR POINT 22   YCALC =          0.986698       DIFF**2 =       0.000004279220
FOR POINT 23   YCALC =          0.989476       DIFF**2 =       0.000000012949
FOR POINT 24   YCALC =          0.990075       DIFF**2 =       0.000009207652
FOR POINT 25   YCALC =          0.996855       DIFF**2 =       0.000001704946

SUM OF SQUARES OF ERRORS FOR CURVE-FIT =       0.000060306266
```

Figure 15. (Continued).

INPUT DATA

TEST CASE 5 X-COORDINATE	TEST CASE 5 Y-COORDINATE
CASE 3 OF 3	
0.2200	0.0119
0.2400	0.0300
0.2600	0.0556
0.2800	0.0896
0.3000	0.1323
0.3200	0.1833
0.3400	0.2418
0.3600	0.3061
0.3800	0.3742
0.4000	0.4441
0.4200	0.5136
0.4400	0.5806
0.4600	0.6437
0.4800	0.7015
0.5000	0.7537
0.5200	0.7996
0.5400	0.8390
0.5600	0.8724
0.5800	0.9001
0.6000	0.9228
0.6200	0.9410
0.6400	0.9554
0.6600	0.9667
0.6800	0.9754
0.7000	0.9820
0.7200	0.9869
0.7400	0.9906
0.7600	0.9933
0.7800	0.9953
0.8000	0.9967

Figure 15. (Continued).

```
*CO-EFFICIENTS FOR POLYNOMIAL OF DEGREE  9*
    -0.848485231399
    20.262203224003
  -197.173971712589
  1059.880441665649
 -3750.140005111694
  9700.356727600097
-17520.517059326171
 20004.520645141601
-12707.314575195312
  3398.905395507812
```

```
*CALCULATED VALUES AND SUM OF ERRORS SQUARED FOR 9 DEGREE POLYNOMIAL*
FOR POINT  1  YCALC =        0.012745     DIFF**2 =     0.000000601850
FOR POINT  2  YCALC =        0.029356     DIFF**2 =     0.000000509098
FOR POINT  3  YCALC =        0.054756     DIFF**2 =     0.000000815690
FOR POINT  4  YCALC =        0.089296     DIFF**2 =     0.000000146885
FOR POINT  5  YCALC =        0.132728     DIFF**2 =     0.000000151046
FOR POINT  6  YCALC =        0.184288     DIFF**2 =     0.000000899720
FOR POINT  7  YCALC =        0.242783     DIFF**2 =     0.000000816801
FOR POINT  8  YCALC =        0.306682     DIFF**2 =     0.000000262453
FOR POINT  9  YCALC =        0.374215     DIFF**2 =     0.000000002024
FOR POINT 10  YCALC =        0.443483     DIFF**2 =     0.000000444748
FOR POINT 11  YCALC =        0.512575     DIFF**2 =     0.000001050407
FOR POINT 12  YCALC =        0.579674     DIFF**2 =     0.000000933148
FOR POINT 13  YCALC =        0.643161     DIFF**2 =     0.000000289499
FOR POINT 14  YCALC =        0.701712     DIFF**2 =     0.000000015045
FOR POINT 15  YCALC =        0.754354     DIFF**2 =     0.000000341927
FOR POINT 16  YCALC =        0.800516     DIFF**2 =     0.000000821663
FOR POINT 17  YCALC =        0.840033     DIFF**2 =     0.000000889492
FOR POINT 18  YCALC =        0.873110     DIFF**2 =     0.000000423290
FOR POINT 19  YCALC =        0.900291     DIFF**2 =     0.000000012349
FOR POINT 20  YCALC =        0.922329     DIFF**2 =     0.000000221075
FOR POINT 21  YCALC =        0.940093     DIFF**2 =     0.000000822142
FOR POINT 22  YCALC =        0.954434     DIFF**2 =     0.000000990398
FOR POINT 23  YCALC =        0.966066     DIFF**2 =     0.000000414041
FOR POINT 24  YCALC =        0.975456     DIFF**2 =     0.000000003223
FOR POINT 25  YCALC =        0.982819     DIFF**2 =     0.000000639291
FOR POINT 26  YCALC =        0.988129     DIFF**2 =     0.000001321069
FOR POINT 27  YCALC =        0.991351     DIFF**2 =     0.000000464910
FOR POINT 28  YCALC =        0.992795     DIFF**2 =     0.000000341957
FOR POINT 29  YCALC =        0.993799     DIFF**2 =     0.000002402732
FOR POINT 30  YCALC =        0.997637     DIFF**2 =     0.000000752988

SUM OF SQUARES OF ERRORS FOR CURVE-FIT =      0.000017800973
```

Figure 15. (Concluded).

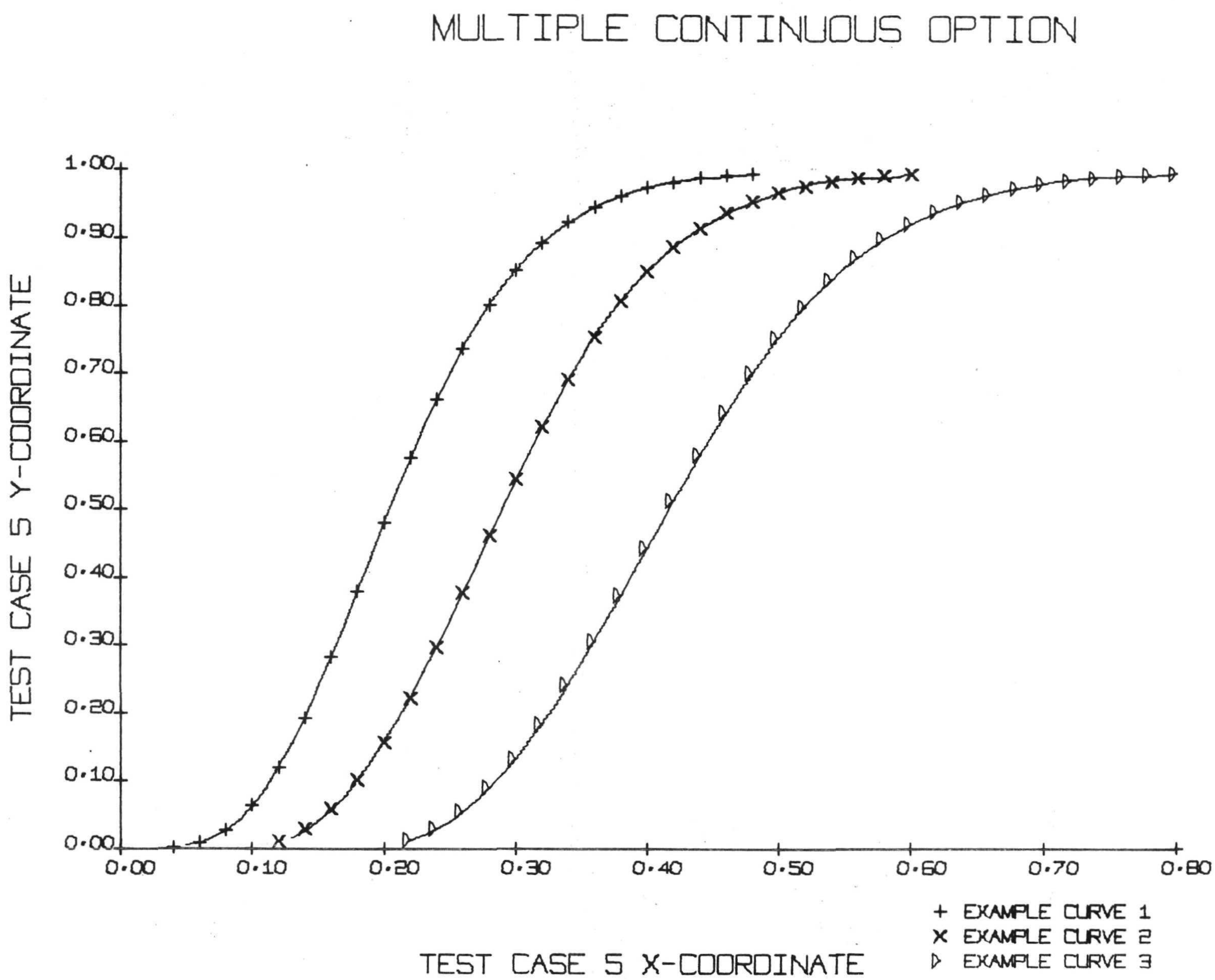

Figure 16. Plotted output for Example Case 5.

SECTION IV. PROGRAM MESSAGES

A. User Messages

The following two user-messages have been included in GECAP.

1.
```
**** POSITION PLOTTER PEN APPROXIMATELY 3 INCHES FROM THE ****
**** RIGHT EDGE OF THE PLOTTER.     . . . PRESS START . . .   ****
```

This message appears on the console typewriter at the beginning of each execution of the program. Its purpose is to instruct the user to make sure the plotter pen is in a proper position relative to the plotting paper. To continue execution the green PROGRAM START button on the console typewriter must be pressed. (The plot origin is established at this position.)

2.
```
***USERS MESSAGE*** THE POLYNOMINAL CALCULATED MAY
OR MAY NOT BE THE BEST FITTED CURVE
```

This message occurs when a statistical test in BESFT has not been satisfied. Execution will continue and the user may determine if the calculated polynomial is acceptable for his purposes. (See the section on program limitations for more information.)

B. Error Messages

GECAP was designed such that input to the program was made as simple as possible. However, various restrictions within the program logic may result in erroneous output if invalid input is used. In order to aid the GECAP user in detecting these errors, various messages are printed on the console typewriter, which indicate some of the common errors that occur from incorrect program instructions or input data. The following list describes these messages and gives the course of action taken by GECAP during their occurrence.

1.
```
****ERROR****  CASE CARD INPUT VARIABLE IS INCORRECT
*NUMBER OF CASES HAS BEEN SET EQUAL TO 1 * EXECUTION RESUMED*
```

This message indicates that an error has been made in the format of the CASE card. Execution will continue, setting the number of plots equal to 1.

2. ````
****ERROR****END POINT OF X AXIS IS LESS THAN STARTING POINT****
          ****EXECUTION DISCONTINUED****
````

This message indicates that an error has been made when specifying the limits for the x axis. Execution will be discontinued at this point.

3. ````
****ERROR****END POINT OF Y AXIS IS LESS THAN STARTING POINT****
          ****EXECUTION DISCONTINUED****
````

This message indicates the same thing as message 4 except for the y axis. Execution will be discontinued.

4. ````
ERROR   NUMBER OF TIC MARKS EXCEEDS LEGIBILITY LIMIT***
STANDARD FIX-UP TAKEN, EXECUTION CONTINUING
````

This message indicates that the values specified for labeling the axes calls for more than 20 'tic' marks or labeling intervals. The program will internally readjust the increments so that 10 'tic' marks are placed on the axes and scale values are recalculated. Execution will continue. The user may increase the increment value XINC or YINC to eliminate this error.

5. ````
**** ERROR. . MORE THAN 100 DATA POINTS ON INPUT ****
 ****  EXECUTION CONTINUING WITHOUT REMAINING POINTS**
````

A limit of 100 data points has been established for this program. If this restriction is violated, the program will discard the extra data points and continue execution with the first 100 values. This message will be printed out for each input data card over 100.

6. ````
  ****ERROR****THE VALUES SPECIFIED FOR LABELING THE AXES
REQUIRE MORE PRECISION THAN THAT ALLOWABLE IN GECAP..........
...CHECK GECAP USER MANUAL FOR DETAILS...
THIS ERROR MAY BE INSIGNIFICANT IN SPECIAL CASES...IF EXECUTION
IS STILL DESIRED PRESS START...IF NOT, PRESS STOP
````
````

This message indicates that more than two decimal places were requested on the program instruction card. In some cases this error will cause invalid results. If execution is desired, the green START button on the console typewriter must be pressed. Six different input parameters (XINC, YINC, XMAX, YMAX, XSTRT, YSTRT) are checked by the program and any one of these values could cause this message to appear. The START button must be pressed each time the message appears, and, only after all six values have been checked, will execution of GECAP be resumed.

7.
```
****ERROR****A DATA POINT WAS FOUND TO EXCEED THE LIMITS OF THE AXES...****
     ****EXECUTION DISCONTINUED****
```

This message indicates that one of the input coordinates does not fall within the user defined limits of the coordinate axes. Execution will be discontinued.

8.
```
  ****ERROR****INCORRECT PLOT OPTION WAS USED FOR
   VARIABLE TYPE(1).****CHECK GECAP USER MANUAL FOR DETAILS.....
****TYPE IN THE DESIRED OPTION FROM THE CONSOLE TYPEWRITER
(C,D,H, OR L).  ****PRESS (EOF) BUTTON TO CONTINUE EXECUTION*****
```

9.
```
  ****ERROR****INCORRECT PLOT OPTION WAS USED FOR
   VARIABLE TYPE(2).****CHECK GECAP USER MANUAL FOR DETAILS.....
****TYPE IN THE DESIRED OPTION FROM THE CONSOLE TYPEWRITER
(C,D,H, OR L).  ****PRESS (EOF) BUTTON TO CONTINUE EXECUTION*****
```

10.
```
  ****ERROR****INCORRECT PLOT OPTION WAS USED FOR
   VARIABLE TYPE(3).****CHECK GECAP USER MANUAL FOR DETAILS.....
****TYPE IN THE DESIRED OPTION FROM THE CONSOLE TYPEWRITER
(C,D,H, OR L).  ****PRESS (EOF) BUTTON TO CONTINUE EXECUTION*****
```

Messages 8, 9, and 10 appear when incorrect symbol options were placed in columns 61, 62, or 63 of the Program Instruction Card. When this message appears, the correct symbol should be typed in from the console typewriter. To resume execution, the EOF button on the typewriter should be pressed.

11. ***ERROR*** ERROR IN ROUTINE CURF. EXECUTION CONTINUING

This error message occurs in routine CURF and indicates that an error resulted while calculating the coefficients for the least squares polynominal. Execution will continue.

SECTION V. PROGRAM DESCRIPTION

A. General

GECAP was designed exclusively for the IBM 1130 computing system. This system was chosen because it is common among industrial and governmental facilities, and therefore, provides easy access to the user. Core requirements for this program are 8K words of storage.[5]

This program has been constructed to provide the user with a rapid and accurate method of plotting a set of data on a grid without having to write any programming instructions. The program reads in information concerning starting values for the coordinate axes, upper limits for the axes, and step increments between 'TIC' marks for the axes. These values are used to calculate scale values for the grid. The scale values calculated by the program are based on the assumption that the plot is small enough to fit an 8.5 × 11 in. sheet of paper. User titles are read in for the x and y axes and also for the grid, and placed at the proper location on the graph.

Several different types of plots are available. The user selects the type of plot, the program generates the requested graph for the given data set. The supplied information and the input data is written on the printer for user reference.

Numerous options concerning the type of plot and the method of labeling the axes are available to the user.

Labeling the axes:

1. Either or both of the coordinate axes may be labeled with integer or real numbers.

---

5. The program is actually larger than 8K but by the use of the LOCAL and LINK system overlay routines, the program is within the 8K limitation of the 1130 computer system.

2. The user may wish to indicate that his labeled values on the axes are scaled to a power of 10. If so, an option is available which will label the last 'TIC' mark on each axis in such a manner as to indicate this scaling factor (ex: 20×10^{-1}).

Type of plot:

1. The data may be plotted as individual points marked by symbols on the grid.

2. The data may be plotted as individual points with straight lines connecting each point to the previous point.

3. A histogram option is available which generates a histogram from the user-supplied frequency information.

4. An option may be used which plots the points individually and then performs a least-squares curve fit on the supplied data. The program uses a statistical test[6] to determine which degree polynomial best fits the data and then plots the curve. The coefficients of the calculated polynomial along with the errors associated with the differences from the calculated curve and the actual data are displayed on the printout.

5. The user may not desire data point symbols on his graph. If so, an option is available, applicable to any of the above options, which causes no data point marks to be placed on the grid.

6. Multiple cases may be plotted on the same grid for options 1, 2, 4, and 5.

7. If data point marks are used, one permits the data symbols to be labeled.

After each set of data is plotted on the grid, the plotter pen is moved to a point below the x axis. The symbol associated with that set of data is drawn and followed by a user-specified name for that particular data set. Only six curves may be labeled in this fashion as the IBM 1130 plotter routines are limited to six different symbols for data point marks.

6. See Appendix A.

B. Program Structure

GECAP is built primarily in two sections. The mainline, GECAP, and supporting routines LABL, ERRO, and HIST perform the operations concerned with drawing and labeling the grid, plotting the individual data points, and generating a plot for all options except the least-squares curve-fit option. The curve-fit option is contained in the second main routine, BESFT, which calculates the polynomial coefficients and plots the corresponding function on the grid. The program is built in this manner due to limited core requirements within the IBM 1130 computing system. The maximum amount of core storage available on this system is 8,000 words of core storage. Since GECAP and BESFT each require approximately 8,000 words of core storage, it is impossible to execute the program with both routines in core together. Execution is achieved through the call LINK system routine which, in effect, stores only one routine in core at a time. If the continuous option is used, GECAP will call BESFT. When this is done, BESFT is brought into core. BESFT executes its function and returns to GECAP. When GECAP returns to core only the values for the variables stored in common will be retained.

Routines: The following is a list of routines called by the mainline GECAP.

1. LABEL - This routine draws and labels the grid and places the titles on the graph.

2. HIST - This routine generates a histogram plot from the input data.

3. ERRO - This routine contains all GECAP error messages and is called only when an error is found in the input data.

4. BESFT - This routine determines the best degree of polynomial through a statistical ANOVA test and plots the least squares function on the grid.

5. F - This function calculates the value of the dependent variable from the polynomial equation found in the curve fit analysis.

6. CURF - This routine calculates the coefficients for all degree polynomials (up to the 11th degree) which fit the data.

7. FINSH - This routine checks to see that all plotting has been completed before further execution of the program proceeds. If the plot buffer is empty, then the program continues; if not, the program waits for all plotting to be completed before continuing.

NOTE: The routines LABEL, HISTO, and ERRO are placed in a system monitor control, LOCAL. This causes these routines to be placed in core only when they are called by the mainline GECAP. At all other times these routines are stored on the disc. The purpose of this monitor control is to prevent the amount of core storage used by GECAP from exceeding 8K words.

C. Program Restrictions and Limitations

1. As previously stated, use of a system routine (CALL LINK) and a system monitor control (LOCAL) have been made in order to limit the amount of core storage required by GECAP. The user should be cautious when making program modifications, for these changes could drive core requirement over the 1130 computer capacity.

2. When executing the continuous (C) or histogram (H) options, the data must be input in order of increasing x value ($x_i < x_i+1 < x_i+2 < x_i+3$, etc).

3. The statistical test in BESFT does not always guarantee that the best degree polynomial has been selected by the program. It is possible that the curve selected may be accurate for the given data points but not the best "eye balled" curve. This circumstance may be avoided by using more data points and by spacing these points evenly with respect to the x axis.

4. A maximum of 100 data points may be input for any case using any option.

5. The maximum degree equation which will be calculated for any case using the continuous option is an 11th degree polynomial. For cases where less than 11 data points are input, the maximum degree polynomial calculated will be (N-1) where N is the number of data points.

D. PROGRAM MODIFICATIONS FOR OTHER COMPUTING SYSTEMS

This program may be modified to run on other systems by making a few minor changes on large computers, rather than calling BESFT through a CALL LINK system routine. BESFT should be a subroutine called by GECAP and placing a GO TO statement after BESFT sending it to the beginning of GECAP. The system monitor, LOCAL, may be deleted. The plotter routines must be changed as these routines are unique to each computing system. Also, the input/output unit numbers for all READ and WRITE statements must be changed to correspond to the particular installation.

APPENDIX A

DETERMINATION OF THE BEST FIT POLYNOMIAL

One of the problems associated with plotting out the results of a least-squares curve-fit is the determination of the degree of polynomial that best fits the empirical data. One possible method is to calculate the least-squares coefficients for as many polynomial equations as one desires, and then plot out each of the different power curves. Selection of the plotted curves can be made against the input data by using the "eye ball" technique.[7] In many cases this is the best method since the analyst has some prior information about the shape of the curve, or because higher degree polynomial coefficients cannot be determined due to a lack of enough data points.

However, a routine has been included in GECAP that statistically determines the best degree of polynomial to be plotted. The test procedure is mathematically defined as follows:

Given a set of n data points x_1, y_1; x_2, y_2; x_3, y_3; + ... x_n, y_n; one can determine the least squares coefficients b_m, for an mth degree polynomial, assuming $m < n-1$. An equation of this type is defined in the following form:

$$y_m (x_i) = b_o + b_1 x_i + b_2 x_i^2 + \ldots b_m x_i^m$$

The problem is to determine what value of m will give a statistically good fit. (i.e., Does the addition of a higher order term to the polynomial equation fit the data significantly better than without using it?)

A test can be set up that will solve this problem on a probabilistic basis. The test uses the ratio of the difference of two independent estimates of the error variance against a single estimate of the error variance. Mathematically, this can be defined as the following F statistic:

$$F_{m-1} = \frac{S^2_{m-1} - S^2_m}{S^2_m}$$

7. The "eye ball" technique (as named by the author) is simply a method in which someone determines the shape of the approximate curve by placing a french curve over the given data and estimating what a good fit would look like.

where

$$S_m^2 = \frac{\sum_{i=1}^{n} (y_m(x_i) - y_i)^2}{n - (m + 1)}$$

is the residual variance after fitting the data with the mth degree polynomial and

$$S_{m-1}^2 = \frac{\sum_{i=1}^{n} (y_{m-1}(x_i) - y_i)^2}{n - (m)}$$

is the residual variance after fitting the data with the m-1th degree polynomial.

The null hypothesis $S_m^2 = 0$ (i.e., the test can be stated as the mth degree polynomial is not significantly better than the m-1th degree polynomial or the null hypothesis $b_m = 0$) is tested for significance at the 5-percent rejection level

where

$$F_{m-1} > F_{0.95,\ \nu_1,\ \nu_2}$$

$$\nu_1 = 1$$

$$\nu_2 = n-m-1$$

to determine if the mth degree curve has a smaller error variance than the m-1th polynomial. If the value of F is less than $F_{\alpha,\ \nu_1,\ \nu_2}$ then it is assumed that the mth degree is satisfactory. This test is also performed for the m-2th error variance. After two successive tests show no significant in the F values, then it is assumed that the mth degree polynomial is a good fit.[8]

It should be pointed out that this test will not always satisfy the user of GECAP in all possible cases. However, it works well when the data to be fitted is evenly distributed along the x axis and the data has less than two

8. The corresponding polynomial coefficients are printed out and the mth polynomial function is plotted on the graph.

inflection points. For data reflecting many (3 or more) inflection points, the desired polynomial fit should have at least 30 or more coordinate values per curve as input to the GECAP Program.

More information on this subject can be obtained from the books Probability and Statistics for Engineers by I. Miller and J. Freund, page 245, and/or Probability and Statistics in Engineering and Management Science by W. Hines and D. Montgomery, pages 332 and 356.

APPENDIX B

PROGRAM LISTING

The following program listing is included for those users wishing to modify the existing program to suit a special need. The listing has been well commented so that a programmer can isolate any individual logic operation within a routine. Information on the IBM 1130/1800 plotter subroutines can be obtained from the IBM System Reference Library, Form C26-3755-0.

```
// FOR
*IOCS(1132 PRINTER,PLOTTER,TYPEWRITER,CARD,KEYBOARD)
*ONE WORD INTEGERS
*EXTENDED PRECISION
C
C***  GECAP REVISION C, 09/05/72
C
C*********************************************************************
C
C*****PROGRAMMED BY                B. R. BEADLE II    (9/71)
C                                  B. D. DOLERHIE     (12/71)
C                                  J. W. OWEN         (9/72)
C                                  R. A. SCHLAGHECK   (9/71 TO 9/72)
C
C*********************************************************************
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C         PROGRAM GECAP IS A GENERALIZED CURVE--FIT AND PLOT PROGRAM
C         FOR USE IN PLOTTING A WIDE VARIETY OF DATA ON A GRID SMALL
C         ENOUGH TO FIT ON AN 8 1/2 X 11 SHEET OF PAPER.  NUMEROUS
C         OPTIONS ARE SUPPLIED TO THE USER, IN ORDER FOR HIM TO OBTAIN
C         A PLOT OF HIS DATA SET.
C
C         OPTIONS'
C
C              1 - EITHER OR BOTH OF THE AXIS ON THE GRID MAY BE NUMB-
C                  ERED IN EITHER INTEGER OR REAL NUMBER FORM.
C
C              2 - THE DATA MAY BE PLOTTED IN ANY ONE OF FOUR WAYS
C                     (1)   DISCRETELY -THE DATA IS PLOTTED AS INDIVIDUAL
C                           POINTS,WITH THE POINT SYMBOLS DIFFERING FROM
C                           CASE TO CASE.
C                     (2) - LINEARLY - THE DATA POINTS ARE CONNECTED BY
C                           STRAIGHT LINES.
C                     (3) - CONTINUOUSLY - THE INDIVIDUAL POINTS ARE
C                           ORIGINALLY PLOTTED AS THE DESCRETE POINTS,BUT
C                           THE POINTS UNDER GO A LEAST-SQUARES CURVE-FIT
C                           AND THE CALCULATED FUNCTION IS PLOTTED.
C                     (4) - HISTOGRAM - THE DATA POINTS ARE NOT PLOTTED
C                           INDIVIDUALLY , BUT A HISTOGRAM IS GENERATED
C                           FROM THE INPUT DATA.
C
C              3 - MULTIPLE CASES MAY BE PLOTTED ON THE SAME GRID WITH-
C                  OUT INPUTTING ADDITIONAL GRID INFORMATION.  THIS
C                  OPTION DOES NOT ALLOW FOR DIFFERENT TYPE PLOTS ON
C                  THE SAME GRID.
C
C                  -NOTE...AN OPTION FOR NO DATA POINT MARKS IS
C                  AVAILABLE FOR ANY TYPE OF PLOT.
C                  -NOTE---IF DATA POINT MARKS ARE USED, AN OPTION MAY
C                  BE USED TO LABEL EACH CURVE INDIVIDUALLY.
C                  -NOTE---IF THE USER DESIRES TO SHOW THAT HIS LABELED
C                  VALUES ARE RAISED TO A POWER OF 10, AN OPTION EXISTS
C                  WHICH LABELS THE LAST 'TIC' MARK ON THE AXIS IN SUCH
C                  A MANNER TO INDICATE HOW THE VALUES SHOULD BE READ.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INTEGER TYPE(3),DESC
      INTEGER CONT
      INTEGER P,R
      INTEGER HISTO,BLNK
      INTEGER O,ERR
```

```
      INTEGER TEN(2)
      DIMENSION XNAME(5)
      COMMON TITLX(7),TITLY(7),TITLE(7)
      COMMON XVALU(101),YVALU(101)
      COMMON KI,XMAX,YMAX,XINC,XSTRT
      COMMON YSTRT,XSCLE,YSCLE
      COMMON NCASE,ICASE
      COMMON NODAT
      DATACONT/'C'/
      DATADESC/'D'/
      DATALINE/'L'/
      DATAHISTO/'H'/
      DATAXCHV/'END'/
      DATABLNK/' '/
      DATABLANK/'      '/
      DATAACK/'OF '/
      ICODE = 0
      J = 0
      K = 0
      R = 2
      P = 3
      TYPE(3)=1
      IDUMY = LINE
C
C***  READ IN THE INFORMATION ABOUT THE NUMBER OF CASES, THE LABELS FOR
C***  THE CURVES, AND THE EXPONENTS FOR THE X AND Y AXES
C
   10 READ(R,20)TITLZ,ICASE,A,NCASE,(XNAME(O),O=1,5),TEN(1),TEN(2)
   20 FORMAT(A4,1X,I1,1X,A3,I1,3X,5A6,3X,I2,3X,I2)
      IF (NCASE-1)40,30,30
   30 IF(A-ACK)40,50,40
   40 ERR=1
      CALL ERRO(TYPE,ERR,K,YINC)
   50 CONTINUE
      IF(ICASE-1)70,70,60
   60 IF(NCASE-ICASE)820,370,370
C
C***  THIS IS THE CONSOLE MESSAGE FOR THE USER
C
   70 WRITE(1,80)
   80 FORMAT(' **** POSITION PLOTTER PEN APPROXIMATELY 3 INCHES FROM THE
     1 ****',/,' **** RIGHT EDGE OF THE PLOTTER.     . . . PRESS START .
     2. .  ****',//)
      PAUSE
C
C***  READ IN THE INFORMATION ABOUT THE GRID
C
      READ(R,90)XINC,YINC,XMAX,YMAX,XSTRT,YSTRT,TYPE(1),TYPE(2),TYPE(3),
     1NODAT
   90 FORMAT(6F10.0,4A1)
C
C***  PRINT OUT THE INPUT DATA
C
      WRITE(P,100)
  100 FORMAT(1H1)
      WRITE(P,110)XINC,YINC,XMAX,YMAX,XSTRT,YSTRT,NCASE
  110 FORMAT(/,' X-AXIS INCREMENT =',F10.2,5X,'Y-AXIS INCREMENT =',F10.2
     1,/,' X-AXIS LIMIT =     ',F10.2,5X,'Y-AXIS LIMIT =     'F10.2,/,' X-
     2AXIS ORIGIN =    ',F10.2,5X,'Y-AXIS ORIGIN =    'F10.2,/,' NCASES
     3      =    ',I10)
C
```

```
C***  CHECK THE VALUES FOR LABELING THE AXES TO SEE IF THE STARTING
C***  POINTS ARE LESS THAN THE END POINTS
C
      IF(XMAX-XSTRT)120,120,130
  120 ERR=2
      CALL ERRO(TYPE,ERR,K,YINC)
      GO TO 150
  130 IF(YMAX-YSTRT)140,140,160
  140 ERR=3
      CALL ERRO(TYPE,ERR,K,YINC)
  150 STOP
  160 ERR=4
      CALL ERRO(TYPE,ERR,K,YINC)
C
C***  CONVERT ALPHA-NUMERIC INPUT DATA TO NUMERIC VALUES
C
      DO 260 K=1,3
      WRITE(P,170)K,TYPE(K)
  170 FORMAT(' TYPE(',I2,')  =    ',6X,A4)
      IF(TYPE(K)-CONT)190,180,190
  180 TYPE(K) = 1
      GO TO 260
  190 IF(TYPE(K)-DESC)210,200,210
  200 TYPE(K) = 2
      GO TO 260
  210 IF (TYPE(K)-HISTO)230,220,230
  220 TYPE(K)=4
      GO TO 260
  230 IF (TYPE(K)-LINE)250,240,250
  240 TYPE(K)=3
      GO TO 260
  250 ERR=5
      CALL ERRO(TYPE,ERR,K,YINC)
  260 CONTINUE
C
C***  READ IN TITLES FOR THE X AND Y AXIS AND ALSO THE GRAPH TITLE
C
      READ(R,270)(TITLX(I),I=1,7),(TITLY(J),J=1,7),(TITLE(K),K=1,7)
  270 FORMAT(6A4,A2,6A4,A2,7A4)
  280 CONTINUE
C***********************************************************************
C
C     THIS SECTION COMPUTES THE VALUES FOR THE GRID FROM THE SPECIFIED
C     USER INFORMATION.
C
      INCX = XINC*10.
      INCY = YINC*10.
      IXI = INCX/10
      IYI = INCY/10
      IF((IYI*10)-INCY)300,290,300
  290 IF((IXI*10)-INCX)310,320,310
  300 IYI = IYI+1
      GO TO 290
  310 IXI = IXI + 1
  320 NXTCS = (XMAX-XSTRT)/XINC
C
C     THIS IF TEST LIMITS THE NUMBER OF 'TIC' MARKS FOR THE X-AXIS
C
      IF(NXTCS-20)340,340,330
  330 ERR=6
      CALL ERRO(TYPE,ERR,K,YINC)
```

```
      GO TO 280
  340 NYTCS = (YMAX-YSTRT)/YINC
C
C     THIS IF TEST LIMITS THE NUMBER OF 'TIC' MARKS FOR THE Y-AXIS
C
      IF(NYTCS - 20)360,360,350
  350 ERR=7
      CALL ERRO(TYPE,ERR,K,YINC)
      GO TO 280
  360 CONTINUE
      CALL LABEL (TYPE,TEN,IXI,IYI,INCX,INCY,NXTCS,NYTCS,YINC)
  370 CALL SCALE(XSCLE,YSCLE,XSTRT,YSTRT)
C
C***  READ IN DATA POINTS AND PLOT THEM
C
      KI=1
      CALL EPLOT(1,XSTRT,YSTRT)
  380 READ(R,390)XVALU(KI),YVALU(KI),XCH
  390 FORMAT(2F10.3,A3)
      IF(XCH-XCHV)400,490,400
  400 IF(YVALU(KI)-YSTRT)440,410,410
  410 IF(YVALU(KI)-YMAX)420,420,440
  420 IF(XVALU(KI)-XSTRT)440,430,430
  430 IF(XVALU(KI)-XMAX)450,450,440
  440 ERR=9
      CALL ERRO(TYPE,ERR,K,YINC)
  450 CONTINUE
  460 IF(KI-101)470,480,480
  470 KI = KI + 1
      GO TO 380
C
C***  A LIMIT OF 100 DATA POINTS IS ESTABLISHED FOR THIS PROGRAM
C
  480 ERR=8
      CALL ERRO(TYPE,ERR,K,YINC)
      GO TO 380
  490 NCK=TYPE(3)
      GO TO (500,510,510,830),NCK
  500 JFK = 1
      GO TO 520
  510 JFK = 2
  520 GO TO(540,530),JFK
  530 ICODE = 2
  540 JG=KI-1
C-----------------------------------------------------------------------
      IF(TYPE(3)-3)550,560,550
  550 IF(NODAT-BLNK)730,560,730
  560 DO 630 LBJ=1,JG
      IF(TYPE(3)-3)600,570,600
  570 IF(LBJ-1)600,580,590
  580 CALL EPLOT(1,XVALU(LBJ),YVALU(LBJ))
      IF(NODAT-BLNK)630,610,630
  590 CALL EPLOT(2,XVALU(LBJ),YVALU(LBJ))
      IF(NODAT-BLNK)630,610,630
  600 CALL EPLOT(1,XVALU(LBJ),YVALU(LBJ))
      CALL EPLOT(2,XVALU(LBJ),YVALU(LBJ))
  610 II=ICASE-1
  620 CALL POINT(II)
  630 CONTINUE
C
C***  LABEL THE CURVE
```

```
C
      DO 640 O=1,5
      IF(XNAME(O)-BLANK)650,640,650
  640 CONTINUE
      GO TO 730
  650 X=(XMAX)-(XMAX-XSTRT)/(4.5)
  660 Y=YSTRT-(.5)/(YSCLE)-(.2)*(II)/(YSCLE)
  670 CALL EPLOT (1,X,Y)
      CALL EPLOT (2,X,Y)
  680 CALL POINT(II)
  690 X=(XMAX)-(XMAX-XSTRT)/(5.)
      Y=Y-(.05)/(YSCLE)
  700 CALL ECHAR (X,Y,.1,.1,0.0)
  710 WRITE (7,720)(XNAME(O),O=1,5)
  720 FORMAT(5A6)
C-----------------------------------------------------------------------
  730 WRITE(P,740) TITLX,TITLY
  740 FORMAT(///,14X,'   ***INPUT DATA***',//,1X,6A4,A2,4X,6A4,A2,//)
      WRITE(P,750)TITLZ,ICASE,A,NCASE
  750 FORMAT(20X,A4,I2,1X,A2,I2)
  760 KI = KI - 1
      DO 780 I=1,KI
      WRITE(P,770)XVALU(I),YVALU(I)
  770 FORMAT(7X,F10.4,14X,7F10.4)
  780 CONTINUE
      KI = KI + 1
      CALL EPLOT(1,XSTRT,YSTRT)
  790 CONTINUE
      IF(ICODE-2)800,810,810
  800 CALL FINSH
      CALL LINK(BESFT)
  810 IF(NCASE-ICASE)820,820,10
C
C***  SET THE PEN FOR THE NEXT PLOT
C
  820 CALL EPLOT(1,XMAX,YSTRT)
      CALL SCALE(1.0,1.0,0.0,0.0)
      CALL EPLOT(1,4.0,0.0)
      CALL EXIT
  830 CALL HIST
      ICODE=2
      GO TO 730
      END
// DUP
*DELETE             GECAP
*STORE      WS  UA  GECAP
```

```
// FOR
*ONE WORD INTEGERS
*EXTENDED PRECISION
C
C***  THIS SUBROUTINE DRAWS THE GRID, LABELS THE GRID, AND
C***  PLACES THE TITLES IN POSITION.
C
      SUBROUTINE LABEL(TYPE,TEN,IXI,IYI,INCX,INCY,NXTCS,NYTCS,YINC)
C
C***  GECAP REVISION C, 09/05/72
C
      INTEGER TYPE(3),TEN(2)
      COMMON TITLX(7),TITLY(7),TITLE(7)
      COMMON XVALU(101),YVALU(101)
      COMMON KI,XMAX,YMAX,XINC,XSTRT
      COMMON YSTRT,XSCLE,YSCLE
      COMMON NCASE,ICASE
      COMMON NODAT
C
C***  THE LENGTH OF THE X AND Y AXES ARE GOVERNED BY THE NEXT TWO STATE-
C***  MENTS . FOR LONGER AXIS, CHANGE THE NUMBERS IN THE FOLLOWING STATE
C***  -MENTS SO THAT THE ACTUAL LENGTH OF THE AXIS (IN INCHES ) WILL BE
C***  IN THE DIVIDEND OF THE STATEMENT.
C
      XSCLE = 8.5/(XMAX - XSTRT)
      YSCLE = 6.0/(YMAX-YSTRT)
      YINT = -.2/YSCLE
      XINT = -.32/XSCLE
C
C
C***  DRAW THE GRID
C
      CALL EPLOT(1,0.,0.)
      CALL EPLOT(2,0.,0.)
      CALL SCALE(XSCLE,YSCLE,0.,0.)
      CALL EGRID(1,0.,0.,YINC,NYTCS)
      CALL EPLOT(1,0.,0.)
      CALL EPLOT(2,0.,0.)
      CALL EGRID(0,0.,0.,XINC,NXTCS)
C
C***  LABEL THE GRID
C
      NXTCS = NXTCS + 1
      NYTCS = NYTCS + 1
      X = XINT
      XNUM = XSTRT
      INIT = XSTRT
      NUM = XSTRT*10.
      IF (TYPE(3)-4)20,10,20
   10 X=(XINT)+(XINC)/(2.)
      NXTCS=NXTCS-1
   20 DO 80 I=1,NXTCS
      CALL ECHAR(X,YINT,.1,.1,0.0)
      IF(TYPE(1)-2)60,30,60
   30 WRITE(7,40)INIT
   40 FORMAT(I4)
   50 FORMAT(I5)
      INIT=INIT+IXI
      GO TO 70
   60 WRITE(7,520)XNUM
   70 CONTINUE
```

```
      X=X+XINC
      NUM=NUM+INCX
      XNUM = FLOAT(NUM)/10.
   80 CONTINUE
C
C***  PLACE EXPONENT AT THE END OF THE X AXIS
C
      IF(TEN(1))90,270,90
   90 IF(TYPE(1)-2)170,100,170
  100 X=XMAX+.2/XSCLE-XSTRT
      Y=YINT
      IF(TYPE(3)-4)120,110,120
  110 X=X-XINC/2.
  120 CALL ECHAR (X,Y,.1,.1,0.0)
  130 WRITE (7,300)
      X=X+.4/XSCLE
      IF(TEN(1))160,160,140
  140 IF(TEN(1)-10)150,160,160
  150 X=X-.1/XSCLE
  160 Y=Y+.05/YSCLE
      CALL ECHAR (X,Y,.1,.1,0.0)
      WRITE (7,310)TEN(1)
      GO TO 270
  170 X=XMAX+.4/XSCLE-XSTRT
      Y=YINT
      IF(TYPE(3)-4)190,180,190
  180 X=X-XINC/2.
  190 CALL ECHAR (X,Y,.1,.1,0.0)
  200 WRITE (7,300)
  210 X=X+.4/XSCLE
      IF(TEN(1))240,240,220
  220 IF(TEN(1)-10)230,240,240
  230 X=X-.1/XSCLE
  240 Y=Y+.05/YSCLE
  250 CALL ECHAR (X,Y,.1,.1,0.0)
  260 WRITE (7,310)TEN(1)
  270 CONTINUE
C
C***  LABEL THE Y AXIS AND PLACE EXPONENT AT THE END OF THE Y AXIS
C
      XINT=-.6/XSCLE
      IF(TYPE(2)-2)280,290,280
  280 XINT=-.65/XSCLE
  290 Y=0.0
      INIT = YSTRT
      NUM = YSTRT
      XNUM= YSTRT
      DO 540 I=1,NYTCS
  300 FORMAT('X 10')
  310 FORMAT(I2)
  320 IF(NYTCS-I)490,330,490
  330 IF(TEN(2))340,490,460
  340 FF=0.0
  350 X=-.7/XSCLE+FF
  360 CALL ECHAR (X,Y,.1,.1,0.0)
  370 WRITE(7,300)
  380 X=-.3/XSCLE
  390 ZZ=Y+.05/YSCLE
  400 CALL ECHAR(X,ZZ,.1,.1,0.0)
  410 WRITE(7,310)TEN(2)
  420 XINT=XINT-.7/XSCLE+FF
```

```
  430 IF(TYPE(2)-2)440,490,440
  440 XINT=XINT-.05/XSCLE
  450 GO TO 490
  460 IF(TEN(2)-10)470,340,340
  470 FF=.1/XSCLE
  480 GO TO 350
  490 CONTINUE
      CALL ECHAR(XINT,Y,.1,.1,0.0)
      IF(TYPE(2)-2)510,500,510
  500 WRITE(7,50)INIT
      INIT = INIT +IYI
      GO TO 530
  510 WRITE(7,520)XNUM
  520 FORMAT(F6.2)
  530 CONTINUE
      Y = Y + YINC
      NUM = NUM + INCY
      XNUM = XNUM + YINC
  540 CONTINUE
C
C***  PLACE THE TITLES ON THE GRAPH
C
      X = 2.4/XSCLE
      Y = -1./YSCLE
      CALL ECHAR(X,Y,.15,.15,0.0)
      WRITE(7,550) (TITLX(L),L=1,7)
  550 FORMAT(7A4)
      ANG = 3.14157/2.0
      X = -.68/XSCLE
      IF(TYPE(2)-2)560,570,560
  560 X=-.73/XSCLE
  570 Y=1.0/YSCLE
      CALL ECHAR(X,Y,.15,.15,ANG)
      WRITE(7,550)(TITLY(M),M=1,7)
      X = 2.5/XSCLE
      Y = 6.5/YSCLE
      CALL ECHAR(X,Y,.2,.2,0.0)
      WRITE(7,550)(TITLE(N),N=1,7)
      CALL EPLOT(1,0.,0.)
      CALL EPLOT(2,0.,0.)
      RETURN
      END
// DUP
*DELETE             LABEL
*STORE      WS  UA  LABEL
```

```
// FOR
*ONE WORD INTEGERS
*EXTENDED PRECISION
C
C***  THIS SUBROUTINE CONTAINS ALL ERROR MESSAGES FOR GECAP AND IS
C***  CALLED ONLY BY GECAP
C
      SUBROUTINE ERRO(TYPE,ERR,K,YINC)
C
C***  GECAP REVISION C, 09/05/72
C
      INTEGER TYPE(3),ERR
      COMMON TITLX(7),TITLY(7),TITLE(7)
      COMMON XVALU(101),YVALU(101)
      COMMON KI,XMAX,YMAX,XINC,XSTRT
      COMMON YSTRT,XSCLE,YSCLE
      COMMON NCASE,ICASE
      COMMON NODAT
      GO TO (10,30,50,70,210,250,270,280,300),ERR
   10 WRITE (1,20)
   20 FORMAT(/,'****ERROR****  CASE CARD INPUT VARIABLE IS INCORRECT ',/
     1,'*NUMBER OF CASES HAS BEEN SET EQUAL TO 1 + EXECUTION RESUMED*')
      NCASE=1
      GO TO 320
   30 WRITE(1,40)
   40 FORMAT(/,'****ERROR****END POINT OF X AXIS IS LESS THAN STARTING P
     1OINT****',/,'             ****EXECUTION DISCONTINUED****')
      GO TO 320
   50 WRITE(1,60)
   60 FORMAT(/,'****ERROR****END POINT OF Y AXIS IS LESS THAN STARTING P
     1OINT****',/,'             ****EXECUTION DISCONTINUED****')
      GO TO 320
C
C***  CHECK THE NUMBER OF DECIMAL PLACES IN THE VALUES USED FOR LABELING
C***  THE AXES AND WRITE OUT AN ERROR MESSAGE IF MORE THAN TWO
C***  DECIMAL PLACES ARE USED
C
   70 NCXI=IFIX(XINC)
      CNX=(XINC-NCXI)*(100.)
      NCX=IFIX(CNX)
      IF(CNX-NCX)80,90,80
   80 WRITE(1,190)
      PAUSE
   90 NCYI=IFIX(YINC)
      CNY=(YINC-NCYI)*(100.)
      NCY=IFIX(CNY)
      IF(CNY-NCY)100,110,100
  100 WRITE(1,190)
      PAUSE
  110 NXX=IFIX(XSTRT)
      XXX=(XSTRT-NXX)*(100.)
      IXX=IFIX(XXX)
      IF(XXX-IXX)120,130,120
  120 WRITE(1,190)
      PAUSE
  130 NYY=IFIX(YSTRT)
      YYY=(YSTRT-NYY)*(100.)
      IYY=IFIX(YYY)
      IF(YYY-IYY)140,150,140
  140 WRITE(1,190)
      PAUSE
```

```
  150 NXM=IFIX(XMAX)
      XXM=(XMAX-NXM)*(100.)
      IXM=IFIX(XXM)
      IF(XXM-IXM)160,170,160
  160 WRITE(1,190)
      PAUSE
  170 NYM=IFIX(YMAX)
      YYM=(YMAX-NYM)*(100.)
      IYM=IFIX(YYM)
      IF(YYM-IYM)180,200,180
  180 WRITE(1,190)
      PAUSE
  190 FORMAT(2X,'****ERROR****THE VALUES SPECIFIED FOR LABELING THE AXES
     1',/,'REQUIRE MORE PRECISION THAN THAT ALLOWABLE IN GECAP..........
     2',/,'...CHECK GECAP USER MANUAL FOR DETAILS...',/,'THIS ERROR MAY
     3BE INSIGNIFICANT IN SPECIAL CASES...IF EXECUTION',/,'IS STILL DESI
     4RED PRESS START...IF NOT, PRESS STOP')
  200 GO TO 320
  210 WRITE(1,220)K
  220 FORMAT(2X,'****ERROR****INCORRECT PLOT OPTION WAS USED FOR',/,'
     1VARIABLE TYPE(',I1,').****CHECK GECAP USER MANUAL FOR DETAILS.....
     2',/,'****TYPE IN THE DESIRED OPTION FROM THE CONSOLE TYPEWRITER
     3',/,'(C,D,H, OR L).   ****PRESS (EOF) BUTTON TO CONTINUE EXECUTION*
     4****')
  230 READ (6,240)TYPE(K)
  240 FORMAT(A1)
      K=K-1
      GO TO 320
  250 WRITE(1,260)
  260 FORMAT(/,' ***ERROR***   NUMBER OF TIC MARKS EXCEEDS LEGIBILITY LI
     1MIT***',/,' STANDARD FIX-UP TAKEN, EXECUTION CONTINUING')
      NXINC = XMAX/10.
      XINC = NXINC
      GO TO 320
  270 WRITE(1,260)
      NYINC = YMAX/10.
      YINC = NYINC
      GO TO 320
  280 WRITE(1,290)
  290 FORMAT(/,' **** ERROR. . MORE THAN 100 DATA POINTS ON INPUT ****',
     1/,'   ****  EXECUTION CONTINUING WITHOUT REMAINING POINTS**')
      GO TO 320
  300 WRITE(1,310)
  310 FORMAT(2X,'****ERROR****A DATA POINT WAS FOUND TO EXCEED THE LIMIT
     1S OF THE AXES...****',/,'          ****EXECUTION DISCONTINUED****')
      GO TO 330
  320 RETURN
  330 STOP
      END
// DUP
*DELETE             ERRO
*STORE      WS  UA  ERRO
```

```
// FOR
*ONE WORD INTEGERS
*EXTENDED PRECISION
C
C***   THIS SUBROUTINE GENERATES A HISTOGRAM PLOT FROM THE INPUT DATA
C
       SUBROUTINE HIST
C
C***   GECAP REVISION C, 09/05/72
C
       COMMON TITLX(7),TITLY(7),TITLE(7)
       COMMON XVALU(101),YVALU(101)
       COMMON KI,XMAX,YMAX,XINC,XSTRT
       COMMON YSTRT,XSCLE,YSCLE
       COMMON NCASE,ICASE
    10 NPTS=KI-1
    20 Z=XVALU(2)-XVALU(1)
    30 IF(XVALU(1)-XSTRT)230,60,40
    40 CALL EPLOT (1,XVALU(1),YVALU(1))
    50 CALL EPLOT (2,XVALU(1),YSTRT)
    60 DO 100 I=1,NPTS
    70 CALL EPLOT (1,XVALU(I),YVALU(I))
    80 PLUS=XVALU(I)+Z
    90 CALL EPLOT (2,PLUS,YVALU(I))
   100 CONTINUE
   110 CALL EPLOT (1,XSTRT,YSTRT)
   120 NPTS=NPTS-1
   130 DO 200 I=1,NPTS
   140 IF (YVALU(I)-YVALU(I+1))180,150,150
   150 CALL EPLOT (1,XVALU(I+1),YVALU(I))
   160 CALL EPLOT (2,XVALU(I+1),YSTRT)
   170 GO TO 200
   180 CALL EPLOT (1,XVALU(I+1),YVALU(I+1))
   190 CALL EPLOT (2,XVALU(I+1),YSTRT)
   200 CONTINUE
   210 CALL EPLOT (1,PLUS,YVALU(I))
   220 CALL EPLOT (2,PLUS,YSTRT)
   230 RETURN
       END
// DUP
*DELETE             HIST
*STORE      WS  UA  HIST
```

```
// FOR
*IOCS(1132 PRINTER,KEYBOARD,TYPEWRITER,CARD,PLOTTER)
*EXTENDED PRECISION
*ONE WORD INTEGERS
C                                                                       BSTF0000
C***  GECAP REVISION C, 09/05/72                                        BSTF0010
C                                                                       BSTF0020
C-----------------------------------------------------------------------BSTF0030
C         PROGRAM BESFT WAS DESIGNED TO SUPPLY A BEST-FIT LEAST SQUARES BSTF0040
C         FUNCTION TO A GIVEN SET OF INPUT DATA.  IT WAS DESIGNED TO BE BSTF0050
C         USED EXCLUSIVELY WITH PROGRAM GECAP.  THE TWO MAINLINES ARE   BSTF0060
C         TIED TOGETHER BY THE USE OF SYSTEM ROUTINE LINK.              BSTF0070
C-----------------------------------------------------------------------BSTF0080
      INTEGER P,R                                                       BSTF0090
      DIMENSION A(12,13),SUM(20),WORK(20),IHLD(20),V(10)                BSTF0100
      DIMENSION F1(49),F2(49)                                           BSTF0110
      COMMON TITLX(7),TITLY(7),TITLE(7)                                 BSTF0120
      COMMON X(101),Y(101)                                              BSTF0130
      COMMON KI,XMAX,YMAX,XINC,XSTRT                                    BSTF0140
      COMMON YSTRT,XSCLE,YSCLE                                          BSTF0150
      COMMON NCASE,ICASE                                                BSTF0160
      DATAA/156*0./                                                     BSTF0170
      DATAF1/161.,18.5,10.1,7.71,6.61,5.99,5.59,5.32,5.12,4.96,4.84,4.75BSTF0180
     1,4.67,4.60,4.54,4.49,4.45,4.41,4.38,4.35,4.32,4.30,4.28,4.26,4.24,BSTF0190
     24.23,4.21,4.20,4.18,4.17,4.16,4.15,4.14,4.13,4.12,4.11,4.105,4.10,BSTF0200
     34.09,4.08,4.075,4.07,4.065,4.06,4.055,4.05,4.045,4.04,4.035/      BSTF0210
      DATAF2/200.,19.0,9.55,6.94,5.79,5.14,4.74,4.46,4.26,4.10,3.98,3.89BSTF0220
     1,3.81,3.74,3.68,3.63,3.59,3.55,3.52,3.49,3.47,3.44,3.42,3.40,3.39,BSTF0230
     23.37,3.35,3.34,3.33,3.32,3.31,3.29,3.285,3.28,3.27,3.26,3.25,3.24,BSTF0240
     33.235,3.23,3.225,3.22,3.215,3.21,3.205,3.20,3.195,3.19,3.185/     BSTF0250
C                                                                       BSTF0260
C***  RE-ESTABLISH THE ORIGIN  ( SEE MAINLINE GECAP )                    BSTF0270
C                                                                       BSTF0280
      CALL SCALE(XSCLE,YSCLE,XSTRT,YSTRT)                               BSTF0290
      KI = KI - 1                                                       BSTF0300
      ERROR = 0.                                                        BSTF0310
      SAVE1 = 0.                                                        BSTF0320
      SAVE2 =9999999.                                                   BSTF0330
      SAVE3 =9999999.                                                   BSTF0340
      NSUB1 = 0                                                         BSTF0350
      NSUB2 = 0                                                         BSTF0360
      NX=-1                                                             BSTF0370
      NCOUT=0                                                           BSTF0380
      NCT=0                                                             BSTF0390
      P = 3                                                             BSTF0400
      R = 2                                                             BSTF0410
   10 CONTINUE                                                          BSTF0420
C                                                                       BSTF0430
C***  BEGIN CURVE-FIT ITERITION                                          BSTF0440
C                                                                       BSTF0450
   20 NX = NX + 1                                                       BSTF0460
      NY = NX + 1                                                       BSTF0470
C                                                                       BSTF0480
C***  LIMIT THE DEGREE OF POLYNOMIAL TO ONE LESS THAN NUMBER OF POINTS  BSTF0490
C***  ON INPUT                                                          BSTF0500
C                                                                       BSTF0510
      IF(NX-KI)30,200,200                                               BSTF0520
   30 IF(NX-9)40,40,200                                                 BSTF0530
   40 CALL CURF(X,Y,KI,NX,NY,A,SUM,V,WORK,IHLD,E)                       BSTF0540
      IF(E)70,70,50                                                     BSTF0550
   50 WRITE(P,60)                                                       BSTF0560
```

```
 60 FORMAT(//,' ***ERROR*** ERROR IN ROUTINE CURF.  EXECUTION CONTINUI
   1NG')
 70 CONTINUE
    ERROR = 0.0
    SIGSQ = 0.0
C
C***  CALCULATE ERRORS FOR THIS POLYNOMIAL AND SUM THEM UP
C
    DO 80 KK = 1,KI
    YCALC = F(A,NX,X(KK))
    SIGSQ = ABS(Y(KK)-YCALC)**2.
 80 ERROR = ERROR + SIGSQ
 90 CONTINUE
100 ERROR=ERROR/(KI-(NX+1))
    NSUB1 = NX
    NSUB2 = NY
    SAVE1=SAVE2
    SAVE2 = SAVE3
    SAVE3 = ERROR
110 TEST1=ABS(SAVE2-ERROR)/ERROR
    TEST2=ABS(SAVE1-ERROR)/ERROR
    I=KI-(NX+1)
    IF(I-49)130,130,120
120 I=49
130 IF(TEST1-F1(I))150,140,140
140 NCOUT=0
    GO TO 160
150 NCOUT=1
160 IF(TEST2-F2(I))180,170,170
170 NCT=NCOUT
    GO TO 20
180 IF(NCT-1)170,190,190
190 NSUB1=NX
    NSUB2 = NX+ 1
    GO TO 220
C
C***  COMPUTE THE BEST-FIT CO-EFFICIENTS
C
200 WRITE(P,210)
210 FORMAT(1X,' ***USERS MESSAGE*** THE POLYNOMINAL CALCULATED MAY OR
   1MAY NOT BE THE BEST FITTED CURVE')
220 CALL CURF(X,Y,KI,NSUB1,NSUB2,A,SUM,V,WORK,IHLD,E)
    WRITE(P,230)
230 FORMAT(1H1)
    WRITE(P,240)NSUB1
240 FORMAT(/,' *CO-EFFICIENTS FOR POLYNOMIAL OF DEGREE',I3,'*')
    DO 260 K=1,NSUB2
    J = NSUB2+1-K
    WRITE(P,250)A(J,1)
250 FORMAT(F20.12)
260 CONTINUE
    WRITE(P,270)
270 FORMAT(/////)
    WRITE(P,280)NSUB1
280 FORMAT(' *CALCULATED VALUES AND SUM OF ERRORS SQUARED FOR',I2,' DE
   1GREE POLYNOMIAL*')
    SIGSQ = 0.0
    ERROR = 0.0
    DO 300 I=1,KI
    YCALC = F(A,NSUB1,X(I))
    SIGSQ = ABS(Y(I)-YCALC)**2.
```

```
      WRITE(P,290)I,YCALC,SIGSQ                                         BSTF1180
  290 FORMAT(' FOR POINT',I3,'  YCALC =',F20.6,10X,'DIFF**2 = ',F20.12) BSTF1190
  300 ERROR = ERROR + SIGSQ                                             BSTF1200
      WRITE(P,310)ERROR                                                 BSTF1210
  310 FORMAT(//,' SUM OF SQUARES OF ERRORS FOR CURVE-FIT =',F20.12)     BSTF1220
      WRITE(P,230)                                                      BSTF1230
C                                                                       BSTF1240
C***  PLOT THE CALCULATED FUNCTION  ON THE SUPPLIED GRID ( SEE MAINLINE BSTF1250
C                                                                       BSTF1260
      XBEGN=X(1)                                                        BSTF1270
      STEP = XBEGN                                                      BSTF1280
      YI = F(A,NSUB1,STEP)                                              BSTF1290
      CALL EPLOT(1,STEP,YI)                                             BSTF1300
      STEP = .1*XINC                                                    BSTF1310
      XLAST=X(KI)                                                       BSTF1320
      XI=XBEGN                                                          BSTF1330
  320 XI=XI+STEP                                                        BSTF1340
      IF(XLAST-XI)330,330,340                                           BSTF1350
  330 XI=XLAST                                                          BSTF1360
  340 YI= F(A,NSUB1,XI)                                                 BSTF1370
      CALL EPLOT(2,XI,YI)                                               BSTF1380
      IF(XLAST-XI)350,350,320                                           BSTF1390
  350 CONTINUE                                                          BSTF1400
  360 IF(NCASE-ICASE)380,380,370                                        BSTF1410
  370 CALL EPLOT(1,XSTRT,YSTRT)                                         BSTF1420
      CALL FINSH                                                        BSTF1430
      CALL LINK(GECAP)                                                  BSTF1440
C                                                                       BSTF1450
C***  SET UP THE PEN FOR THE NEXT PLOT                                  BSTF1460
C                                                                       BSTF1470
  380 CALL EPLOT(1,XI,YI)                                               BSTF1480
      CALL EPLOT(1,XMAX,YSTRT)                                          BSTF1490
      CALL SCALE(1.0,1.0,0.0,0.0)                                       BSTF1500
      CALL EPLOT(1,4.0,0.0)                                             BSTF1510
      CALL EXIT                                                         BSTF1520
      END                                                               BSTF1530
// DUP
*DELETE             BESFT
*STORE      WS  UA  BESFT
```

```
// FOR
*EXTENDED PRECISION
*ONE WORD INTEGERS
      FUNCTION F(A,N,X)                                                 F0000
C                                                                       F0010
C***  GECAP REVISION C, 09/05/72                                        F0020
C                                                                       F0030
C                                                                       F0040
C*** FUNCTION F CALCULATES A Y VALUE FROM A GIVEN X VALUE USING THE LEAST F0050
C*** SQUARES CO-EFFICIENTS                                              F0060
C                                                                       F0070
      DIMENSION A(12,13)                                                F0080
   10 F =A(N+1,1)                                                       F0090
      L = N                                                             F0100
      IF(N)20,40,20                                                     F0110
   20 DO 30 I = 1,N                                                     F0120
      F= A(L,1)*X**I + F                                                F0140
   30 L = L - 1                                                         F0150
   40 RETURN                                                            F0160
      END                                                               F0170
// DUP
*DELETE             F
*STORE      WS  UA  F
```

```
// FOR
*ONE WORD INTEGERS
*EXTENDED PRECISION
      SUBROUTINE CURF (X,Y,N,M,MX,A,SUM,V,WORK,IHLD,E)
C
C***  GECAP REVISION C, 09/05/72
C
C     CATEGORY
C         MATHEMATICAL
C     PURPOSE
C         FIT TABULAR DATA TO POLYNOMIAL OF TYPE Y=A0+A1*X+A2*X**2+...
C     DESCRIPTION
C         THIS SUBROUTINE WILL FIT N DATA POINTS IN X AND Y TO A
C         POLYNOMIAL OF THE TYPE  Y=A0+A1*X+A2*X**2+...+AM*X**M,
C         WHERE N IS GREATER THAN OR EQUAL TO M+1.  THE COEFFICIENTS
C         A0,A1,...,AM ARE DETERMINED BY THE METHOD OF LEAST SQUARES.
C         THE ROUTINE MAY BE RUN IN EITHER SINGLE OR DOUBLE PRECISION
C         ARITHMETIC.  THE DATA POINTS NEED NOT BE EQUALLY SPACED.
C     INPUT
C         1  X        FIRST LOCAT+ON OF AN ARRAY CONTAINING N
C                     INDEPENDENT VARIABLE DATA POINTS.
C         2  Y        FIRST LOCATION OF AN ARRAY CONTAINING N
C                     DEPENDENT VARIABLE DATA POINTS.
C         3  N        NUMBER OF DATA POINTS TO BE FITTED.
C         4  M        DEGREE OF POLYNOMIAL TO WHICH DATA IS TO BE
C                     FITTED.
C         5  MX       VARIABLE DIMENSION INTEGER.  MUST BE EQUAL TO
C                     MAXIMUM M USED PLUS 1.
C         6  A        A TWO DIMENSIONAL ARRAY USED BY CURFIT TO SOLVE
C                     THE LEAST SQUARES NORMAL EQUATIONS.  A MUST BE
C                     DIMENSIONED TO (MX,MX+1) IN THE CALLING PROGRAM.
C         7  SUM      A TEMPORARY STORAGE AREA.  MUST BE DIMENSIONED
C                     TO (2*MX) IN CALLING PROGRAM.
C         8  V        A TEMPORARY STORAGE AREA.  MUST BE DIMENSIONED
C                     TO (MX) IN CALLING PROGRAM.
C         9  WORK     SEE 8 ABOVE.
C         10 IHLD     SEE 8 ABOVE.
C     OUTPUT
C         1  A        A(1,1) CONTAINS THE COEFFICIENT AM, A(2,1)
C                     CONTAINS AM-1,...,A(M+1,1) CONTAINS A0 OF THE
C                     EQUATION Y=A0+A1*X+A2*X**2+...+AM*X**M.
C         2  E        ERROR CHECK.  IF E=0.,O.K.  IF E=1. AN ERROR
C                     HAS OCCURED IN THE CALCULATION OF THE
C                     COEFFICIENTS.
C     REMARKS
C         THIS ROUTINE CONTAINS A METHOD OF SCALING WHICH PREVENTS
C         OVERFLOW IN CASE THE DATA POINTS ARE LARGE, AND PREVENTS
C         LOSS OF ACCURACY IN CASE THE INDEPENDENT DATA POINTS
C         AGREE TO SEVERAL SIGNIFICANT FIGURES.
C     REFERENCES
C         INTERNAL TECHNICAL NOTE NO. TN-65-01
C         LEAST SQUARES POLYNOMIAL CURVE FIT NO. 2
C         BILLY G. GIBBS
C         SCIENTIFIC DIGITAL PROGRAMMING BRANCH
C         ARMY COMPUTATION CENTER
C         REDSTONE ARSENAL, ALABAMA
C     CONTACT
C         CHIEF, THEORETICAL PROBLEMS SECTION
C         SCIENTIFIC DIGITAL PROGRAMMING BRANCH
C         ARMY COMPUTATION CENTER
C
```

```
C     SUBROUTINE CURF (X,Y,N,M,MX,A,SUM,V,WORK,IHLD,E)                 CURF0580
      DIMENSION X(1),Y(1),SUM(1),V(1),A(12,1),WORK(1),IHLD(1)          CURF0590
      E = 0.0                                                          CURF0600
      LS = 2*M + 1                                                     CURF0610
      LB = M + 2                                                       CURF0620
      LV = M + 1                                                       CURF0630
      XH = ABS(X(1))                                                   CURF0640
      DO 20 I=2,N                                                      CURF0650
      IF(XH- ABS(X(I)))10,20,20                                        CURF0660
   10 XH= ABS(X(I))                                                    CURF0670
   20 CONTINUE                                                         CURF0680
      DO 30 I=1,N                                                      CURF0690
   30 X(I)=X(I)/XH                                                     CURF0700
      FHH=X(1)                                                         CURF0710
      DO 50 I=2,N                                                      CURF0720
      IF(FHH-X(I))50,50,40                                             CURF0730
   40 FHH=X(I)                                                         CURF0740
   50 CONTINUE                                                         CURF0750
      DO 60 I=1,N                                                      CURF0760
   60 X(I)=X(I)-FHH                                                    CURF0770
      DO 70 J= 2, LS                                                   CURF0780
   70 SUM(J)=0.                                                        CURF0790
      SUM(1)= N                                                        CURF0800
      DO 80 J=1,LV                                                     CURF0810
   80 V(J)=0.                                                          CURF0820
      DO 100 I=1,N                                                     CURF0830
      P=1.                                                             CURF0840
      V(1)= V(1)+ Y(I)                                                 CURF0850
      DO 90 J= 2,LV                                                    CURF0860
      P = X(I) * P                                                     CURF0870
      SUM(J) = SUM(J) + P                                              CURF0880
   90 V(J) = V(J) + Y(I)* P                                            CURF0890
      DO 100 J= LB,LS                                                  CURF0900
      P = X(I) * P                                                     CURF0910
  100 SUM(J) = SUM(J) + P                                              CURF0920
      KK = LV                                                          CURF0930
      DO 120 I=1,LV                                                    CURF0940
      L = I                                                            CURF0950
      DO 110 K=1,LV                                                    CURF0960
      A(K,KK) = SUM(L)                                                 CURF0970
  110 L = L+1                                                          CURF0980
  120 KK = KK - 1                                                      CURF0990
      DO 130 I=1,LV                                                    CURF1000
  130 A(I,LB) = V(I)                                                   CURF1010
      DO 140 I=1,LV                                                    CURF1020
  140 IHLD(I)=I                                                        CURF1030
      JJ=LB                                                            CURF1040
      DO 280 I=1,LV                                                    CURF1050
      KK=LV-I                                                          CURF1060
      IF(KK)240,240,150                                                CURF1070
  150 LL=KK+1                                                          CURF1080
      IJJ=1                                                            CURF1090
      L=I                                                              CURF1100
      WORK(1)=A(1,1)                                                   CURF1110
      DO 170 II=1,LL                                                   CURF1120
      DO 170 J=1,LL                                                    CURF1130
      IF(ABS(WORK(1))-ABS(A(II,J)))160,170,170                         CURF1140
  160 WORK(1)=A(II,J)                                                  CURF1150
      L=J+I-1                                                          CURF1160
      IJJ=J                                                            CURF1170
  170 CONTINUE                                                         CURF1180
```

```
      IF(IJJ-1)200,200,180
  180 DO 190 II=1,LV
      Z=A(II,1)
      A(II,1)=A(II,IJJ)
  190 A(II,IJJ)=Z
      IY=IHLD(I)
      IHLD(I)=IHLD(L)
      IHLD(L)=IY
  200 DO 230 L=1,KK
      IF(ABS(A(1,1))-ABS(A(L+1,1)))210,230,230
  210 DO 220 J=1,JJ
      Z=A(1,J)
      A(1,J)=A(L+1,J)
  220 A(L+1,J)=Z
  230 CONTINUE
  240 JJ=JJ-1
      IF(A(1,1))250,430,250
  250 DO 260 J=1,JJ
  260 WORK(J)=A(1,J+1)/A(1,1)
      KK=JJ+1
      DO 270 K=1,M
      DO 270 J=2,KK
  270 A(K,J-1)=A(K+1,J)-A(K+1,1)*WORK(J-1)
      DO 280 J=1,JJ
  280 A(LV,J)=WORK(J)
      LB=LV-1
      DO 310 I=1,LB
      L=I+1
      DO 310 J=L,LV
      IF(IHLD(I)-IHLD(J))310,310,290
  290 IY=IHLD(I)
      IHLD(I)=IHLD(J)
      IHLD(J)=IY
      DO 300 K=1,1
      Z=A(I,K)
      A(I,K)=A(J,K)
  300 A(J,K)=Z
  310 CONTINUE
      DO 320 I=1,N
  320 X(I)=X(I)+FHH
      NN=LV
      DO 380 I=1,LV
      SUM(1)=0.0
      NN=NN-1
      KK=NN+1
      IF(KK-M)330,330,380
  330 L=M-KK+1
      DO 370 K=KK,M
      II=K-NN
      SIGN=1.
      FAC=1.
      IF(II-0)340,360,340
  340 SIGN=(-1)**II
      IF=K
      DO 350 J=1,II
      Z=J
      WORK(1)=IF
      FAC=FAC*WORK(1)/Z
  350 IF=IF-1
  360 SUM(1)=SUM(1)+FAC*SIGN*FHH**II*A(L,1)
  370 L=L-1
```

```
  380 A(I,2)=A(I,1)+SUM(1)
      DO 390 I=1,LV
  390 A(I,1)=A(I,2)
      DO 400 I=1,N
  400 X(I)=X(I)*XH
      K=LV+1
      DO 410 I=1,LV
      K=K-1
      NN=I-1
  410 A(K,1)=A(K,1)/(XH**NN)
  420 RETURN
  430 E=1.
      GO TO 420
      END
// DUP
*DELETE             CURF
*STORE        WS  UA  CURF
```

```
// ASM
                    ENT       FINSH
              FINSH DC        0
                    MDX   L   50,0
                    MDX       *-3
                    BSC   I   FINSH
                    END

// DUP
*STORE      WS  UA  FINSH
```

APPENDIX C

SYMBOLS AVAILABLE FOR THE IBM 1130 PLOTTER ROUTINES

Alphabetic Characters	A - Z
Numeric Characters	0 - 9

<u>Special Characters</u>

Characters	Punch
.	12-8-3
(	12-8-5
+	12-8-6
$	11-8-3
*	11-8-4
)	11-8-5
-	11
/	0-1
,	0-8-3
.	8-5
=	8-6
Space	Blank

APPROVAL

COMPUTER USER'S MANUAL FOR A GENERALIZED CURVE FIT AND PLOTTING PROGRAM

By Ronald A. Schlagheck, B. D. Beadle II,
B. D. Dolerhie, Jr., and J. W. Owen

The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

This document has been reviewed and approved for technical accuracy.

H. R. Sells

H. R. SELLS
Chief, Systems Operations Branch

T. P. Isbell

T. P. ISBELL
Chief, Mechanical and Crew Systems
Integration Division

Karl L. Heimburg

KARL L. HEIMBURG
Director, Astronautics Laboratory

Page Intentionally Left Blank

Page Intentionally Left Blank