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

January 1, 1973

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama
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, scientists, 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.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>INTRODUCTION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECTION II.</td>
<td>CONTROL CARDS, INPUT DATA AND OPERATING INSTRUCTIONS</td>
<td>2</td>
</tr>
<tr>
<td>A. Control Cards</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>B. Input Data</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>C. Operating Instructions</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>SECTION III.</td>
<td>EXAMPLE CASES</td>
<td>8</td>
</tr>
<tr>
<td>SECTION IV.</td>
<td>PROGRAM MESSAGES</td>
<td>28</td>
</tr>
<tr>
<td>A. User Messages</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>B. Error Messages</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>SECTION V.</td>
<td>PROGRAM DESCRIPTION</td>
<td>31</td>
</tr>
<tr>
<td>A. General</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>B. Program Structure</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>C. Program Restrictions and Limitations</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>D. Program Modifications for Other Computing Systems</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>APPENDIX A:</td>
<td>DETERMINATION OF THE BEST FIT POLYNOMIAL</td>
<td>36</td>
</tr>
<tr>
<td>APPENDIX B:</td>
<td>PROGRAM LISTING</td>
<td>39</td>
</tr>
<tr>
<td>APPENDIX C:</td>
<td>SYMBOLS AVAILABLE FOR THE IBM 1130 PLOTTER ROUTINES</td>
<td>60</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Generalized GECAP deck set-up</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Listing of input deck for Example Case 1</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Listing of printed output for Example Case 1</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Plotted output for Example Case 1</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Listing of input deck for Example Case 2</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>Listing of printed output for Example Case 2</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Plotted output for Example Case 2</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>Listing of input deck for Example Case 3</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>Listing of printed output for Example Case 3</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>Plotted output for Example Case 3</td>
<td>14</td>
</tr>
<tr>
<td>11</td>
<td>Listing of input deck for Example Case 4</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>Listing of printed output for Example Case 4</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>Plotted output for Example Case 4</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>Listing of input deck for Example Case 5</td>
<td>19</td>
</tr>
<tr>
<td>15</td>
<td>Listing of printed output for Example Case 6</td>
<td>21</td>
</tr>
<tr>
<td>16</td>
<td>Plotted output for Example Case 5</td>
<td>27</td>
</tr>
</tbody>
</table>
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 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 x 11 in. standard 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, IX, 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. (Optional, leave blank if not needed)

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

Columns 53-54: These columns are reserved for the exponent used to label the y axis. (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:

<table>
<thead>
<tr>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4 6 8-9 11 15-44 48-49 53-54</td>
</tr>
<tr>
<td>CASE 1 OF 4 TEST RUN 24D -1 -1</td>
</tr>
</tbody>
</table>

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 \( \times 10^{\text{exp}} \), where exp is the input value. \((-9 \leq \text{exp} \leq 99\)). Example: \(20 \times 10^{-1}\). 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:

<table>
<thead>
<tr>
<th>Punch</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>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.</td>
</tr>
<tr>
<td>C</td>
<td>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.)</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>1-10</th>
<th>11-20</th>
<th>21-30</th>
<th>31-40</th>
<th>41-50</th>
<th>51-60</th>
<th>61</th>
<th>62</th>
<th>63</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2.0</td>
<td>9.0</td>
<td>90.0</td>
<td>1.0</td>
<td>60.0</td>
<td>D</td>
<td>C</td>
<td>C</td>
<td>*</td>
</tr>
</tbody>
</table>

• 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

<table>
<thead>
<tr>
<th>1-26</th>
<th>27-52</th>
<th>53-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRESSURE CO2 MM-HG</td>
<td>EFFICIENCY-PCT.</td>
<td>MOLECULAR SIEVE</td>
</tr>
</tbody>
</table>

• 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

<table>
<thead>
<tr>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
</tr>
<tr>
<td>1.53</td>
</tr>
</tbody>
</table>

- End Card

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

<table>
<thead>
<tr>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>21-23</td>
</tr>
<tr>
<td>END</td>
</tr>
</tbody>
</table>

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.
4. Press the NPRO button on card reader.
5. Load the GECAP deck.
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.

```
// JOB OR ///**COLD START CARD**///
// XEU GECAP 1
*LOCALGECAP*LABEL*HIST*ERROR
CASE 1 OF 1
1. 5. 10. 100. U. U. 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 symbols 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) = 0
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 5. Listing of input deck for Example Case 2.

<table>
<thead>
<tr>
<th>X-AXIS INCREMENT</th>
<th>2.00</th>
<th>Y-AXIS INCREMENT</th>
<th>10.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-AXIS LIMIT</td>
<td>20.00</td>
<td>Y-AXIS LIMIT</td>
<td>200.00</td>
</tr>
<tr>
<td>X-AXIS ORIGIN</td>
<td>0.00</td>
<td>Y-AXIS ORIGIN</td>
<td>50.00</td>
</tr>
<tr>
<td>NCASES</td>
<td>1</td>
<td>TYPE( 1)</td>
<td>D</td>
</tr>
<tr>
<td>TYPE( 2)</td>
<td>D</td>
<td>TYPE( 3)</td>
<td>D</td>
</tr>
</tbody>
</table>

***INPUT DATA***

**TEST CASE 2 X-COORDINATE**

<table>
<thead>
<tr>
<th>CASE 1 OF 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00000</td>
</tr>
<tr>
<td>4.00000</td>
</tr>
<tr>
<td>6.00000</td>
</tr>
<tr>
<td>8.00000</td>
</tr>
<tr>
<td>9.50000</td>
</tr>
<tr>
<td>10.0000</td>
</tr>
<tr>
<td>12.0000</td>
</tr>
<tr>
<td>14.0000</td>
</tr>
<tr>
<td>16.0000</td>
</tr>
<tr>
<td>18.0000</td>
</tr>
<tr>
<td>20.0000</td>
</tr>
<tr>
<td>110.0000</td>
</tr>
<tr>
<td>120.0000</td>
</tr>
<tr>
<td>115.5600</td>
</tr>
<tr>
<td>170.2000</td>
</tr>
<tr>
<td>120.0000</td>
</tr>
<tr>
<td>125.5000</td>
</tr>
<tr>
<td>130.0000</td>
</tr>
<tr>
<td>105.0000</td>
</tr>
<tr>
<td>190.0000</td>
</tr>
<tr>
<td>200.0000</td>
</tr>
<tr>
<td>160.0000</td>
</tr>
</tbody>
</table>

Figure 6. Listing of printed output for Example Case 2.
Figure 7. Plotted output for Example Case 2.
Figure 8. Listing of input deck for Example Case 3.

<table>
<thead>
<tr>
<th>X-AXIS INCREMENT</th>
<th>Y-AXIS INCREMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>20.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X-AXIS LIMIT</th>
<th>Y-AXIS LIMIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.00</td>
<td>120.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X-AXIS ORIGIN</th>
<th>Y-AXIS ORIGIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>20.00</td>
</tr>
</tbody>
</table>

NCASES = 1

<table>
<thead>
<tr>
<th>TYPE( 1 )</th>
<th>TYPE( 2 )</th>
<th>TYPE( 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>C</td>
<td>H</td>
</tr>
</tbody>
</table>

***INPUT DATA***

<table>
<thead>
<tr>
<th>TEST CASE 3--CELL MEAN</th>
<th>TEST CASE 3--FREQUENCY</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CASE 1 OF 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
</tr>
<tr>
<td>2.0000</td>
</tr>
<tr>
<td>3.0000</td>
</tr>
<tr>
<td>4.0000</td>
</tr>
<tr>
<td>5.0000</td>
</tr>
<tr>
<td>6.0000</td>
</tr>
<tr>
<td>7.0000</td>
</tr>
<tr>
<td>8.0000</td>
</tr>
<tr>
<td>9.0000</td>
</tr>
<tr>
<td>10.0000</td>
</tr>
<tr>
<td>11.0000</td>
</tr>
<tr>
<td>12.0000</td>
</tr>
<tr>
<td>13.0000</td>
</tr>
<tr>
<td>14.0000</td>
</tr>
<tr>
<td>15.0000</td>
</tr>
<tr>
<td>16.0000</td>
</tr>
<tr>
<td>17.0000</td>
</tr>
<tr>
<td>18.0000</td>
</tr>
<tr>
<td>19.0000</td>
</tr>
</tbody>
</table>

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

TEST CASE 3—CELL MEAN

Figure 10. Plotted output for Example Case 3.
Figure 11. Listing of input deck for Example Case 4.
Figure 12. Listing of printed output for Example Case 4.
COEFFICIENTS FOR POLYNOMIAL OF DEGREE 10:

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:

<table>
<thead>
<tr>
<th>POINT</th>
<th>YCALC</th>
<th>DIFF**2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.000709</td>
<td>0.000000502803</td>
</tr>
<tr>
<td>2</td>
<td>9.947923</td>
<td>0.000004313213</td>
</tr>
<tr>
<td>3</td>
<td>9.819390</td>
<td>0.000003716019</td>
</tr>
<tr>
<td>4</td>
<td>9.615908</td>
<td>0.000004915439</td>
</tr>
<tr>
<td>5</td>
<td>9.335927</td>
<td>0.000016588000</td>
</tr>
<tr>
<td>6</td>
<td>8.988198</td>
<td>0.000003247140</td>
</tr>
<tr>
<td>7</td>
<td>8.572699</td>
<td>0.000007289361</td>
</tr>
<tr>
<td>8</td>
<td>8.092588</td>
<td>0.000006702353</td>
</tr>
<tr>
<td>9</td>
<td>7.544891</td>
<td>0.000026091918</td>
</tr>
<tr>
<td>10</td>
<td>6.946120</td>
<td>0.000015053024</td>
</tr>
<tr>
<td>11</td>
<td>6.294849</td>
<td>0.000023513143</td>
</tr>
<tr>
<td>12</td>
<td>5.595945</td>
<td>0.000035346223</td>
</tr>
<tr>
<td>13</td>
<td>4.845912</td>
<td>0.000016704003</td>
</tr>
<tr>
<td>14</td>
<td>4.067449</td>
<td>0.000006506254</td>
</tr>
<tr>
<td>15</td>
<td>3.258165</td>
<td>0.000003365552</td>
</tr>
<tr>
<td>16</td>
<td>2.424215</td>
<td>0.000017769932</td>
</tr>
<tr>
<td>17</td>
<td>1.562073</td>
<td>0.000004300971</td>
</tr>
<tr>
<td>18</td>
<td>0.697891</td>
<td>0.000004445361</td>
</tr>
<tr>
<td>19</td>
<td>-0.001492</td>
<td>0.000002266602</td>
</tr>
<tr>
<td>20</td>
<td>-1.049323</td>
<td>0.00000457832</td>
</tr>
<tr>
<td>21</td>
<td>-1.909124</td>
<td>0.000001767067</td>
</tr>
<tr>
<td>22</td>
<td>-2.754392</td>
<td>0.000031449202</td>
</tr>
<tr>
<td>23</td>
<td>-3.588137</td>
<td>0.000065218449</td>
</tr>
<tr>
<td>24</td>
<td>-4.385229</td>
<td>0.000027344544</td>
</tr>
<tr>
<td>25</td>
<td>-5.149389</td>
<td>0.00000372938</td>
</tr>
<tr>
<td>26</td>
<td>-5.875004</td>
<td>0.000024950876</td>
</tr>
<tr>
<td>27</td>
<td>-6.565761</td>
<td>0.000010490893</td>
</tr>
<tr>
<td>28</td>
<td>-7.189556</td>
<td>0.00000118064</td>
</tr>
<tr>
<td>29</td>
<td>-7.769007</td>
<td>0.000009885186</td>
</tr>
<tr>
<td>30</td>
<td>-8.290437</td>
<td>0.00000191150</td>
</tr>
<tr>
<td>31</td>
<td>-8.749856</td>
<td>0.000000206060</td>
</tr>
<tr>
<td>32</td>
<td>-9.143423</td>
<td>0.000011718138</td>
</tr>
<tr>
<td>33</td>
<td>-9.467486</td>
<td>0.000056053369</td>
</tr>
<tr>
<td>34</td>
<td>-9.693488</td>
<td>0.000042395776</td>
</tr>
<tr>
<td>35</td>
<td>-9.876957</td>
<td>0.000009259824</td>
</tr>
<tr>
<td>36</td>
<td>-9.980715</td>
<td>0.00000511452</td>
</tr>
<tr>
<td>37</td>
<td>-10.000982</td>
<td>0.00000964893</td>
</tr>
</tbody>
</table>

SUM OF SQUARES OF ERRORS FOR CURVE-FIT = 0.000513523724

Figure 12. (Concluded)
CONT. OPTION, NO DATA MARKS

TEST CASE 4 X-COORDINATE

Figure 13. Plotted output for Example Case 4.
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**

**X-AXIS LIMIT = 0.80**

**X-AXIS ORIGIN = 0.00**

**NCASES = 3**

**TYPE( 1) = C**

**TYPE( 2) = C**

**TYPE( 3) = C**

---

***INPUT DATA***

**TEST CASE 5 X-COORDINATE**

<table>
<thead>
<tr>
<th>X-Coordinate</th>
<th>Y-Coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0400</td>
<td>0.0016</td>
</tr>
<tr>
<td>0.0600</td>
<td>0.0088</td>
</tr>
<tr>
<td>0.0800</td>
<td>0.0275</td>
</tr>
<tr>
<td>0.1000</td>
<td>0.0633</td>
</tr>
<tr>
<td>0.1200</td>
<td>0.1189</td>
</tr>
<tr>
<td>0.1400</td>
<td>0.1933</td>
</tr>
<tr>
<td>0.1600</td>
<td>0.2825</td>
</tr>
<tr>
<td>0.1800</td>
<td>0.3802</td>
</tr>
<tr>
<td>0.2000</td>
<td>0.4797</td>
</tr>
<tr>
<td>0.2200</td>
<td>0.5750</td>
</tr>
<tr>
<td>0.2400</td>
<td>0.6618</td>
</tr>
<tr>
<td>0.2600</td>
<td>0.7374</td>
</tr>
<tr>
<td>0.2800</td>
<td>0.8008</td>
</tr>
<tr>
<td>0.3000</td>
<td>0.8521</td>
</tr>
<tr>
<td>0.3200</td>
<td>0.8924</td>
</tr>
<tr>
<td>0.3400</td>
<td>0.9233</td>
</tr>
<tr>
<td>0.3600</td>
<td>0.9463</td>
</tr>
<tr>
<td>0.3800</td>
<td>0.9630</td>
</tr>
<tr>
<td>0.4000</td>
<td>0.9750</td>
</tr>
<tr>
<td>0.4200</td>
<td>0.9833</td>
</tr>
<tr>
<td>0.4400</td>
<td>0.9891</td>
</tr>
<tr>
<td>0.4600</td>
<td>0.9930</td>
</tr>
<tr>
<td>0.4800</td>
<td>0.9956</td>
</tr>
</tbody>
</table>

**Figure 15.** Listing of printed output for Example Case 5.
### Coefficients for Polynomial of Degree 9

-0.039928562066
+2.680569725111
-65.717747807506
+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

<table>
<thead>
<tr>
<th>Point</th>
<th>YCALC</th>
<th>DIFF**2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001927</td>
<td>0.000000082703</td>
</tr>
<tr>
<td>2</td>
<td>0.008772</td>
<td>0.00000003316</td>
</tr>
<tr>
<td>3</td>
<td>0.026791</td>
<td>0.000000621963</td>
</tr>
<tr>
<td>4</td>
<td>0.063058</td>
<td>0.00000091133</td>
</tr>
<tr>
<td>5</td>
<td>0.119515</td>
<td>0.000000354491</td>
</tr>
<tr>
<td>6</td>
<td>0.194297</td>
<td>0.000000841481</td>
</tr>
<tr>
<td>7</td>
<td>0.283006</td>
<td>0.00000217405</td>
</tr>
<tr>
<td>8</td>
<td>0.379886</td>
<td>0.00000111237</td>
</tr>
<tr>
<td>9</td>
<td>0.478844</td>
<td>0.00000766828</td>
</tr>
<tr>
<td>10</td>
<td>0.574275</td>
<td>0.00000647926</td>
</tr>
<tr>
<td>11</td>
<td>0.661666</td>
<td>0.00000456999</td>
</tr>
<tr>
<td>12</td>
<td>0.737961</td>
<td>0.00000231697</td>
</tr>
<tr>
<td>13</td>
<td>0.801681</td>
<td>0.00000691816</td>
</tr>
<tr>
<td>14</td>
<td>0.852820</td>
<td>0.00000436204</td>
</tr>
<tr>
<td>15</td>
<td>0.892541</td>
<td>0.00000005145</td>
</tr>
<tr>
<td>16</td>
<td>0.922735</td>
<td>0.00000330597</td>
</tr>
<tr>
<td>17</td>
<td>0.945499</td>
<td>0.00000640990</td>
</tr>
<tr>
<td>18</td>
<td>0.962653</td>
<td>0.00000157419</td>
</tr>
<tr>
<td>19</td>
<td>0.975387</td>
<td>0.00000142702</td>
</tr>
<tr>
<td>20</td>
<td>0.984216</td>
<td>0.000000716604</td>
</tr>
<tr>
<td>21</td>
<td>0.989394</td>
<td>0.00000069805</td>
</tr>
<tr>
<td>22</td>
<td>0.992002</td>
<td>0.000001015472</td>
</tr>
<tr>
<td>23</td>
<td>0.995954</td>
<td>0.000000125563</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>X-COORDINATE</th>
<th>Y-COORDINATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0102</td>
<td></td>
</tr>
<tr>
<td>0.0287</td>
<td></td>
</tr>
<tr>
<td>0.0583</td>
<td></td>
</tr>
<tr>
<td>0.1004</td>
<td></td>
</tr>
<tr>
<td>0.1552</td>
<td></td>
</tr>
<tr>
<td>0.2217</td>
<td></td>
</tr>
<tr>
<td>0.2973</td>
<td></td>
</tr>
<tr>
<td>0.3785</td>
<td></td>
</tr>
<tr>
<td>0.4619</td>
<td></td>
</tr>
<tr>
<td>0.5436</td>
<td></td>
</tr>
<tr>
<td>0.6207</td>
<td></td>
</tr>
<tr>
<td>0.6909</td>
<td></td>
</tr>
<tr>
<td>0.7530</td>
<td></td>
</tr>
<tr>
<td>0.8062</td>
<td></td>
</tr>
<tr>
<td>0.8506</td>
<td></td>
</tr>
<tr>
<td>0.8868</td>
<td></td>
</tr>
<tr>
<td>0.9155</td>
<td></td>
</tr>
<tr>
<td>0.9380</td>
<td></td>
</tr>
<tr>
<td>0.9552</td>
<td></td>
</tr>
<tr>
<td>0.9681</td>
<td></td>
</tr>
<tr>
<td>0.9776</td>
<td></td>
</tr>
<tr>
<td>0.9846</td>
<td></td>
</tr>
<tr>
<td>0.9895</td>
<td></td>
</tr>
<tr>
<td>0.9931</td>
<td></td>
</tr>
<tr>
<td>0.9955</td>
<td></td>
</tr>
</tbody>
</table>

**USERS MESSAGE**  THE POLYNOMIAL CALCULATED MAY OR MAY NOT BE THE BEST FITTED CURVE

Figure 15.  (Continued).
• COEFFICIENTS 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

<table>
<thead>
<tr>
<th>POINT</th>
<th>YCALC</th>
<th>DIFF**2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.011873</td>
<td>0.000002733528</td>
</tr>
<tr>
<td>2</td>
<td>0.026310</td>
<td>0.000006146741</td>
</tr>
<tr>
<td>3</td>
<td>0.056916</td>
<td>0.000001969616</td>
</tr>
<tr>
<td>4</td>
<td>0.101147</td>
<td>0.00000558885</td>
</tr>
<tr>
<td>5</td>
<td>0.157209</td>
<td>0.000003763071</td>
</tr>
<tr>
<td>6</td>
<td>0.223401</td>
<td>0.000002793994</td>
</tr>
<tr>
<td>7</td>
<td>0.297697</td>
<td>0.00000157887</td>
</tr>
<tr>
<td>8</td>
<td>0.377576</td>
<td>0.000001006416</td>
</tr>
<tr>
<td>9</td>
<td>0.460065</td>
<td>0.000003477294</td>
</tr>
<tr>
<td>10</td>
<td>0.541940</td>
<td>0.000002853150</td>
</tr>
<tr>
<td>11</td>
<td>0.620046</td>
<td>0.00000509448</td>
</tr>
<tr>
<td>12</td>
<td>0.691831</td>
<td>0.00000425034</td>
</tr>
<tr>
<td>13</td>
<td>0.754671</td>
<td>0.000002630334</td>
</tr>
<tr>
<td>14</td>
<td>0.808083</td>
<td>0.000003472011</td>
</tr>
<tr>
<td>15</td>
<td>0.851795</td>
<td>0.000001312294</td>
</tr>
<tr>
<td>16</td>
<td>0.886659</td>
<td>0.00000019819</td>
</tr>
<tr>
<td>17</td>
<td>0.914176</td>
<td>0.000001914500</td>
</tr>
<tr>
<td>18</td>
<td>0.936084</td>
<td>0.000003901386</td>
</tr>
<tr>
<td>19</td>
<td>0.953881</td>
<td>0.000001901406</td>
</tr>
<tr>
<td>20</td>
<td>0.968385</td>
<td>0.00000046550</td>
</tr>
<tr>
<td>21</td>
<td>0.979562</td>
<td>0.0000002508122</td>
</tr>
<tr>
<td>22</td>
<td>0.986698</td>
<td>0.0000004279220</td>
</tr>
<tr>
<td>23</td>
<td>0.989476</td>
<td>0.000000012949</td>
</tr>
<tr>
<td>24</td>
<td>0.990075</td>
<td>0.0000009207652</td>
</tr>
<tr>
<td>25</td>
<td>0.996855</td>
<td>0.0000001704946</td>
</tr>
</tbody>
</table>

SUM OF SQUARES OF ERRORS FOR CURVE-FIT = 0.000060306266

Figure 15. (Continued).
## ***INPUT DATA***

### TEST CASE 5 X-COORDINATE

<table>
<thead>
<tr>
<th>X-COORDINATE</th>
<th>Y-COORDINATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2200</td>
<td>0.0119</td>
</tr>
<tr>
<td>0.2400</td>
<td>0.0300</td>
</tr>
<tr>
<td>0.2600</td>
<td>0.0556</td>
</tr>
<tr>
<td>0.2800</td>
<td>0.0896</td>
</tr>
<tr>
<td>0.3000</td>
<td>0.1323</td>
</tr>
<tr>
<td>0.3200</td>
<td>0.1833</td>
</tr>
<tr>
<td>0.3400</td>
<td>0.2418</td>
</tr>
<tr>
<td>0.3600</td>
<td>0.3061</td>
</tr>
<tr>
<td>0.3800</td>
<td>0.3742</td>
</tr>
<tr>
<td>0.4000</td>
<td>0.4441</td>
</tr>
<tr>
<td>0.4200</td>
<td>0.5136</td>
</tr>
<tr>
<td>0.4400</td>
<td>0.5806</td>
</tr>
<tr>
<td>0.4600</td>
<td>0.6437</td>
</tr>
<tr>
<td>0.4800</td>
<td>0.7015</td>
</tr>
<tr>
<td>0.5000</td>
<td>0.7537</td>
</tr>
<tr>
<td>0.5200</td>
<td>0.7996</td>
</tr>
<tr>
<td>0.5400</td>
<td>0.8396</td>
</tr>
<tr>
<td>0.5600</td>
<td>0.8724</td>
</tr>
<tr>
<td>0.5800</td>
<td>0.9001</td>
</tr>
<tr>
<td>0.6000</td>
<td>0.9228</td>
</tr>
<tr>
<td>0.6200</td>
<td>0.9410</td>
</tr>
<tr>
<td>0.6400</td>
<td>0.9554</td>
</tr>
<tr>
<td>0.6600</td>
<td>0.9667</td>
</tr>
<tr>
<td>0.6800</td>
<td>0.9754</td>
</tr>
<tr>
<td>0.7000</td>
<td>0.9820</td>
</tr>
<tr>
<td>0.7200</td>
<td>0.9869</td>
</tr>
<tr>
<td>0.7400</td>
<td>0.9906</td>
</tr>
<tr>
<td>0.7600</td>
<td>0.9933</td>
</tr>
<tr>
<td>0.7800</td>
<td>0.9953</td>
</tr>
<tr>
<td>0.8000</td>
<td>0.9967</td>
</tr>
</tbody>
</table>

### CASE 3 OF 3

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*

<table>
<thead>
<tr>
<th>POINT</th>
<th>YCALC</th>
<th>DIFF**2</th>
<th>SUM OF SQUARES OF ERRORS FOR CURVE-FIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.012745</td>
<td>0.0000000601850</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.029356</td>
<td>0.0000000509098</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.054756</td>
<td>0.0000000819690</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.089296</td>
<td>0.0000001468855</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.132728</td>
<td>0.0000001510464</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.184288</td>
<td>0.0000000899720</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.242783</td>
<td>0.0000000816801</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.306682</td>
<td>0.0000000262453</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.374215</td>
<td>0.0000000020240</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.443483</td>
<td>0.0000000444748</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.512575</td>
<td>0.0000001050407</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.579674</td>
<td>0.0000000933148</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.643161</td>
<td>0.0000000289499</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.701712</td>
<td>0.0000000150445</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.754354</td>
<td>0.0000000341927</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.800516</td>
<td>0.0000000821663</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.840033</td>
<td>0.0000000889492</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.873110</td>
<td>0.0000000423290</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.900291</td>
<td>0.0000000123499</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.922329</td>
<td>0.0000000221075</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.940093</td>
<td>0.0000000822142</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0.954434</td>
<td>0.0000000990398</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.966066</td>
<td>0.0000000414041</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.975456</td>
<td>0.0000000032233</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.982819</td>
<td>0.0000000639291</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.988129</td>
<td>0.0000001321069</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>0.991391</td>
<td>0.0000000464910</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>0.992795</td>
<td>0.0000000341957</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0.993799</td>
<td>0.0000002402732</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.997637</td>
<td>0.0000000752988</td>
<td></td>
</tr>
</tbody>
</table>

SUM OF SQUARES OF ERRORS FOR CURVE-FIT = 0.000017800973

Figure 15. (Concluded).
MULTIPLE CONTINUOUS OPTION

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 POLYNOMIAL 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.
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: $20 \times 10^{-1}$).

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 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 \((x_i < x_{i+1} < x_{i+2} < x_{i+3}, \text{ etc})\).

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.
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. 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 $x_1, y_1; x_2, y_2; x_3, y_3; \ldots x_n, y_n$; 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:

$$y_m(x_i) = b_0 + b_1 x_i + b_2 x_i^2 + \ldots + b_m x_i^m$$

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.
PROGRAM GECAP IS A GENERALISED CURVE-FIT AND PLOT PROGRAM FOR USE IN PLOTTING A WIDE VARIETY OF DATA ON A GRID SMALL ENOUGH TO FIT ON AN 8 1/2 X 11 SHEET OF PAPER. NUMEROUS OPTIONS ARE SUPPLIED TO THE USER IN ORDER FOR HIM TO OBTAIN A PLOT OF HIS DATA SET.

OPTIONS:

1 - EITHER OR BOTH OF THE AXIS ON THE GRID MAY BE NUMB-ERED IN EITHER INTEGER OR REAL NUMBER FORM.

2 - THE DATA MAY BE PLOTTED IN ANY ONE OF FOUR WAYS
   (1) DISCRETELY - THE DATA IS PLOTTED AS INDIVIDUAL POINTS WITH THE POINT SYMBOLS DIFFERING FROM CASE TO CASE.
   (2) LINEARLY - THE DATA POINTS ARE CONNECTED BY STRAIGHT LINES.
   (3) CONTINUOUSLY - THE INDIVIDUAL POINTS ARE ORIGINALLY PLOTTED AS THE DISCRETE POINTS BUT THE POINTS UNDER GO A LEAST-SQUARES CURVE-FIT AND THE CALCULATED FUNCTION IS PLOTTED.
   (4) HISTOGRAM - THE DATA POINTS ARE NOT PLOTTED INDIVIDUALLY BUT A HISTOGRAM IS GENERATED FROM THE INPUT DATA.

3 - MULTIPLE CASES MAY BE PLOTTED ON THE SAME GRID WITHOUT INPUTTING ADDITIONAL GRID INFORMATION. THIS OPTION DOES NOT ALLOW FOR DIFFERENT TYPE PLOTS ON THE SAME GRID.

-NOTE--AN OPTION FOR NO DATA POINT MARKS IS AVAILABLE FOR ANY TYPE OF PLOT.
-NOTE--IF DATA POINT MARKS ARE USED, AN OPTION MAY BE USED TO LABEL EACH CURVE INDIVIDUALLY.
-NOTE--IF THE USER DESIRES TO SHOW THAT HIS LABELED VALUES ARE RAISED TO A POWER OF 10, AN OPTION EXISTS WHICH LABELS THE LAST 'TIC' MARK ON THE AXIS IN SUCH A MANNER TO INDICATE HOW THE VALUES SHOULD BE READ.

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(10),YVALU(10)
COMMON XI,XMAX,XINC,XSTRT
COMMON YSTRT,YSCALE,YSCALE
COMMON NCASE,ICASE
COMMON NODAT
DATA CONT/'C'/
DATADESC/'D'/
DATALINE/'L'/
DATAMUST/'X'/
DATAZCHV/'END'/
DATABLACK/'1'/
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 X, Y
C C*** THE CURVES, AND THE EXONENTS FOR THE X AND Y AXES
C
10 READ(R*20)TITLZ,ICASE,A,NCASE,(XNAME(I),I=1,5),TEN(1),TEN(2)
20 FORMAT(A4*1X,11I13*13*525*12*12)
30 IF (ICASE=1)40,100,30
40 ERN=1
CALL ERRRO(TYPE,ERR,K,YINC)
50 CONTINUE
60 IF (ICASE=1)70,100,60
70 WRITE(1*80)
80 FORMAT(1X,15A10,y X-AXIS LIMIT = ',F10.2,6X,Y-AXIS LIMIT = ',F10.2,6X)
90 FORMAT(6F10.0,4A1)

C C*** THIS IS THE CONSOLE MESSAGE FOR THE USER
C
70 WRITE(1*80)
80 FORMAT(1X,15A10,y X-AXIS LIMIT = ',F10.2,6X,Y-AXIS LIMIT = ',F10.2,6X)
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(6F10.0,4A1)
CHECK THE VALUES FOR LABELING THE AXES TO SEE IF THE STARTING POINTS ARE LESS THAN THE END POINTS

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)

CONVERT ALPHA-NUMERIC INPUT DATA TO NUMERIC VALUES

DO 260 K=1, 3
   WRITE(K), TYPE(K)
170 FORMAT(1X, 'TYPE', 4X, '1I2')
   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

READ IN TITLES FOR THE X AND Y AXIS AND ALSO THE GRAPH TITLE

READ (R, 270) (TITLX(I), I=1,7), (TITLY(J), J=1,7), (TITLE(K), K=1,7)
270 FORMAT (6A* A2, 6A* 4, A2, 7A4)
280 CONTINUE

THIS SECTION COMPUTES THE VALUES FOR THE GRID FROM THE SPECIFIED USER INFORMATION.

INX = XINC*10.
INCY = YINC*10.
IXI = INX/10
IYI = INCY/10
IF ((IXI*10) = INCY) 300, 290, 300
290 IF ((IYI*10) = INXC) 310, 320, 310
300 IYI = IYI + 1
   GO TO 290
310 IXI = IXI + 1
320 NXTCS = (XMAX-XSTRT)/INX

THIS IF TEST LIMITS THE NUMBER OF 'TIC' MARKS FOR THE X-AXIS

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 = 201360,360,350
350 ERR=7
CALL ERROIOTYPE,ERR,K,YINC)
GO TO 280
360 CONTINUE
C
C IF(NYTC3 = 201360,360,350
370 CALL SCALE(XSCLE,YSCLE,XSTRT,YSTRT)
C
C*** READ IN DATA POINTS AND PLOT THEM
C
380 KI=1
CALL EPL01(1,XSTRT,YSTRT)
390 READ(R=390)XVALU(KI),YVALU(KI),XCH
390 FORMAT(2F10.3,1A3)
400 IF(XCH='XCHV') THEN
400 IF(YVALU(KI)-YSTRT)440,410,410
410 IF(YVALU(KI)-YMAX)420,430,420
420 IF(XVALU(KI)-XSTRT)440,430,440
430 IF(XVALU(KI)-XMAX)450,450,440
440 ERR=9
CALL ERROIOTYPE,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 ERROIOTYPE,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(540,550),JFK
530 ICODE = 2
540 JG=KJ-1
C
C---------------------------
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)640,570,600
570 IF(LBJ=1600)580,590
580 CALL EPL0T(1,XVALU(LBJ),YVALU(LBJ))
590 CALL EPL0T(2,XVALU(LBJ),YVALU(LBJ))
600 IF(NODAT=BLNK)630,610,630
610 IF(ICODE=1)
620 CALL POINT(II)
630 CONTINUE
C
C*** LABEL THE CURVE

C----------------------------------
DO 640 O=15
IF(XNAME(I)-ULANK)=1650 640 CONTINUE
GO TO 730
650 X=(XMAX)-(XMAX-XSTRT)/(4.5)
660 Y=YSTRT-(YSCLE)/(YSCLE)-Y/(YSCLE)
670 CALL EPLT 1 (X,Y)
680 CALL POINT 11
690 X=(XMAX)-(XMAX-XSTRT)/(5.0)
Y=Y-.05/(YSCLE)
700 CALL ECHAR X,Y,1,1,0,0
710 WRITE (7,720) (XNAME(I)+O,5)
720 FORMAT(A6)
730 WRITE(P,740) TITLX,TITLY
740 FORMAT(A,14X,***INPUT DATA***,1X,B4,A2,4X,A4,A2,4X)
WRITE(P,750) TITLX,ICASE,ANCASE
750 FORMAT(A,20X,A4,12X,A2,12)
760 K1=KI-1
DO 780 I=KI,1
WRITE(P,770) XVALU(I),YVALU(I)
770 FORMAT(F10.4,F10.4)
780 CONTINUE
KI=KI+1
CALL EPLT (1,XSTRT,YSTRT)
790 CONTINUE
IF(ICODE=2) 800,810,810
800 CALL FINSH
810 CALL LINK(BESFT)
820 IF(4CASE-ICASE) 820,820,10
C*** SET THE PEN FOR THE NEXT PLOT
C
820 CALL EPLT (1,XMAX,YSTRT)
CALL SCALE (1.0,1.0,0.0,0.0)
CALL EPLT (1,4.0,0.0,0.0)
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
GECAP REVISION C, 09/05/72
C
INTEGER TYPE,TEN
COMMON TITLE(7),TITLY(7),TITLE(7)
COMMON XVAL1(101),YVAL1(101)
COMMON K1*MAX,MAX,INC*XSTART
COMMON YSTART,XSCALE,YSCALE
COMMON NCASE,ICASE
COMMON NODAT
C
C*** THE LENGTH OF THE X AND Y AXES ARE GOVERNED BY THE NEXT TWO STATEMENTS.
C*** MENTS SO THAT THE ACTUAL LENGTH OF THE AXIS (IN INCHES) WILL BE
C*** IN THE DIVIDEND OF THE STATEMENTS.
C
XSCALE = 8.5/(XMAX - XSTART)
YSCALE = 6.0/(YMAX - YSTART)
YINT = -.32/YSCALE
XINT = -.32/XSCALE
C
C*** DRAW THE GRID
C
CALL EPL0T(1,0,0,0)
CALL EPL0T(2,0,0,0)
CALL SCALE(XSCALE,YSCALE,0,0)
CALL EGRID(1,0,0,0,0,YINC,NYTCS)
CALL EPL0T(1,0,0,0,0)
CALL EPL0T(2,0,0,0,0)
CALL EGRID(0,0,0,0,XINC,NXTCS)
C
C*** LABEL THE GRID
C
NXTCS = NXTCS + 1
NYTCS = NYTCS + 1
X = XINT
XNUM = XSTART
INIT = XSTART
NUM = XSTART +10
IF (TYPE(3)>0) X = XINT + (XINC)/(2.0)
NXTC5=NXTCS-1
20 DO 80 I=1,NXTCS
   CALL ECHAR(X*YINT+1*1+I*0,0)
   IF TYPE(1)=2)60+30*60
   30 WRITE(7,*40)XINT
   40 FORMAT(14)
   50 FORMAT(15)
   INIT=INIT+I XI
   GO TO 70
   60 WRITE(7,*520)XNUM
   70 CONTINUE

45
C*** PLACE EXPONENT AT THE END OF THE X AXIS
C
90 IF(TEN(1)<190*270*90
100 X=XMAX*2/XSCALE=XSTRT
110 X=X-XINC/2.
120 CALL ECHAR (X*Y+1*1*10*0)
130 WRITE (7*300)
140 IF(TEN(1)=10150*160*160
150 X=X-XINC/2.
160 Y=Y+0.05/YSCALE
CALL ECHAR (X*Y+1*1*10*0)
WRITE (7*310)TEN(1)
GO TO 270
170 X=XMAX*4/XSCALE-XSTRT
Y=YINT
IF(TYPE(3)<-4190*180*190
180 X=X-XINC/2.
190 CALL ECHAR (X*Y+1*1*10*0)
200 WRITE (7*300)
210 X=X+4/XSCALE
IF(TEN(1)>=1240*240*220
220 IF(TEN(1)<-101230*240*240
230 X=X+1/XSCALE
240 Y=Y+0.05/YSCALE
250 CALL ECHAR (X*Y+1*1*10*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
280 XINT=-.6/XSCALE
IF(TYPE(2)<-280*290*280
290 Y=0.0
300 INIT = YSTRT
NUM = YSTRT
XNUM= YSTRT
DO 540 I=1,NYTCS
300 FORMAT(1*10')
310 FORMAT(12)
320 IF(NYTCS<1)490*330*490
330 IF(TEN(2)<1)340*490*460
340 FF=0.0
350 X=-.7/XSCALE+FF
360 CALL ECHAR (X*Y+1*1*10*0)
370 WRITE (7*300)
380 X=-.3/XSCALE
390 ZZ=Y+0.05/YSCALE
400 CALL ECHAR (X*ZZ+1*1*10*0)
410 WRITE (7*310)TEN(2)
420 XINT=XINT-.7/XSCALE+FF
430 IF (TYPE(2) = 2) GO TO 490
440 XINT = XINT + 0.5 / XSCLE
450 GO TO 490
460 IF (NEN(2) = 1) GO TO 340
470 FF = 1 / XSCLE
480 GO TO 350
490 CONTINUE
C
CALL ECHAR(XINT, YINT, 'l'.1,.10,0,0)
CALL ECHAR(XINT, YINT, 't'.1,.10,0,0)
500 WRITE(7, 50) INIT
INIT = INIT + 1
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 = -6.66 / XSCLE
IF (TYPE(2) = 2) GO TO 560
560 X = -7.3 / 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) (TITLYN(N), N = 1, 7)
CALL EPLOT(1.0, 0.0)
CALL EPLOT(2.0, 0.0)
RETURN
END
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 ERR0(TYPE, ERR, K, YINC)
C
GECAP REVISION C, 09/05/72
C
INTEGER TYPE(3), ERR
COMMON TITLX(7), TITYL(7), TITLE(7)
COMMON XVALU(101), YVALU(101)
COMMON K1, XMAX, YMAX, XINC, XSTRT
COMMON YSTRT, XSCLE, YSICLE
COMMON NCASE, ICASE
COMMON NODAT
GO TO (10, 30, 50, 70, 210, 280, 250, 270, 300, 300, 300, 280, 300, 250, 300, 270, 300, 210, 300, 280, 300, 250, 300, 270, 300)
10 WRITE (1, 20)
20 FORMAT ('**ERROR** CASE CARD INPUT VARIABLE IS INCORRECT *ERROR0017*
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 PERRO0220
10INT****', '/**EXECUTION DISCONTINUED****')
   GO TO 320
50 WRITE(1, 60)
60 FORMAT ('**ERROR** END POINT OF Y AXIS IS LESS THAN STARTING PERRO0260
10INT****', '/**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
   NXI=IFIX(XINC)
   CNX=(XINC-NXI)*(100.)
   NCX=IFIX(CNX)
   IF(CNX-NCX)80, 90, 80
   WRITE(1, 190)
   PAUSE
90 NCY=IFIX(YINC)
   CNY=(YINC-NCY)*(100.)
   NCY=IFIX(CNY)
   IF(CNY-NCY)100, 110, 100
   WRITE(1, 190)
   PAUSE
100 NCX=IFIX(XSTRT)
   XXX=(XSTRT-NCX)*(100.)
   IXX=IFIX(XXX)
   IF(IXX=IXX)120, 130, 120
110 WRITE(1, 190)
   PAUSE
120 NYY=IFIX(YSTRT)
   YYY=(YSTRT-NYY)*(100.)
   IYY=IFIX(YYY)
   IF(IYY=IYY)140, 150, 140
130 WRITE(1, 190)
   PAUSE
140 WRITE(1, 190)

48
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 ARE NOT
1**REQUIRE MORE PRECISION THAN THAT ALLOWABLE IN GECAP***ERROR0700
2**CHECK GECAP USER MANUAL FOR DETAILS***ERROR0710
3**BE INsignificant IN SPECIAL CASES**IF EXECUTION***ERROR0730
4**RED PRESS START**IF NOT PRESS STOP**ERROR0740
200 GO TO 320
210 WRITE(1,220)K
220 FORMAT(1X)ERROR**INCORRECT PLOT OPTION WAS USED FOR**ERROR0770
1**CHECK GECAP USER MANUAL FOR DETAILS**ERROR0780
2**TYPE IN THE DESIRED OPTION FROM THE CONSOLE TYPEWRITER ERROR0790
3**PRESS (EOF) BUTTON TO CONTINUE EXECUTION**ERROR0800
4**ERROR0810
230 READ (6,240)TYPE(K)
240 FORMAT(A1)
K=K-1
GO TO 320
250 WRITE(1,260)
260 FORMAT(2X,***ERROR*** NUMBER OF TIC MARKS EXCEEDS LEGIBILITY LIMIT**ERROR0870
1**STANDARD FIX-UP TAKEN; EXECUTION CONTINUING**ERROR0880
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(1X,*** ERROR: MORE THAN 100 DATA POINTS ON INPUT ***ERROR0900
1**EXECUTION CONTINUING WITHOUT REMAINING POINTS**ERROR0910
GO TO 320
300 WRITE(1,310)
310 FORMAT(2X,***ERROR***A DATA POINT WAS FOUND TO EXCEED THE LIMIT**ERROR1010
1**OF THE AXES***EXECUTION DISCONTINUED**ERROR1020
GO TO 330
320 RETURN
330 STOP
END
//DUP
*DELETE WS UA ERROR
*STORE WS UA ERROR
SUBROUTINE HIST

COMMON TITLX(7), TITLY(7), TITLE(7)
COMMON XVALU(101), YVALU(101)
COMMON XMAX, YMAX, XINC, XSTRT
COMMON YSTRT, XSCLE, YSCLE
COMMON NCASE, ICASE

NPTS = 1
20 Z = XVALU(2) - XVALU(1)
30 IF (XVALU(1) - XSTRT) < 60.40
40 CALL EPL0T (1, XVALU(1), YVALU(1))
50 CALL EPL0T (2, XVALU(1), YSTRT)
60 DO 100 I = 1, NPTS
70 CALL EPL0T (1, XVALU(I), YVALU(I))
80 PLUS = XVALU(I) + Z
90 CALL EPL0T (2, PLUS, YVALU(I))
100 CONTINUE
110 CALL EPL0T (1, XSTRT, YSTRT)
120 NPTS = NPTS - 1
130 DO 200 I = 1, NPTS
140 IF (YVALU(I) - YVALU(I + 1)) < 150, 150
150 CALL EPL0T (1, XVALU(I + 1), YVALU(I))
160 CALL EPL0T (2, XVALU(I + 1), YSTRT)
170 GO TO 200
180 CALL EPL0T (1, XVALU(I + 1), YVALU(I + 1))
190 CALL EPL0T (2, XVALU(I + 1), YSTRT)
200 CONTINUE
210 CALL EPL0T (1, PLUS, YVALU(I))
220 CALL EPL0T (2, PLUS, YSTRT)
230 RETURN

END

// FOR
* ONE WORD INTEGERS
* EXTENDED PRECISION
C
C*** THIS SUBROUTINE GENERATES A HISTOGRAM PLOT FROM THE INPUT DATA
C
SUBROUTINE HIST
C
C*** GECCP REVISION C, 09/05/72
C
COMMON TITLX(7), TITLY(7), TITLE(7)
COMMON XVALU(101), YVALU(101)
COMMON XMAX, YMAX, XINC, XSTRT
COMMON YSTRT, XSCLE, YSCLE
COMMON NCASE, ICASE
NPTS = 1
20 Z = XVALU(2) - XVALU(1)
30 IF (XVALU(1) - XSTRT) < 60.40
40 CALL EPL0T (1, XVALU(1), YVALU(1))
50 CALL EPL0T (2, XVALU(1), YSTRT)
60 DO 100 I = 1, NPTS
70 CALL EPL0T (1, XVALU(I), YVALU(I))
80 PLUS = XVALU(I) + Z
90 CALL EPL0T (2, PLUS, YVALU(I))
100 CONTINUE
110 CALL EPL0T (1, XSTRT, YSTRT)
120 NPTS = NPTS - 1
130 DO 200 I = 1, NPTS
140 IF (YVALU(I) - YVALU(I + 1)) < 150, 150
150 CALL EPL0T (1, XVALU(I + 1), YVALU(I))
160 CALL EPL0T (2, XVALU(I + 1), YSTRT)
170 GO TO 200
180 CALL EPL0T (1, XVALU(I + 1), YVALU(I + 1))
190 CALL EPL0T (2, XVALU(I + 1), YSTRT)
200 CONTINUE
210 CALL EPL0T (1, PLUS, YVALU(I))
220 CALL EPL0T (2, PLUS, YSTRT)
230 RETURN

END

// DUP
* DELETE
* STORE WS UA HIST
FOR

• I/OCS (1132 PRINTER, KEYBOARD, TYPEWRITER, CARD, PLOTTER)

• EXTENDED PRECISION

• ONE WORD INTEGERS

C

*** GECAP REVISION C, 09/05/72

***

PROGRAM BESTFT WAS DESIGNED TO SUPPLY A BEST-FIT LEAST SQUARES FUNCTION TO A GIVEN SET OF INPUT DATA. IT WAS DESIGNED TO BE USED EXCLUSIVELY WITH PROGRAM GECAP. THE TWO MAINLINES ARE TIED TOGETHER BY THE USE OF SYSTEM ROUTINE LINK.

INTEGER P,R

DIMENSION A(12,13), SUM(20), WORK(20), IHLD(20), V(10)

DIMENSION F1(49), F2(49)

COMMON TITLX(7), TITY(7), TITLE(7)

COMMON XI(101), Y(101)

COMMON XMAX, YMAX, XINC, YINC, XSTRT

COMMON YSTRT, XSCALE, YSCALE

COMMON NCASE, ICASE

DATA A/156*0./

DATA F1/161., 18.5, 10.1, 7.7, 6.6, 5.99, 5.59, 5.32, 5.12, 4.96, 4.84, 4.75/

DATA F2/200., 19.0, 18.5, 17.4, 16.4, 15.4, 14.4, 13.4, 12.4, 11.4, 10.5, 9.5/

ERROR 0.

SAVE1 0.

SAVE2 9999999.

SAVE3 9999999.

NSUB1 0

NSUB2 0

NX = 1

NCOUT = 0

NCT = 0

P = 3

R = 2

10 CONTINUE

CALL SCALE (XSCALE, YSCALE, XSTRT, YSTRT)

BEGIN CURVE-FIT ITERATION

20 NX = NX + 1

NY = NY + 1

LIMIT THE DEGREE OF POLYNOMIAL TO ONE LESS THAN NUMBER OF POINTS ON INPUT

IF (NX = K1) 30, 200, 200

30 IF (NX = 9) 40, 60, 200

40 CALL CURF (X, Y, K1, NX, NY, A, SUM, V, WORK, IHLD, E)

IF (E) 70, 70, 50

50 WRITE (P, 60)
60 FORMAT(/,'***ERROR*** ERROR IN ROUTINE CURF. EXECUTION CONTINUES')
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/(K-1-(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 = FL(1)) 150, 140, 0
   NCOUT = 0
140 GO TO 160
   NCOUT = 1
150 GO TO 20
160 IF(TEST2 = FL(2)) 180, 170, 170
   NCT = NCOUT
   GO TO 20
170 IF(NCT = 1) 170, 190, 190
180 NSUB1 = NX
190 NSUB2 = NX + 1
   GO TO 220
C
C*** COMPUTE THE BEST-FIT CO-EFFICIENTS
C
200 WRITE(P,210)
210 FORMAT(IX,#*USERS MESSAGE*** THE POLYNOMIAL CALCULATED MAY OR MAY NOT BE THE BEST FITTED CURVE')
220 CALL CURF(X,Y,KI,NSUB1,NSUB2,A,SUM,V,WORK,IML,I,E)
230 WRITE(P,230)
230 FORMAT(IH1)
240 FORMAT(/,'#CO-EFFICIENTS FOR POLYNOMIAL OF DEGREE',13,'**')
   DO 260 K = 1, NSUB2
      J = NSUB2 + 1 - K
      WRITE(P,250) A(J+1)
   250 FORMAT(F20.12)
   CONTINUE
260 WRITE(P,270)
270 FORMAT(/,'####')
280 FORMAT(##*CALCULATED VALUES AND SUM OF ERRORS SQUARED FOR##')
   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.
300 CONTINUE
WRITE(1,290) YCALC, SIGSQ

290 FORMAT(' FOR POINT ', 13, ' YCALC = ', F20.6, '10X', 'DIFF**2 = ', F20.12)

300 ERROR = ERROR + SIGSQ

WRITE(1,310) ERROR

310 FORMAT(' SUM OF SQUARES OF ERRORS FOR CURVE-FIT = ', F20.12)

C

C*** PLOT THE CALCULATED FUNCTION ON THE SUPPLIED GRID (SEE MAINLINE)

C

XBEGN = X(1)
STEP = XBEGN
YI = F(A, NSUB1, STEP)
CALL EPlot(1, STEP, YI)
STEP = +1 * XINC
XLAST = X(KI)
XI = XBEGN

320 XI = XI + STEP
IF (XLAST = XI) 330, 330, 340

330 XI = XLAST

340 YI = F(A, NSUB1, XI)
CALL EPlot(2, XI, YI)
IF (XLAST = XI) 350, 350, 320

350 CONTINUE

360 IF (INCASE = ICASE) 380, 380, 370

370 CALL EPlot(1, XSTRT, YSTRT)
CALL FINISH
CALL LINK(GECAP)

C

C*** SET UP THE PEN FOR THE NEXT PLOT

C

380 CALL EPlot(1, XI, YI)
CALL EPlot(1, XMAX, YSTRT)
CALL SCALE(1, 0.1, 0, 0.1, 0)
CALL EPlot(1, 0, 0, 0)
CALL EXIT
END

// DUP
*DELETE        BESFT
*STORE          WS UA    BESFT
// FOR
*EXTENDED PRECISION
*ONE WORD ENTEGERS
FUNCTION F(A,N,X)
C
C*** GECAP REVISION C, 39/05/72
C
C
C*** FUNCTION F CALCULATES A Y VALUE FROM A GIVEN X VALUE USING THE LAST
C*** SQUARES CO-EFFICIenis

DIMENSION A(12,13)
10 F = A(N+1,1)
    L = N
    IF(N)20,40,20
20 DO 30 I = 1,N
    F = A(L,1)*X**I + F
30 L = L - 1
40 RETURN
END

// DUP
*DELETE F
*STORE WS UA F
FOR ONE WORD INTEGERS
*EXTENDED PRECISION

SUBROUTINE CURF (X,Y,N,M,X,M,A,SUM,V,WORK,IHLDE)

C C C
C GECAP REVISION C 09/05/72
C C C
C CATEGORY
C MATHEMATICAL
C PURPOSE
FIT TABULAR DATA TO POLYNOMIAL OF TYPE Y = AO + A1*X + A2*X**2 + ...
DESCRIPTION
THIS SUBROUTINE WILL FIT N DATA POINTS IN X AND Y TO A POLYNOMIAL OF THE TYPE
Y = AO + A1*X + A2*X**2 + ... + AM*X**M,
WHERE N IS GREATER THAN OR EQUAL TO M+1. THE COEFFICIENTS AO, A1, ..., AM ARE DETERMINED BY THE METHOD OF LEAST SQUARES.
The routine may be run in either single or double precision arithmetic. The data points need not be equally spaced.

INPUT
1 X FIRST LOCATION OF AN ARRAY CONTAINING N INDEPENDENT VARIABLE DATA POINTS.
2 Y FIRST LOCATION OF AN ARRAY CONTAINING N DEPENDENT VARIABLE DATA POINTS.
3 N NUMBER OF DATA POINTS TO BE FITTED.
4 M DEGREE OF POLYNOMIAL TO WHICH DATA IS TO BE FITTED.
5 MX VARIABLE DIMENSION INTEGER. MUST BE EQUAL TO MAXIMUM M USED PLUS 1.
6 A A TWO DIMENSIONAL ARRAY USED BY CURFIT TO SOLVE THE LEAST SQUARES NORMAL EQUATIONS. A MUST BE DIMENSIONED TO (MX+MX+1) IN THE CALLING PROGRAM.
7 SUM A TEMPORARY STORAGE AREA. MUST BE DIMENSIONED TO (2*MX) IN CALLING PROGRAM.
8 V A TEMPORARY STORAGE AREA. MUST BE DIMENSIONED TO (MX) IN CALLING PROGRAM.
9 WORK SEE 8 ABOVE.
10 IHLDE SEE 8 ABOVE.

OUTPUT
1 A A(1,1) CONTAINS THE COEFFICIENT AM. A(2,1)
CONTAINS AM-1, ..., A(M+1,1) CONTAINS AO OF THE EQUATION Y = AO + A1*X + A2*X**2 + ... + AM*X**M.
2 E ERROR CHECK. IF E = 0.0, OK. IF E = 1.0, AN ERROR HAS OCCURRED IN THE CALCULATION OF THE COEFFICIENTS.

REMARKS
THIS ROUTINE CONTAINS A METHOD OF SCALING WHICH PREVENTS OVERFLOW IN CASE THE DATA POINTS ARE LARGE, AND PREVENTS LOSS OF ACCURACY IN CASE THE INDEPENDENT DATA POINTS AGREE TO SEVERAL SIGNIFICANT FIGURES.

REFERENCES
INTERNAL TECHNICAL NOTE NO. TN-65-01
LEAST SQUARES POLYNOMIAL CURVE FIT NO. 2
BILLY G. GIBBS
SCIENTIFIC DIGITAL PROGRAMMING BRANCH
ARMY COMPUTATION CENTER
REDSTONE ARSENAL, ALABAMA

CONTACT
CHIEF, THEORETICAL PROBLEMS SECTION
SCIENTIFIC DIGITAL PROGRAMMING BRANCH
ARMY COMPUTATION CENTER

55
SUBROUTINE CURF (X, Y, N, M, X, Y, N, M, A, SUM, V, WORK, IHL, E)

DIMENSION X(1), Y(1), SUM(1), V(1), A(12, 1), WORK(1), IHL(1)

E = 0.0
LS = 2*M + 1
LB = M + 2
LV = M + 1
XH = ABS(X(1))

DO 20 I = 1, N
IF (XH . LE. ABS(X(I))) 10, 20, 20
10 XH = ABS(X(I))
20 CONTINUE

DO 30 I = 1, N
30 X(I) = X(I) / XH
FHH = X(I)

DO 50 I = 1, N
IF (FHH . LE. X(I)) 150, 50, 40
40 FHH = X(I)
50 CONTINUE

DO 60 I = 1, N
60 X(I) = X(I) - FHH

DO 80 J = 1, LV
80 SUM(J) = 0.

DO 110 I = 1, N
110 V(I) = 0.

P = 1.
V(I) = V(I) + Y(I)
DO 90 J = 2, LS
90 V(J) = V(J) + Y(I)* P
P = X(I) * P
SUM(J) = SUM(J) + P
DO 100 J = LB, LS
100 SUM(J) = SUM(J) + P

KK = LV
DO 120 I = 1, LV
120 SUM(J) = SUM(J) + 1

L = I
DO 110 K = 1, LV
A(K, KK) = SUM(L)
110 L = L + 1

KK = L - 1
DO 130 I = 1, LV
130 SUM(J) = SUM(J) + V(I)

DO 140 I = 1, LV
140 IHL(I) = I

JJ = LB
DO 280 I = 1, LV
280 SUM(J) = SUM(J) + 1

KK = LV - 1
IF (KK . LE. 240) 240, 240, 150
150 LL = KK + 1

IJJ = I
L = I
WORK(I) = A(I, I)
DO 170 I = 1, LL
170 WORK(I) = A(I, J)
L = J + 1 - 1
IJJ = J

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

DELETE CURF
STORE WS UA CURF
## APPENDIX C

### SYMBOLS AVAILABLE FOR THE IBM 1130 PLOTTER ROUTINES

<table>
<thead>
<tr>
<th>Alphabetic Characters</th>
<th>A – Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Characters</td>
<td>0 – 9</td>
</tr>
<tr>
<td>Special Characters</td>
<td></td>
</tr>
<tr>
<td>Characters</td>
<td>Punch</td>
</tr>
<tr>
<td>.</td>
<td>12–8–3</td>
</tr>
<tr>
<td>(</td>
<td>12–8–5</td>
</tr>
<tr>
<td>+</td>
<td>12–8–6</td>
</tr>
<tr>
<td>$</td>
<td>11–8–3</td>
</tr>
<tr>
<td>*</td>
<td>11–8–4</td>
</tr>
<tr>
<td>)</td>
<td>11–8–5</td>
</tr>
<tr>
<td>–</td>
<td>11</td>
</tr>
<tr>
<td>/</td>
<td>0–1</td>
</tr>
<tr>
<td>,</td>
<td>0–8–3</td>
</tr>
<tr>
<td>.</td>
<td>8–5</td>
</tr>
<tr>
<td>=</td>
<td>8–6</td>
</tr>
<tr>
<td>Space</td>
<td>Blank</td>
</tr>
</tbody>
</table>
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
Chief, Systems Operations Branch

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

KARL L. HEIMBURG
Director, Astronautics Laboratory
Page Intentionally Left Blank