MGRAPH:

A MULTIPLE CURVE PLOT PROGRAM FOR USE ON A LINE PRINTER

T. L. TUCKER

GREAT LAKES FOREST RESEARCH CENTRE

SAULT STE. MARIE, ONTARIO

INFORMATION REPORT O-X-202

CANADIAN FORESTRY SERVICE

DEPARTMENT OF THE ENVIRONMENT

MARCH 1974

Copies of this report may be obtained from

Director,
Great Lakes Forest Research Centre,
Canadian Forestry Service,
Department of the Environment,
Box 490, Sault Ste. Marie, Ontario.
P6A 5M7

ABSTRACT

This report outlines a computer program which is designed to facilitate researchers who want to obtain a quick visual comparison of several data sets or see if a functional form may exist in their data.

MGRAPH can plot up to five curves or data sets per graph, which are read in and/or generated from a function. Because most data used by researchers are coded on computer cards for a wide variety of data processing other than plotting, MGRAPH is designed to accept a wide range of data formats. It accepts coded and uncoded data of varying record lengths, can transform the Y variables and can handle data which are read in and generated simultaneously.

Several applications of MGRAPH are presented and data cards are shown to aid the user.

TABLE OF CONTENTS

INTRODUCTION

Researchers generally require a quick and simple way of graphically illustrating data for rapid comparisons or interpretation of general functional forms. A number of plotters attached to computers or programs utilizing line printers are available for plotting. However, plotters are slow and not readily accessible to researchers using remote terminals. While plots obtained from line printers are not as neat or precise, they are quick and readily available. Nevertheless, most of these plot programs either a) are built into a general statistical canned program and are useable only in that manner, have no versatility in form of data input, require the user to precalculate the scale or interval for the X and Y axes and read these values into the program, or b) do not allow multiple curves, functions or data sets to be plotted on one graph. The plot program (MGRAPH) presented here is designed to eliminate these difficulties.

None of the features presented in MGRAPH is unique in itself but in combination they provide a rather short program which will work on any computer using Basic Fortran IV or higher forms of Fortran, accept a wide variety of data formats, calculate its own scale for the X and Y axes, give the user the option of specifying the desired graph size, and plot up to five curves, function or data sets per graph. This latter feature not only saves space, paper and printing time but also allows a quick comparison of several curves on one graph.

MAIN PROGRAM

MGRAPH will accept X and Y coordinates which are read in or generated, either simultaneously or separately. The option is also available for transforming variables after they are read in. Because data are seldom put on cards for plotting only, MGRAPH allows for a wide variety of data formats which may be more applicable for other forms of analysis. MGRAPH is therefore written to accept (1) single and multiple X-Y observations per card[1], (2) single or multiple X-Y observations per card which are coded, (3) two or more Y's plotted against one X, i.e., (X_1, Y_1, Y_2 ... Y_5) per card, (4) one or more Y's plotted against one X with one or several transformations of any Y variable, (5) one to five functions (and to generate the data), and (6) several combinations of these five general forms. Table 1 provides a more detailed list of input combinations which are acceptable to MGRAPH. Tables A1-A5 in Appendix I give examples of data cards corresponding to each input type described in Table 1.

In Type 1, the simplest input type, one or more X-Y coordinates are read in per card and the order remains the same over all cards

---

[1] MGRAPH is not confined to reading cards but can read records from tapes or disks with the appropriate change in logical unit number.

Table 1   Data input combinations

Input type	Number of curves plotted per graph by mode of specification: Total	Read in	Transform	Generate	Number read per card: X	Y	Code	Form of input	Remarks
1	1-5	1-5	0	0	1-5	1-5	0	$X_1, Y_1 \ldots X_i, Y_i$ $i = 1, 5$	any ordering of variables is acceptable but must be maintained for all data cards
2[a]	1-5	1-5	0	0	1-5	1-5	1-5	$X_1, Y_1, C_1 \ldots X_i, Y_i, C_i$ $i = 1, 5$	
	1-5	1-5	0	0	1	1	1	X_2, Y_2, C_2	
								X_5, Y_5, C_5 one observation per card	
								X_3, Y_3, C_3	
								X_1, Y_1, C_1	
								X_4, Y_4, C_4	
3	2-5	2-5	0	0	1	2-5	0	$X_1, Y_1 \ldots Y_i$ $i = 1, 5$	all Y's and/or Y_t's plotted against X_1
4	2	1	1	0	1	1	0	$X_1, Y_1 : Y_{t1} = f(Y_1)$	
	3-5	1	2-4	0	1	1	0	$X_1, Y_1 : Y_{ti} = f_i(Y_1)$ $i = 1, 2 \ldots 4$	
	3-5	2	1-3	0	1	2	0	$X_1, Y_1, Y_2 : Y_{ti} = f_i(Y_1 \mid Y_2)$ $i = 1 \ldots 3$	
5	1-5	0	0	1-5	0	0	0	$Y_i = f_i(X)$ $i = 1 \ldots 5$	
6[b]	3	1	1	1	1	1	0	$X_1, Y_1 : Y_{t1} = f(Y_1)$	
								$Y_3 = f(X)$	
	3	2	0	1	1	1	1	X_1, Y_1, C_1	
								X_2, Y_2, C_2	
								$Y_3 = f(X)$	

[a] If only one X and Y are plotted ($i = 1$) a code is not needed.

[b] Only two of the more complex combinations of data (Types 2-5 and 4-5) are given here, but combinations of types 1-5, and 3-5 are also acceptable and input should be self explanatory.

(presorted data). Type 2 is similar to Type 1 except that the data are coded, e.g., a species code. The code allows the user to have either several X-Y observations per card in any order with the order changing from card to card[2], or single X-Y observations per card with the cards arranged in any order, i.e., the code allows proper sorting of data. In Type 3 several Y's are plotted against one X. Type 4 is the same as Type 3 except that one or more of the Y's can be transformed and plotted against X.[3] Type 5 allows the user to specify one or several functions which generate the X-Y coordinates. Type 6 shows where Types 1-5, 2-5, 3-5, and 4-5 combinations are acceptable.

When the data are read in, the user is required to specify a format statement which corresponds to his data.[4] It will take the form

5 FORMAT(C_1, C_2...C_n)

where: C's are format codes
n = the number of variables read per card.

In addition, if the user desires to transform the data (input Type 4), he must also specify the desired transformation(s). They will be of the form:

```
ZT(1) = f(Z(N))
  :       :
  :       :
ZT(t) = f(Z(N))
```

where: t = the number of transformations being done
Z(N) = the Y variable read in, upon which the transformation will be done, e.g., if the user desired the reciprocal of the second variable read in, ZT(1) = 1/Z(2).

---

[2] The X's, Y's and C's must always occupy the same columns from card to card, e.g., Card 1 X_1 Y_1 C_1 X_2 Y_2 C_2 X_3 Y_3 C_3

Card 2 X_3 Y_3 C_3 X_1 Y_1 C_1 X_2 Y_2 C_2

The order of X, Y and C is changing but within the appropriate columns (see Table 2, Appendix I).

[3] Note: Transformations cannot be done when more than one X is read in (Types 1 and 2) because the program has no way of distinguishing which X the transformed Y is to be plotted against.

[4] Even when no data are to be read in, statement number 5 is required because all sections of the compiled program, including that which reads cards, are scanned to see if all referenced statements (e.g., statement 5) are present.
Note: A 5 CONTINUE statement placed where the format statement goes will suffice when data are not read in.

When the data are to be generated as in Types 5 and 6 a function must be specified by the user for each set of data to be generated, and will take the form:

$$Z(1) = f_1(XX)$$
$$\vdots \qquad \vdots$$
$$Z(t) = f_t(XX)$$

where: t = the number of curves to be generated
XX = the X axis variable of which Y is a function.

The following is an example for generating two sets of data:

Z(1) = 73.5 + .15*XX + .05*XX**2

Z(2) = 50.0 + .15*XX + .075*XX**2.

XX is generated by the program from data concerning the size and range of the X axis. Wherever X enters the function the user must insert the variable name XX (see above).

Table 2 gives a checklist of the format, transformation, and generating function cards which the user must specify and insert in the main program for each input type. The proper place for inserting these is given by COMMENT cards in the main program (Appendix IV).

Table 2 Main program cards which user specifies, by input type

Input type	Format	Number of cards required (transformation)	Generating functions	Total
1	1	0	0	1
2	1	0	0	1
3	1	0	0	1
4	1	1-5	0	2-6
5	1[a]	0	1-5	2-6
6	1	1-4	1-4	3-6

[a] See footnote 4, page 3.

DATA INPUT INSTRUCTIONS

In order that MGRAPH may read in, transform, generate, and arrange and plot the data the user must specify a number of program variables. The names and definitions of these user-specified variables are given below and the input instructions for them are given in abbreviated form in Table 3. A more detailed description can be found in Appendix II along with examples of data cards corresponding to Table 3 (Table A6, Appendix II).

The user-specified variables are of two types: 1) for specifying and processing the data and 2) for plotting the data (see the following list). They will occupy from four to five data cards, and they *must* precede the data deck containing the X-Y observations (see Figure A5, Appendix IV).

List of User-specified Variables

INPUT = Number of curves to be generated from functions
MA = Number of curves, up to five, to be plotted per graph
NVB = Number of variables per data card
NVA = Number of variables after transformation ($NVA \geq NVB$)
IX = Which variables per card or record are X coordinates
IY = Which variables per card or record are Y coordinates[5]
ICV = Which variables per card or record are data codes
SV = Starting X value when Y is to be generated
EV = Ending X value when Y is to be generated
FLG = Value of flag card which signals the end of data set
ICODE = Value of data codes for separating X-Y data (MAX = 5)
SYMB = Symbols for plotting curves (MAX = 5)
KX = Size of X axis - number of spaces wide (MAX = 100)
KY = Size of Y axis - number of lines high
ANX = X axis label
ANY = Y axis label
ACR = Label for each curve (equal to MA)
FAMNA = Label for graph (name of family of curves)
IVX = Code which indicates if integer values or real values are desired for scaling the X axis
IVY = Code which indicates if integer or real values are desired for scaling the Y axis
SY1 = Minus sign (-) for delineating the X axis
SY2 = Plus sign (+) for delineating the X axis

---

[5] Only those Y's which are read in and will be plotted are specified here, i.e., if a Y is used in a transformation and subsequently disregarded it must not be specified.

Table 3 Input instructions[a]

Card	Column	Variable	Input example	Remarks
1	1	INPUT	3	INPUT $\leq$ MA (three curves to be generated)
1	2	MA	5	
1	4,5	NVB	06	
1	6,7	NVA	06	NVA > NVB
1	9-18	IX	01 04 00 00 00	Of the six (NVB) variables read in variables 1 and 4 are X's
1	19-28	IY[a]	03 06 00 00 00	Of the six (NVB) variables read in variables 3 and 6 are Y's
1	29-38	ICV	02 05 00 00 00	Of the six (NVB) variables read in variables 2 and 5 are Codes
1	40-47	SV	10	If INPUT = 0, SV = 0
1	48-55	EV	100	If INPUT = 0, EV = 0
1	56-64	FLG	99	Must correspond to format for reading first variable from X-Y data cards
1	66-80	ICODE	050 060 000 000 000	Codes for the two ICV variables
2	1-5	SYMB	12AB*	If < 5 leave remaining blank
2	7,8	KY	40	
2	10-12	KX	075	Maximum = 100
2	14	IVY	0	0 = real scale acceptable; 1 = integer scale desired
2	15	IVX	1	0 = real scale acceptable; 1 = integer scale desired
2	17	SY1	- (minus)	{ for delineating the X axis
2	18	SY2	+ (plus)	{
2	20-49	ANY	VOLUME PER ACRE	Start in column 20
2	51-80	ANX	SITE INDEX	centre between columns 51-80
3	1-20	ACR	WHITE PINE	names { 1 }
	21-40	ACR	RED PINE	for { 2 } centre all five names
	41-60	ACR	JACK PINE	curves { 3 } between the 20
	61-80	ACR	BLACK SPRUCE	{ 4 } columns provided
4	1-20	ACR	BALSAM FIR	{ 5 If not needed delete card 4.
5	1-20	FAMNA	YIELD COMPARISONS	Start in column 1
5	21-29	ATT	123456789	{ Must always be punched
5	30	SPS	&	{ as written

[a] A more detailed description can be found in Appendix I along with examples of data cards.

ATT = } Symbols required if more than one point occupies
SPS = } the same coordinates on the curve

SUBPROGRAMS

The main program calls the subroutine GRAPH and through the common storage makes all the required arguments available to it. GRAPH in turn calls three other subroutines (SORT, LIMIT, and SCALE), which either calculate needed variables or arrange the variable arrays in the proper order for plotting. GRAPH utilizes these values to arrange and print out the entire graph with all labels, symbols and a legend. Because several X-Y coordinates may be identical or so nearly identical that they occupy the same point on the graph, GRAPH is written to check for this eventuality, and where it occurs it prints the number of points occupying the same coordinates on the graph up to a maximum of nine superimposed points (see Figure A1, Appendix III). If more than nine occur the symbol "&" is printed. The user is therefore advised not to use the symbol "&" or the numbers 2 through 9 as plotting symbols when superimposed points are suspected.

SORT is a "bubble sorter", which arranges all the Y coordinates in descending order and rearranges the X values to correspond to the proper Y's. Where more than one curve is to be plotted per graph SORT also rearranges the symbols which correspond to the Y's for each curve.

LIMIT is used to find the maximum and minimum values of X. Where such a subprogram is built into a computer system, this subroutine can be eliminated.

SCALE takes the user's specifications for graph size and calculates a well-behaved scale and starting value (rounds and truncates) for the X and Y axis. It also allows the user to specify if he desires the X or Y scales to be only integer values.

APPLICATION

Figures A1-A4 in Appendix III illustrate the versatility of MGRAPH. Their corresponding INPUT cards specifying both program and data cards required by the user are given in tables A7-A10, Appendix III. Figure A1 illustrates a 4 in. x 5 in. graph[6],

---

[6] The specifications for a 4 in. x 5 in. graph are that it be 25 lines tall (punched in columns 7 and 8 of card 2) and 50 spaces wide (punched 050 in columns 10, 11 and 12 of card 2). A 6.5 in. x 7.5 in. graph, i.e., 40 lines tall and 75 spaces wide, and a 8.5 in. x 10 in. graph, 50 lines tall and 100 lines wide (maximum width) are the two other recommended sizes.

(smallest recommended size) which is the plot of four site index curves generated from functions[7]. Figure A1 is the smallest recommended size.

Figure A2 illustrates a scatter diagram (stem map) of tree locations identified (coded) by species and read in. Figure A3 gives a graph of residual values obtained from a typical regression program and illustrates MGRAPH's ability to handle both positive and negative values simultaneously. In this case it shows that the model used was inappropriate. Figure A4 shows a plot of actual data plus a fitted least squares regression line. This illustrates the versatility of combining both actual (read in) and generated data simultaneously on one graph.

Appendix IV gives a printout of the MGRAPH program and shows the arrangement of the data cards. It was originally written for the IBM 360/25 in basic Fortran IV and with the appropriate change in logical unit number can be used on any machine which accepts basic Fortran IV or higher forms of Fortran.

---

[7] Functions obtained from Lundgren, A. L., and W. A. Dolid. 1970. Biological growth functions describe published site index curves for Lake States timber species. USDA Forest Serv., North Central Forest Exp. Stn., St. Paul, Minn. Res. Pap. NC-36. 9 p.

APPENDICES

APPENDIX I

The data used in the following five examples are identical for the reader's convenience in comparing input types. All five examples illustrate data input for two sets of data (two curves) each containing five X-Y observations (five points per curve). Note: Up to five data sets can be handled at one time, but for ease of presentation only two are illustrated.

Table A1. An example of Type 1 data[a]

Column	12345678911111111112222222222333333333344444444445555555555666666666677777777778 01234567890123456789012345678901234567890123456789012345678901234567890			
Card 1	75	1430	105	1700
Card 2	100	1500	95	1625
Card 3	20	1100	10	1050
Card 4	30	1175	70	1500
Card 5	60	1305	40	1375
Card 6	999[b]			

For each of these cards the four figures are: X_1, Y_1, X_2, Y_2

[a] All X-Y coordinates are placed on cards in a sorted form.

[b] Flag Card - signals end of data.

Table A2 An example of Type 2 data[a]

Column	12345678911111111112222222222333333333344444444445555555555666666666677777777778 01234567890123456789012345678901234567890123456789012345678901234567890					
Card 1	75	1430	10	105	1700	20
Card 2	95	1625	20	100	1500	10
Card 3	10	1050	20	70	1500	20
Card 4	20	1100	10	30	1175	10
Card 5	60	1305	10	40	1375	20
Card 6	999[b]					

For each of these cards the six figures are:

Card 1	X_1, Y_1, $CODE_1$, X_2, Y_2, $CODE_2$
Card 2	X_2, Y_2, $CODE_2$, X_1, Y_1, $CODE_1$
Card 3	X_2, Y_2, $CODE_2$, X_2, Y_2, $CODE_2$
Card 4	X_1, Y_1, $CODE_1$, X_1, Y_1, $CODE_1$
Card 5	X_1, Y_1, $CODE_1$, X_2, Y_2, $CODE_2$

[a] Data are the same as in Table 1 except that they are not prearranged in sorted form; therefore, a code is required by the computer for sorting.

[b] Flag Card - signals end of data.

Table A3 An example of Type 2 data[a]

Column	12345678911111111112222222222333333333344444444445555555555666666666677777777778 0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
Card 1	75 1430 10
Card 2	95 1625 20
Card 3	10 1050 20
Card 4	20 1100 10
Card 5	60 1305 10
Card 6	105 1700 20
Card 7	100 1500 10
Card 8	70 1500 20
Card 9	30 1175 10
Card 10	40 1375 20
Card 11	999[b]

For each of these cards the three figures are: Cards 1,4,5,7,9 X_1, Y_1, $CODE_1$

Cards 2,3,6,8,10 X_2, Y_2, $CODE_2$

[a] Data are the same as in Table 1 except that each card has only one data set and the order is shuffled; therefore, a code is required by the computer for sorting.

[b] Flag Card - signals end of data.

Table A4   An example of Type 3 data

Column	12345678911111111112222222222333333333344444444445555555555666666666677777777778 01234567890123456789012345678901234567890123456789012345678901234567890		
Card 1	75	1430	1450
Card 2	100	1500	1520
Card 3	20	1100	1120
Card 4	30	1175	1195
Card 5	60	1305	1325
Card 6	999[a]		

For each of these cards the three figures are: X_1, Y_1, Y_2

[a] Flag Card - signals end of data.

Table A5 An example of Type 4 data

Column	12345678911111111112222222222333333333344444444445555555555666666666677777777778 01234567890123456789012345678901234567890123456789012345678901234567890
Card 1	75 1430
Card 2	100 1500
Card 3	20 1100
Card 4	30 1175
Card 5	60 1305
Card 6	999[a]

For each of these cards the two figures are: X_1, Y_2

The transformed variable $Y_{t1} = Y_1 + 20$ (Note: Y_{t1} is equal to Y_2 in Table A4)

[a] Flag Card - signals end of data.

Table A6 Data cards corresponding to Table 3

```
Column      12345678911111111112222222222333333333344444444445555555555666666666677777777778
                     01234567890123456789012345678901234567890123456789012345678901234567890

Card 1[a]   35 0606 0104      0306      0205       0000001000000010000000099. 050060
Card 2      12AB* 40 075 01 -+ VOLUME PER ACRE                          SITE INDEX
Card 3          WHITE PINE           RED PINE           JACK PINE          BLACK SPRUCE
Card 4          BALSAM FIR
Card 5      YIELD COMPARISONS   123456789&
```

[a] Columns 13-18, 23-28, 33-39 are left blank rather than punched zero.

APPENDIX II

Detailed Data Input Instructions

CARD 1:

Column: 1 is used to specify the number of curves or data sets (var. INPUT) which will be generated (Max. = 5). If all data are read in leave column blank.

Column: 2 is used to specify the total number of curves (var. MA) to be plotted per graph (Max. = 5).

Columns: 4, 5 are used to specify the number of variables which will be read per data card (var. NVB). If all data are generated leave columns blank.

Columns: 6, 7 are used to specify the total number of variables which are read in and transformed (var. NVA). These must always be equal to or greater than NVB, i.e., if no data transformation is done NVA = NVB; if one transformation is done NVA = NVB + 1. If all data are generated leave columns blank.

Columns: 9-18 are used to identify which of the variables (NVB) read in are X's (var. IX). A maximum of five X variables can be identified with two columns reserved for each variable. If 15 variables were read in (NVB = 15) and the first, fourth, seventh, tenth and thirteenth variables were X's columns 9-18 would be punched 0104071013. If only the first and fourth variable were X's columns 9-12 would be punched 0104 and the remainder left blank or punched zero. If all data are generated leave columns blank.

Columns: 19-28 are used to identify which of the NVB variables are Y's which *will be used in plotting* (var. IY). The input is exactly like the preceding variable (IX), with one exception. Some Y's may be read in, used in a transformation, and then discarded. In this case the Y so transformed is *not* used in plotting and therefore *must not be* specified as a Y variable. If all data are generated leave volumns blank.

Columns: 29-38 are used to identify which of the NVB variables are data codes (var. ICV). The data input instructions are exactly like those for variable IX. If codes are not used leave columns blank.

Columns: 40-47 are used to specify the starting value of X (var. SV) which will be used when *generating* the X-Y data. If all data are read leave columns blank.

Columns: 48-55 are used to specify the ending value of X (var. EV) which will be used when *generating* the X-Y data. If all data are read in leave columns blank.

Columns: 56-64 are used to specify a flag card variable (var. FLG) which signals the end of the data deck. It must conform exactly with the format specifications for the first variable read in from the X-Y data cards. FLG usually takes the form 999.9. If the first variable read in from the X-Y cards has a format of F5.1 the FLG would be punched 0009999.9. Note that the decimal point is always punched. If all data are generated leave columns blank.

Columns: 66-80 are used to identify from one to five one-, two- or three-digit data codes which will be used to separate and sort the X-Y observations (var. ICODE), e.g., if five two-digit codes are needed columns 66-80 would be punched 010020030040050, if two three-digit codes are needed columns 66-71 would be punched 111222 while the remaining columns (72-80) would be left blank or punched zero. If codes are not needed leave these columns blank.

CARD 2:

Columns: 1, 2, 3, 4 and 5 are reserved for the symbols (SYMB) used to identify and plot the five curves or data sets. They can be any alphabetic, numeric or special character acceptable in basic Fortran IV. If less than five are required the remainder should be left blank, e.g., if four symbols are needed, columns 1, 2, 3 and 4 will be punched with column 5 left blank.

Columns: 7 and 8 are used to specify the number of single-spaced lines (variable KY) to be allocated to the Y axis (height of Y axis).

Columns: 10, 11 and 12 are used to specify the number of horizontal spaces (variable KX) to be allocated to the X axis (width of X axis) (Maximum = 100).

Column: 14 is used to specify if an integer scale is desired for the Y axis (variable IVY), i.e., it is punched "1" if an integer scale is desired; otherwise it is punched "0" (See Figures A1 and A3, Appendix III).

Column: 15 is used to specify if an integer scale is desired for the X axis (variable IVX), i.e., it is punched "1" if an integer scale is desired; otherwise it is punched "0".

Columns: 17 and 18 are always punched - (minus), and + (plus), i.e. - +. This is for delineating the X axis.

Columns: 20-49 are reserved for the label of the Y axis (variable ANY). If fewer than 30 spaces are required it is suggested that the user start the name in column 20.

Columns: 51-80 are reserved for the label of the X axis (variable ANX). If fewer than 30 spaces are required it is suggested that the user centre the name between columns 51 and 80.

CARDS 3-4 are reserved for the labels identifying the curves or data sets to be plotted (variable ACR).

Columns: 1-20 are reserved for the label identifying the curve or data set which corresponds to the points plotted by the symbol found in column 1 of card 2. If the name is less than 20 characters it should be centred between columns 1 and 20.

Columns: 21-40 are reserved for the label identifying the second curve or data set which corresponds to the points plotted by the symbol found in column 2 of card 2. If the name is less than 20 characters it should be centred between columns 21 and 40.

Columns: 41-60 and 61-80 are reserved for the names of the third and fourth data sets. The fifth data set has its label on card 4, which is punched in the same manner as columns 1-20 on card 3. If four or fewer data sets are required card 4 is omitted.

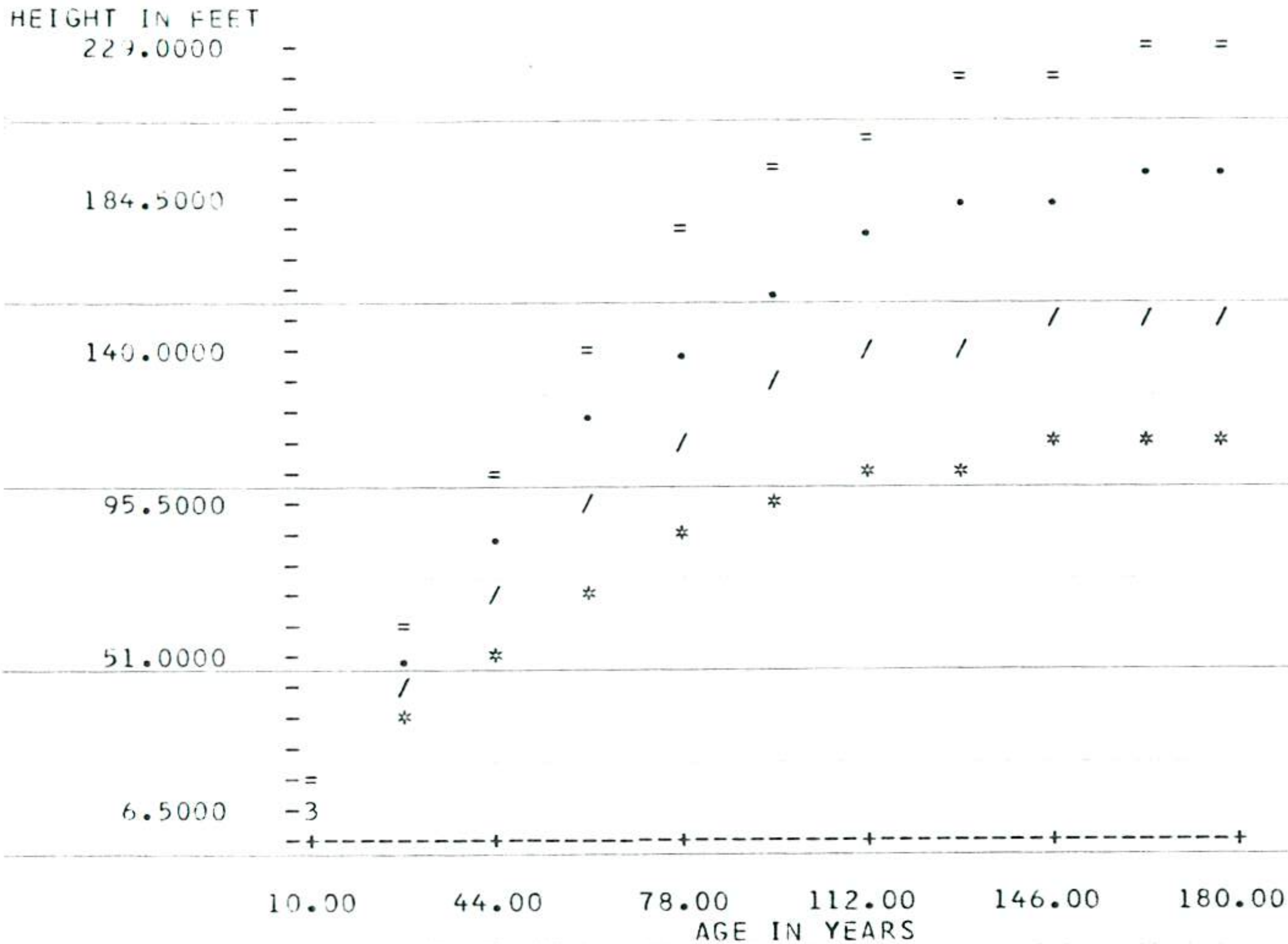
CARD 5:

Columns: 1-20 are reserved for the label identifying the graph or name of the family of curves (variable FAMNA) which are found on the graph. This label will be printed out to the right of the legend. Start name in column 1.

Columns: 21-29 are reserved for the numerical characters (123456789) used for plotting when two or more points are superimposed (variable ATT). They must always be punched on this card.

Column: 30 is reserved for the special character "&" used if more than nine points are superimposed (variable SPS). They must always be punched on this card.

APPENDIX III



SYMBOL   CURVE IDENTITY   PLOT IDENTITY   W. PINE LAKE STATES

SYMBOL	CURVE IDENTITY
*	SITE INDEX 60
/	SITE INDEX 80
.	SITE INDEX 100
=	SITE INDEX 120

Figure A1. Four site index curves for white pine in the Lake States.

Table A7 Program and data cards required to generate four site index curves

Column	12345678911111111112222222222333333333344444444445555555555666666666677777777778 01234567890123456789012345678901234567890123456789012345678901234567890
Program Cards	
1	Z(1)=(1.966*60.0)*(1.0-(EXP(-.02399*XX)))**1.8942
2	Z(2)=(1.966*80.0)*(1.0-(EXP(-.02399*XX)))**1.8942
3	Z(3)=(1.966*100.)*(1.0-(EXP(-.02399*XX)))**1.8942
4	Z(4)=(1.966*120.)*(1.0-(EX)(-.02399*XX)))**1.8942
5	5 CONTINUE
Data Cards	
User-specified Variables	
1	44 10 180
2	*/.= 25 050 01 -+ HEIGHT IN FEET AGE IN YEARS
3	SITE INDEX 60 SITE INDEX 80 SITE INDEX 100 SITE INDEX 120
4	W. PINE LAKE STATES 123456789&

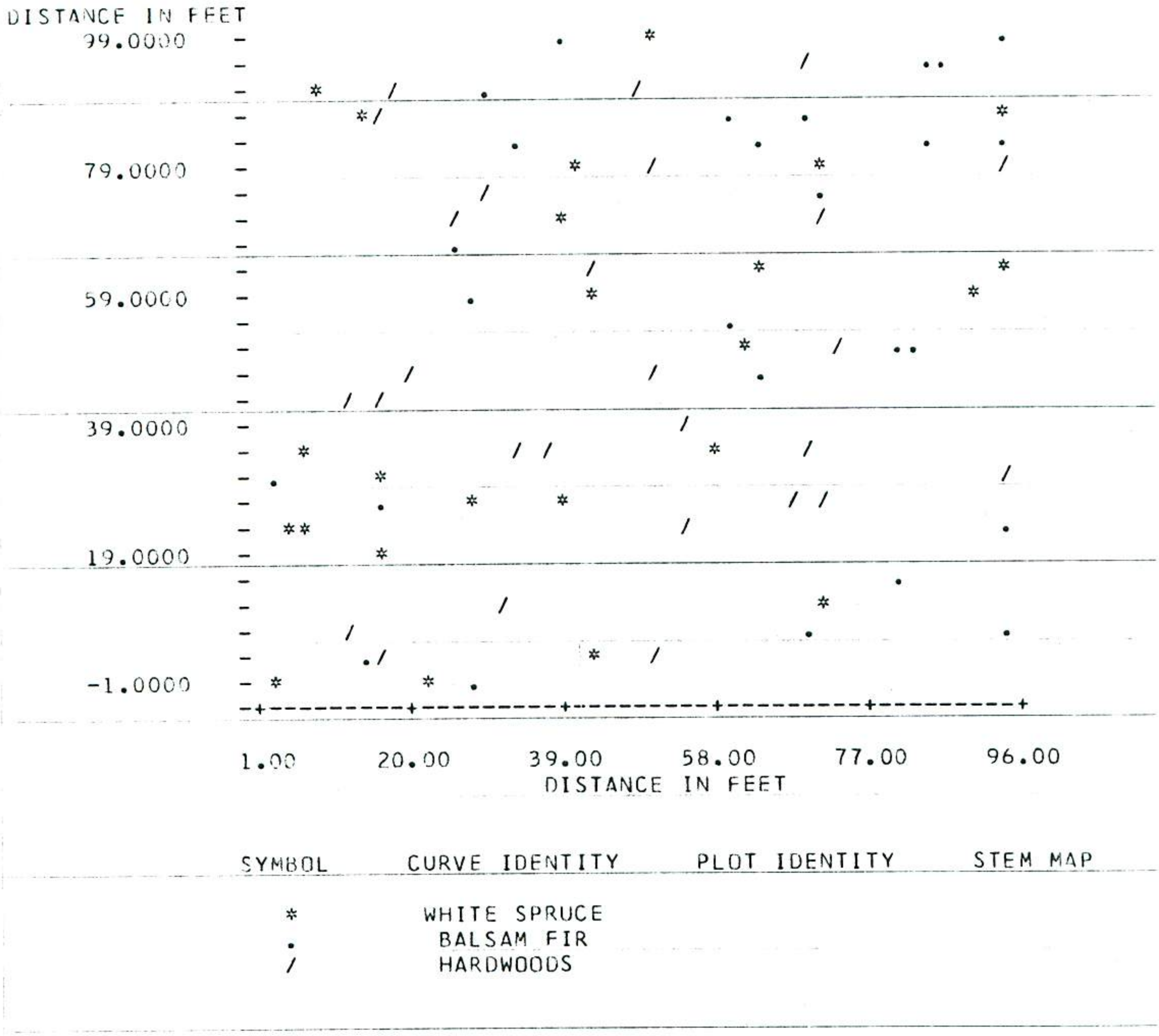


Figure A2. Stem map.

Table A8   Program and data cards required to translate and plot tree locations

```
Column        12345678911111111112222222222333333333344444444445555555555666666666677777777778
                       01234567890123456789012345678901234567890123456789012345678901234567890

Program
 Cards
   1              5 FORMAT(5(F4.2,1X,F4.2,1X,F3.0,2X))

Data Cards

User-specified
Variables
   1          03 1515 010407101302050811140306091215                     99.99 100200300
   2          *./   25 050 00 -+ DISTANCE IN FEET                       DISTANCE IN FEET
   3              WHITE SPRUCE         BALSAM FIR          HARDWOODS
   4          STEM MAP            123456789&

  X-Y
Coordinates
   1          5134  225 300  9693 6445 100   695 3676 100  1636  225 300  2019 2024 200
   .
   .
   .
   20         6978 3587 300  2060 4725 300  4287  465 100  9855  605 200  7213 3390 100
   21         9999
```

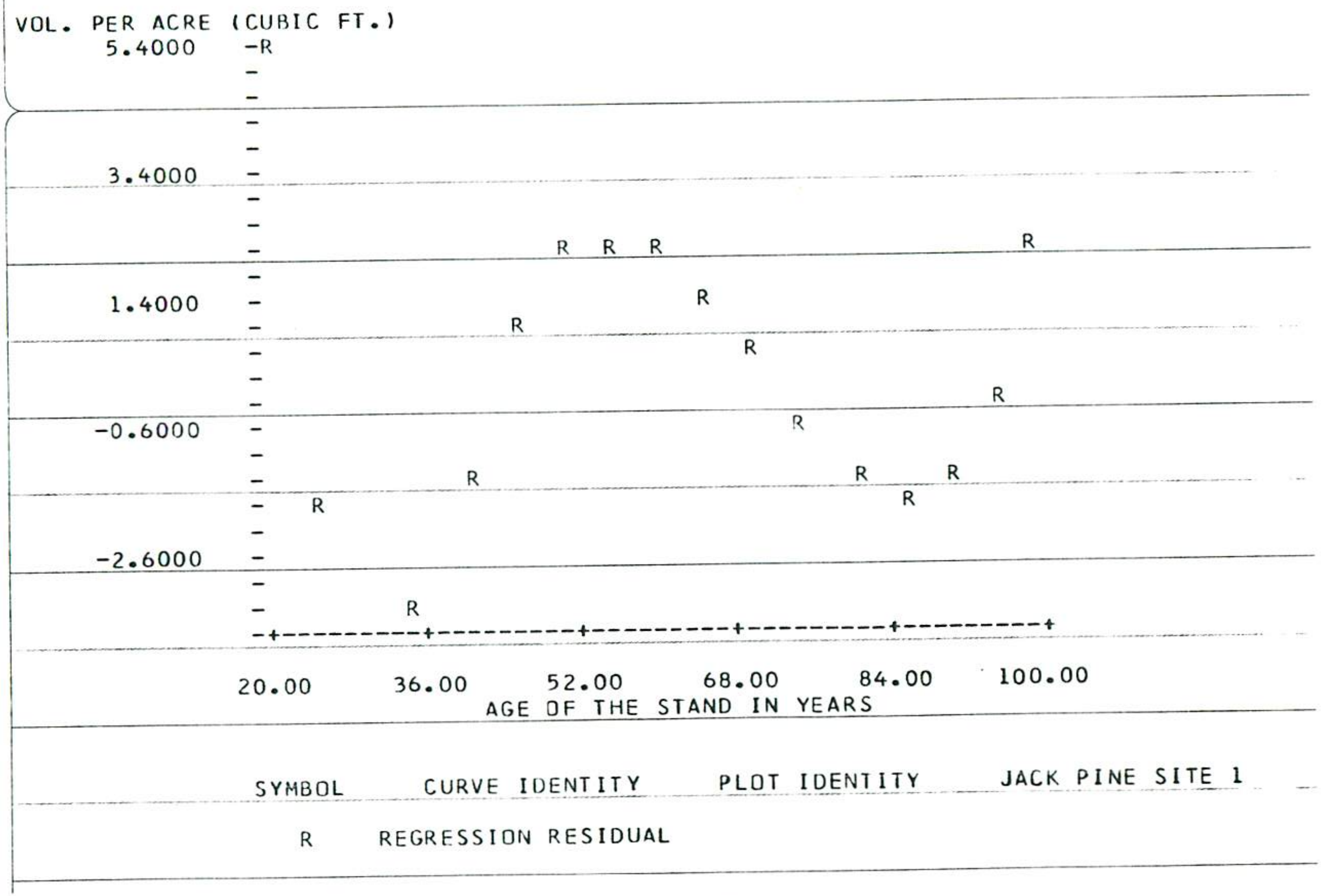


Figure A3. Plotted regression residuals.

Table A9 Program and data cards required to plot regression residuals

```
Column       12345678911111111112222222222333333333344444444445555555555666666666677777777778
                      01234567890123456789012345678901234567890123456789012345678901234567890

Program
 Cards
   1             5 FORMAT(F3.0,2X,F5.3)

Data Cards

User-specified
Variables
   1         01 0202 01        02                                        999.
   2         R     25 050 01 -+ VOL. PER ACRE (CUBIC FT.)        AGE OF THE STAND IN YEARS
   3         REGRESSION RESIDUAL
   4         JACK PINE SITE 1    123456789&

  X-Y
Coordinates
   1          20   53237
   .
   .
   .
   17        100   21164
   18        999
```

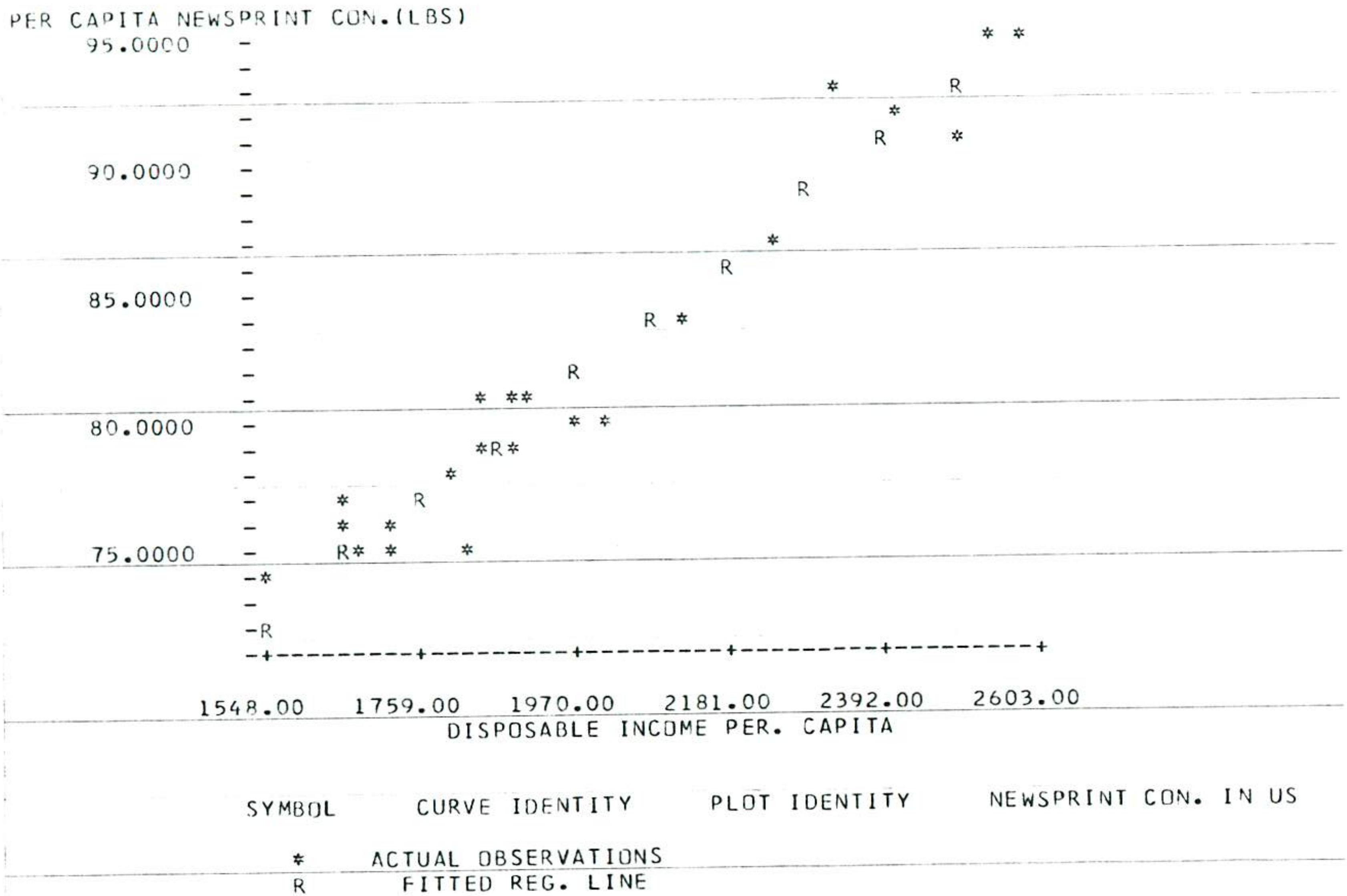


Figure A4. Per capita newsprint consumption in the United States: actual observations and fitted least squares regression line.

Table A10 Program and data cards required to generate and read in United States per capita newsprint consumption

```
Column        12345678911111111112222222222333333333344444444445555555555666666666677777777778
                       01234567890123456789012345678901234567890123456789012345678901234567890

Program
 Cards
   1                Z(1)=37.71834+0.02228*XX
   2              5 FORMAT(F4.0,F2.0)

Data Cards

User-specified
Variables
   1          12 0202 01        02                       1550    2600    9999.
   2          *R    25 050 01 -+ PER CAPITA NEWSPRINT CON.(LBS) DISPOSABLE INCOME PER CAPITA
   3          ACTUAL OBSERVATIONS   FITTED REGRESSION LINE
   4          NEWSPRINT CON. IN US123456789&

  X-Y
Coordinates
   1          156770
   .
   .
   .
   23         259595
   24         9999
```

APPENDIX IV

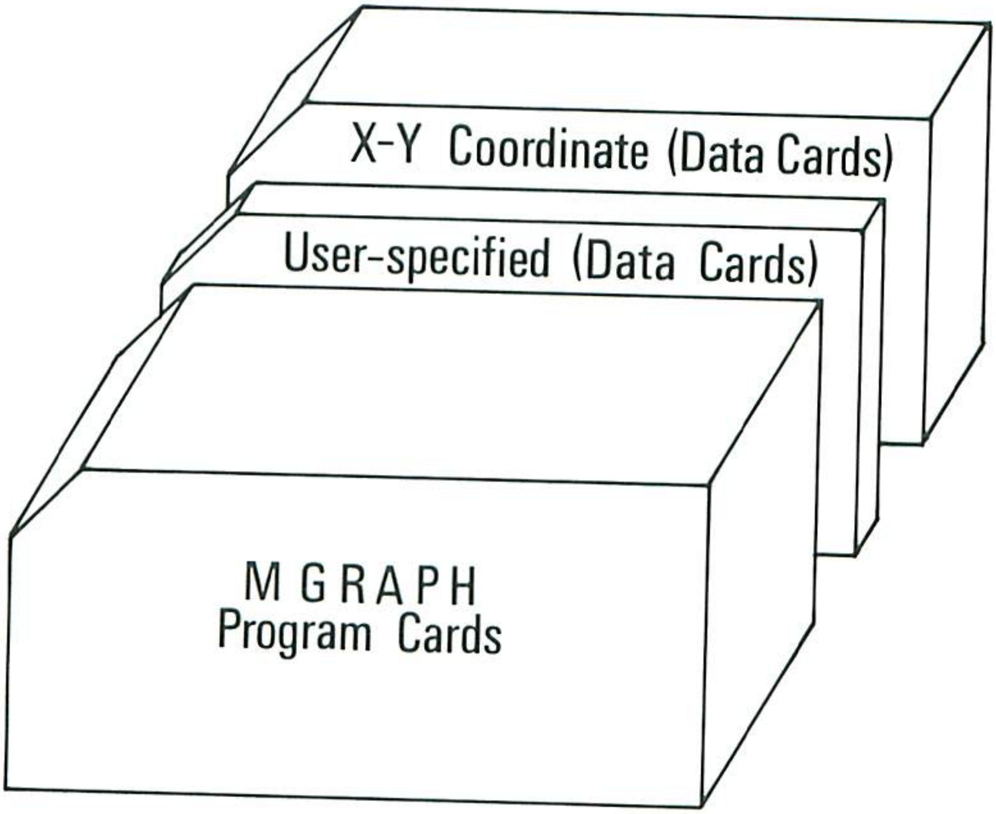


Figure A5. Arrangement of program and data cards.

Printout of the MGRAPH Program

```
      COMMON NZ,X(250),Y(250),ASY(250),SYMB(5),ANY(8),ANX(8),ACR(5,5),FA
     1MNA(5),MA,SYMBB,KY,KX,SY1,SY2,IVY,IVX,SPS,ATT(9)
      DOUBLE PRECISION C(2),RX,SX
      DIMENSION IX(5),IY(5),ICV(5),Z(15),ZT(5),ICODE(5),NS(3)
      READ(1,900) INPUT,MA,NVB,NVA,(IX(I),I=1,5),(IY(I),I=1,5),(ICV(I),I
     1=1,5),SV,FV,FLG,(ICODE(I),I=1,5)
      READ(1,901)(SYMB(I),I=1,5),KY,KX,IVY,IVX,SY1,SY2,SYMBB,(ANY(I),I=1
     1,8),(ANX(I),I=1,8)
      READ(1,902) ((ACR(I,J),J=1,5),I=1,MA)
      READ(1,903)(FAMNA(I),I=1,5),(ATT(I),I=1,9),SPS
      NZ=0
      IR=MA-INPUT
      IF(IR) 3,2,1
    1 DO 7 J=1,3
      DO 8 I=1,5
      GO TO (9,10,11),J
    9 IF(ICV(I)) 8,12,8
   10 IF(IX(I)) 8,12,8
   11 IF(IY(I)) 8,12,8
    8 CONTINUE
      NS(J)=5
      GO TO 7
   12 NS(J)=I-1
    7 CONTINUE
      NYY=NS(3)
      DO 16 I=1,1000
      READ(1,5)(Z(K),K=1,NVB)
      IF(Z(1)-FLG) 17,18,17
   17 IF(NVB-NVA) 19,20,3
   19 NV=NVB+1
C--   PLACE TRANSFORMATION CARDS HERE
      ZT(1)=Z(2)-20.0
      DO 21 K=NV,NVA
      KB= K- NVB
      Z(K)=ZT(KB)
      NYY=NS(3)+(K-NVB)
   21 IY(NYY)=K
   20 DO 22 J=1,NYY
      NZ=NZ+1
      IF(NS(2)-1) 24,23,24
   23 IJ=IX(1)
      GO TO 25
   24 IJ=IX(J)
   25 X(NZ)=Z(IJ)
      IJ=IY(J)
      Y(NZ)=Z(IJ)
      IF(NS(1)) 26,27,26
   26 IJ=ICV(J)
      DO 28 K=1,IR
      IF(Z(IJ)-ICODE(K)) 28,29,28
   28 CONTINUE
```

```
   29 ASY(NZ)=SYMB(K)
      GO TO 22
   27 IF(MA-1) 35,34,35
   34 ASY(NZ)=SYMB(1)
      GO TO 22
   35 ASY(NZ)=SYMB(J)
   22 CONTINUE
   16 CONTINUE
   18 IF(INPUT) 3,30,2
    2 C(1)=SV
      C(2)=EV
      CALL SCALE (KX,C,IVX,RX,SX)
      AV=RX*5.0
      NP=((EV-SV)/AV)+1
      DO 31 K=1,NP
      XX=SV+(K-1)*AV
C--   PLACE GENERATING FUNCTIONS HERE
      J=IR+1
      DO 32 I=J,MA
      NZ=NZ+1
      X(NZ)=XX
      KB=I-IR
      Y(NZ)=Z(KB)
   32 ASY(NZ)=SYMB(I)
   31 CONTINUE
   30 CALL GRAPH
      GO TO 33
  900 FORMAT(2I1,1X,2I2,1X,15I2,1X,2F8.0,F9.0,1X,5I3)
  901 FORMAT(5A1,1X,I2,1X,I3,1X,2I1,1X,3A1,7A4,A2,1X,7A4,A2)
  902 FORMAT(20A4)
  903 FORMAT(5A4,10A1)
    3 WRITE(3,999)
  999 FORMAT(1H1,'TEST INDICATES AN ERROR IN DATA FOR EITHER VARIABLE(S)
     1---MA, INPUT, NVB, OR NVA')
C--   PLACE FORMAT FOR X-Y DATA INPUT HERE
    5 FORMAT(F3.0,1X,F4.0)
   33 CONTINUE
      END
```

```
      SUBROUTINE SORT
      COMMON N,W(250),Z(250),ASY(250)
      I=N
    1 K=0
      DO 2 J=2,I
      IF(Z(J)-Z(J-1)) 2,2,3
    3 V=Z(J)
      Z(J)=Z(J-1)
      Z(J-1)=V
      V=W(J)
      W(J)=W(J-1)
      W(J-1)=V
      V=ASY(J)
      ASY(J)=ASY(J-1)
      ASY(J-1)=V
      K=1
    2 CONTINUE
      I=I-1
      IF(K) 4,4,1
    4 RETURN
      END
```

```
      SUBROUTINE LIMIT (ZMIN,ZMAX)
      COMMON N,Z(250)
      ZMIN=Z(1)
      ZMAX=Z(1)
      DO 1 J=2,N
      IF(ZMAX-Z(J)) 2,3,3
    2 ZMAX=Z(J)
      GO TO 1
    3 IF(ZMIN-Z(J)) 1,1,4
    4 ZMIN=Z(J)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SCALE (KK,C,INV,RR,SS)
      DIMENSION D(2)
      DOUBLE PRECISION RR,SS,C(2),RZ,AO,BBO,BAO,CO,TSIZ,SIZE,CHNG
      AO=1.
      FO=1.
      RR=DABS(C(1)-C(2))/KK
      IF(RR-.05) 5,5,4
    4 IRR=RR
      ARR=RR-IRR
      DO 6 I=1,10
      IF(ARR-(.1*I)) 7,8,6
    6 CONTINUE
      I=10
    7 IF(ARR-((.1*I)-.05)) 9,8,8
    9 RR=IRR+.1*(I-1)
      GO TO 10
    8 RR=IRR+.1*I
      GO TO 10
    5 DO 12 I=1,10
      AO=AO*.1
      FO=FO*10.
      IF(RR-AO) 12,13,13
   12 CONTINUE
   13 IRR=RR*FO
      RZ=IRR/FO
      DO 14 I=1,3
      AAO=(AO*.1)*(2.5*I)
      IF(RZ+AAO-RR) 14,15,15
   14 CONTINUE
      RR=RZ+AO
      GO TO 10
   15 GO TO (16,17,17),I
   16 RR=RZ
      GO TO 10
   17 RR=RZ+(AO*.1)*5.
   10 SIZE=RR*KK
      AO=1.
      TSIZ=DABS(C(1)-C(2))
      CHNG=DABS(SIZE-TSIZ)/2
      IF(SIZE-TSIZ) 18,19,20
   18 SS=C(1)-CHNG
      GO TO 21
   19 SS=C(1)
      GO TO 21
   20 SS=C(1)+CHNG
   21 DO 100 I=1,2
      IF(C(I)-0.0) 101,102,102
  101 D(I)=-1.0
      GO TO 100
  102 D(I)=1.0
```

```
100 CONTINUE
    IF(INV) 24,24,23
 23 IF(DABS(SS)-1.0) 25,26,27
 27 ISS=SS
    ASS=SS-ISS
    IF(ABS(ASS)-.50) 29,28,28
 28 SS=ISS+(1.*D(1))
    GO TO 30
 24 IF(SS-0.0) 33,26,33
 33 IF(DABS(SS)-1.0) 25,26,36
 25 DO 37 I=1,10
    A0=A0*.1
    IF(DABS(SS)-A0) 37,39,38
 37 CONTINUE
 38 DO 40 I=1,10
    BB0=A0*I
    IF(DABS(SS)-BB0) 41,42,40
 40 CONTINUE
 41 B00=A0*.1
    DO 43 I=1,10
    BA0=B00*I
    IF((BB0-BA0)-DABS(SS)) 44,45,43
 43 CONTINUE
 44 DO 46 I=1,1000
    IF(DABS(SS)-((BB0-BA0)+I*(RR/2))) 47,46,46
 46 CONTINUE
    I=1000
 47 SS=(BB0-BA0+(I-1)*(RR/2))*D(1)
    GO TO 30
 36 ISS=DABS(SS)
    IF(DABS(SS)-(ISS+RR/2)) 29,29,49
 29 SS=ISS*D(1)
    GO TO 30
 49 DO 50 I=1,1000
    C0=I*RR/2
    IF(DABS(SS)-(ISS+C0)) 51,51,50
 50 CONTINUE
 51 SS=(ISS+C0-RR/2)*D(1)
    GO TO 30
 26 SS=C(1)
    GO TO 30
 39 SS=A0*D(1)
    GO TO 30
 42 SS=BB0*D(1)
    GO TO 30
 45 SS=(BB0+BA0)*D(1)
 30 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE GRAPH
      COMMON N,X(250),Y(250),ASY(250),SYMB(5),ANY(8),ANX(8),ACR(5,5),FAM
     1NA(5),MA,SYMBB,KY,KX,SY1,SY2,IVY,IVX,SPS,ATT(9)
      DIMENSION B(101),IT(101)
      DOUBLE PRECISION SXX(10),C(2),WY,WX,RX,RY,SX,SY
      WRITE(3,99)(ANY(I),I=1,8)
   99 FORMAT(1H1,7A4,A2)
      CALL SORT
      CALL LIMIT (XMIN,XMAX)
      C(1)=Y(1)
      C(2)=Y(N)
      CALL SCALE (KY,C,IVY,RY,SY)
      C(1)=XMIN
      C(2)=XMAX
      CALL SCALE (KX,C,IVX,RX,SX)
      NO=5
      KAA=KX-1
      DO 1 J=1,101
      IT(J)=0
    1 B(J)=SYMBB
      WY=SY-RY/2
      JZ=1
      OUT=0
      DO 2 I=1,N
      KO=0
    5 IF(Y(I)-WY) 7,7,3
    3 KO=1
      WX=SX+RX/2
      DO 4 J=1,KAA
      IF(X(I)-WX) 6,4,4
    4 WX=WX+RX
      J=KX
    6 B(J)=ASY(I)
      IT(J)=IT(J)+1
      IF(I-N) 2,18,18
   18 OUT=1
    7 IF(KO-0) 11,10,11
   11 DO 12 J=1,KX
      IF(IT(J)-1) 12,12,13
   13 IF(IT(J)-9) 14,14,15
   14 IND=IT(J)
      B(J)=ATT(IND)
      GO TO 10
   15 B(J)=SPS
   12 CONTINUE
   10 IF(NO-5) 94,95,95
   94 WRITE(3,100)(B(J),J=1,101)
  100 FORMAT(1H ,15X,1H-,101A1)
      NO=NO+1
      IF(OUT) 90,90,2
```

```
 95 IF(JZ-1) 97,96,97
 97 SY=SY-(5*RY)
 96 WRITE(3,102) SY,(B(J),J=1,101)
102 FORMAT(1H ,F12.4,3X,1H-,101A1)
    JZ=2
    NO=1
    IF(OUT) 90,90,2
 90 DO 8 J=1,101
    IT(J)=0
  8 B(J)=SYMBB
    WY=WY-RY
    GO TO 5
  2 CONTINUE
    DO 9 I=1,KX
  9 B(I)=SY1
    NB=0
    KZ=KX/10
    KZ=(KZ*10)+1
    DO 40 I=1,KZ,10
    B(I)=SY2
    NB=NB+1
    IF(NB-1) 42,41,42 .
 41 SXX(NB)=SX
    GO TO 40
 42 SXX(NB)=((I-1)*RX)+SX
 40 CONTINUE
    WRITE(3,101) (B(J),J=1,101)
101 FORMAT(1H ,15X,1H-,101A1)
    WRITE(3,103) (SXX(I),I=1,NB)
103 FORMAT(1H0,9X,11(2X,F8.2))
    WRITE(3,106)(ANX(I),I=1,8)
106 FORMAT(1H ,28X,7A4,A2)
    WRITE(3,104)(FAMNA(I),I=1,5)
104 FORMAT(1H0,/,16X,'SYMBOL',5X,'CURVE IDENTITY',5X,'PLOT IDENTITY',5
   1X,5A4,/)
    DO 115 J=1,MA
    WRITE(3,105) SYMB(J),(ACR(J,I),I=1,5)
105 FORMAT(1H ,18X,A1,4X,5A4)
115 CONTINUE
    RETURN
    END
```