AD-A032 579   RAND CORP SANTA MONICA CALIF   F/G 12/1
CURVES: A COST ANALYSIS CURVE-FITTING PROGRAM.(U)
SEP 76   H E BOREN, G W CORWIN   F44620-73-C-0011
UNCLASSIFIED   R-1753-1-PR   NL

1 OF 2
AD
A032579

FC (12)

R-1753-1-PR
September 1976

CURVES: A Cost Analysis Curve-Fitting Program

H. E. Boren, Jr., and Capt. G. W. Corwin

See Form 1473

A report prepared for

UNITED STATES AIR FORCE PROJECT RAND

Rand
SANTA MONICA, CA. 90406

The research described in this report was sponsored by the United States Air Force under Contract No. F44620-73-C-0011 — Monitored by the Directorate of Planning, Programming and Analysis, Deputy Chief of Staff, Research and Development, Hq USAF.

R-1753-1-PR
September 1976

CURVES: A Cost Analysis Curve-Fitting Program

H. E. Boren, Jr., and Capt. G. W. Corwin

A report prepared for

UNITED STATES AIR FORCE PROJECT RAND

Rand
SANTA MONICA, CA. 90406

PREFACE

The CURVES computer program described in this report is an outgrowth of Rand's military cost analysis research activity. The program provides a user-oriented tool for estimating by least-squares procedures the parameters and statistical characteristics of several equations commonly used in the derivation of cost-estimating relationships.

A central feature of cost analysis research is the development of predictive relationships by which, for example, the costs of new military equipment or activities can be estimated from data on past equipment and activities. Although several standard computer programs are available for curve-fitting and statistical analysis of data, they often are not well suited to the purposes of cost analysis. Most standard statistical programs are designed to accommodate very large data sets and to provide a wide range of appropriate statistical tests. The typical cost analysis application, however, involves few data points (usually less than 100) and requires rather selective curve-fitting routines and statistical tests, circumstances that make the standard programs both cumbersome and fairly expensive to operate. Moreover, no single standard statistical program is likely to include all of the functional forms most useful to the cost analyst. Hence, a special-purpose program such as CURVES is both more convenient and more economical to operate than a standard program.

The CURVES program described here is a substantial updating and extension of the program in an earlier Rand report: RM-5762-PR, *CURVES: A Five-Function Curve-Fitting Computer Program* (December 1968), by one of the present authors, H. E. Boren, Jr. The new CURVES program adds three functions (logarithmic-linear and two semilogarithmic-linear forms) and several new statistical and operational features.

The CURVES program was developed as a by-product of research on the cost of advanced military aircraft and missiles. Its use is not restricted to advanced hardware, however, or even to cost analysis applications. It should be useful to analysts throughout the Air Force and elsewhere in the Defense Department who are concerned with describing causal relationships in functional form. This report was undertaken as part of the Project RAND research task entitled "Cost Analysis Methods for Air Force Systems."

Every effort has been made to remove errors from the CURVES program described in this report. However, no guarantee, expressed or implied, is made as to either the numerical or the logical accuracy of the program. Information concerning any errors or difficulties found in the use of the program or documentation will be greatly appreciated by the authors.

During the period when this report was prepared, Captain Gerald W. Corwin was on duty at The Rand Corporation in the Management Sciences Department. He is at present with the Cost Analysis Division, Directorate of Management Analysis, Office of the Comptroller of the Air Force, Headquarters United States Air Force.

This report supersedes R-1753-PR, which was first distributed in December 1975. It is being reissued in its present form to acquaint users with several recent major modifications to the CURVES program. The modifications are discussed in Appendix C, and the updated program listing is provided in Appendix D.

SUMMARY

This report describes an extension of a FORTRAN-IV curve-fitting (regression analysis) computer program (CURVES) developed in 1968 to facilitate the derivation of cost-estimating relationships for advanced military equipments. The program makes least-squares determinations of the parameters of any of eight types of equations selected by the user, given a set of observations on the dependent and independent variables of interest. The types of equations that can be fitted using CURVES are: linear, quadratic, power, asymptotic-power, exponential, logarithmic-linear, and two types of semilogarithmic-linear. Except for the quadratic and asymptotic-power equations, up to seven independent variables may be used.

Because of the importance of learning curve theory in cost analysis work, CURVES has been expanded to include logarithmic-linear as well as semilogarithmic-linear functions. Although the power form counterpart of the logarithmic-linear equation was included in the previous version, estimates of the parameters obtained from the regressed logarithmic-linear equation are often preferred to those obtained from the regressed power equation.

Other new features have been added. A correlation matrix of the input data is provided for all fitted equations using more than one independent variable, as well as a variance-covariance matrix of the estimated coefficients. Also included are standard errors and Student's t-ratios of the parameters, significance levels, beta coefficients, and the Durbin-Watson statistic. A plot routine is incorporated for providing various plots of the data. The plots available to the user are: (a) residual Y versus fitted Y, (b) observed Y versus fitted Y, and (c) observed Y versus any one of the independent variables. In the case of a single independent variable, plot (c) can be used to provide a plot of the regression equation. The program is fairly small (about 92,000 bytes of core), fast in execution time, and hence cheap to operate.

Section I describes the overall features of CURVES and the options available to the user. Section II discusses the equations available for regression in the program, including an examination of the nonlinear ones,

which require special methods for solution. The section also contains a brief summary of the statistics used in the program. Section III presents specific details of the operation of the program, including the options available to the user. Section IV describes the program outputs. A sample output for four runs is shown, together with a card listing of the deck setup required for the runs.

Appendices A and B treat the mathematical considerations relating to nonlinear-least-squares solutions. Appendix C discusses recent modifications that have been made to CURVES, and Appendix D presents an updated listing of the CURVES computer program.

ACKNOWLEDGMENTS

The authors would like to express their appreciation to Rand colleague D. C. Kephart for his helpful suggestions and comments concerning the text. Particular thanks are due to Gus Haggstrom, also of Rand, not only for his careful review and constructive criticism of the text material but also for suggesting a programmable method for calculating standard error statistics for the nonlinear cases.

CONTENTS

FIGURES

TABLES

I. INTRODUCTION

One of the central tasks of cost analysis is the development of cost-estimating relationships, predictive models that mathematically describe the cause and effect connections affecting the resources (costs) required to produce given outputs. In military cost analysis, one often needs to estimate approximate relationships that describe the cost of advanced aircraft in terms of weight, speed, production rate, and similar parameters. One of the several means by which estimating relationships may be derived is through application of curve-fitting and statistical analysis techniques to empirical (historical) data. The CURVES computer program described in this report is specifically designed to provide the computational facility, mathematical equations, and descriptive statistics most often needed in cost analysis.

The widespread use of similar techniques in many fields of research has led to development of numerous standard curve-fitting and statistical computer programs. Most of these standard programs, which provide many desirable features and options, often are not well suited to the needs of cost analysts. The typical cost analysis problem involves small amounts of data, often fewer than 100 data points. Most standard programs are intended to accommodate very large data sets. One consequence is that they frequently are cumbersome and expensive to operate, involving large computer core storage needs and frequent disk operations, when applied to small data sets. Moreover, such programs generally do not treat the nonlinear equations encountered in cost analysis; and the CURVES program is intended to provide a more economical, more convenient, and better tailored tool for cost analysis. Although it was developed in the context of research on the cost of advanced military aircraft and missiles, it is applicable to a much wider range of situations in which mathematically described causal relationships are needed.

The CURVES program was written originally for the purpose of making ordinary least-squares determinations for the parameters of five types of equations: linear, quadratic, asymptotic-power, and exponential.[1]

[1] H. E. Boren, Jr., *CURVES: A Five-Function Curve-Fitting Computer Program*, The Rand Corporation, RM-5762-PR, December 1968.

This report presents an updated version of CURVES that is much more powerful and flexible, yet cheaper to operate, than the previous version. In addition to the original five equations, three logarithmic equations have been added. They are: Ln Y vs Ln X (logarithmic-linear), Ln Y vs X (semilogarithmic-linear), and Y vs Ln X (semilogarithmic-linear).[1] Except for the quadratic and asymptotic-power equations, all equations may now be fitted using up to *seven* independent variables. In addition, values of the Y-intercept may be prespecified for all equations except the power and exponential. The program has been rewritten to be as user-oriented as possible in terms of input procedures. CURVES is a fairly small program (about 92,000 bytes[2] of core), is very fast in execution time, and is designed to minimize disk input/output operations. It is currently in use on the Rand IBM 370/158 computer but is adaptable to any computer system that accepts FORTRAN and has enough core capacity to handle the program.

A plot routine has been added to provide the following plots: (a) residual Y versus fitted Y, (b) observed Y versus fitted Y, and (c) observed Y versus any one of the independent variables. The routine may also be used to plot the regression equation for a one-independent-variable case. Plots use letter and numeral symbols so that each data point may be easily identified.

The statistics calculated in CURVES include those relating to "goodness-of-fit measures," such as sum of squares of Y residuals, total sum of squares, coefficient of determination (R^2), standard error of estimate of Y (SEY), coefficient of variation (ratio of SEY to sample mean of Y), mean of absolute relative deviations of Y, and the F-statistic. Also included are standard errors of the parameter estimates, Student's t-ratios, significance levels,[3] beta coefficients, and the Durbin-Watson statistic. The printout of significance levels is very

---

[1]In this report the abbreviation "Ln" is used to denote a natural logarithm (to base e = 2.71828...).

[2]This includes about 25,000 bytes for two variable-dimensioned arrays, whose sizes can be changed to suit the user's needs.

[3]Formulas for calculating significance levels were obtained from a study at Rand by D. Tihansky and F. Timson in April 1972.

useful because it obviates the need to obtain the values from a Student's t-table. Means and standard deviations of the dependent and independent variables are printed as well as a correlation matrix in the multivariate case. The variance-covariance matrix of the estimated coefficients is now printed.

CURVES can treat up to several hundred data points depending on the type of equation being fitted, the number of independent variables being used, and whether plotting is done. A set of data needs to be entered only once even if several regressions are to be run on it. A variable-format procedure is provided the user so that data may be entered in any order on the input cards. Data may also be entered from tape or disk provided that the data are in the appropriate format. CURVES also provides for variable transformations as discussed in Appendix C.

The CURVES program is written in FORTRAN-IV (G/H level) except for one assembler subroutine that permits reading the data from memory; however, the subroutine is not required for normal program operation. If the assembler subroutine cannot be made available, the user can use a "scratch" disk for data storage and retrieval.

One major change in this edition of CURVES is that the Gauss-Jordan method of solving simultaneous equations is used instead of Cramer's Rule. All solutions are made in double-precision arithmetic either through standard algebraic methods for the linear, quadratic, and logarithmic equations or through iterative methods for the other equations.

II. PROGRAM CONSIDERATIONS

EQUATION TYPES

The equations available in CURVES were chosen principally on the basis of their application to the derivation of cost analysis estimating relationships. The Y-intercept value A (or Ln A for the logarithmic-linear equation) may be specified for all equations except the power and exponential.[1] The equations are:

1. Linear

$$Y = A + B \cdot X1 + C \cdot X2 + \ldots + H \cdot X7,$$

2. Quadratic

$$Y = A + B \cdot X1 + C \cdot X1^2,$$

3. Power

$$Y = A \cdot X1^B \cdot X2^C \cdot \ldots \cdot X7^H,$$

4. Asymptotic-Power

$$Y = A + B \cdot X1^C,$$

5. Exponential

$$Y = e^{(A + B \cdot X1 + C \cdot X2 + \ldots + H \cdot X7)},$$

6. Logarithmic-Linear--Ln Dependent vs Ln Independent

$$\mathrm{Ln}\ Y = \mathrm{Ln}\ A + B \cdot \mathrm{Ln}\ X1 + C \cdot \mathrm{Ln}\ X2 + \ldots + H \cdot \mathrm{Ln}\ X7,$$

7. Semilogarithmic-Linear--Ln Dependent vs Independent

$$\mathrm{Ln}\ Y = A + B \cdot X1 + C \cdot X2 + \ldots + H \cdot X7,$$

8. Semilogarithmic-Linear--Dependent vs Ln Independent

$$Y = A + B \cdot \mathrm{Ln}\ X1 + C \cdot \mathrm{Ln}\ X2 + \ldots + H \cdot \mathrm{Ln}\ X7,$$

---

[1] In this report the dependent variable is represented by Y and the independent variables by X1, X2, ..., X7. When only one independent variable is considered, X1 is used.

where Y = dependent variable,

X1, X2, ..., X7 = independent variables,

A, B, C, ..., H = parameters to be estimated by least-squares methods,

Ln = natural logarithm, base e.

EQUATION CHARACTERISTICS

Examples of types of curves that can be fitted in the program are shown in Fig. 1.

Linear (1)

The linear form is the simplest treated here. Its characteristics are well known and, in our opinion, need no further elaboration. The user has the option of specifying the Y-intercept A.

Quadratic (2)

Sometimes the quadratic equation is used to represent points that lie along a parabola. However, one must be aware that a quadratic equation always has a maximum or minimum point (vertex). This means that the effects of changes in the independent variable (X1) on the dependent variable (Y) are different in sign on either side of the vertex. The coordinates of the vertex are printed in the output. Again, the user has the option of specifying the Y-intercept A.

Power (3)

The power equation is one of the more common equations used in cost analysis work. A plot of its logarithmic counterpart, the logarithmic-linear form, is known as the "learning curve" or "improvement cost curve." Conceptually, a regression of the power form does not result in the same estimates of the parameters as a regression of its logarithmic-linear counterpart. Appendix A discusses the differences between the power and logarithmic-linear regressions. For positive exponent B, the graph of the power equation always passes through the origin. Therefore, it should never be used if a positive Y-intercept is desired or logically required. For negative B, the equation is undefined at X1 = 0, has a negative slope, and approaches zero asymptotically as X1 becomes infinite.

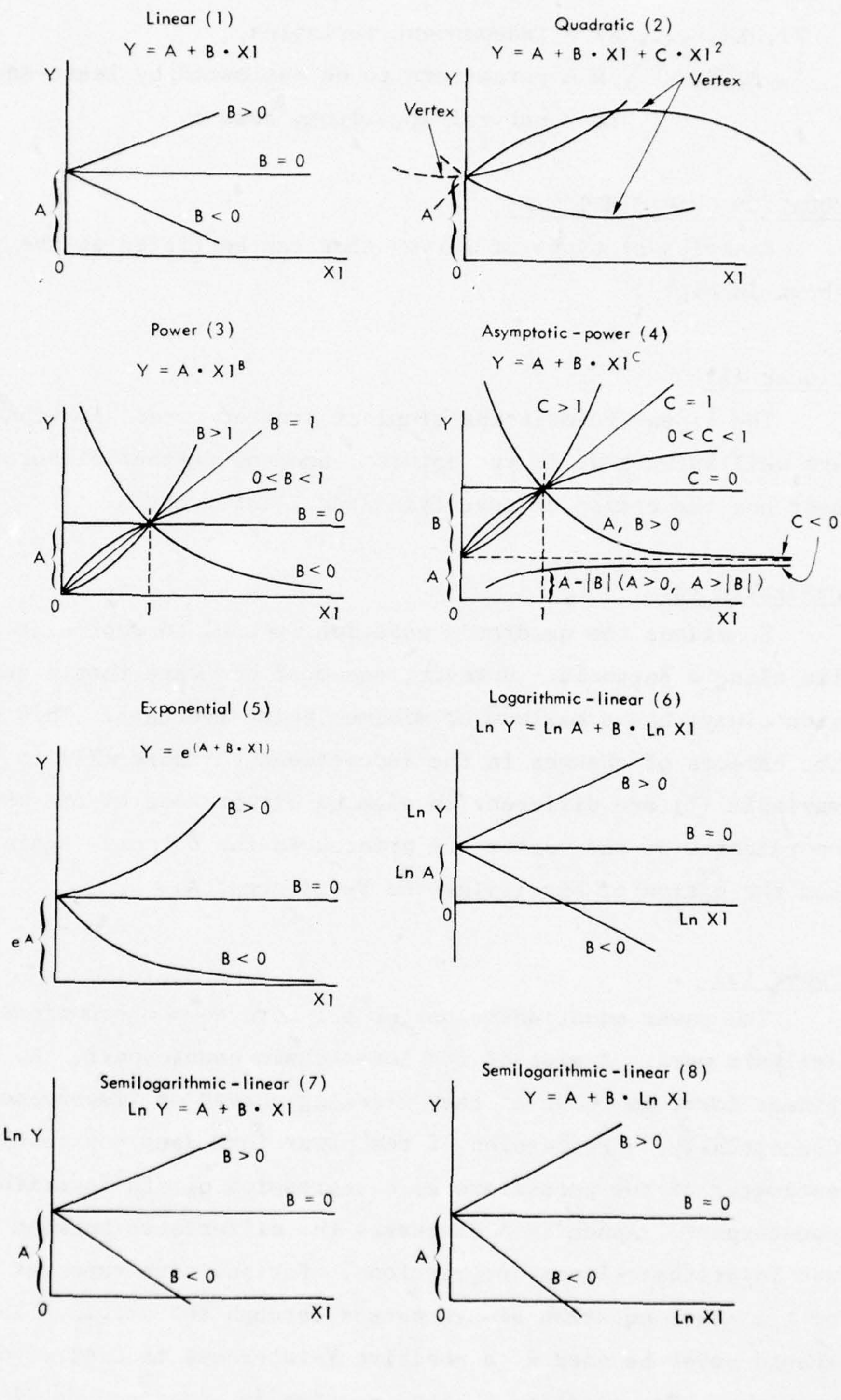


Fig. 1--Examples of curves used in program for a one-independent-variable case

Asymptotic-Power (4)

An examination of the graph of the asymptotic-power equation shows that the curve has a horizontal asymptote of Y = A for negative C. That is, as X1 becomes large, the second term $(B \cdot X1^C)$ approaches zero, and hence the value of Y tends to A. Consequently, there is a leveling-off effect for negative C. This equation may thus be used to represent points that lie along a curve either increasing or decreasing to a horizontal asymptote. Like the power equation, this equation is undefined at X1 = 0 for negative C. For positive C, the Y-intercept is equal to A. As X1 becomes large, the second term $(B \cdot X1^C)$ ultimately becomes large compared with A, and therefore the equation behaves like the power function $(B \cdot X1^C)$ in this region of X1.

Exponential (5)

The exponential form is used to represent points that lie along a curve having a positive Y-intercept (e^A). As X1 increases, the graph of the equation rises for $B > 0$ and falls for $B < 0$. In the latter case, the X1-axis is the asymptote of the curve. The logarithmic counterpart of the exponential equation is the semilogarithmic equation in which Ln Y is a linear function of X1. However, for the same reasons as stated for the power and logarithmic-linear cases, a least-squares regression of the exponential form produces different estimates of the parameters than a regression of its semilogarithmic-linear counterpart.

Logarithmic-Linear (6)

The logarithmic-linear equation, also known as the learning curve, is an equation in which the logarithm of the dependent variable is a linear function of the logarithms of the independent variables. In the program, it is fitted using the same technique as for the linear equation. The constant term may be specified in the form of Ln A.

The "learning" process is a phenomenon that prevails in many industries, and its existence has been verified by empirical data and controlled tests. The basis of learning curve theory is that each time the total quantity of items produced is increased by a constant percentage, the cost per item, or average cost of all items produced, is reduced by some constant percentage. If the number of items produced is doubled, then the percentage to which the cost is reduced is known as the learning curve slope. For example, if the number of items is increased from 120 to 240, and the cost reduces from \$100 to \$80, then the learning curve slope is 80 percent. Equations of this type may be applied either to the cost per Nth item produced (unit cost curve) or to the average cost for the first N items produced (cumulative average cost curve).

The learning curve slope (S) can be expressed as a function of the exponent (B) as follows. From the above definition

$$S = \frac{Y_{2N}}{Y_N} ,$$

$$S = \frac{A \cdot (2N)^B}{A \cdot N^B} ,$$

$$S = 2^B ,$$

or

$$B = \frac{\text{Log } S}{\text{Log } 2} ,$$

where S = learning curve slope (decimal),
B = exponent of quantity,
N = quantity,
Y = dependent variable (cost, manhours, people, etc.),
Log = logarithm to any base.

With the use of the power and logarithmic functions and the plot routine provided in CURVES, the user now has the capability to examine learning curve regressions and select the equation most appropriate for the set of data under study.

Semilogarithmic (7, 8)

The semilogarithmic equations exist in two forms. In one form, the logarithm of the dependent variable is a linear function of the independent variables. In the other, the dependent variable is a linear function of the logarithms of the independent variables. The graphs of the two equations produce straight lines on rectangular coordinate paper or, in terms of X and Y, straight lines on logarithmic paper, provided that the proper axis is scaled in logarithms. The constant A may be prespecified in fitting equations using these functional forms. Note that for the logarithmic-linear equation (Eq. (6)), the constant term is specified as Ln A, whereas for the semilogarithmic cases, it is specified as A.

NONLINEAR-LEAST-SQUARES ESTIMATES

Least-squares estimates of the parameters of an equation are always unique with a closed, algebraic solution provided the equation is *linear* with respect to all of its *parameters*. Therefore, for this program, regressions of the linear, the quadratic, and the three logarithmic equations produce such estimates of the parameters.[1] However, the power, asymptotic-power, and exponential equations are not linear in terms of all of their parameters. Thus, least-squares estimates of their parameters cannot usually be obtained by simple, algebraic methods and, as shown later, may not represent absolute minimums of the sum of squares of Y residuals. They must be obtained in some other way, usually through some type of iterative procedure. (The general principles of such procedures and other mathematical considerations relating to nonlinear-least-squares regressions are presented in Appendix A.)

For the power and exponential equations, a modified Gauss-Newton method is used, in which initial estimates of the parameters are obtained from the logarithmic-linear regressions; and then corrections, guaranteed to produce convergence to a solution, are applied to those initial estimates. For all iterative procedures, a solution is reached when the

---

[1]Under this definition, the quadratic function is considered to be a linear function because it is linear with respect to the parameters that are to be estimated.

ratio of the value of each parameter to its value corresponding to the previous iteration differs from unity by some predetermined value ($\leq 10^{-7}$ or as otherwise specified).[1]

Least-squares estimates of the parameters of the asymptotic-power equation are based on another type of iterative procedure because there appears to be no easy way to obtain the initial guesses of the parameters required for the modified Gauss-Newton method. This procedure is treated in Appendix B. Because the modified Gauss-Newton method cannot be used in this case, the equation is restricted to one independent variable.

STATISTICAL CONSIDERATIONS

Program statistics contain the standard measures relating to goodness of fit, such as sum of squares of residuals, sum of squares total, coefficient of determination (R^2), standard error of estimate of Y (SEY), coefficient of variation (ratio of SEY to sample mean of Y), mean of absolute relative deviations of Y, F statistic, and the Durbin-Watson statistic. The relative deviation of Y at the ith point is the ratio of the Y residual at that point to the observed value of Y. Also included are standard errors of the parameters, t-ratios, significance levels, and beta coefficients. Means, standard deviations, and correlation coefficients are printed for the input data.

The standard errors of the parameters for the nonlinear cases (power, asymptotic-power, and exponential) are calculated as follows. In obtaining the least-squares estimate of the parameters for the power and exponential equations, a matrix of partial derivatives with respect to the parameters is calculated and inverted during each iteration in order to obtain correlations to the parameter values. As the iterations converge, the square roots of the diagonal terms of the inverted matrix multiplied by the standard error of estimate of Y converge to the standard errors of the parameters obtained by the least-squares estimates. For the

---

[1]This procedure is described in detail in C. A. Graver and H. E. Boren, Jr., *Multivariate Logarithmic and Exponential Regression Models*, The Rand Corporation, RM-4879-PR, July 1967. The term "exponential" there is equivalent to the term "power" in this report.

asymptotic-power equation for which the Gauss-Newton method is not used, the inverted matrix of partial derivatives is obtained only after the estimates of the parameters are computed using another type of iterative procedure discussed in Appendix B.

The Durbin-Watson statistic is used to test for serial correlation.[1] It is based on successive differences in the Y residuals. Therefore, the statistic may not be useful unless the data are ordered in some meaningful way.

The statistics that are printed out by CURVES are defined in Table 1.

---

[1] J. Durbin and G. S. Watson, "Testing for Serial Correlation in Least Squares Regression II," *Biometrika*, Vol. 38, 1951.

Table 1

STATISTICAL EQUATIONS USED IN PROGRAM

Statistic	Equation
Degrees of freedom for error	DF1 = N - M where N = number of data points M = number of parameters estimated
Degrees of freedom due to regression	DF2 = number of independent variables
Total degrees of freedom	DFT = DF1 + DF2
Sum of squares total (unspecified Y-intercept)	$SST = \sum_{i=1}^{N} (Y_i - \bar{Y})^2$, where Y_i = Y value for ith observation $\bar{Y}$ = mean of observed Y values
Sum of squares total (specified Y-intercept A)[a]	$SST = \sum_{i=1}^{N} (Y_i - A)^2$,
Sum of squares of residuals	$SSE = \sum_{i=1}^{N} (Y_i - Y_{ci})^2$, where Y_{ci} = fitted value for ith observation
Standard error of estimate	$S = \sqrt{\frac{SSE}{DF1}}$

Table 1 -- (cont.)

Statistic	Equation
Coefficient of variation	$CV = S/\bar{Y}$
Coefficient of determination	$R^2 = 1 - \frac{SSE}{SST}$
Relative deviation for ith residual	$D_i = \frac{Y_i - Y_{ci}}{Y_i}$, $Y_i \neq 0$
Mean of absolute relative Y deviations[b]	$DM = \frac{\sum_{i=1}^{N} \lvert D_i \rvert}{N}$
Standard deviation of input variables	$SDEV = \sqrt{\frac{\sum_{i=1}^{N} (V_i - \bar{V})^2}{N-1}}$, where V_i = value of input variable under consideration $\bar{V}$ = mean value of input variable
F value	$F = \frac{DF1}{DF2} \cdot \left(\frac{R^2}{1-R^2}\right)$
Student's t-ratio of parameter estimates	$t = \frac{B}{SE}$, where B = parameter estimate SE = standard error of estimated parameter

Table 1 -- (cont.)

Statistic	Equation
Level of significance relating to t-ratio	$$SIGLEV = 1 - \frac{\Gamma\left(\frac{DF1+1}{2}\right)}{\sqrt{\pi \cdot DF1} \cdot \Gamma\left(\frac{DF1}{2}\right)} \cdot \int_{-t}^{t} \left(1 + \frac{X^2}{DF1}\right)^{-(DF1+1)/2} dX ,$$ where Γ denotes the gamma function. (Above formula approximated by series expansion as used in a study by D. Tihansky and F. Timson at Rand in April 1972.)
β-coefficient of regression coefficient on variable V	$$\beta = B \cdot \sqrt{\frac{\sum_{i=1}^{N} (V_i - \bar{V})^2}{\sum_{i=1}^{N} (Y_i - \bar{Y})^2}} ,$$ $$= B \cdot \frac{SDEV_V}{SDEV_Y}$$
Durbin-Watson statistic	$$DW = \frac{\sum_{i=2}^{N} [(Y_i - Y_{ci}) - (Y_{i-1} - Y_{ci-1})]^2}{SSE}$$

[a] For the logarithmic-linear case, the Y-intercept is Ln A.

[b] If any Y_i is zero, the corresponding D_i cannot be calculated. In such a case the summation is reduced to fewer than N data points.

III. INPUT PROCEDURES

The flow of operations within the program is depicted in Fig. 2. The program is structured so that many sets of data may be entered in which each set constitutes a run.[1] Table 2 lists the maximum number of data points that can be used for each equation. As was stated previously, the maximum number depends on the type of equation being regressed, the number of independent variables being used, and whether plots are to be made. As soon as each set is read in, the program operates on that set before proceeding to the next set of input data. Each set of data may be entered on a separate deck of cards.[2] However, several or all of the sets of data, if space on the cards permits, may be entered on one deck of cards, thus effecting considerable savings in the use of cards and in the effort of duplicating a deck of cards containing data for several runs. A variable format procedure is used, allowing much flexibility in the format of the input data. Data may also be entered from disk or tape provided all format requirements are met.

Table 3 lists the types of cards used for a job consisting of one or more runs (steps 6-10 may be repeated as often as desired).

TITLE CARD

The Title card must be entered for each run. It contains the title for the current run (which is listed at the top of each output page) and may consist of any valid characters; all 80 columns may be used. An example of a Title card is shown in Fig. 3 as it might appear in a card arrangement for a CURVES run.

CONTROL CARD

The Control card is the second input card for each regression and

---

[1] A series of one or more runs is defined here as a job constituting one session on the computer.

[2] The use of the word card in this report, while usually meaning the normal punch card, can in the case of data "cards" or blank "card" mean physical record (card, disk, or tape).

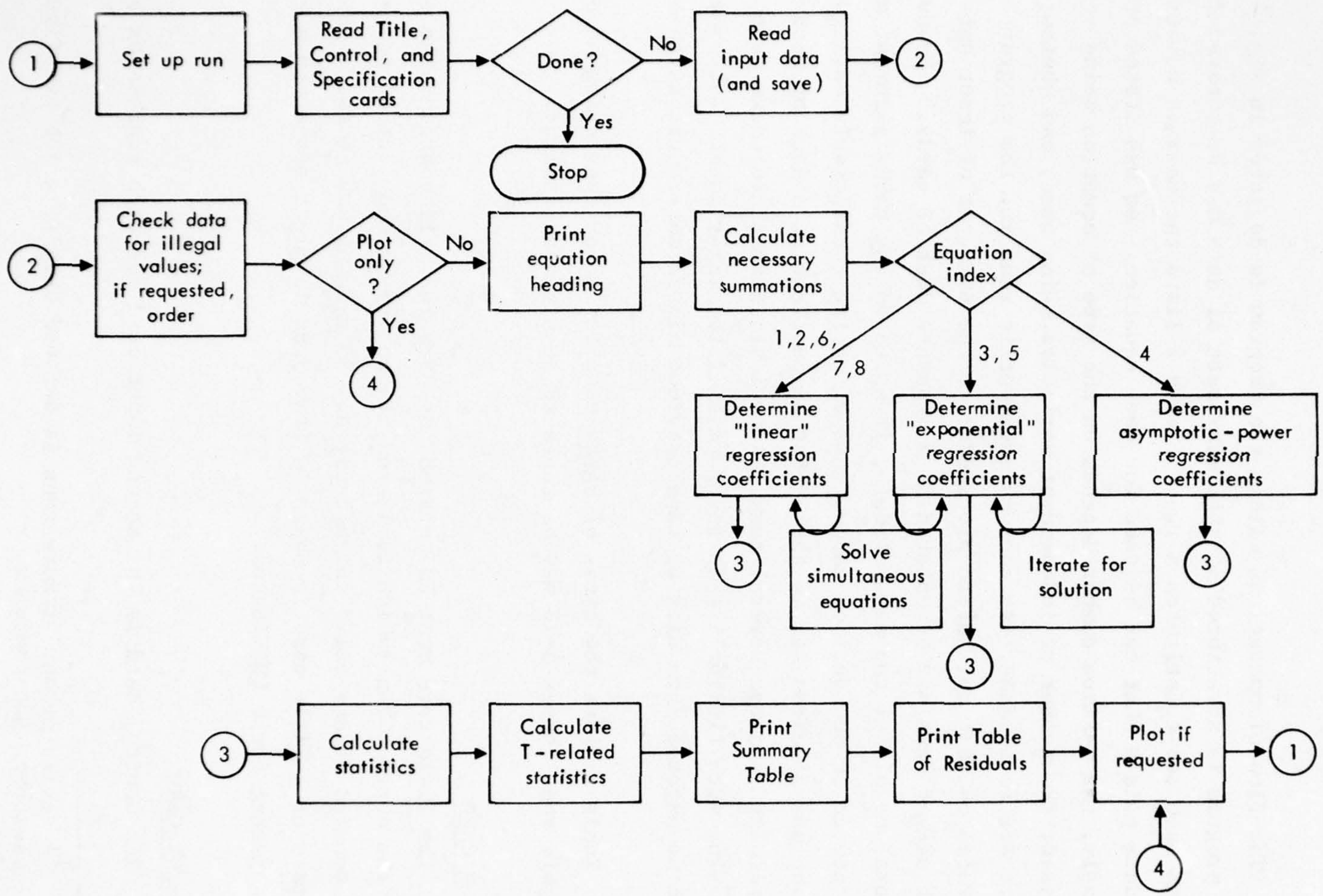


Fig. 2--Flow of operations

Table 2

MAXIMUM NUMBER OF DATA POINTS BY EQUATION, PLOT OPTION, AND NUMBER OF INDEPENDENT VARIABLES

Equation	Number of Independent Variables	Without Plotting		With Plotting	
		Without ID[a]	With ID	Without ID	With ID
Linear	1	551	441	381	306
	3	367	314	254	218
	7	220	200	152	138
Quadratic[b] Exponential Semilogarithmic-linear (Ln dependent variable)	1	441	367	306	254
	3	314	215	218	190
	7	200	183	138	126
Power Asymptotic power[b] Ln-linear Semilogarithmic-linear (Ln independent variable)	1	367	314	254	218
	3	220	200	152	138
	7	121	115	84	79

[a] ID indicates data-point identifier.

[b] For one independent variable only.

NOTE: This table is based on

$$N_{max} = \frac{C_t - p \cdot C_p}{k} - 1 ,$$

where

N_{max} = maximum number of data points

C_t = maximum number of cells of data storage = 2211

C_p = number of cells reserved for plotting = 676

p = plotting requirement ($p = 0$ for no plotting; $p = 1$ for plotting)

k = number of columns required for data; columns are required for:

(a) dependent variable Y (always)
(b) each independent variable (always)
(c) identifier (only if used; see Control Card, columns 2 to 10)
(d) computed Y (always)
(e) residual Y (always)
(f) Ln Y for all equations except 1 and 2
(g) Ln X for each independent variable for equations 3, 4, 6, 8

Table 3

CARD TYPES FOR CURVES PROGRAM

First Run		
1.	Title	Required.
2.	Control card	Required.
3.	Specification cards	Required; used to supply special control information such as format, labels, data location, and initial guesses for iterative solutions. A "Read" or "Read8" specification card is always required.
4.	Data cards	Required.
5.	Blank card	Required; used only at end of data cards.
Succeeding Runs		
6.	Title card	Required.
7.	Control card	Required.
8.	Specification card(s)	"Read" or "Read8" card is always required.
9.	Data cards	Required only if input was not saved from previous run.
10.	Blank card	Required only at end of input data cards.
After Final Run		
11.	Done card	Word DONE entered in Cols. 1-4 after last run. Terminates job.

is required (even if it is blank). It contains the control data to perform the desired regression. Table 4 summarizes all the information on the Control card. An example of a control card is shown in Fig. 3.

Equation Designator

Column 1 is used for the equation designator. An integer from 1 through 8 is entered to designate which equation is to be used for the regression in the run. (A zero indicates a plot-only option.) The equation designators are:

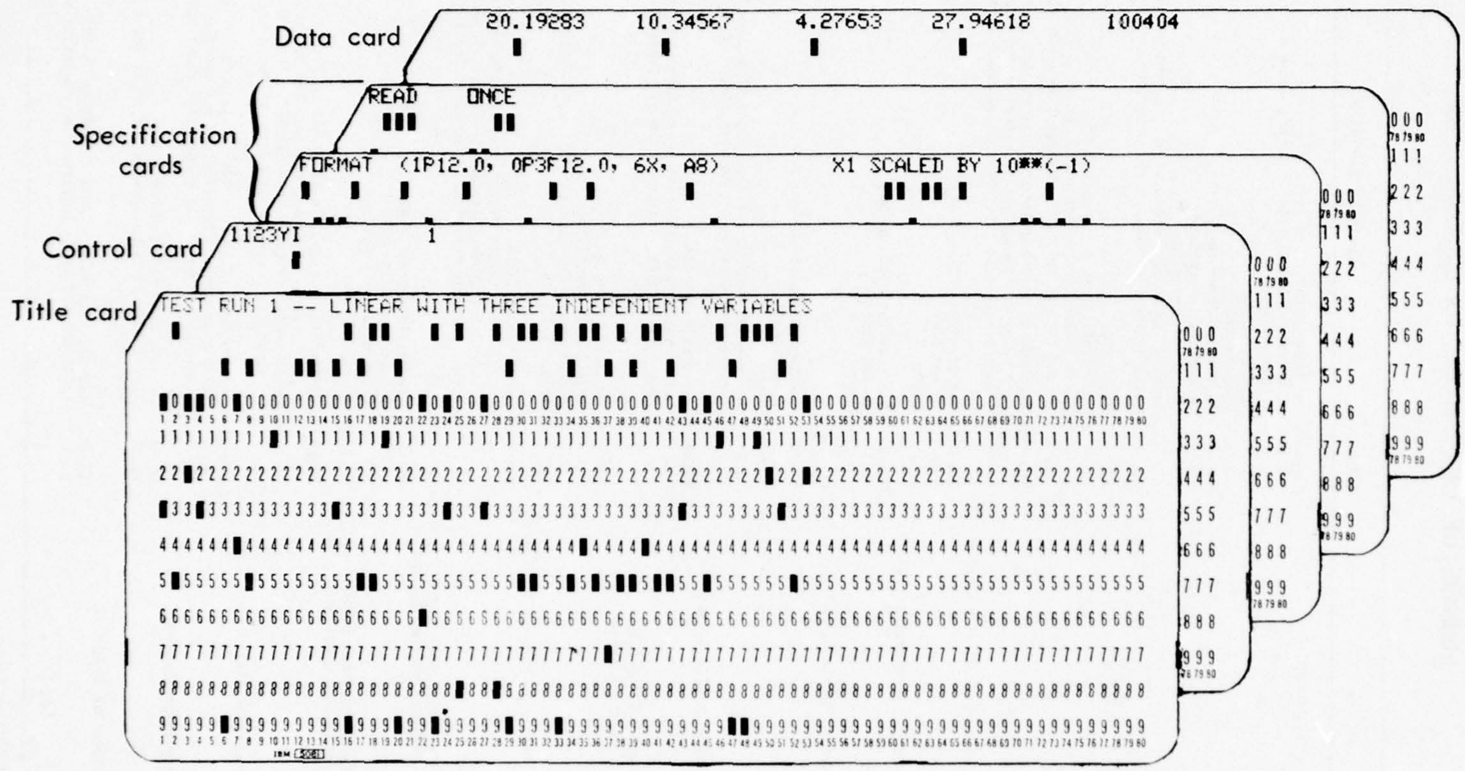


Fig. 3--Examples of Title, Control, Specification, and Data cards

Table 4

SUMMARY OF INFORMATION ON CONTROL CARD

Card Column(s)	Use	Value	Control Description
1	Equation index	Blank	Use previous equation index.[a]
		0	No regression,[b] only plot Y vs XM, where M = 1, 2, ..., 7
		1	Linear regression
		2	Quadratic regression
		3	Power regression
		4	Asymptotic-power regression
		5	Exponential regression
		6	Ln-linear regression
		7	Semilog-linear (Ln dependent vs independent) regression
		8	Semilog-linear (dependent vs Ln independent) regression
2-10	Variable order		
		I	Identifier (optional)
		Y	Dependent variable (required)
		1	First independent variable (required)
		2-7	Second through seventh independent variables (optional)
			(Above values may be in any order and indicate the order of the data fields specified on the format card. Values must be packed, left-justified; nine blanks mean "use previous variable order".)
11	Plot Y residual vs fitted Y		
		Blank	Use previous plot option.[a]
		0	Do not plot.
		1	Plot.
12	Plot Y observed vs fitted Y		
		Blank	Use previous plot option.[a]
		0	Do not plot.
		1	Plot.
13	Plot Y observed vs XM		
		Blank	Use previous plot option.[a]
		0	Do not plot.
		1-7	Plot.
		8-9	For one-independent variable regressions, plot Y vs X1 and regression equation; otherwise plot Y vs X1 only.
14	Zero/zero plot option		
		Blank	Use previous index.[a]
		0	Plot Y vs XM data with minimum rectangular grid.
		1	Plot Y vs XM data, including the additional point (0,0).
15	Zero/negative data acceptance option		
		Blank	Use previous index.[a]
		0	Reject all nonpositive data.
		1	Accept zero-value data points, reject negative values.[c]
		2	Accept all data.[c]
16	Format for Table of Residuals		
		Blank	Use previous index.[a]
		0	Print actual values rather than transformed logarithms.
		1	Print Ln-transformed variables and calculated values for Ln-linear functions. (Equation index $\geq$ 6.)
17	Order input data in ascending dependent-variable order		
		Blank	Use previous order index.[a]
		0	Do not order data.
		1	Order data.
18	Number of input cards per data-point record		
		Blank	Use previous value. Initial setting is 1 card/record (i.e., each input card has one data point).
		1-9	Value indicates the number of card images for each input record (as described in the input format statement).
19-20	Iteration limit		
		Blank	Use previous value. Initial setting is 20.
		1-99	Maximum number of iterations before aborting an iterative solution of power and exponential equations.
21-30	Specified intercept		
		Blank	Normal, unconstrained regression
		Real	Specified intercept (invalid for power and exponential equations)
31-40	Iteration tolerance		
		Blank	Use previous value. Initial setting is 10^{-7}.
		Real	Solution-difference tolerance for iterative solutions
41-80	(d)		

[a] Initial setting is zero.

[b] See description of Col. 13.

[c] Zero and negative values cannot be used safely for any equation except linear and quadratic because regressions of all other equations require logarithms.

[d] See Appendix C, p. 58.

Equation Designator	Equation
0	(Plot only)
1	Linear
2	Quadratic
3	Power
4	Asymptotic-power
5	Exponential
6	Logarithmic-linear
7	Semilogarithmic-linear (Ln Y vs X1, X2, ..., X7)
8	Semilogarithmic-linear (Y vs Ln X1, Ln X2, ..., Ln X7)

One of the above integers should be entered in Col. 1 for the first run (default is zero). If Col. 1 is blank after the first run, then the value for the previous run is used. Thus, if the same type of equation is to be used for a series of runs, its designator needs to be entered only for the first run.[1] If a zero is entered as an equation designator, no regression is run; instead, the input data are plotted as Y versus XM, where M is defined in Col. 13 as one of the independent variables being used (if Col. 13 is left blank, a plot against X1 is assumed). It is important to note that for this version of CURVES a distinction is made on the Control card between a blank and a zero (0). A blank field on the Control card always causes the program to retain the previous value of the associated indicator, except for the specified intercept (which reverts to the condition of "no specified intercept").

Order of Variables

Columns 2 through 10 indicate the order of the variables on each data card as described in the variable-format specification statement. Depending on the number of independent variables being used and on whether data-point identifiers are being used, Cols. 4 to 10 may be

---

[1]This procedure has great advantages in simplifying the input operations for a job consisting of more than one run, but it also has its drawbacks. If, for example, data for a run are removed from a block of data representing a series of regressions and then rerun, control information that should be passed on to future runs may be lost and subsequent blanks will default back to the last run with a nonblank entry.

left blank. The symbols used to show the order must be left-justified (with no imbedded blanks) and are as follows:

Symbol	Type of Data	
I	Identifier (alphanumeric) (optional)	May be in any order from left to right.
Y	Dependent variable (required)	
1	First independent variable (required)	
2,3,4,5,6,7	Second through seventh independent variables (optional)	

The independent variables are identified ordinarily and thus require that no numerical symbol be skipped; i.e., use of 3 implies that 1 and 2 exist.

Suppose that a set of data is to be entered in which values for three independent variables are located in Cols. 1-12, 13-24, and 25-36, and values for the independent variable are located in Cols. 37-48. Suppose also that an identifier (a six-digit integer) is in Cols. 55-60. Then 1, 2, 3, Y, and I (123YI) would be entered in Cols. 2-6 to show the above order for a format of "(4F12.0, 6X, A6)."

Note that if the format statement is written with tab formats (i.e., "T"), the order applies to the order of fields as defined in the format statement. Thus, for the above example, a format of "(T55, A6, T1, 4F12.0)" would require an order of I123Y.

The data-point identifiers may be up to eight characters long, with a corresponding format specification as large as A8.

Plot of Residuals vs Fitted Values

Column 11 is used to indicate whether a plot of the residuals versus the fitted values is desired. If a 1 (or any positive digit) is entered in Col. 11, the plot is generated; if a 0 (zero) is entered, the plot is not generated. As with all indices on the control card, a blank causes the previous value to be retained.

Plot of Observed Y vs Fitted Values

Column 12 is used similarly to Col. 11 to generate a plot of observed values of Y versus their fitted values.

Plot of Observed Y vs Independent Variable or Plot of Regression Equation

Column 13 is used to indicate the independent variable to be plotted with the observed Y. In this case, one of the digits 1 through M is entered, where M is the index of the independent variable to be plotted. In addition, for the case of a single independent variable, if Col. 13 contains an 8 or 9 instead of a 1, the resulting regression equation is plotted on the same graph using dots as the plot symbols. In all cases where the plotting index value is determined to be invalid, the plot is not generated.

For all plots except that of the regression equation (dots), 26 letters and nine numerals, or a total of 35, are used. The numeral zero is excluded so as not to conflict with the letter "O". For plots of data with more than 35 points, the symbols are repeated in blocks to account for all data points. As an example, for 37 data points, the points in order from 1 through 37 would be represented by A, A, B, B, C, D, E, ..., Z, 1, 2, 3, ..., 9. Each symbol used in a plot is also listed beside the corresponding data point in the Table of Residuals so that the data point can be readily identified.

Plot-size Option

Column 14 is used as a designator for a plot-size option for the Y versus X plot. Normal operation of all plotting is to determine the minimum and maximum ordinate and abscissa, and then produce the minimum size (largest scale) rectangular plot containing all the data. In the case of the Y vs X plot, especially for a positive Y-intercept, the point (0,0) can be added to the plot-points (to depict the first quadrant) by entering a 1 in Col. 14.

Data Value Acceptance

Column 15 is used as an indicator for data value acceptance. Normally, a negative value for a data point terminates the job because negative values cannot be used in some cases for the logarithmic equations and in all cases involving the power, asymptotic power, and exponential equations. A zero value causes the program to reject the data point (except the origin, which has a special meaning as described below) but to continue the processing (reading). The latter condition allows a data set to be processed even though there are some missing data. (A blank is read as zero in a floating-point format.)

In some cases, it is desirable to accept zero as a valid data value (e.g., a linear regression with a zero/one dummy variable); to do so, a 1 is entered in Col. 15. However, note that a regression of any equation except the linear and quadratic requires logarithms of some or all of the input variables. If the program attempts to take the logarithm of a zero or negative value, the job terminates. If Col. 15 contains a 2, all data values (positive, zero, and negative) are accepted except the origin, which is interpreted to signify that the reading of data is complete.

Output Status for Logarithmic Equations

Column 16 is used to indicate the output status for the logarithmic-equations--equation designators 6, 7, and 8. For each of these equations, the regression is performed on the logarithms of some or all of the variables. If the column is left blank, all logarithmic results for the Table of Residuals are printed as nonlogarithmic (antilog) data for better readability and understandability. That is, the original input data (before logarithmic transformation) are printed, the fitted Y values are exponentiated if in Ln Y form (equation designator value of 6 or 7), and the residuals and relative Y deviations are calculated based on nonlogarithmic data. To use a logarithmic model with all appropriate output in the Table of Residuals in logarithmic data, a 1 is entered in Col. 16. Regardless of the value entered in Col. 16, the Summary Table always produces statistics for the regressed equation whether or not the equation is logarithmic.

Ordering of Input Data

Column 17 is used to designate whether the input data are to be ordered from low to high values of Y. A value of 1 signifies that the data are to be ordered. For the first run a blank (or zero) signifies that the data are not to be ordered; for subsequent runs a blank signifies that the value of the order designator for the preceding run is to be used. Again, this is done so that if all runs in a series are to be either ordered or unordered, the order designator need only to be entered for the first run. To reset the order indicator to zero (no ordering), simply enter a 0 (zero).

Figure 3 also shows an example of a control card in which a linear regression is to be made on the input data (1 in Col. 1). The data are to be ordered with respect to Y (1 in Col. 17).

Number of Input Cards per Data-Point Record

Column 18 is used to indicate the number of input cards per data-point record. If the input data are entered from the card reader, the entry in Col. 18 is the number of physical (card) records per logical record (data point); for disk or tape input data, see Format Card in the next subsection under SPECIFICATION CARDS. The initial setting is one card per record; if a value of zero is entered, a value of 1 is substituted. The maximum number of cards per record is nine. This designator is used primarily to block the input data into individual records when the READ MEMORY option is used; a slash is not permitted in the variable format statement when using memory because it repeats the same record instead of advancing. This value has no use for the READ ONCE or READ8 ONCE options. The initial setting is one card per record. A blank value repeats the previous value.

Maximum Number of Iterations Allowed

Columns 19 and 20 are used to specify the maximum number of iterations to be allowed before aborting the iterative solution of the power and exponential equations. The default setting is 100.

Y-Intercept Specification

Columns 21 through 30 are used to specify the equation intercept term (regression constant). This is the A value for all equations except the Ln-linear, for which it is Ln A. The intercept for the power and exponential equations may not be specified. The desired intercept is entered in floating-point format anywhere within the field. The implied decimal point location is at the right end of the field, after Col. 30. If the field is blank, the unspecified regression equation is assumed.

Iteration Tolerance

Columns 31 through 40 are used to specify the iteration tolerance (DELTA). For regressions of the nonlinear equations, an interative solution is achieved when the ratio of the value of each parameter to its value for the previous iteration differs from unity by an amount equal to or less than DELTA. DELTA is initialized at 10^{-7}. The user may change this by specifying another constant in Cols. 31-40, preferably in decimal format, for example, 0.00000001. The implied decimal point is prior to Col. 31. Any value of DELTA less than 10^{-12} or greater than 10^{-1} is reset to 10^{-7}. The authors recommend that DELTA not be changed, eccept for very special reasons; a DELTA of 10^{-7} should be sufficient to obtain all parameters within the accuracy printed.

SPECIFICATION CARDS

There are basically four specification cards that provide control infcrmation for the run similar to the control card but that requires more space to express. They are: Format, Label, Guess, and Read. They follow the control card and may be in any order except that the Read card must be the last one. For any such card, the specification name is entered beginning in Col. 1; the information following the name should begin in Col. 9.

Format Card

The Format card indicates where the data are located on the data cards. This card must begin with the word FORMAT in Cols. 1-6 followed by a left parenthesis. A matching right parenthesis closes the format specification. Information within the parentheses must conform to the rules for FORTRAN formats. In addition, except for the alphanumeric identifiers, all input data must be in real-number (floating-point) formats. One or two continuation cards may be used, each of which must likewise contain the word FORMAT in Cols. 1-6. After the final right parenthesis of the format, the user may make any comments desired (e.g., identifying the variables the format refers to), because data beyond the last parenthesis are ignored by the program. If a Format card is omitted after the first run, format specifications are carried over from the previous run. The Format card shown in Fig. 3 could be used for the previous example, as discussed before.

Label Card

The Label card is used to identify the variable with an eight-character label. The labels are placed next to the regression parameter or variable to which they refer in the output Summary Table; they also appear as table headings in the Table of Residuals. If an alternate form (LABELS) is used, the Summary Table also contains a printout of the names of the regression variables immediately below the title. (This alternative may allow the user to use the same title with several runs, but to distinguish the runs by the variable list; it may also reduce the requirements of Title card preparation.)

If no Label (or Labels) card is used, the default (for each run) is listed below. Labels do *not* carry over from one run to the next; they must be entered each time. Table 5 shows the format of the Label card.

Guess Card

The Guess card is used to provide initial guesses of the parameter values for the iterative solutions in fitting the power and exponential equations. The normal solution procedure in the program is first to make a least-squares regression of the corresponding logarithmic-linear (or semilogarithmic-linear) equation to obtain preliminary estimates of the parameters (the values are printed out under the heading INITIAL GUESS in the Summary Table) and then iterate. The user may make the initial guesses by entering a Guess specification card in the format shown in Table 6. The implied decimal point is at the right end of each field. *The authors recommend that, except for special circumstances, the user should not attempt initial guesses for the iterative*

Table 5

FORMAT OF LABEL SPECIFICATION CARD

Card Columns	Value	Specification Description[a]
1-6	LABELb̸ or LABELS	Indicates the type of specification card
7-8		Not used
9-16	Alphanumeric	Heading for identifier, if used (default is LABELb̸b̸b̸); if no identifier, b̸b̸b̸b̸b̸b̸b̸b̸.
17-24	Alphanumeric	Heading for dependent variable (default is b̸b̸b̸Yb̸b̸b̸b̸).
25-32	Alphanumeric	Heading for first independent variable (default is b̸b̸b̸X1b̸b̸b̸).
33-40	Alphanumeric	Heading for second independent variable, if used (default is b̸b̸b̸X2b̸b̸b̸ for all functions except quadratic and asymptotic; quadratic default is b̸b̸X1**2b̸; asymptotic default is EXPONENT).
.	.	.
.	.	.
.	.	.
73-80	Alphanumeric	Heading for seventh independent variable, if used (default is b̸b̸b̸X7b̸b̸b̸).

[a] A blank column is indicated by b̸. (See pp. 21-22 for definition of variables.)

Table 6

FORMAT OF GUESS SPECIFICATION CARD

Card Columns	Value	Specification Description[a]
1-6	GUESSb̸	Indicates the type of specification card
7-8		Not used
9-16	Real	Initial guess for A
17-24	Real	Initial guess for B
25-32	Real	Initial guess for C, if used
.	.	.
.	.	.
.	.	.
65-72	Real	Initial guess for H, if used

[a] See p. 5 for description of A, B, C, ... H use.

cases. It is quite likely that by using the Guess card, it will take the program much longer to converge to the solution, if indeed it converges at all.

Read Card

The final specification card is the Read card. This card (which is a required input) signifies that the data are ready to be processed. (An alternate reading file for the data cards is discussed below under the READ8 option.) The format of the Read card is shown in Table 7.

Anything other than ONCE, MEMO, or DISK in Cols. 9-12 is equivalent to ƀƀƀƀ and will result in the data being reread from the save-data file last used (disk or memory). If the last option was ONCE, a new set of cards will be read (because no data were stored). The initial setting is ONCE. Since rereading of saved data requires no more data input, data cards are not expected nor may they be present in the input

Table 7

FORMAT OF READ SPECIFICATION CARD

Card Columns	Value	Specification Description
1-6	READƀƀ or READ8ƀ	Indicates the type of specification card; data are read in from card unit 5 (i.e., sequentially from the normal card reader) or from user-defined unit 8.
7-8		Not used
9-12		The save-data option parameter:
	ONCE	Do not save the data, just process it for this one run.
	MEMO	Save the card images in memory (≤ 100 cards) for rapid access on later runs.
	DISK	Save the card images on scratch disk unit 4 (up to the maximum number of regression data points) for access on subsequent runs.
	ƀƀƀƀ	Use saved data
13-80		Not used

stream when this option is in effect. Figure 3 shows an example of the READ ONCE option.

In the event that a data input stream other than the normal sequential card input (i.e., unit 5) is desired, the user may substitute READ8b̸ for READb̸b̸ in Cols. 1-6. This alternative sets up unit 8 for the input of data cards only (including the BLANK card). The Title, Control, and Specifications cards always come from unit 5.

Any data saved must be in card images. When using memory, data are stored in a record length (in bytes) of 80 times the number of cards per record (maximum of 720 bytes--9 cards--in the program). Therefore, in reading back such stored data from memory, the same record length must be used in the format statement; e.g., FORMAT (T200, 3F8.0).[1] A slash mark is not permitted in this case, because it repeats the same 80-byte record instead of advancing to a new record. For a scratch disk, data are stored in 80-byte records and hence must be read back the same way. Therefore, slash marks must be used in the format statement for multiple cards per record when using disk.

DATA CARDS

Each data point must contain at least a pair of values, one for the dependent variable (Y) and one for the independent variable (X1). Each set of data constituting a run must contain at least as many data points as the number of parameters being estimated and may contain up to the maximum allowable as indicated in Table 1. If the maximum is exceeded, an error message explains this fact. The location of the data on the card must be in exact agreement with the information entered on the Control and Format cards, or else the data will not be read properly. The numerical data (dependent and independent variable values) are read as real (floating-point) numbers and data-point identifiers (if used) as alphanumeric data.

---

[1] The largest repetitive skip that can be used in a FORTRAN format statement is 255 (i.e., 255X or T255). If a larger skip is required, simply break it up into two or more skips. For example, a skip of 300 columns may be entered in the format statement as 150X, 150X; or 100X, 200X; etc.

For any data card, if either the Y field or any of the X fields, but not all, is blank or contains the value zero, that card is usually skipped.[1] However, if all X and Y fields are blank (zeros), and the identifier field, if used, is blank or contains the word BLANK, the reading of input data for the run is terminated at that point (see BLANK CARD below).

If the identifier field is neither blank nor contains the word BLANK and all the data are zero, the data point is merely rejected. Thus, the user may identify places where data are needed in the data set, but for which data are not yet available.

Figure 3 also shows a data card containing data in the specific format.

BLANK CARD

Each set of data cards constituting a run must always end with a blank card. This card is used to terminate the reading of the input data for a given run. There must be as many blank cards as there are cards per record (see description under CONTROL CARD, Column 18). As an option, the user may, if saving data in memory or on disk, type the word BLANK in Cols. 1-5 of a single blank card.

DONE CARD

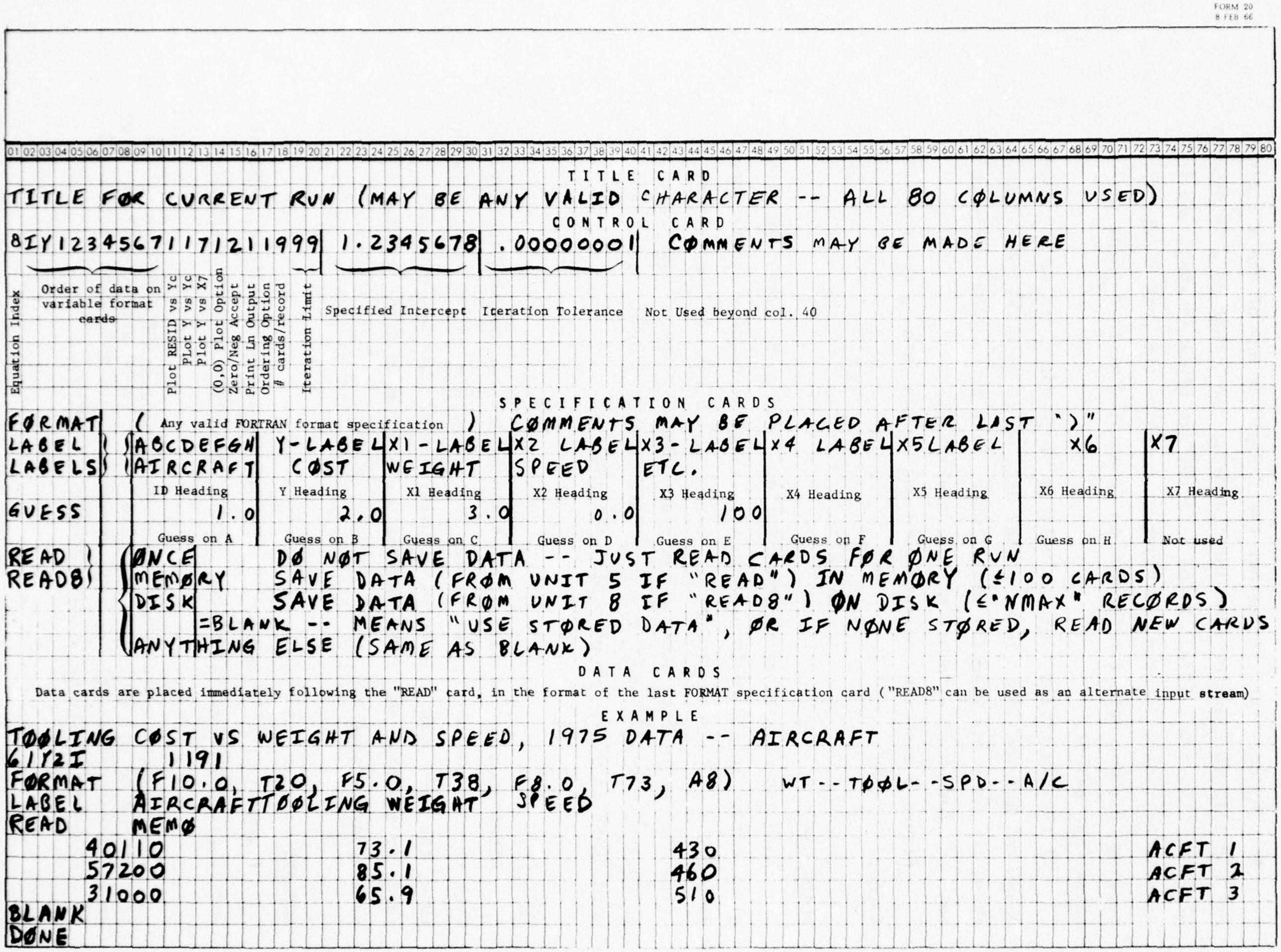
After each set of data is read and processed, the machine cycles back to read another set of data. To terminate a run or a series of runs, the word DONE is entered in Cols. 1-4 on what would be the Title card for the next run. This causes the program to print a large DONE on a separate page and stop.

SUMMARY

Figure 4 briefly summarizes the information on the Title, Control, Specification, Data, Blank, and Done cards. Note that the Read card

---

[1] As was stated previously under Data Value Acceptance, an option is provided to allow zero or negative values to be accepted by the program. Nevertheless, the default condition is to skip zero fields and to terminate the run on reading negative values.



FORM 20
8 FEB 66

01 02 03 … 80

TITLE CARD

TITLE FØR CURRENT RUN (MAY BE ANY VALID CHARACTER -- ALL 80 CØLUMNS USED)

CONTROL CARD

8IY1234567117121199 | 1.2345678 | .00000001 | CØMMENTS MAY BE MADE HERE

Equation Index
Order of data on variable format cards
Plot RESID vs Yc
Plot Y vs Yc
Plot Y vs X7
(0,0) Plot Option
Zero/Neg Accept
Print Ln Output
Ordering Option
\# cards/record
Iteration Limit
Specified Intercept
Iteration Tolerance
Not Used beyond col. 40

SPECIFICATION CARDS

FØRMAT (Any valid FORTRAN format specification) CØMMENTS MAY BE PLACED AFTER LAST ")"

	ID Heading	Y Heading	X1 Heading	X2 Heading	X3 Heading	X4 Heading	X5 Heading	X6 Heading	X7 Heading
LABEL	ABCDEFGH	Y-LABEL	X1-LABEL	X2 LABEL	X3-LABEL	X4 LABEL	X5LABEL	X6	X7
LABELS	AIRCRAFT	CØST	WEIGHT	SPEED	ETC.				
GUESS	1.0	2.0	3.0	0.0	100				
	Guess on A	Guess on B	Guess on C	Guess on D	Guess on E	Guess on F	Guess on G	Guess on H	Not used

READ / READ8
- ØNCE — DØ NØT SAVE DATA -- JUST READ CARDS FØR ØNE RUN
- MEMØRY — SAVE DATA (FRØM UNIT 5 IF "READ") IN MEMØRY (≤100 CARDS)
- DISK — SAVE DATA (FRØM UNIT 8 IF "READ8") ØN DISK (≤"NMAX" RECØRDS)
- =BLANK -- MEANS "USE STØRED DATA", ØR IF NØNE STØRED, READ NEW CARDS
- ANYTHING ELSE (SAME AS BLANK)

DATA CARDS

Data cards are placed immediately following the "READ" card, in the format of the last FORMAT specification card ("READ8" can be used as an alternate input stream)

EXAMPLE

```
TØØLING CØST VS WEIGHT AND SPEED, 1975 DATA -- AIRCRAFT
61Y2I         1191
FØRMAT   (F10.0, T20, F5.0, T38, F8.0, T73, A8)   WT--TØØL--SPD--A/C
LABEL    AIRCRAFTTØØLING WEIGHT   SPEED
READ     MEMØ
     40110           73.1             430               ACFT 1
     57200           85.1             460               ACFT 2
     31000           65.9             510               ACFT 3
BLANK
DØNE
```

Fig. 4--Summary of Title, Control, Specification, Data, Blank, and Done cards

must always be the last Specification card for any run. Figure 5 shows the order of two data decks for a series of three runs. In the figure, the second deck also contains data for the third run. Hence, it is to be saved for rereading during the third run.

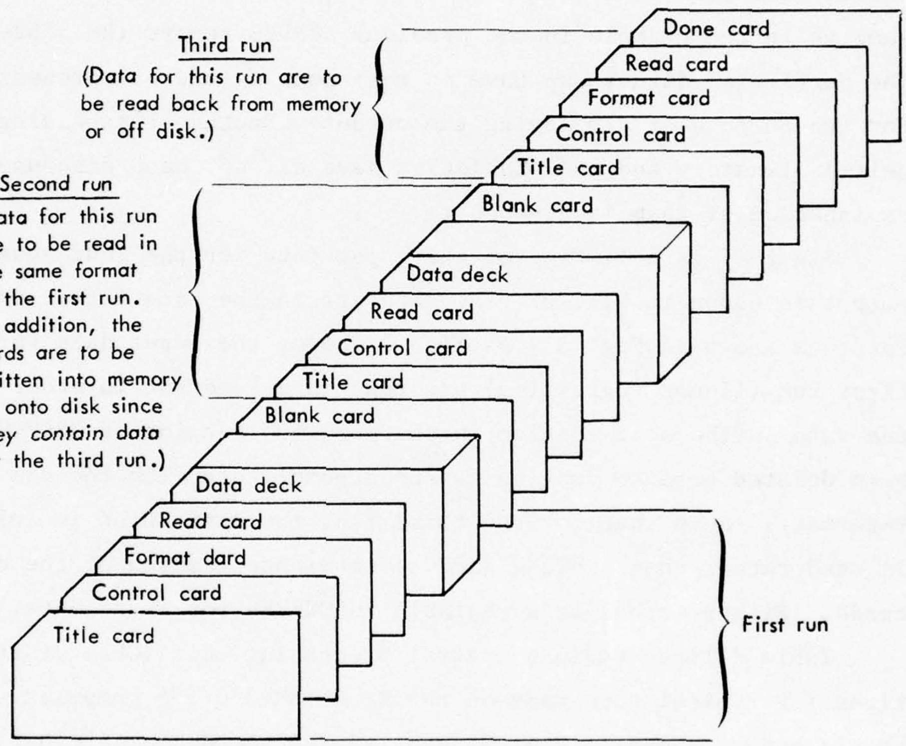


Fig. 5--Arrangement of two data card packs for three runs

IV. PROGRAM OUTPUT

Figure 6 is an example of program output for fitting linear, power, logarithmic-linear, and quadratic equations. The output also includes examples of the plots available in the program. For comparability and continuity, the input data for the first two runs are the same as in the example in the previous CURVES report (RM-5762-PR). The particular data shown have no real meaning and are presented only for the purpose of displaying the output. Because the headings are self-explanatory and the statistics have already been discussed, no explanation of them is given here.

Figure 7 is a listing of the input data for the four runs whose output is shown in Fig. 6. The data are in the same format on the cards as shown in Fig. 3. As Fig. 7 shows, the input data for the first run (linear regression) are read in a P-format in order to scale the data. (The scale option in the previous version of CURVES has been deleted because scaling can be accomplished with the use of the P-format.) Note that for the third run, the word BLANK in Cols. 1-5 is used rather than a blank card to terminate reading of the data cards. Either method is acceptable to CURVES for saved data.

Table 8 lists various central processing unit (CPU) execution times for typical runs made on the Rand IBM 370/158 computer. (The sample output shown in Fig. 6 required 2.5 seconds of execution time and 92,000 bytes of core storage,* based on an H-compiled program.)

---

*If data are stored on a scratch disk or tape, instead of in memory, additional core storage may be required.

CURVES REGRESSION ANALYSIS COMPUTER PROGRAM
(JULY 1976)

TEST RUN 1 -- LINEAR WITH THREE INDEPENDENT VARIABLES   DATE: 76272   TIME: 1350   PAGE: 1

LINEAR REGRESSION -- Y = A + B * X1 + C * X2 + D * X3

SUMMARY TABLE

PARAMETER	VALUE	STANDARD ERROR	T-RATIO	SIGNIF LEVEL	BETA COEFF
A (CONSTANT)	10.88889	6.09182	1.78746	0.08553	
B X1	11.59367	1.02263	11.33708	0.00000	0.97455
C X2	-0.76324	0.08622	-8.85198	0.00000	-0.80713
D X3	0.64312	0.07958	8.08104	0.00000	0.59656

CORRELATION MATRIX

VARIABLE	MEAN	STANDARD DEVIATION	Y	X1	X2	X3
Y	80.87119	39.70236	1.00000	0.64276	0.07657	0.53870
X1	6.32042	3.33733	0.64276	1.00000	0.63698	0.30566
X2	45.39383	41.98536	0.07657	0.63698	1.00000	0.44074
X3	48.75018	36.82854	0.53870	0.30566	0.44074	1.00000

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	0.88598	MEAN OF ABSOLUTE RELATIVE DEVIATIONS	0.10984
STANDARD ERROR OF ESTIMATE	14.15868	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.17508
SUM OF SQUARES OF RESIDUALS	5212.17710	SUM OF SQUARES TOTAL	45712.04526
F VALUE	67.34208	DURBIN-WATSON STATISTIC	2.26350
DEGREES OF FREEDOM FOR ERROR	26	DEGREES OF FREEDOM DUE TO REGRESSION	3
TOTAL DEGREES OF FREEDOM	29	NUMBER OF DATA POINTS	30

VARIANCE-COVARIANCE MATRIX

	A	B	C	D
A	0.37110D 02	-0.39228D 01	-0.17801D 00	-0.24846D 00
B	-0.39228D 01	0.10458D 01	-0.51817D-01	-0.29300D-02
C	-0.17801D 00	-0.51817D-01	0.74344D-02	-0.23002D-02
D	-0.24846D 00	-0.29300D-02	-0.23002D-02	0.63335D-02

Fig. 6--Program output for sample case

TEST RUN 1 -- LINEAR WITH THREE INDEPENDENT VARIABLES DATE: 76272 TIME: 1350 PAGE: 2

TABLE OF RESIDUALS

LABEL	OBSERVED Y	X1	X2	X3	COMPUTED Y	RESIDUAL Y	RELATIVE DEVIATION
1001	14.39608	5.12227	86.12357	24.16273	20.08084	-5.68476	-0.39488
1002	19.44080	0.23766	15.28765	30.98716	21.90432	-2.46352	-0.12672
1003	24.06789	1.52672	40.15672	63.17772	38.57032	-14.50243	-0.60256
1004	27.94618	2.01928	10.34567	4.27653	29.15382	-1.20764	-0.04321
1005	33.60992	3.52567	42.16543	27.17864	37.06082	-3.45090	-0.10267
1006	40.40568	3.82762	27.37677	3.28716	36.48384	3.92184	0.09706
1007	40.71304	1.21549	3.26751	31.26884	42.59641	-1.88337	-0.04626
1008	40.97917	12.11763	175.26876	52.17625	51.15903	-10.17986	-0.24842
1009	51.71629	3.22762	8.18761	16.27615	52.52708	-0.81079	-0.01568
1010	54.53670	8.92672	109.26547	44.27861	59.46227	-4.92557	-0.09032
1011	59.00000	1.00000	20.00000	101.00000	72.17232	-13.17232	-0.22326
1012	71.09687	4.01982	15.00000	39.22218	71.26912	-0.17225	-0.00242
1013	80.09382	5.03729	10.00000	20.00000	74.51941	5.57441	0.06960
1014	83.40204	8.02456	35.25411	15.25672	86.82731	-3.42527	-0.04107
1015	84.84002	6.53382	16.27865	1.27553	75.03557	9.80445	0.11556
1016	88.26627	5.62718	19.26713	41.26517	87.96131	0.30496	0.00346
1017	89.92674	6.21786	18.88888	29.17654	87.32377	2.60297	0.02895
1018	90.28862	7.51235	76.11111	88.99112	97.12469	-6.83607	-0.07571
1019	92.51133	6.22452	10.27625	17.24561	86.30152	6.20981	0.06712
1020	94.38172	4.92882	14.11167	54.28817	92.17486	2.20686	0.02338
1021	100.47238	10.53547	55.27618	12.16547	98.66816	1.80422	0.01796
1022	104.27262	9.42469	48.23418	35.28861	106.03578	-1.76316	-0.01691
1023	107.75168	10.00000	100.00000	100.00000	114.81265	-7.06097	-0.06553
1024	111.35247	9.72617	84.23456	93.25671	119.33417	-7.98170	-0.07168
1025	122.69347	8.92219	12.16524	8.01187	110.19726	12.49621	0.10185
1026	131.24108	11.73392	95.18293	100.00000	138.59177	-7.35069	-0.05601
1027	137.00000	10.62679	81.27543	107.26784	141.04492	-4.04492	-0.02952
1028	138.46147	7.42819	18.22184	97.22215	145.62623	-7.16476	-0.05175
1029	143.72068	6.41625	102.33728	114.23477	80.63456	63.08612	0.43895
1030	147.55076	7.92618	12.25418	90.26713	151.48166	-3.93090	-0.02664

MINIMUM RELATIVE DEVIATION = -0.60256, MEAN ABSOLUTE RELATIVE DEVIATION = 0.10984, MAXIMUM RELATIVE DEVIATION = 0.43895

Fig. 6 (cont.)

TEST RUN 2 -- POWER WITH THREE INDEPENDENT VARIABLES DATE: 76272 TIME: 1350 PAGE: 3

POWER REGRESSION -- Y = A * X1**B * X2**C * X3**D

SUMMARY TABLE

PARAMETER	VALUE	INITIAL GUESS	STANDARD ERROR	T-RATIO	SIGNIF LEVEL
A (CONSTANT)	15.11017	13.27392	0.99490	15.18761	0.00000
B X1	1.18696	1.20099	0.02070	57.35095	0.00000
C X2	0.27451	0.29947	0.01036	26.50234	0.00000
D X3	-0.55490	-0.56444	0.00926	-59.93611	0.00000

CORRELATION MATRIX

VARIABLE	MEAN	STANDARD DEVIATION	Y	X1	X2	X3
Y	773.54965	528.96465	1.00000	0.62711	0.51727	-0.10302
X1	65.46327	39.00866	0.62711	1.00000	0.35631	0.56950
X2	65.01891	40.25761	0.51727	0.35631	1.00000	0.29407
X3	65.94354	48.50178	-0.10302	0.56950	0.29407	1.00000

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	0.99649	MEAN OF ABSOLUTE RELATIVE DEVIATIONS	0.03995
STANDARD ERROR OF ESTIMATE	33.08462	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.04277
SUM OF SQUARES OF RESIDUALS	28459.39895	SUM OF SQUARES TOTAL	8114304.55697
F VALUE	2462.36137	DURBIN-WATSON STATISTIC	1.98651
DEGREES OF FREEDOM FOR ERROR	26	DEGREES OF FREEDOM DUE TO REGRESSION	3
TOTAL DEGREES OF FREEDOM	29	NUMBER OF DATA POINTS	30

VARIANCE-COVARIANCE MATRIX

	A	B	C	D
A	0.989830D 00	-0.154340D-01	-0.145210D-02	0.274680D-02
B	-0.154340D-01	0.428340D-03	-0.966740D-04	-0.125950D-03
C	-0.145210D-02	-0.966740D-04	0.107290D-03	0.153210D-04
D	0.274680D-02	-0.125950D-03	0.153210D-04	0.857130D-04

Fig. 6 (cont.)

TEST RUN 2 -- POWER WITH THREE INDEPENDENT VARIABLES DATE: 76272 TIME: 1350 PAGE: 4

TABLE OF RESIDUALS

LABEL	OBSERVED Y	X1	X2	X3	COMPUTED Y	RESIDUAL Y	RELATIVE DEVIATION
200101	60.25169	10.21822	65.27819	95.11118	59.95322	0.29847	0.00495
200202	115.19201	29.12678	8.16279	75.25619	133.76559	-18.57358	-0.16124
200303	143.04087	25.14557	34.25671	90.00000	150.82050	-7.77963	-0.05439
200404	247.96037	38.29918	27.18827	75.27164	257.55163	-9.59126	-0.03868
200505	342.15482	35.12311	19.23518	37.26153	312.20043	29.95439	0.08755
200606	351.20949	35.18762	10.26781	16.25673	417.28460	-66.07511	-0.18814
200707	370.88634	17.26155	76.25144	10.15782	403.32732	-32.44098	-0.08747
200808	425.47155	41.15237	118.26132	77.23518	413.96958	11.50197	0.02703
200909	460.16664	20.23457	14.16289	4.25617	497.17746	-37.01082	-0.08043
201010	461.97847	23.15678	104.28715	19.29175	436.37698	25.60149	0.05542
201111	530.19078	67.22218	48.18273	92.23116	524.94852	5.24226	0.00989
201212	531.15436	71.16253	100.18273	145.27168	533.66005	-2.50569	-0.00472
201313	535.55565	70.27168	108.26152	132.17817	565.95153	-30.39588	-0.05676
201414	585.28353	89.26615	5.27715	45.23519	594.75845	-9.47492	-0.01619
201515	661.12500	78.28719	66.26718	100.00000	656.40565	4.71935	0.00714
201616	755.79487	44.27651	85.28716	27.18279	736.83582	18.95905	0.02508
201717	779.62800	28.12816	9.18826	3.27715	754.58938	25.03862	0.03212
201818	800.00000	43.13425	59.17236	18.21926	806.74399	-6.74399	-0.00843
201919	814.87900	111.25411	23.18726	76.16233	868.48898	-53.60998	-0.06579
202020	880.20145	102.17628	112.27162	130.18719	898.95228	-18.75083	-0.02130
202121	972.57176	34.18273	26.18273	5.26174	974.85942	-2.28766	-0.00235
202222	980.76950	100.00000	100.00000	100.00000	982.68094	-1.91144	-0.00195
202323	1000.58891	121.27157	55.24351	110.23145	994.51041	6.07850	0.00607
202424	1125.67637	151.28761	93.27615	178.29977	1143.32694	-17.65057	-0.01568
202525	1138.31287	120.11234	77.19283	105.25172	1105.81779	32.49508	0.02855
202626	1158.45163	120.17865	90.24133	105.25671	1154.98485	3.46678	0.00299
202727	1345.54035	57.19287	102.00000	17.26153	1349.24647	-3.70612	-0.00275
202828	1472.22902	65.24138	110.27816	21.25517	1435.85787	36.37115	0.02470
202929	1510.21359	102.15672	54.28716	40.25617	1412.08655	98.12704	0.06498
203030	2650.01076	112.18882	147.23557	25.18892	2692.08979	-42.07903	-0.01588

MINIMUM RELATIVE DEVIATION = -0.18814, MEAN ABSOLUTE RELATIVE DEVIATION = 0.03995, MAXIMUM RELATIVE DEVIATION = 0.08755

Fig. 6 (cont.)

TEST RUN 3 -- LN-LINEAR WITH SEVEN INDEPENDENT VARIABLES (MAXIMUM)   DATE: 76272   TIME: 1350   PAGE: 5

LOG-LINEAR REGRESSION -- LN Y = LN A + B * LN X1 + C * LN X2 + D * LN X3 + E * LN X4 + F * LN X5 + G * LN X6 + H * LN X7

SUMMARY TABLE

NOTE -- STATISTICS ARE BASED ON LOGARITHMS

PARAMETER		VALUE	STANDARD ERROR	T-RATIO	SIGNIF LEVEL	BETA COEFF
A		1.00077				
LN A	(CONSTANT)	0.76846D-03	0.38661	0.00199	0.99849	
B	X1	1.30306	0.13254	9.83130	0.00019	1.23982
C	X2	-0.52434	0.21479	-2.44113	0.05857	-0.33397
D	X3	-0.80013D-01	0.06075	-1.31699	0.24497	-0.06820
E	X4	-0.15377D-01	0.06543	-0.23502	0.82351	-0.01492
F	X5	-0.24842D-01	0.04131	-0.60135	0.57383	-0.02944
G	X6	-0.16084	0.06860	-2.34442	0.06602	-0.11117
H	X7	0.22546D-01	0.06859	0.32872	0.75569	0.01927

CORRELATION MATRIX

VARIABLE	MEAN	STANDARD DEVIATION	LN Y	LN X1	LN X2	LN X3	LN X4	LN X5	LN X6	LN X7
LN Y	1.90624	0.89619	1.00000	0.98114	0.84033	-0.33833	-0.63312	-0.51181	-0.08066	-0.06402
LN X1	2.61087	0.85268	0.98114	1.00000	0.90594	-0.26396	-0.65811	-0.47919	-0.01115	0.03706
LN X2	1.47808	0.57082	0.84033	0.90594	1.00000	-0.17965	-0.67309	-0.37613	-0.11463	0.25882
LN X3	1.37565	0.76389	-0.33833	-0.26396	-0.17965	1.00000	0.39066	0.22639	-0.01514	0.41224
LN X4	-1.03020	0.86946	-0.63312	-0.65811	-0.67309	0.39066	1.00000	0.34593	-0.09241	-0.02588
LN X5	3.80359	1.06205	-0.51181	-0.47919	-0.37613	0.22639	0.34593	1.00000	-0.04777	0.07327
LN X6	3.84640	0.61945	-0.08066	-0.01115	-0.11463	-0.01514	-0.09241	-0.04777	1.00000	0.11618
LN X7	3.79923	0.76611	-0.06402	0.03706	0.25882	0.41224	-0.02588	0.07327	0.11618	1.00000

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	0.99111	MEAN OF ABSOLUTE RELATIVE DEVIATIONS (LN)	0.04016
STANDARD ERROR OF ESTIMATE	0.13089	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.06866
SUM OF SQUARES OF RESIDUALS (LN)	0.08566	SUM OF SQUARES TOTAL (LN)	9.63779
F VALUE	79.65198	DURBIN-WATSON STATISTIC	2.35611
DEGREES OF FREEDOM FOR ERROR	5	DEGREES OF FREEDOM DUE TO REGRESSION	7
TOTAL DEGREES OF FREEDOM	12	NUMBER OF DATA POINTS	13

VARIANCE-COVARIANCE MATRIX

	LN A	B	C	D	E	F	G	H
LN A	0.14947D 00	-0.51267D-02	-0.64764D-02	-0.33657D-02	0.19456D-02	-0.48143D-02	-0.14311D-01	-0.92414D-02
B	-0.51267D-02	0.17568D-01	-0.24984D-01	-0.35107D-03	-0.32823D-03	0.15473D-02	-0.29162D-02	0.43449D-02
C	-0.64764D-02	-0.24984D-01	0.46136D-01	0.66172D-03	0.47209D-02	-0.11085D-02	0.62413D-02	-0.84734D-02
D	-0.33657D-02	-0.35107D-03	0.66172D-03	0.36910D-02	-0.11477D-02	-0.18348D-03	0.20876D-03	-0.16650D-02
E	0.19456D-02	-0.32823D-03	0.47209D-02	-0.11477D-02	0.42806D-02	-0.14712D-03	0.10707D-02	-0.38496D-03
F	-0.48143D-02	0.15473D-02	-0.11085D-02	-0.18348D-03	-0.14712D-03	0.17065D-02	0.17286D-04	0.46095D-04
G	-0.14311D-01	-0.29162D-02	0.62413D-02	0.20876D-03	0.10707D-02	0.17286D-04	0.47065D-02	-0.15815D-02
H	-0.92414D-02	0.43449D-02	-0.84734D-02	-0.16650D-02	-0.38496D-03	0.46095D-04	-0.15815D-02	0.47042D-02

Fig. 6 (cont.)

TEST RUN 3 -- LN-LINEAR WITH SEVEN INDEPENDENT VARIABLES (MAXIMUM) DATE: 76272 TIME: 1350 PAGE: 6

TABLE OF RESIDUALS

LABEL	OBSERVED Y	X1	X2	X3	X4	X5	X6	X7	COMPUTED Y	RESIDUAL Y	RELATIVE DEVIATION
ALPHA	1.000	2.100	1.100	6.600	0.940	77.000	76.000	14.000	1.023	-0.023	-0.023
BETA	2.000	4.100	2.900	8.900	0.810	86.000	12.000	92.000	2.016	-0.016	-0.008
GAMMA	3.100	9.000	3.700	3.200	0.610	98.000	96.000	64.000	3.812	-0.712	-0.230
DELTA	4.000	8.000	3.500	3.700	0.630	45.000	66.000	75.000	3.614	0.386	0.096
EPSILON	7.000	13.500	4.100	3.900	0.220	71.000	64.000	91.000	6.644	0.356	0.051
ZETA	5.000	9.900	3.800	7.400	0.990	81.000	42.000	43.000	4.494	0.506	0.101
ETA	9.900	19.000	3.700	7.600	0.360	84.000	93.000	76.000	9.618	0.282	0.029
THETA	10.000	21.000	6.900	2.600	0.400	59.000	29.000	80.000	10.473	-0.473	-0.047
IOTA	11.000	24.000	8.500	1.800	0.120	65.000	69.000	33.000	9.970	1.030	0.094
KAPPA	12.000	25.000	7.300	2.800	0.050	41.000	34.000	37.000	12.662	-0.662	-0.055
LAMBDA	15.000	19.000	3.400	0.600	0.350	14.000	29.000	9.000	14.811	0.189	0.013
MU	17.000	36.500	8.400	5.600	0.180	2.000	73.000	83.000	17.345	-0.345	-0.020
OMEGA	20.000	40.000	7.600	8.700	0.440	48.000	25.000	21.000	20.875	-0.875	-0.044

MINIMUM RELATIVE DEVIATION = -0.22975, MEAN ABSOLUTE RELATIVE DEVIATION = 0.06234, MAXIMUM RELATIVE DEVIATION = 0.10130

Fig. 6 (cont.)

TEST RUN 4 -- QUADRATIC WITH ALL PLOTS, SAME Y AND X2 (NOW X1) DATA AS RUN 3 DATE: 76272 TIME: 1350 PAGE: 7

QUADRATIC REGRESSION -- Y = A + B * X1 + C * X1**2

SUMMARY TABLE

PARAMETER	VALUE	STANDARD ERROR	T-RATIO	SIGNIF LEVEL
A (CONSTANT)	-2.07212	2.94159	-0.70442	0.49725
B POPULATN	2.72044	0.27508	9.88949	0.00000
C POP**2	-0.82889D-01	2.02012	-0.04103	0.96808

VARIABLE	MEAN	STANDARD DEVIATION
Y TOTLCOST	9.00000	5.97648
X1 POPULATN	4.99231	2.40709

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	0.55808	MEAN OF ABSOLUTE RELATIVE DEVIATIONS	0.47349
STANDARD ERROR OF ESTIMATE	4.35217	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.48357
SUM OF SQUARES OF RESIDUALS	189.41377	SUM OF SQUARES TOTAL	428.62000
F VALUE	6.31438	DURBIN-WATSON STATISTIC	1.32871
X COORDINATE OF VERTEX	16.41017	Y COORDINATE OF VERTEX	20.24935
DEGREES OF FREEDOM FOR ERROR	10	DEGREES OF FREEDOM DUE TO REGRESSION	2
TOTAL DEGREES OF FREEDOM	12	NUMBER OF DATA POINTS	13

VARIANCE-COVARIANCE MATRIX

	A	B	C
A	0.86529D 01	-0.79634D 00	-0.38815D 00
B	-0.79634D 00	0.75671D-01	0.73161D 00
C	-0.38815D 00	0.73161D 00	0.40809D 01

Fig. 6 (cont.)

TEST RUN 4 -- QUADRATIC WITH ALL PLOTS, SAME Y AND X2 (NOW X1) DATA AS RUN 3 DATE: 76272 TIME: 1350 PAGE: 8

TABLE OF RESIDUALS

HEADING	OBSERVED TOTLCOST	POPULATN	COMPUTED TOTLCOST	RESIDUAL TOTLCOST	RELATIVE DEVIATION
A ALPHA	1.00000	1.10000	0.82008	0.17992	0.17992
B BETA	2.00000	2.90000	5.12007	-3.12007	-1.56004
C GAMMA	3.10000	3.70000	6.85878	-3.75878	-1.21251
D DELTA	4.00000	3.50000	6.43405	-2.43405	-0.60851
E EPSILON	7.00000	4.10000	7.68834	-0.68834	-0.09833
F ZETA	5.00000	3.80000	7.06865	-2.06865	-0.41373
G ETA	9.90000	3.70000	6.85878	3.04122	0.30719
H THETA	10.00000	6.90000	12.75260	-2.75260	-0.27526
I IOTA	11.00000	8.50000	15.06293	-4.06293	-0.36936
J KAPPA	12.00000	7.30000	13.36997	-1.36997	-0.11416
K LAMBDA	15.00000	3.40000	6.21920	8.78080	0.58539
L MU	17.00000	8.40000	14.93097	2.06903	0.12171
M OMEGA	20.00000	7.60000	13.81559	6.18441	0.30922

MINIMUM RELATIVE DEVIATION = -1.56004, MEAN ABSOLUTE RELATIVE DEVIATION = 0.47349, MAXIMUM RELATIVE DEVIATION = 0.58539

Fig. 6 (cont.)

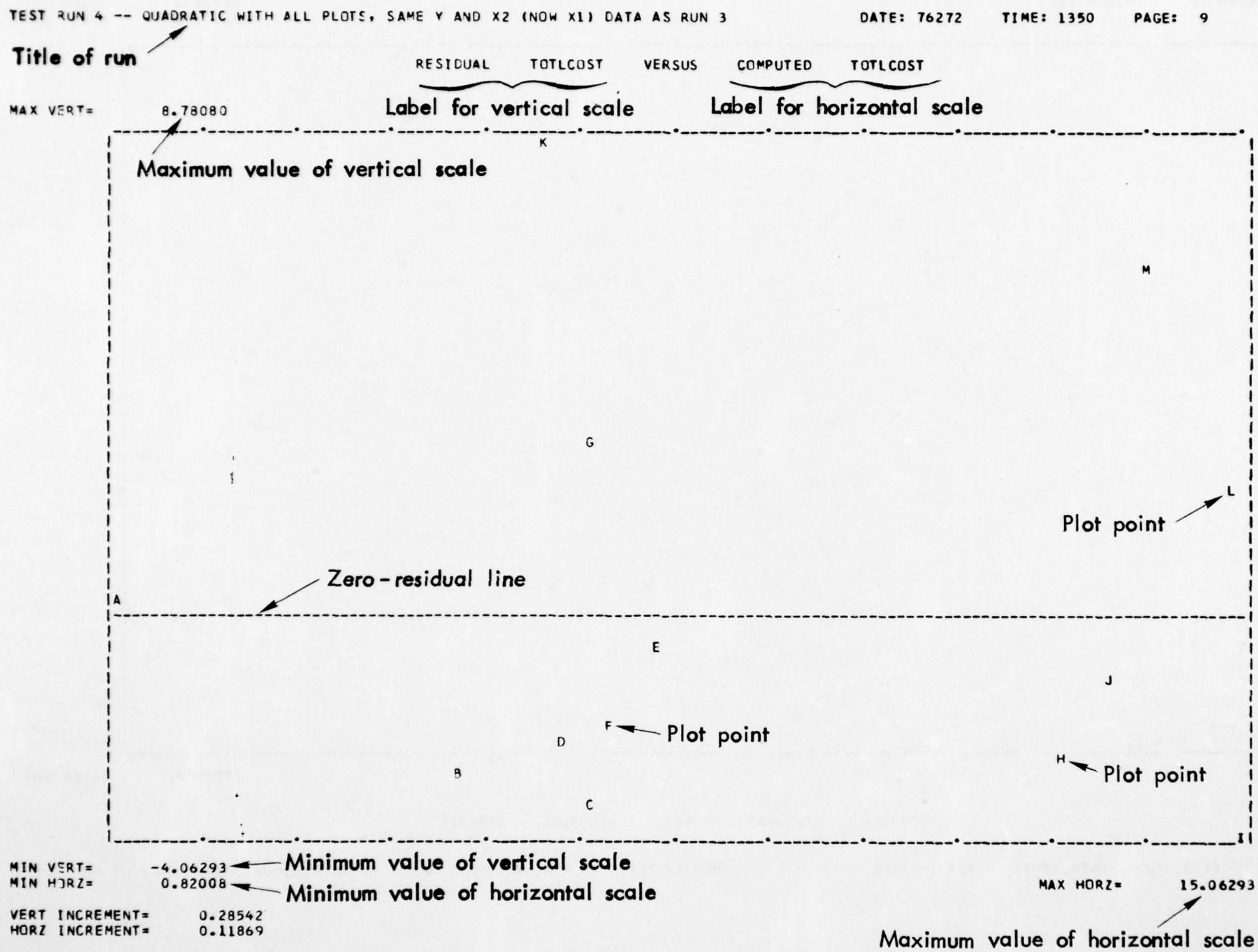
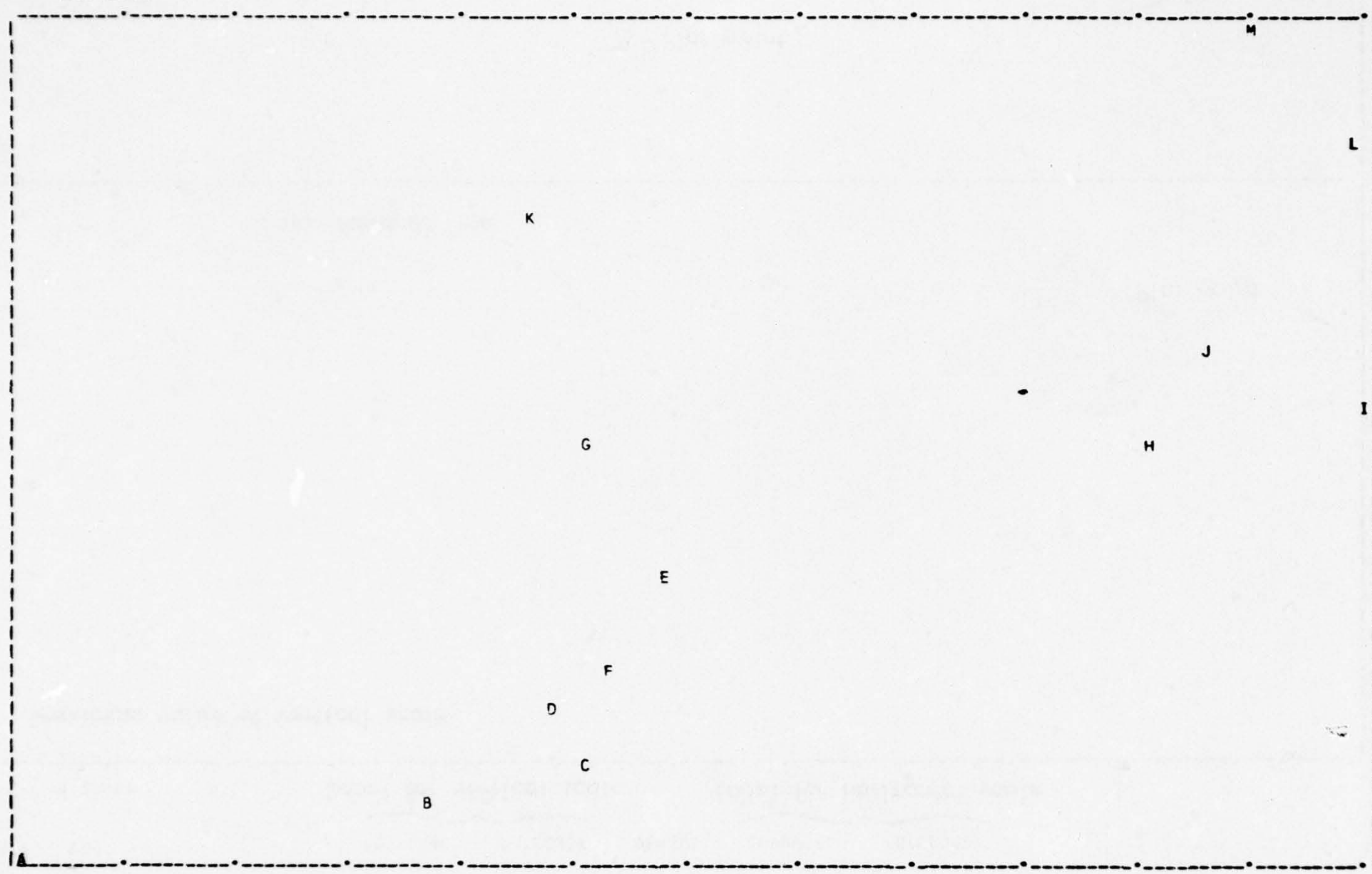

TEST RUN 4 -- QUADRATIC WITH ALL PLOTS, SAME Y AND X2 (NOW X1) DATA AS RUN 3
DATE: 76272
TIME: 1350
PAGE: 9
Title of run
RESIDUAL TOTLCOST
VERSUS
COMPUTED TOTLCOST
Label for vertical scale
Label for horizontal scale
MAX VERT= 8.78080
Maximum value of vertical scale
Zero-residual line
Plot point
Plot point
Plot point
MIN VERT= -4.06293
Minimum value of vertical scale
MIN HORZ= 0.82008
Minimum value of horizontal scale
MAX HORZ= 15.06293
Maximum value of horizontal scale
VERT INCREMENT= 0.28542
HORZ INCREMENT= 0.11869

Fig. 6 (cont.)

TEST RUN 4 -- QUADRATIC WITH ALL PLOTS, SAME Y AND X2 (NOW X1) DATA AS RUN 3 DATE: 76272 TIME: 1350 PAGE: 10

OBSERVED TOTLCOST VERSUS COMPUTED TOTLCOST

MAX VERT= 20.00000



MIN VERT= 1.00000
MIN HORZ= 0.82008 MAX HORZ= 15.06293

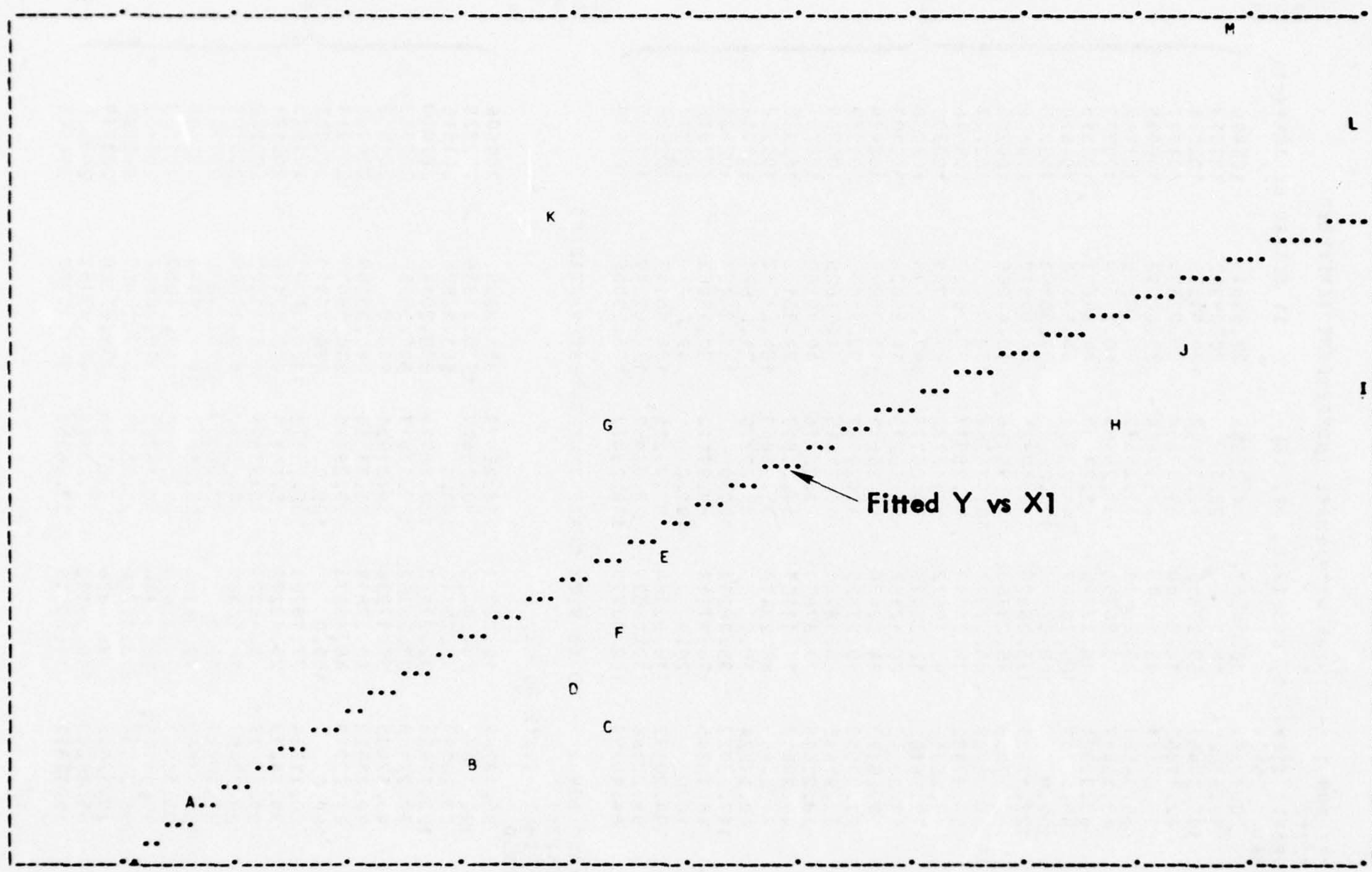
VERT INCREMENT= 0.42222
HORZ INCREMENT= 0.11869

Fig. 6 (cont.)

TEST RUN 4 -- QUADRATIC WITH ALL PLOTS, SAME Y AND X2 (NOW X1) DATA AS RUN 3 DATE: 76272 TIME: 1350 PAGE: 11

OBSERVED TOTLCOST VERSUS OBSERVED POPULATN

MAX VERT= 20.00000



MIN VERT= 0.0
MIN HORZ= 0.0 MAX HORZ= 8.50000

VERT INCREMENT= 0.44444
HORZ INCREMENT= 0.07083

Fig. 6 (cont.)

```
TEST RUN 1 -- LINEAR WITH THREE INDEPENDENT VARIABLES                      Title Card
1123YI         1                                                           Control Card
FORMAT  (1PF12.0, 0P3F12.0, 6X, A8)           X1 SCALED BY 10**(-1)        Format Card
READ    ONCE                                                               Read Card
    20.19283    10.34567     4.27653    27.94618      100404   \
    50.37289    10.0        20.0        80.09382      101313   |
   105.35467    55.27618    12.16547   100.47238      102121   |
    62.17862    18.88888    29.17654    89.92674      101717   |
    35.25671    42.16543    27.17864    33.60992      100505   |
    80.24561    35.25411    15.25672    83.40204      101414   |
    12.15487     3.26751    31.26884    40.71304      100707   |
    65.33821    16.27865     1.27553    84.84002      101515   |
    89.26718   109.26547    44.27861    54.53670      101010   |
   100.0       100.0       100.0       107.75168      102323   |
   121.17625   175.26876    52.17625    40.97917      100808   |
     2.37658    15.28765    30.98716    19.44080      100202   |
   106.26789    81.27543   107.26784   137.0          102727   |
    56.27182    19.26713    41.26517    88.26627      101616   |
    15.26718    40.15672    63.17772    24.06789      100303   |
    79.26182    12.25418    90.26713   147.55076      103030    > Data Cards
    51.22268    86.12357    24.16273    14.39608      100101   |
    97.26173    84.23456    93.25671   111.35247      102424   |
    62.24518    10.27625    17.24561    92.51133      101919   |
    32.27615     8.18761    16.27615    51.71629      100909   |
    38.27615    27.37677     3.28716    40.40568      100606   |
    49.28817    14.11167    54.28817    94.38172      102020   |
    94.24689    48.23418    35.28861   104.27262      102222   |
    40.19824    15.0        39.22218    71.09687      101212   |
   117.33922    95.18293   100.0       131.24108      102626   |
    75.12345    76.11111    88.99112    90.28862      101818   |
    10.0        20.0       101.0        59.0          101111   |
    74.28192    18.22184    97.22215   138.46147      102828   |
    89.22186    12.16524     8.01187   122.69347      102525   |
    64.16253   102.33728   114.23477   143.72068      102929   /
                                                                           Blank Card
TEST RUN 2 -- POWER WITH THREE INDEPENDENT VARIABLES                       Title Card
3123YI                                                                     Control Card
FORMAT  (4F12.0, 6X, A6)                                                   Format Card
READ                                                                       Read Card
    35.18762    10.26781    16.25673   351.20949      200606   \
   102.15672    54.28716    40.25617  1510.21359      202929   |
    78.28719    66.26718   100.0       661.12500      201515   |
   102.17628   112.27162   130.18719   880.20145      202020   |
    70.27168   108.26152   132.17817   535.55565      201313   |
    43.13425    59.17236    18.21926   800.0          201818   |
    10.21822    65.27819    95.11118    60.25169      200101   |
    67.22218    48.18273    92.23116   530.19078      201111   |
   100.0       100.0       100.0       980.76950      202222   |
   120.11234    77.19283   105.25172  1138.31287      202525    > Data Cards
    34.18273    26.18273     5.26174   972.57176      202121   |
    29.12678     8.16279    75.25619   115.19201      200202   |
    20.23457    14.16289     4.25617   460.16664      200909   |
    57.19287   102.0        17.26153  1345.54035      202727   |
   120.17865    90.24133   105.25671  1158.45163      202626   |
    25.14567    34.25671    90.0       143.04087      200303   |
    17.26155    76.25144    10.15782   370.88634      200707   |
    65.24138   110.27816    21.25517  1472.22902      202828   |
    28.12816     9.18826     3.27715   779.62800      201717   |
    41.15287   118.26132    77.23518   425.47155      200808   |
    35.12311    19.23518    37.26153   342.15482      200505   /
```

Fig. 7--Deck setup for sample runs shown in Fig. 6

```
 121.27157     55.24351    110.23145   1000.58891        202323  }
  23.15678    104.28715     19.29175    461.97847        201010  }
 111.25411     23.18726     76.16233    814.67900        201919  }
 151.28761     93.27615    178.29977   1125.67637        202424  }
  44.27651     85.28716     27.18279    755.79487        201616  }  Data Cards
  71.16253    100.18273    145.27168    531.15436        201212  }  (Continued)
 112.18882    147.23557     25.18892   2650.01076        203030  }
  38.29918     27.18827     75.27164    247.96037        200404  }
  89.26615      5.27715     45.23519    585.28353        201414  }
                                                                    Blank Card
TEST RUN 3 -- LN-LINEAR WITH SEVEN INDEPENDENT VARIABLES (MAXIMUM)  Title Card
6IY1234567     0                                                    Control Card
FORMAT  (A7, 3X, 8F5.0)                                             Format Card
READ    MEMORY      I.E., READ CARDS, SAVE IN MEMORY                Read Card
AIPHA        1.0  2.1  1.1  6.6  .94   77   76   14  }
BETA         2.0  4.1  2.9  8.9  .81   86   12   92  }
GAMMA        3.1  9.0  3.7  3.2  .61   98   96   64  }
DELTA        4.0  8.0  3.5  3.7  .63   45   66   75  }
EPSILON      7.0 13.5  4.1  3.9  .22   71   64   91  }
ZETA         5.0  9.9  3.8  7.4  .99   81   42   43  }
ETA          9.9 19.0  3.7  7.6  .36   84   93   76  }  Data Cards
THETA       10.0 21.0  6.9  2.6  .40   59   29   80  }
IOTA        11.0 24.0  8.5  1.8  .12   65   69   33  }
KAPPA       12.0 25.0  7.3  2.8  .05   41   34   37  }
LAMBDA      15.0 19.0  3.4  0.6  .35   14   29   09  }
MU          17.0 36.5  8.4  5.6  .18   02   73   83  }
OMEGA       20.0 40.0  7.6  8.7  .44   48   25   21  }
BLANK                                                               Blank Card
TEST RUN 4 -- QUADRATIC WITH ALL PLOTS, SAME Y AND X2 (NOW X1) DATA AS RUN 3
2IY1        1191                                                    Control Card
FORMAT  (A8, 2X, F5.0, 5X, F5.0)                                    Format Card
LABEL   HEADING TOTLCOSTPOPULATNPOP**2                              Label Card
READ                                                                Read Card
DONE                                                                Done Card
```

Fig. 7 (cont.)

Table 8

APPROXIMATE CPU TIMES (SECONDS) FOR VARIOUS CURVES RUNS ON IBM 370/158

Equation Index	Number of Variables	Number of Data Points 20	60	100
1,2,6,7,8[a]	1	.3	.6	1.0
	3	.4	.8	1.3
	7	.6	1.2	1.6
3[b]	1	.6	1.7	2.3
	3	1.6	5.5	7.0
	7	4.6	12.7	17.3
4[c]	1	1.9	4.8	7.7
5[b]	1	.5	1.0	1.3
	3	.7	1.4	2.2
	7	1.3	2.5	3.4

[a]Linear, quadratic, and logarithmic equations, each having an algebraic solution for parameter estimates.

[b]Power and exponential equations, each using modified Gauss-Newton iterative procedure to obtain parameter estimates.

[c]Asymptotic-power equations using iterative, incremental-stepping procedures to obtain parameter estimates.

Appendix A

NONLINEAR-LEAST-SQUARES CONSIDERATIONS

LOGARITHMIC AND NONLOGARITHMIC EQUATIONS

The usual procedure for deriving least-squares estimates of the parameters of the power or exponential equation is first to convert the equation into a logarithmic-linear (or semilogarithmic-linear) equation. One then has an equation for which least-squares estimates of the parameters can be obtained by simple algebraic means. However, note that these least-squares estimates are not the same as the least-squares estimates of the parameters that specify the original equations. This may be seen by considering, for example, the power equation and its logarithmic form.

Let

$$Y = A \cdot X1^{B} \cdot X2^{C} \cdot X3^{D} \cdot \ldots \cdot X7^{H}$$

and

$$\mathrm{Ln}\ Y = \mathrm{Ln}\ A + B \cdot \mathrm{Ln}\ X1 + C \cdot \mathrm{Ln}\ X2 + D \cdot \mathrm{Ln}\ X3 + \ldots + H \cdot \mathrm{Ln}\ X7 .$$

For a least-squares solution, one is interested in minimizing the sum of squares of the Y residuals (denoted by Q).[1] Therefore, for the power equation,

$$Q = \sum_{i=1}^{N} (Y_i - Y_{ci})^2 ,$$

and for the logarithmic equation

$$Q' = \sum_{i=1}^{N} (\mathrm{Ln}\ Y_i - \mathrm{Ln}\ Y_{ci})^2$$

[1] Throughout this discussion, Q is used to represent the sum of squares of the Y residuals.

or

$$Q' = \sum_{i=1}^{N} \text{Ln}\left(\frac{Y_i}{Y_{ci}}\right)^2 ,$$

where N = number of data points,

Y_i = observed value of dependent variable for ith data point,

Y_{ci} = fitted value of dependent variable for ith data point.

In the logarithmic case, the sum of squares of the actual *differences* (residuals) between the observed and fitted Y values is *not* being minimized, rather the sum of squares of the logarithms of the *ratios* of those values is being minimized. Depending on the observations, the two procedures may produce substantially different estimates of the parameters A, B, C, ..., H.

It may also be seen that any statistic based on the sum of squares of Y residuals, such as the coefficient of determination, may be misleading if used to compare the logarithmic form with its nonlogarithmic counterpart. For the logarithmic form, such statistics are based on logarithms and hence have different meanings.

Regression theory states that if the error term on the dependent variable Y is an additive, normally distributed random variable with a mean of zero, then a least-squares fit will lead to maximum likelihood estimates of the regression coefficients. Therefore, for the power form, the following is assumed:

$$Y = A \cdot X1^B \cdot X2^C \cdot X3^D \cdot \ldots \cdot X7^H + \varepsilon$$

where the errors are independent, normally distributed random variables with mean zero and a common variance. For the logarithmic form, one has

$$\text{Ln } Y = \text{Ln } A + B \text{ Ln } X1 + C \text{ Ln } X2 + D \text{ Ln } X3 + \ldots + H \text{ Ln } X7 + \text{Ln } \delta$$

or

$$Y = A \cdot X1^{B} \cdot X2^{C} \cdot X3^{D} \cdot \ldots, X7^{H} \cdot \delta ,$$

where the error terms Ln δ satisfy the conditions specified for the errors in the previous case. In this case, the error term is multiplicative.

The question as to whether the regressed power equation or its regressed logarithmic form is more appropriate for a set of data depends on many factors including the error term associated with the data and what criterion is used for a "good fit."[1] However, one of the best tests for comparison is to examine the plot of Y residuals versus fitted values (or residuals of Log Y versus fitted Log Y for logarithmic case). The "better" model (power versus log) will show a more random normal distribution of the Y residuals around the zero line. This plot is available in the program for such an examination.

NONLINEAR SOLUTIONS

It is a necessary condition that the first partial derivatives of Q with respect to the parameters must be zero in order that Q be minimized. This is not, unfortunately, a sufficient condition for a function that is not linear with respect to all of its parameters. The reason for this is that if Q could be graphed (in multi-dimensional space) for a nonlinear function, there might be other critical points--such as saddle points or relative maxima or minima points--where the first partial derivatives would also be zero. A test that checks for this possibility involves examining the matrix of second partial derivatives of Q, which is a generalization of the second-derivative test for a one-parameter case. If this matrix is positive-definite for all parameters in a region containing a solution, it can be shown that the solution represents an absolute minimum for Q in that region and is the only solution in that region.[2] However, if the matrix is not positive-definite

---

[1] For the interested reader, this question is treated in Graver and Boren, RM-4879-PR.

[2] H. O. Hartley, "The Modified Gauss-Newton Method for the Fitting of Non-Linear Regression Functions by Least-Squares," *Technometrics*, Vol. 3, No. 2, May 1961, pp. 273-274.

at all points in that region, then there may be other "solutions" for the same set of data.

For regressions of the power and exponential equation involving very large ($\geq 10^6$) or very small ($\leq 10^{-6}$) values of X or Y input data, the matrix of partial derivatives may not be inverted accurately enough to give reasonable corrections to the parameters. Consequently, in such cases there may be no convergence to a solution of the parameters. This problem can usually be remedied by rescaling the input data (using the P-format). If input data are rescaled, only the estimate of the parameter A is changed in the regression.

In summary, one should be aware that for a nonlinear equation as defined in this report, the "solution" obtained may not represent an absolute minimum for Q. The only sure way to know is to try all combinations of the parameters for each data sample to determine all "solutions" and to then determine which solution gives the lowest sum of squares of Y residuals. For practical reasons this is very difficult to do. However, one must remember that an attempt is being made to find a solution to a function that adequately represents the data. Whether or not there are solutions in other unknown regions may be rather unimportant if the solution that is found is satisfactory to the analyst--that is, if it satisfies the analyst's criterion for a good fit.[1]

---

[1]For further information on nonlinear least-squares solutions, see N. R. Draper and H. Smith, *Applied Regression Analysis*, John Wiley & Sons, Inc., New York, London, Sydney, Chap. 10, 1966, pp. 263-304.

Appendix B

LEAST-SQUARES ESTIMATION FOR ASYMPTOTIC-POWER EQUATION

$$Y = A + B \cdot X1^{C}$$

REGRESSION EQUATIONS

To obtain least-squares estimates of the parameters A, B, and C of the asymptotic-power equation, the following procedure is used. First, let the residual corresponding to Y_i be defined by

$$e_i = Y_i - Y_{ci} = Y_i - (A + B \cdot X1_i^C) \ , \tag{1}$$

where A, B, and C are least-squares estimates of the parameters.

The requirement for a least-squares fit for N data points is that the sum of squares of the Y residuals (denoted by Q) shall be a minimum; here,

$$Q = \sum_{i=1}^{N} (Y_i - A - B \cdot X1_i^C)^2 \ . \tag{2}$$

If Q is to be a minimum, the partial derivatives of Q with respect to the parameters A, B, and C must be zero:

$$\frac{\partial Q}{\partial A} = Q_A = -2 \cdot \sum (Y_i - A - B \cdot X1_i^C) = 0 \ ,$$

$$Q_B = -2 \cdot \sum (Y_i - A - B \cdot X1_i^C) \cdot X1_i^C = 0 \ ,$$

$$Q_C = -2 \cdot \sum (Y_i - A - B \cdot X1_i^C) \cdot B \cdot X1_i^C \cdot \text{Ln } X1_i = 0 \ .$$

Simplifying and rearranging terms gives:

$$\sum Y_i = A \cdot N + B \cdot \sum X1_i^C , \tag{3}$$

$$\sum Y_i \cdot X1_i^C = A \cdot \sum X1_i^C + B \cdot \sum X1_i^{2C} , \tag{4}$$

$$\sum Y_i \cdot X1_i^C \cdot \text{Ln } X1_i = A \cdot \sum X1_i^C \cdot \text{Ln } X1_i + B \cdot \sum X1_i^{2C} \cdot \text{Ln } X1_i . \tag{5}$$

The problem then becomes one of solving Eqs. (3), (4), and (5) for the parameters A, B, and C, given a set of independent observations of Y and X1. Except in very special cases, the equations cannot be solved by ordinary algebraic methods but must be solved by iterative techniques. First, A can be eliminated from Eqs. (3) and (4) by multiplying Eq. (3) by $\sum X1_i^C$ and Eq. (4) by N and then subtracting the two equations. Having done this, one can solve for B in terms of C. That is

$$B = \frac{\sum Y_i \cdot \sum X1_i^C - N \cdot \sum Y_i \cdot X1_i^C}{\left(\sum X1_i^C\right)^2 - N \cdot \sum X1_i^{2C}} . \tag{6}$$

Therefore, for a given set of observations of Y and X1, if C is known, B can be solved from Eq. (6), and A can then be solved from Eq. (3).

$$A = \frac{\sum Y_i - B \cdot \sum X1_i^C}{N} . \tag{7}$$

The solution of A, B, and C must also satisfy Eq. (5). Let G represent the difference of the members in Eq. (5) as follows:

$$G = \sum Y_i \cdot X1_i^C \cdot \text{Ln } X1_i - A \cdot \sum X1_i^C \cdot \text{Ln } X1_i - B \cdot \sum X1_i^{2C} \cdot \text{Ln } X1_i . \tag{8}$$

G will be zero only when A, B, and C are a solution.

PROGRAM SEQUENCE OF OPERATIONS

The sequence of operations in the computer program is as follows.[1] First, the various summations involved in Eqs. (6), (7), and (8) are obtained using C = -8.001 (initially). Then B and A are determined from Eqs. (6) and (7).[2] After these calculations are made, the value of G is obtained from Eq. (8), and its algebraic sign is noted. Unless A, B, and C are a solution, G will not be zero. The machine then steps the value of C by +0.1, repeats all of the summations and calculations, and checks the algebraic sign of G again. This procedure is continued until the algebraic sign of G is reversed, signifying that a solution lies somewhere between the previous value of C and the value of C at this cross-over point.

At this point, the program begins an iterative operation in which at each cross-over point the incremental step is halved and the direction of advance is reversed. This iterative procedure is done as many times as desired to give any degree of accuracy required for C. In the program, this procedure is repeated until the ratio of the value of each of the parameters A, B, and C from one iteration to the next differs from unity by an amount equal to, or less than, 10^{-7} (or as otherwise specified).

The search for roots continues to C = -0.001.[3] After this point is reached, the program begins another search starting at C = +0.001 and proceeding by increments of +0.1 out to +8.001. If no solution at all is found within these limits, a statement to this effect is printed, and the program continues on to the next run. Any time a solution is found for A, B, and C, the sum of squares of Y residuals (Q) is determined and compared with the corresponding value for the

---

[1] Acknowledgment is made to James Johnston (formerly at Rand) for his suggestions in the initial programming aspects of this problem.

[2] If A is specified, then that value is used instead of calculating A from Eq. (7). Also, the equations for B and G are changed. However, the procedure of solving for B and C is similar to the case in which A is not specified.

[3] Because a zero value for C results in a degenerate case, the search purposely avoids a region very close to zero for C.

previous solution (if there was one). The solution that gives the lowest sum of squares of Y residuals is stored temporarily for comparison with any future solution so obtained. In this way, when the search is completed and if there is a solution, that solution will generally represent the lowest sum of squares of Y residuals in the region searched.

Any "solution" found in the specified range for C represents a solution for which the partial derivatives of Q with respect to the parameters are zero. The Q value for that solution is also compared with the Q values for the end points of C to make sure that Q is not decreasing to some other minimum outside the range of C. As of now, we have not been able to determine any requirements for Q to have a unique minimum but have observed that for various sets of data, Q seems to have a unique minimum in the region searched. Even if it does not, the minimum of the relative minima will usually be found. As stated before, there is apparently no proof that other minima cannot exist outside the range searched, which cannot be determined by the above method, even when a solution has been found in the prescribed range. However, this may be unimportant if the "solution" found satisfies the analyst's criterion for a good fit.

The above limits on C and the increments of 0.1 were chosen on the basis of what is believed to be a reasonable search range for C, of economic computer operating time, and of the extent to which the search range should be covered in order to lessen the chances of missing a root. Although two roots could conceivably be missed in the increment of 0.1, indicating that the G function goes from, say, a positive to a negative to a positive value within an interval of C equal to 0.1, this seems rather unlikely. Such a function would have to behave very erratically, and test results indicate that this function does not behave in this manner.

Perhaps it should be noted that a degenerate, or trivial, case results if C = 0 or if all Y values are constant or if all X1 values are constant. Any of these conditions results in:

$$Y = \text{constant}.$$

Appendix C

MODIFICATIONS TO CURVES

INTRODUCTION

The purpose of this appendix is to report on several modifications that have been made to the CURVES Cost Analysis Curve-Fitting Program, reported in an earlier Rand Report R-1753-PR.* A new listing of the modified CURVES computer program is presented in Appendix D.

VARIABLE TRANSFORMATIONS

A major modification has been incorporated in CURVES to allow for variable transformations of the following kinds: (a) power, (b) logarithmic, and (c) binary. Each of these is discussed below.

Power

A power transformation may be made on any variable as follows:

$$V_p \rightarrow (V_p)^{E_p} ,$$

where V_p = variable p,

p = 0 through 7 (p = 0: Y variable, p = 1: X1 variable, p = 2: X2 variable, ..., p = 7: X7 variable),

E_p = real-number exponent for variable p, with range $-10.0 < E_p < 10.0$,

$\rightarrow$ = "transformed to."

To fit, for example,

$$1 / Y = A + B \cdot X1 ,$$

-1.0 is entered for E_0. To fit

---

*H. E. Boren, Jr., and Capt. G. W. Corwin, *CURVES: A Cost Analysis Curve-Fitting Program*, The Rand Corporation, R-1753-PR, December 1975.

$$Y = A + B \cdot X1 + C \cdot \sqrt{X2} ,$$

0.5 is entered for E_2. To fit

$$Y = A + B / \sqrt[3]{X1} ,$$

-.333 is entered for E_1. To fit an equation of the form

$$Y = A + B \cdot X1 + C / X2 + D / \sqrt{X3} ,$$

data for Y and the three X variables are entered as if a linear regression is to be run. Then a value of -1.0 is entered for E_2 and -0.5 for E_3. The values for each variable are then raised to the appropriate exponent and stored in the same cells as the original values, thereby replacing the original values. For CURVES, any time a variable is transformed, the original values of the variable are lost for that run. However, if the original values are initially stored in memory or on disk, they can be recovered for subsequent runs. Any variable so transformed is indicated as such in the Table of Residuals of the output by the word MODIFIED over the name of the variable. All transformation factors are also printed at the end of the Table of Residuals.

Values for E_p are entered on the Control card (see Table 4--p. 20) in 5-column fields beginning in Col. 41. The fields are listed in Table C.1 below.

Table C.1

TRANSFORMATION FIELDS ON CONTROL CARD

Columns	Transformation Factor	Variable to be Transformed
41-45	E_0	Y
46-50	E_1	X1
51-55	E_2	X2
56-60	E_3	X3
61-65	E_4	X4
66-70	E_5	X5
71-75	E_6	X6
76-80	E_7	X7

If a 0 (zero) or 1.0 is entered, or the field is left blank, no transformation takes place for the variable corresponding to that field. The implied decimal-point location is at the right end of each field; a punched decimal point overrides the implied location. Similar to most of the other information on the Control card, any transformation factor that has been entered is retained from run to run unless superseded by a new value. Thus, if the same transformations are to be made for a series of regressions, the transformation factors need only be entered for the first run.

Logarithmic

Any variable may now be transformed to its logarithm as follows:

$$V_p \rightarrow \text{Ln}\ (V_p) \ ,$$

where V, p = same as before,
Ln = natural logarithm.

Although the latest published version of CURVES treats logarithmic equations, the program cannot treat logarithms of individual variables. For example, it is now possible to fit an equation of the form

$$Y = A + B \cdot X1 + C \cdot \text{Ln}\ (X2) + D \cdot X3 \ ,$$

in which logarithms are taken only of the X2 variable. To designate a logarithmic transformation, a value of 88.0 is entered in the transformation field corresponding to the variable to be transformed by logarithms. Thus, for the above equation, an 88.0 would be entered for E_2 in Cols. 51-55 of the Control card (Table C.1). In that case, natural logarithms would be taken of the values entered for the X2 variable.

Binary

The following kinds of binary operations may be made in CURVES:

$$1.\quad V_p \rightarrow V_p \cdot V_q \ ,$$

2. $V_p \rightarrow V_p / V_q$,

3. $V_p \rightarrow V_p + V_q$,

4. $V_p \rightarrow V_p - V_q$,

where V, p = same as before,

q = subscript of second variable involved in operation (0 through 7).

To effect any of the above binary operations, a two-digit number ranging from 10.0 through 47.0 is entered in the E_p field corresponding to the variable to be transformed. The left digit of the number indicates which of the above binary operations is to take place. A 1 signifies multiplication, a 2 division, a 3 addition, and a 4 subtraction. The second digit is the q subscript representing the second variable involved in the operation. For example, suppose that the following equation is to be fitted:

$$(Y / X1) = A + B \cdot X1 .$$

A 21.0 would be entered for E_0 on the Control card in Cols. 41-45 corresponding to the Y field. The digit 2 in 21 signifies that divisions are to be made on the Y values, and the digit 1 in 21 signifies that the divisors are to be the X1 values. In this case, after the Y and X1 values are entered, each Y value is replaced by the division of it by the corresponding X1 value. The division thus becomes the new Y value.

Another example of a binary operation is

$$Y = A + B \cdot (X1 / X3) + C \cdot (X2 + X4) .$$

Here, Y and the "four" independent variables X1 through X4 must be entered through the usual input process for CURVES. Thus, on the Control card a linear regression with four independent variables would be indicated. However, after the operations are performed in accordance with the equation, only two independent variables are to be used in the regression, namely, (X1 / X3) and (X2 + X4). This is accomplished as follows:

In Cols. 46-50 of the Control card, corresponding to the X1 variable, the two-digit transformation factor 23 is entered for E_1. This signifies to the program that each value of the original X1 variable is to be replaced by the value of X1 / X3. In Cols. 51-55, corresponding to the X2 variable, the two-digit number 34 is entered for E_2. This signifies that each value of the original X2 variable is to be replaced by adding to that value the corresponding value of X4. To prevent X3 and X4 from being used as independent variables in the regression, a 99.0 is entered for E_3 in the next field on the Control card, corresponding to the X3 variable, namely, Cols. 56-60. Any time a 99.0 is used in a transformation field, the variables corresponding to that field and to any remaining transformation fields are not used in the regression. Therefore, if a 99.0 is used, it must be the last transformation factor entered on the Control card. Also, whenever a 99.0 is entered, the transformation factors for that field and for the remaining fields are set to zero. This prevents unwanted carryovers of transformation factors from previous runs. Obviously, a 99.0 can never be entered on the Control card in the fields corresponding to the Y or X1 variables because those two variables are always required for a regression. If a 99.0 is to be entered on the Control card in the transformation fields, it must be entered for an E_q after Col. 50 and only after a transformation factor E_p has been entered with a value in the range of $10.0 \leq E_p \leq 47.0$, where $p < q$.

In the above example in which Y is regressed against (X1 / X3) and (X2 + X4), the values of X3 and X4 are shown in the Table of Residuals of the output with the words NOT USED over the (X3) and (X4) headings, because they are not used as independent variables in the regression.

SAMPLE OUTPUTS

Examples of outputs resulting from regressions involving the transformations discussed here are given in Figs. C.1 through C.6. The first regression involves an equation of the form:

$$Y = A + B / X1 + C \cdot \text{Ln}(X2) + D \cdot \sqrt{X3} + E \cdot \sqrt[3]{X4} .$$

CURVES REGRESSION ANALYSIS COMPUTER PROGRAM
(JULY 1976)

TEST RUN 5 -- Y = A + B/X1 + C * LN (X2) + D * SQRT(X3) + E * CUBE ROOT(X4)   DATE: 76272   TIME: 1433   PAGE: 1

LINEAR REGRESSION -- Y = A + B * X1 + C * X2 + D * X3 + E * X4

SUMMARY TABLE

PARAMETER		VALUE	STANDARD ERROR	T-RATIO	SIGNIF LEVEL	BETA COEFF
A	(CONSTANT)	10.21604	0.00000	2157220.21845	0.00000	
B	X1	-5.32811	0.00000	-1257251.09863	0.00000	-0.52490
C	X2	0.85629	0.00000	1142738.96037	0.00000	0.37173
D	X3	1.76310	0.00000	3011641.65194	0.00000	0.94255
E	X4	-0.25177	0.00000	-114730.46432	0.00000	-0.05198

CORRELATION MATRIX

VARIABLE	MEAN	STANDARD DEVIATION	Y	X1	X2	X3	X4
Y	14.28625	3.23622	1.00000	-0.21759	0.36784	0.78926	-0.09869
X1	0.20821	0.31882	-0.21759	1.00000	0.15746	0.30081	0.66854
X2	1.93733	1.40490	0.36784	0.15746	1.00000	0.07055	-0.23591
X3	2.30465	1.73007	0.78926	0.30081	0.07055	1.00000	0.41580
X4	2.15548	0.66819	-0.09869	0.66854	-0.23591	0.41580	1.00000

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	1.00000	MEAN OF ABSOLUTE RELATIVE DEVIATIONS	0.00000
STANDARD ERROR OF ESTIMATE	0.00000	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.00000
SUM OF SQUARES OF RESIDUALS	0.00000	SUM OF SQUARES TOTAL	83.78478
F VALUE	> 10**8	DURBIN-WATSON STATISTIC	3.17034
DEGREES OF FREEDOM FOR ERROR	4	DEGREES OF FREEDOM DUE TO REGRESSION	4
TOTAL DEGREES OF FREEDOM	8	NUMBER OF DATA POINTS	9

VARIANCE-COVARIANCE MATRIX

	A	B	C	D	E
A	0.22427D-10	-0.16695D-11	-0.79998D-12	-0.62754D-12	-0.39347D-11
B	-0.16695D-11	0.17960D-10	-0.13908D-11	0.13892D-12	-0.65684D-11
C	-0.79998D-12	-0.13908D-11	0.56150D-12	-0.85894D-13	0.81465D-12
D	-0.62754D-12	0.13892D-12	-0.85894D-13	0.34273D-12	-0.45589D-12
E	-0.39347D-11	-0.65684D-11	0.81465D-12	-0.45589D-12	0.48155D-11

Fig. C.1--Regression of $Y = A + B / X1 + C \cdot \text{Ln}(X2) + D \cdot \sqrt{X3} + E \cdot \sqrt[3]{X4}$

TEST RUN 5 -- Y = A + B/X1 + C * LN (X2) + D * SQRT(X3) + E * CUBE ROOT(X4)   DATE: 76272   TIME: 1433   PAGE: 2

TABLE OF RESIDUALS

OBSERVED Y	MODIFIED X1	MODIFIED X2	MODIFIED X3	MODIFIED X4	COMPUTED Y	RESIDUAL Y	RELATIVE DEVIATION
11.13243	0.07851	0.25278	0.94478	2.17468	11.13242	0.00000	0.00000
11.26916	0.38956	2.16619	1.12960	2.85084	11.26916	-0.00000	-0.00000
12.37792	0.18755	3.07440	0.61025	2.17405	12.37792	0.00000	0.00000
13.02871	0.03882	2.30259	0.73348	0.97470	13.02871	-0.00000	-0.00000
13.23252	0.06667	2.30259	1.00000	1.44220	13.23251	0.00000	0.00000
13.96222	1.00000	2.30259	4.47214	3.10688	13.96222	0.00000	0.00000
14.21820	0.03319	-0.94881	3.20734	2.63487	14.21820	-0.00000	-0.00000
19.39509	0.03790	2.42046	4.42210	1.93931	19.39509	0.00000	0.00000
19.96005	0.04169	3.56320	4.22220	2.10181	19.96005	-0.00000	-0.00000

MINIMUM RELATIVE DEVIATION = -0.00000,   MEAN ABSOLUTE RELATIVE DEVIATION = 0.00000,   MAXIMUM RELATIVE DEVIATION = 0.00000

TRANSFORMATION FACTORS -- Y: 0.0   X1: -1.00000   X2: 88.00000   X3: 0.50000   X4: 0.33330

Fig. C.2--Test run 5 (cont.)

TEST RUN 6 -- Y = A + B * (X1/X3) + C * (X2*X4)   DATE: 76272   TIME: 1433   PAGE: 3

LINEAR REGRESSION -- Y = A + B * X1 + C * X2

SUMMARY TABLE

PARAMETER	VALUE	STANDARD ERROR	T-RATIO	SIGNIF LEVEL	BETA COEFF
A (CONSTANT)	7.28610	0.00000	40453707.78680	0.00000	
B X1	-3.45210	0.00000	***************	0.00000	-0.22833
C X2	0.77721	0.00000	582785229.83033	0.00000	0.96422

CORRELATION MATRIX

VARIABLE	MEAN	STANDARD DEVIATION	Y	X1	X2
Y	69.85716	75.48862	1.00000	-0.26808	0.97363
X1	2.49903	4.99299	-0.26808	1.00000	-0.04123
X2	91.60709	93.65211	0.97363	-0.04123	1.00000

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	1.00000	MEAN OF ABSOLUTE RELATIVE DEVIATIONS	0.00000
STANDARD ERROR OF ESTIMATE	0.00000	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.00000
SUM OF SQUARES OF RESIDUALS	0.00000	SUM OF SQUARES TOTAL	28492.65863
F VALUE	> 10**8	DURBIN-WATSON STATISTIC	2.22489
DEGREES OF FREEDOM FOR ERROR	3	DEGREES OF FREEDOM DUE TO REGRESSION	2
TOTAL DEGREES OF FREEDOM	5	NUMBER OF DATA POINTS	6

VARIANCE-COVARIANCE MATRIX

	A	B	C
A	0.32439D-13	-0.15610D-14	-0.16265D-15
B	-0.15610D-14	0.62571D-15	0.13753D-17
C	-0.16265D-15	0.13753D-17	0.17785D-17

TABLE OF RESIDUALS

OBSERVED Y	MODIFIED X1	MODIFIED X2	NOT USED (X3)	NOT USED (X4)	COMPUTED Y	RESIDUAL Y	RELATIVE DEVIATION
7.21035	0.28618	1.17365	8.27100	1.34500	7.21035	0.00000	0.00000
7.79736	0.09870	1.09620	10.00000	3.00000	7.79736	0.00000	0.00000
14.64670	12.60942	65.47725	0.87165	12.38200	14.64670	-0.00000	-0.00000
73.01063	0.15653	85.25997	5.28700	8.29700	73.01063	-0.00000	-0.00000
126.33447	0.11299	153.67588	8.35100	14.54300	126.33447	-0.00000	-0.00000
190.14345	1.73033	242.95956	9.11100	11.70100	190.14344	0.00000	0.00000

MINIMUM RELATIVE DEVIATION = -0.00000,   MEAN ABSOLUTE RELATIVE DEVIATION = 0.00000,   MAXIMUM RELATIVE DEVIATION = 0.00000

TRANSFORMATION FACTORS -- Y: 0.0   X1: 23.00000   X2: 14.00000

Fig. C.3--Regression of $Y = A + B \cdot (X1 / X3) + C \cdot (X2 \cdot X4)$

TEST RUN 7 -- Y = A + B * (X1+X3) + C * (X2-X4) DATE: 76272 TIME: 1433 PAGE: 4

LINEAR REGRESSION -- Y = A + B * X1 + C * X2

SUMMARY TABLE

PARAMETER		VALUE	STANDARD ERROR	T-RATIO	SIGNIF LEVEL	BETA COEFF
A	(CONSTANT)	7.28599	0.00023	31163.11603	0.00000	
B	X1	-3.45210	0.00004	-94031.62174	0.00000	-1.04835
C	X2	0.77721	0.00002	49724.32826	0.00000	0.55437

CORRELATION MATRIX

VARIABLE	MEAN	STANDARD DEVIATION	Y	X1	X2
Y	7.61738	6.10004	1.00000	-0.85454	0.18787
X1	3.32990	1.85249	-0.85454	1.00000	0.34960
X2	15.21657	4.35104	0.18787	0.34960	1.00000

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	1.00000	MEAN OF ABSOLUTE RELATIVE DEVIATIONS	0.00001
STANDARD ERROR OF ESTIMATE	0.00016	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.00002
SUM OF SQUARES OF RESIDUALS	0.00000	SUM OF SQUARES TOTAL	223.26310
F VALUE	> 10**8	DURBIN-WATSON STATISTIC	2.46134
DEGREES OF FREEDOM FOR ERROR	4	DEGREES OF FREEDOM DUE TO REGRESSION	2
TOTAL DEGREES OF FREEDOM	6	NUMBER OF DATA POINTS	7

VARIANCE-COVARIANCE MATRIX

	A	B	C
A	0.54663D-07	-0.39394D-08	-0.32632D-08
B	-0.39394D-08	0.13478D-08	-0.20061D-09
C	-0.32632D-08	-0.20061D-09	0.24431D-09

TABLE OF RESIDUALS

OBSERVED Y	MODIFIED X1	MODIFIED X2	NOT USED (X3)	NOT USED (X4)	COMPUTED Y	RESIDUAL Y	RELATIVE DEVIATION
1.83025	4.94000	14.92200	1.28700	1.00000	1.83021	0.00004	0.00002
2.50904	6.46300	22.56000	4.67800	3.22400	2.50902	0.00002	0.00001
4.21554	3.24130	10.44600	2.56700	9.55400	4.21548	0.00006	0.00002
4.76104	2.99300	10.04500	0.88200	4.00000	4.76097	0.00007	0.00001
7.81040	3.00000	14.00000	2.00000	3.00000	7.81069	-0.00029	-0.00004
15.90047	1.33200	17.00000	1.00000	3.00000	15.90042	0.00005	0.00000
16.29488	1.34000	17.54300	0.55500	0.88800	16.29483	0.00005	0.00000

MINIMUM RELATIVE DEVIATION = -0.00004, MEAN ABSOLUTE RELATIVE DEVIATION = 0.00001, MAXIMUM RELATIVE DEVIATION = 0.00002

TRANSFORMATION FACTORS -- Y: 0.0 X1: 33.00000 X2: 44.00000

Fig. C.4--Regression of $Y = A + B \cdot (X1 + X3) + C \cdot (X2 - X4)$

TEST RUN 8 -- Y = A + B * (X1+X3) + C * (X2-X4) + D * X3 + E * X4   DATE: 76272   TIME: 1433   PAGE: 5

LINEAR REGRESSION -- Y = A + B * X1 + C * X2 + D * X3 + E * X4

SUMMARY TABLE

PARAMETER		VALUE	STANDARD ERROR	T-RATIO	SIGNIF LEVEL	BETA COEFF
A	(CONSTANT)	7.28474	0.00047	15441.64389	0.00000	
B	X1	-3.45196	0.00005	-63178.89254	0.00000	-1.04831
C	X2	0.77728	0.00003	30083.18196	0.00000	0.55442
D	X3	-0.31899D-03	0.00011	-2.78381	0.10845	-0.00007
E	X4	0.10788D-03	0.00004	2.74079	0.11133	0.00005

CORRELATION MATRIX

VARIABLE	MEAN	STANDARD DEVIATION	Y	X1	X2	X3	X4
Y	7.61738	6.10004	1.00000	-0.85454	0.18787	-0.55808	-0.31110
X1	3.32990	1.85249	-0.85454	1.00000	0.34960	0.78543	0.02377
X2	15.21657	4.35104	0.18787	0.34960	1.00000	0.47860	-0.51625
X3	1.85271	1.42523	-0.55808	0.78543	0.47860	1.00000	0.33101
X4	3.52371	2.90246	-0.31110	0.02377	-0.51625	0.33101	1.00000

COEFFICIENT OF DETERMINATION (UNADJ), R SQ	1.00000	MEAN OF ABSOLUTE RELATIVE DEVIATIONS	0.00001
STANDARD ERROR OF ESTIMATE	0.00010	COEFF VARIATION (STD ERR EST / MEAN Y OBS)	0.00001
SUM OF SQUARES OF RESIDUALS	0.00000	SUM OF SQUARES TOTAL	223.26310
F VALUE	> 10**8	DURBIN-WATSON STATISTIC	2.88307
DEGREES OF FREEDOM FOR ERROR	2	DEGREES OF FREEDOM DUE TO REGRESSION	4
TOTAL DEGREES OF FREEDOM	6	NUMBER OF DATA POINTS	7

VARIANCE-COVARIANCE MATRIX

	A	B	C	D	E
A	0.22256D-06	-0.15725D-08	-0.13026D-08	-0.14781D-08	-0.67787D-09
B	-0.15725D-08	0.29853D-08	0.10045D-08	-0.56253D-08	0.16465D-08
C	-0.13026D-08	0.10045D-08	0.66759D-09	-0.26268D-08	0.92836D-09
D	-0.14781D-08	-0.56253D-08	-0.26268D-08	0.13130D-07	-0.40817D-08
E	-0.67787D-09	0.16465D-08	0.92836D-09	-0.40817D-08	0.15493D-08

Fig. C.5--Regression of $Y = A + B \cdot (X1 + X3) + C \cdot (X2 - X4) + D \cdot X3 + E \cdot X4$

TEST RUN 8 -- Y = A + B * (X1+X3) + C * (X2-X4) + D * X3 + E * X4 DATE: 76272 TIME: 1433 PAGE: 6

TABLE OF RESIDUALS

OBSERVED Y	MODIFIED X1	MODIFIED X2	X3	X4	COMPUTED Y	RESIDUAL Y	RELATIVE DEVIATION
1.83025	4.94000	14.92200	1.28700	1.00000	1.83031	-0.00006	-0.00003
2.50904	6.46300	22.56000	4.67800	3.22400	2.50900	0.00004	0.00001
4.21554	3.24130	10.44600	2.56700	9.55400	4.21557	-0.00003	-0.00001
4.76104	2.99300	10.04500	0.88200	4.00000	4.76094	0.00010	0.00002
7.81040	3.00000	14.00000	2.00000	3.00000	7.81046	-0.00006	-0.00001
15.90047	1.33200	17.00000	1.00000	3.00000	15.90049	-0.00002	-0.00000
16.29488	1.34000	17.54300	0.55500	0.88800	16.29485	0.00003	0.00000

MINIMUM RELATIVE DEVIATION = -0.00003, MEAN ABSOLUTE RELATIVE DEVIATION = 0.00001, MAXIMUM RELATIVE DEVIATION = 0.00002

TRANSFORMATION FACTORS -- Y: 0.0 X1: 33.00000 X2: 44.00000 X3: 0.0 X4: 0.0

Fig. C.6--Test run 8 (cont.)

Data for the examples were taken from precalculated values and therefore represent near-perfect fits. This was done in order to check the results. Because X1, X2, X3, and X4 were all transformed, the word MODIFIED appears over each of their headings in the Table of Residuals shown in Fig. C.2. The transformation factors are shown at the bottom of the figure. To obtain the original X values, one simply reverses the transformation process. That is, to obtain the original X values for the first run, the X1 values are raised to the -1 power, the X2 values are exponentiated (using base e), the X3 values are squared, and the X4 values are cubed. However, as was mentioned before, if the original values are stored in memory or on disk, they can be reread (recovered) for another run.

The third page of output (Fig. C.3) shows a regression involving binary operations of the form:

$$Y = A + B \cdot (X1 / X3) + C \cdot (X2 \cdot X4) .$$

In the Table of Residuals, the word MODIFIED appears over the X1 and X2 headings, because those values were transformed. The values of X3 and X4 are listed but with the heading NOT USED over (X3) and (X4) to indicate that they were not used as independent variables in the regression. Because each original value of X1 was divided by each value of X3, the original values of X1 can be calculated as follows:

$$X1\ (\text{new}) = X1\ (\text{old}) / X3 ,$$
$$\text{or } X1\ (\text{old}) = X1\ (\text{new}) \cdot X3 .$$

Therefore, if each X1 value shown in the Table of Residuals is multiplied by the corresponding value of the X3 variable, the original value of X1 is obtained for each data point. A similar process can be used to obtain the original values of X2. Figure C.4 shows another output involving a regression of an equation of the form:

$$Y = A + B \cdot (X1 + X3) + C \cdot (X2 - X4) .$$

Lastly, Figs. C.5 and C.6 show an extension of the results of Fig. C.4 in which X3 and X4 are allowed to remain in the regression. The equation is then:

$$Y = A + B \cdot (X1 + X3) + C \cdot (X2 - X4) + D \cdot X3 + E \cdot X4 .$$

Because X3 and X4 are allowed to remain in the regression, the heading NOT USED does not appear in the X3 and X4 headings in the Table of Residuals. A listing of the input data for the four runs shown in Figs. C.1 through C.6 is shown in Fig. C.7. For convenience, card column numbers are shown at the top of the figure.

A summary of the transformation factors is given in Table C.2. An updated listing of the CURVES program including a new subroutine TRANS is included in Appendix D.

VARIANCE-COVARIANCE MATRIX

Except for a one-parameter case, the CURVES program now prints (in scientific notation) the variance-covariance matrix of the estimated coefficients on the first page of the output. For a one-parameter case, e.g.,

$$Y = B \cdot X1 ,$$

the variance of B is simply the square of the standard error of B, already printed at the top of the page to the right of the value of B.

MISCELLANEOUS

For the power and exponential cases, the default value of the iteration limit has been changed from 20 to 100. It was found that a limit of 20 is not always sufficient for such regressions.

```
         10        20        30        40        50        60
123456789012345678901234567890123456789012345678901234567890123456789012345 <----COLUMNS

TEST RUN 5 -- Y = A + B/X1 + C * LN (X2) + D * SQRT(X3) + E * CUBE ROOT(X4)
1Y1234         1                                -1   88   .5.3333
FORMAT   (5F10.0)
READ    ONCE
11.2691599 2.567     8.725     1.276     23.1768
11.1324268 12.738    1.2876    .8926     10.287
19.9600498 23.987    35.276    17.827    9.287
13.2325164 15.       10.       1.        3.
13.0287067 25.762    10.       .538      .926
14.2181985 30.127    .3872     10.287    18.298
13.9622199 1.        10.       20.       30.
19.3950887 26.387    11.251    19.555    7.295
12.3779224 5.332     21.637    .3724     10.278

TEST RUN 6 -- Y = A + B * (X1/X3) + C * (X2*X4)
                                                23   14   99
READ
7.21034603 2.367     .8726     8.271     1.345
14.6467017 10.991    5.2881    .87165    12.382
73.0106287 .8276     10.276    5.287     8.297
190.143445 15.765    20.764    9.111     11.701
7.79735533 .987      .3654     10.       3.
126.33447 .9436      10.567    8.351     14.543

TEST RUN 7 -- Y = A + B * (X1+X3) + C * (X2-X4)
                                                33   44   99
READ     MEMORY
4.21554396 .6743     20.       2.567     9.554
2.5090353 1.785      25.784    4.678     3.224
1.8302536 3.653      15.922    1.287     1.
15.9004728 .332      20.       1.        3.
16.294881 .785       18.431    .555      .888
7.8104    1.         17.       2.        3.
4.76103915 2.111     14.045    .882      4.

TEST RUN 8 -- Y = A + B * (X1+X3) + C * (X2-X4) + D * X3 + E * X4
                                                33   44
READ
DONE
```

Fig. C.7--Input card arrangement to generate outputs shown in Figs. C.1 through C.6

Table C.2

SUMMARY OF TRANSFORMATION FACTORS

Factor (E_p)	Type of Transformation
$-10.0 < E_p < 10.0$	$(V_p)^{E_p}$
$E_p = 88.0$	$Ln (V_p)$
$10.0 \le E_p \le 17.0$	$V_p \cdot V_q$
$20.0 \le E_p \le 27.0$	V_p / V_q
$30.0 \le E_p \le 37.0$	$V_p + V_q$
$40.0 \le E_p \le 47.0$	$V_p - V_q$
$E_p = 99.0$	Do not use V_p or any remaining variables in regression. E_p and remaining transformation factors (E_{p+1}, ... E_7) set to zero.
E_p = other values	Error

NOTES: V_p = variable p.

V_q = variable q; q is designated by second digit of E_p; first digit of E_p (1, 2, 3, or 4) designates type of combination--multiplication, division, addition, or subtraction, respectively, when $10.0 \le E_p \le 47.0$.

E_p = transformation factor for variable V_p.

p, q = 0 through 7 (0 = Y variable, 1 = X1 variable, 2 = X2 variable, ..., 7 = X7 variable).

Ln = natural logarithm.

Appendix D

UPDATED LISTING OF CURVES PROGRAM

```
C     CURVES: A COST ANALYSIS CURVE-FITTING COMPUTER PROGRAM, RAND
C     REPORT R-1753-1-PR, BY H.E. BOREN, JR. AND G.W. CORWIN,
C     SEPTEMBER 1976
C
      IMPLICIT REAL*8(A-H,O-Z)
C     THE FOLLOWING IS THE COMPLETE COMMON
      COMMON /C1/ N, NMAX, KOLUMN, NIV, NIVP1, NVAR, NP, LABEL, ID
      COMMON /C2/ IVAL(10), IND, IFV, IN, I8, JX(9), NPAGE
      COMMON /C3/ ABORT, LINEAR, IA, IGUESS, NXSET, NYCOL, NYC, NYDEV
      COMMON /C4/ S(8), SYX(8,8), COV(8,8), FMT(30), HEAD(9), LNH(8)
      COMMON /C5/ AN, P(8), SE(8), TR(8), SIGLEV(8), BETA(8), VMSQ(8)
      COMMON /C6/ VMEAN(8), T(8), H(8,8), P1(8), SDEV(8), RM(8,8)
      COMMON /C7/ YCEPT, DELTA, RDEVM, EA, DW, XV, YV, SEYSQ
      COMMON /C8/ DFT, DF1, DF2, CD, SEY, CV, YDEVSQ, FVALUE, SST
      COMMON /C9/ V(2211)
      COMMON /C0/ DATA(10,101)
      COMMON /CX/ EXV(8), NEXV, NVT
      LOGICAL*4   ABORT, LINEAR, IA, IGUESS
      EQUIVALENCE (IEQ, IVAL(1)), (NCARDS, IVAL(9)), (LIM, IVAL(10))
      [redacted]
      DO 10 I = 1, 8
      EXV(I) = 0.D0
      RM(I,I) = 1.D0
   10 IVAL(I) = 0
      NCARDS = 1
      LIM = 100
      NPAGE = 1
      DELTA = 1.D-7
      WRITE (6, 20)
   20 FORMAT (1H1, 42X, 'CURVES REGRESSION ANALYSIS COMPUTER PROGRAM'/
     1 59X, '(JULY 1976)' )
C     SET ABORT-FOR-ERROR INDICATOR TO FALSE
   30 ABORT = .FALSE.
      CALL READ
      CALL INPUT(V)
C     CHECK ERROR INDICATOR
      IF (ABORT) GO TO 30
      IF (IEQ .EQ. 0) GO TO 80
      CALL PRINT
      CALL SUMS(V)
      GO TO (40, 40, 50, 60, 50, 40, 40, 40), IEQ
   40 CALL LINE(V)
      GO TO 70
   50 CALL EXPO(V)
      GO TO 70
   60 CALL ASYM(V)
   70 IF (ABORT) GO TO 30
      CALL STAT(V)
      CALL TVAL(V)
      CALL OUT1
   80 CALL OUT2(V)
      GO TO 30
      END
```

```
      SUBROUTINE READ
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /C1/ M1, NMAX, KOLUMN, NIV, NIVP1, NVAR, NP, LABEL, ID
      COMMON /C2/ IVAL(10), M2(2), IN, I8, JX(9), NPAGE
      COMMON /C3/ ABORT, LINEAR, IA, IGUESS, NXSET, NYCOL, NYC, NYDEV
      COMMON /C4/ Z1(136), FMT(30), HEAD(9)
      COMMON /C6/ Z2(80), P1(8)
      COMMON /C7/ YCEPT, DELTA
      COMMON /CX/ EXV(8)
      LOGICAL*4   ABORT, LINEAR, IA, IGUESS
      DIMENSION   EXV1(8), EXV2(8), SVHEAD(9), KX(9), KV(10), JXSAVE(9)
      DIMENSION   ALPHA(4), IVAL1(10), LPHA(10), ANAMES(6), ALIST(10)
      EQUIVALENCE (LIST1,ALIST(2)), (IEQ, IVAL(1))
      DATA        KV/'Y','1','2','3','4','5','6','7','I',' '/
      DATA        LONCE/'ONCE'/, MEMO/'MEMO'/, LDISK/'DISK'/
      DATA        PARLB/' X1**2'/, EXPON/'EXPONENT'/
      DATA        IBLANK /' '/, BLANK/' '/
      DATA        SVHEAD/'LABEL',' Y',' X1',' X2',' X3',
     1            '   X4','   X5','   X6','   X7'/
      DATA        ANAMES/'READ','FORMAT','LABEL','LABELS','GUESS',
     1            'READ8'/, JXSAVE/' ','?','O','R','D','E','R','?',' '/
C
C     SUBROUTINE FOR READING TITLE, CONTROL, AND SPECIFICATION CARDS
C
C     IVAL: (1)  (2)  (3)  (4)   (5)     (6)    (7)   (8)    (9)   (10)
C           IEQ NP1 NP2 NP3 LZZYES KEEPZ LNOUT IORD NCARDS LIM
C
      CALL TITLER
      READ (5, 10) KX, LPHA, ALPHA, IVAL1, YCEPT, DELTA1, EXV1, EXV2
   10 FORMAT (1X,9A1,T1,A1,9X,8A1,A2,4A5,T1,I1,9X,8I1,I2,F10.0,F10.10,
     1 8F5.0, T41, 8A5)
C     CHECK EACH FIELD OF CONTROL CARD FROM COLUMN 1 THROUGH COLUMN 40.
C     IF NOT BLANK, SET APPROPRIATE DESIGNATOR TO VALUE SO ENTERED.
C     (VARIABLE TRANSFORMATION FACTORS ARE ENTERED IN COLUMNS 41-80.)
      DO 20 J = 1, 10
      IF (LPHA(J) .NE. IBLANK) IVAL(J) = IVAL1(J)
   20 CONTINUE
      IA = .FALSE.
      IF (ALPHA(2) .NE. BLANK .OR. ALPHA(1) .NE. BLANK) IA = .TRUE.
      IF (ALPHA(4) .NE. BLANK .OR. ALPHA(3) .NE. BLANK) DELTA = DELTA1
      IF (DELTA .GT. 0.1 .OR. DELTA .LT. 1.D-12) DELTA = 1.D-7
      IFMT = -10
      LABEL = 0
      IN = 0
      I8 = 5
      IGUESS = .FALSE.
      DO 30 I = 1, 9
   30 HEAD(I) = SVHEAD(I)
      IF (IEQ .EQ. 2) HEAD(4) = PARLB
   40 READ (5, 50) ANAME, ALIST
   50 FORMAT (A6, A2, 9A8)
      DO 60 J =1, 6
      IF (ANAME .EQ. ANAMES(J)) GO TO (160, 90, 110, 110, 130, 150), J
   60 CONTINUE
   70 WRITE (6, 80) ANAME, ALIST
   80 FORMAT ('OILLEGAL NAME ON SPECIFICATION CARD.  THIS JOB HAS ',
     1 'BEEN TERMINATED.'/ 1H , A6, A2, 9A8)
      STOP
   90 IFMT = IFMT + 10
      IF (IFMT .EQ. 30) GO TO 70
```

```
      DO 100 I = 1, 10
  100 FMT(I+IFMT) = ALIST(I)
      GO TO 40
  110 LABEL = J - 2
      DO 120 I = 1, 9
  120 HEAD(I) = ALIST(I+1)
      GO TO 40
  130 IGUESS = .TRUE.
C     CONVERT ALPHANUMERIC TO NUMERIC
      CALL MEMRE (ALIST(2), 64)
      READ (1, 140) P1
  140 FORMAT (8F8.0)
      GO TO 40
  150 I8 = 8
  160 IF (LIST1 .EQ. LONCE) IN = 1
      IF (LIST1 .EQ. MEMO ) IN = 2
      IF (LIST1 .EQ. LDISK) IN = 3
      IF (IEQ .EQ. 4) HEAD(4) = EXPON
      IF (NPAGE .GT. 2 .OR. IFMT .GE. 0) GC TO 180
      WRITE (6, 170)
  170 FORMAT (//'OFIRST RUN MUST HAVE FORMAT CARD.  THIS JOB HAS ',
     1 'BEEN TERMINATED.')
      STOP
C     USE PREVIOUS ORDER IF VARIABLE-CRDER INDEX AIL BLANK
  180 DO 190 I = 1, 9
      IF (KX(I) .NE. IBLANK) GO TO 200
  190 CONTINUE
C     ERROR IF FIRST RUN DOES NCT HAVE VARIABLE-CRDER INDEXES
      IF (NPAGE .LE. 2) GO TO 330
      GO TO 290
C     DETERMINE VARIABLE ORDER
  200 JMAX = 2
      NBLNKS = 0
      DO 250 K = 1, 9
      DO 210 J = 1, 10
      IF (KX(K) .EQ. KV(J)) GO TO 240
  210 CONTINUE
      GO TO 340
  240 JXSAVE(K) = J
      IF (J .EQ. 10) NBLNKS = 1
C     ERROR FOR IMBEDDED BLANKS IN VARIABLE-CRDER INDEX
      IF (NBLNKS .GT. 0 .AND. J .LT. 10) GC TC 340
      IF (J .LE. 8 .AND. JMAX .LT. J) JMAX = J
  250 CONTINUE
C     ERROR FOR A REPEATED VARIABLE INDEX
      DO 260 I = 1, 9
      IP1 = I + 1
      DO 260 J = IP1, 9
      IF (JXSAVE(I) .NE. JXSAVE(J)) GO TO 260
      IF (JXSAVE(I) .NE. 10) GO TO 340
  260 CONTINUE
C     ERROR IF ALL APPROPRIATE VARIABLES NCT SPECIFIED
      DO 280 J = 1, JMAX
      DO 270 I = 1, 9
      IF (JXSAVE(I) .EQ. J) GO TO 280
  270 CONTINUE
      GO TO 340
  280 CONTINUE
C     NUMBER OF INDEPENDENT VARIABLES
      NIV = JMAX - 1
      IF (NIV .LE. 0) GO TC 340
```

```
      NIVP1 = JMAX
      NIVP2 = JMAX + 1
C     TOTAL NUMBER OF PARAMETERS (NP)
  290 NP = NIVP1
      IF (NIV .GT. 1 .AND. (IEQ .EQ. 2 .OR. IEQ .EQ. 4)) GO TO 340
      IF (IEQ .EQ. 2 .OR. IEQ .EQ. 4) NP = 3
      DO 292 I = 1, NIVP1
      IF (EXV2(I) .NE. BLANK) EXV(I) = EXV1(I)
  292 CONTINUE
      ID = 0
      DO 300 I = 1, NIVP2
      JX(I) = JXSAVE(I)
      IF (JX(I) .NE. 9) GO TO 300
      JX(I) = NIVP2
      ID = 1
  300 CONTINUE
      IF (LABEL .EQ. 2) WRITE (6, 310) (HEAD(J+1), J = 1, NIVP1)
  310 FORMAT (1H0, 21X, A8, '  WITH  ', 7(A8,2X))
      IF (ID .EQ. 0) HEAD(1) = BLANK
      NVAR = NIVP1 + ID
      NYC = NVAR + 1
      IF (IEQ .EQ. 2) NYC = NYC + 1
      NYDEV = NYC + 1
      KOLUMN = NYDEV
      IF (IEQ .GE. 3) KOLUMN = KOLUMN + NIVP1
      IF (IEQ .EQ. 5 .OR. IEQ .EQ. 7) KOLUMN = KOLUMN - NIV
      NMAX = 2211
C     REDUCE V-SIZE BY 676 IF PLOTTING.
      IF (IVAL(2) + IVAL(3) + IVAL(4) .GT. 0) NMAX = 1535
      NMAX = NMAX / KOLUMN
      NXSET = 0
      IF (IEQ .EQ. 3 .OR. IEQ .EQ. 6 .OR. IEQ .EQ. 8) NXSET = NYDEV
      NYCOL = 1
      IF (IEQ .EQ. 3 .OR. (IEQ.GE.5 .AND. IEQ.LE.7)) NYCOL = NYDEV + 1
      IF (.NOT.IA .OR. (IEQ .NE. 3 .AND. IEQ .NE. 5)) RETURN
      WRITE (6, 320)
  320 FORMAT ('0INTERCEPT MAY NOT BE SPECIFIED FOR THIS FUNCTION.',
     1 /1H , 'THIS JOB HAS BEEN TERMINATED.')
      STOP
  330 WRITE (6, 350) JXSAVE
      STOP
  340 WRITE (6, 350) KX
  350 FORMAT (//'0THERE IS AN ERROR IN THE VARIABLE-ORDER INDEX (',
     1 'CONTROL CARD) ***',9A1,'***'/' THIS JOB HAS BEEN TERMINATED.')
      STOP
      END
```

```
      SUBROUTINE TITLER
      COMMON /C2/ M1(23), NPAGE
      REAL*4 TITLE(20), DATE(2)
      DATA IH/' '/, IH1/'1'/, DONE/'DONE'/
      READ (5, 10) TITLE
   10 FORMAT (20A4)
      IF (TITLE(1) .EQ. DONE) GO TO 30
      IF (NPAGE .EQ. 1) CALL DATER(DATE)
      ENTRY TITLE2
      WRITE (6, 20) IH, TITLE, DATE, NPAGE
   20 FORMAT (A1/1H ,20A4,10X,'DATE: ',Z5,4X,'TIME: ',Z4,4X,'PAGE:',I3/)
      NPAGE = NPAGE + 1
      IH = IH1
      RETURN
C     PRINT TERMINATION STATEMENT IF ALL DATA HAVE BEEN PROCESSED.
   30 WRITE (6, 40)
   40 FORMAT (1H1 / 10(1H0/),
     1 44X, '*******      ****      *     *    ********' /
     2 44X, '*     *     *    *     **    *    *       ' /
     3 44X, '*      *   *      *    * *   *    *       ' /
     4 44X, '*      *   *      *    *  *  *    ******  ' /
     5 44X, '*      *   *      *    *  *  *    *       ' /
     6 44X, '*      *   *      *    *   * *    *       ' /
     7 44X, '*     *     *    *     *    **    *       ' /
     8 44X, '*******      ****      *     *    ********' )
      STOP
      END
```

```
      SUBROUTINE INPUT(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION   V(NMAX,KOLUMN)
      COMMON /C1/ N, NMAX, KOLUMN, M1, NIVE1, NVAR, NP, M2, ID
      COMMON /C2/ M3(5), KEEPZ, M4, ICRD, NCARDS, M5(3), IN, I8, JX(9)
      COMMON /C3/ ABORT, M6, IA
      COMMON /C4/ Z1(136), FMT(30)
      COMMON /C5/ AN
      COMMON /C8/ DFT, DF1, DF2
      COMMON /CX/ EXV(8), NEXV, NVT
      COMMON /C0/ DATA(10,101)
      LOGICAL*4   ABORT, IE
      DIMENSION   VDATA(10)
      DATA        BLANK/' '/, IM/5/, ABLANK/'BLANK'/, I101/101/
C
C     SUBROUTINE FOR READING IN AND SAVING DATA
C
      IF (IN .EQ. 0) GO TO 160
      IF (IN .EQ. 1) GO TO 150
      KOUNTB = 0
      IF (IN .GE. 3) GO TO 80
      IBYTES = 80 * NCARDS
C     SET INPUT MEDIUM NUMBER FOR MEMORY.
      IM = 1
C     SAVE DATA, LATER IT WILL BE READ FROM MEMORY USING 'MEMRE'.
      DO 40 I = 1, I101
      READ (I8, 10) (DATA(K,I), K = 1, 10)
   10 FORMAT (10A8)
C     CHECK FOR BLANK CARD(S) (MUST BE COMPLETELY BLANK) OR 'BLANK'.
      IF (DATA(1,I) .EQ. ABLANK) GO TO 60
      DO 20 K = 1, 10
      IF (DATA(K,I) .NE. BLANK) GO TO 30
   20 CONTINUE
      KOUNTB = KOUNTB + 1
      IF (KOUNTB .EQ. NCARDS) GO TO 160
      GO TO 40
   30 KOUNTB = 0
   40 CONTINUE
   50 NMAX = I101 / NCARDS - 1
      GO TO 260
   60 JCARDS = I + NCARDS - 1
      IF (JCARDS .GT. I101) GO TO 50
      DO 70 J = I, JCARDS
      DO 70 K = 1, 10
   70 DATA(K,J) = BLANK
      GO TO 160
C     SET INPUT MEDIUM FOR DISK.
   80 IM = 4
      REWIND 4
      NREC = NCARDS * NMAX
      DO 110 I = 1, NREC
C     READ INPUT DATA AS ALPHANUMERIC DATA
      READ (I8, 10) VDATA
      IF (VDATA(1) .EQ. ABLANK) GO TO 120
C     WRITE INPUT DATA ONTO UTILITY DISK.
      WRITE (4, 10) VDATA
      DO 90 K = 1, 10
      IF (VDATA(K) .NE. BLANK) GO TO 100
   90 CONTINUE
      KOUNTB = KOUNTB + 1
```

```
      IF (KOUNTB .EQ. NCARDS) GO TO 160
      GO TO 110
  100 KOUNTB = 0
  110 CONTINUE
      GO TO 260
  120 DO 130 K = 1, 10
  130 VDATA(K) = BLANK
      DO 140 J = 1, NCARDS
  140 WRITE (4, 10) VDATA
      GO TO 160
C     SET INPUT MEDIUM FOR CARDS.
  150 IM = 5
  160 KRECRD = 1 - NCARDS
      IF (IM .GE. 5) IM = I8
      IF (IM .EQ. 4) REWIND 4
C     IF SUBROUTINE 'MEMRE' IS NOT ASSEMBLED, DO NOT USE 'READ MEMORY'
C     OR 'GUESS' SPECIFICATION OPTIONS.  FURTHER, THE DIMENSIONS ON
C     COMMON /CO/ MAY BE REDUCED TO 'DATA(1,1)' IN MAIN AND INPUT.
C     CREATE DUMMY PROGRAMS FOR 'MEMRE' AND 'DATER'.
      NEG = 0
      DO 200 I = 1, NMAX
  170 KRECRD = KRECRD + NCARDS
      IF (IM .EQ. 1) CALL MEMRE(DATA(1,KRECRD),IBYTES)
C     READ INPUT DATA FROM CARDS, FROM MEMORY, OR FROM UTILITY DISK
      READ (IM, FMT) (V(I, JX(J)), J = 1, NVAR)
      NZ = 0
      DO 180 K = 1, NIVP1
      IF (V(I,K) .EQ. 0.D0) NZ = NZ + 1
      IF (V(I,K) .LT. 0.D0 .AND. KEEPZ .NE. 2) NEG = 1
  180 CONTINUE
      IF (NZ .LT. NIVP1) GO TO 190
      IF (.NOT.ID .OR. V(I,NVAR) .EQ. BLANK .OR.
     1 V(I,NVAR) .EQ. ABLANK) GO TO 210
  190 IF (NZ .GT. 0 .AND. KEEPZ .EQ. 0) GO TO 170
  200 CONTINUE
      GO TO 260
C     SET N EQUAL TO NUMBER OF DATA POINTS.
  210 N = I - 1
      IF (NEG .EQ. 1) GO TO 300
      AN = N
      NEXV = 0
      NVT = 0
      DO 212 J = 1, NIVP1
      IF (EXV(J) .NE. 0.D0 .AND. EXV(J) .NE. 1.D0) GO TO 214
  212 CONTINUE
      GO TO 216
  214 NVT = 1
      CALL TRANS(V)
      IF (ABORT) RETURN
C     TOTAL DEGREES OF FREEDOM
  216 DFT = AN - 1.D0 + IA
C     DEGREES OF FREEDOM FOR ERROR
      DF1 = N - NP + IA
C     DEGREES OF FREEDOM DUE TO REGRESSION
      DF2 = DFT - DF1
      IF (DF1 .LT. 0.D0) GO TO 280
      IF (IORD .EQ. 0) RETURN
C     ORDER THE DATA FROM LOW TO HIGH VALUES OF Y.
      NK = N - 1
      DO 250 I = 1, NK
      IP1 = I + 1
```

```
      DO 240 J = IP1, N
      IF (V(I,1) .LE. V(J,1)) GO TO 240
      DO 230 K = 1, NVAR
      TEMP = V(J,K)
      L = J
  220 V(L,K) = V(L-1,K)
      L = L - 1
      IF (L .GT. I) GO TO 220
  230 V(I,K) = TEMP
  240 CONTINUE
  250 CONTINUE
      RETURN
C     ERROR MESSAGES
  260 WRITE (6, 270) NMAX
  270 FORMAT (//'0NUMBER OF INPUT DATA POINTS HAS EXCEEDED ',
     1 'MAXIMUM ALLOWABLE (',I3,').  THIS JOB HAS BEEN TERMINATED.')
      STOP
  280 WRITE (6, 290) N
  290 FORMAT (//'0THE NUMBER OF DATA POINTS (',I1, ') IS LESS ',
     1 'THAN THE NUMBER OF PARAMETERS TO BE SOLVED.  THIS RUN HAS ',
     2 'BEEN TERMINATED.' )
      ABORT = .TRUE.
      RETURN
  300 WRITE (6, 310) I
  310 FORMAT (//'0A NEGATIVE VALUE EXISTS IN THE INPUT DATA',
     1 ' (FOR EXAMPLE, DATA CARD NUMBER ', I3,').  THIS RUN HAS BEEN ',
     2 'TERMINATED.' )
      ABORT = .TRUE.
      RETURN
      END
```

```
      SUBROUTINE TRANS(V)
      IMPLICIT REAL*8(A-H,C-Z)
      DIMENSION   V(NMAX,KCLUMN)
      COMMON /C1/ N, NMAX, KCLUMN, NIV, NIVP1, M1, NP
      COMMON /C2/ IEQ
      COMMON /C3/ ABORT
      COMMON /CX/ EXV(8), NEXV, NVT
      LOGICAL*4   ABORT
   10 DO 140 J = 1, NIVP1
      IF (EXV(J) .EQ. 0.D0) GO TO 140
      IF (EXV(J) .EQ. 88.D0) GO TO 30
      IF (EXV(J) .EQ. 99.D0) GO TO 150
      IF (EXV(J) .LE. (-10.D0) .OR. EXV(J) .GT. 47.D0) GO TO 160
      IF (EXV(J) .GE. 10.D0) GO TO 50
C     EXPONENTIAL TRANSFORMATIONS
      DO 20 I = 1, N
   20 V(I,J) = V(I,J)**EXV(J)
      GO TO 140
C     LOGARITHMIC TRANSFORMATIONS
   30 DO 40 I = 1, N
   40 V(I,J) = DLOG(V(I,J))
      GO TO 140
   50 KEXV = EXV(J)
      IF (FLOAT(KEXV) .NE. EXV(J)) GO TO 160
      NEXV = 1
      KLV = KEXV / 10
      KMV = KEXV - 10 * KLV + 1
      IF (KMV .GT. 8) GO TO 160
C     COMBINATIONS OF VARIABLES
      GO TO (60, 80, 100, 120), KLV
   60 DO 70 I = 1, N
   70 V(I,J) = V(I,J) * V(I,KMV)
      GO TO 140
   80 DO 90 I = 1, N
   90 V(I,J) = V(I,J) / V(I,KMV)
      GO TO 140
  100 DO 110 I = 1, N
  110 V(I,J) = V(I,J) + V(I,KMV)
      GO TO 140
  120 DO 130 I = 1, N
  130 V(I,J) = V(I,J) - V(I,KMV)
  140 CONTINUE
      NEXV = 0
      RETURN
  150 IF (J .LE. 2 .OR. NEXV .EQ. 0) GO TO 160
      NEXV = NIVP1 + 1 - J
      NIV = NIV - NEXV
      NIVP1 = NIV + 1
      IF (IEQ .NE. 2 .AND. IEQ .NE. 4) NP = NP - NEXV
      RETURN
C     WRITE ERROR MESSAGE.
  160 WRITE (6, 170)
  170 FORMAT (//'0A TRANSFORMATION FACTOR HAS NOT BEEN ENTERED ',
     1 'CORRECTLY.  THIS RUN HAS BEEN TERMINATED.' )
      ABORT = .TRUE.
      RETURN
      END
```

```
      SUBROUTINE PRINT
      IMPLICIT REAL*8(A-H,C-Z)
      COMMON /C1/ M1(3), NIV
      COMMON /C2/ IEQ
      DATA       IB/' '/, IP/'+'/, IR/')'/
C
C     SUBROUTINE FOR PRINTING SUBHEADINGS
C
      GO TO (10, 30, 50, 70, 90, 110, 130, 150, 170), IEQ
   10 WRITE (6, 20) (IB, I = 1, NIV)
   20 FORMAT ('OLINEAR REGRESSION -- Y = A',A1,'+ B * X1',A1,
     1 '+ C * X2',A1,'+ D * X3',A1,'+ E * X4',A1,'+ F * X5',A1,
     2 '+ G * X6',A1,'+ H * X7')
      RETURN
   30 WRITE (6, 40)
   40 FORMAT ('OQUADRATIC REGRESSION -- Y = A + B * X1 + C * X1**2')
      RETURN
   50 WRITE (6, 60) (IB, I = 1, NIV)
   60 FORMAT ('OPOWER REGRESSION -- Y = A',A1,'* X1**B',A1,'* X2**C',
     1 A1,'* X3**D',A1,'* X4**E',A1,'* X5**F',A1,'* X6**G',A1,'* X7**H')
      RETURN
   70 WRITE (6, 80)
   80 FORMAT ('OASYMPTOTIC-POWER REGRESSION -- Y = A + B * X1**C')
      RETURN
   90 WRITE (6, 100) (IB, IP, I = 1, NIV), IR
  100 FORMAT ('OEXPONENTIAL REGRESSION -- Y = EXP(A',2A1,' B * X1',2A1,
     1 ' C * X2',2A1,' D * X3',2A1,' E * X4',2A1,' F * X5',2A1,
     2 ' G * X6',2A1,' H * X7',A1)
      RETURN
  110 WRITE (6, 120) (IB, I = 1, NIV)
  120 FORMAT ('OLOG-LINEAR REGRESSION -- LN Y = LN A',A1,'+ B * LN X1',
     1 A1,'+ C * LN X2',A1,'+ D * LN X3',A1,'+ E * LN X4',
     2 A1,'+ F * LN X5',A1,'+ G * LN X6',A1,'+ H * LN X7')
      RETURN
  130 WRITE (6, 140) (IB, I = 1, NIV)
  140 FORMAT ('OSEMILOG REGRESSION -- LN Y = A',A1,'+ B * X1',A1,
     1 '+ C * X2',A1,'+ D * X3',A1,'+ E * X4',A1,'+ F * X5',A1,
     2 '+ G * X6',A1,'+ H * X7')
      RETURN
  150 WRITE (6, 160) (IB, I = 1, NIV)
  160 FORMAT ('OSEMILOG REGRESSION -- Y = A',A1,'+ B * LN X1',
     1 A1,'+ C * LN X2',A1,'+ D * LN X3',A1,'+ E * LN X4',
     2 A1,'+ F * LN X5',A1,'+ G * LN X6',A1,'+ H * LN X7')
  170 RETURN
      END
```

```
      SUBROUTINE SUMS(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION   V(NMAX,KOLUMN)
      COMMON /C1/ N, NMAX, KOLUMN, NIV, NIVP1, NVAR, M1(2), ID
      COMMON /C2/ IEQ
      COMMON /C3/ M2, LINEAR, IA, M3, NXSET, NYCOL, M4, NYDEV
      COMMON /C4/ S(8), SYX(8,8)
      COMMON /C5/ AN, BB(40), VMSQ(8)
      COMMON /C6/ VMEAN(8), T(8), H(8,8), Z1(8), SDEV(8)
      COMMON /C7/ YCEPT
      COMMON /C8/ DFT, Z2(7), SST
      LOGICAL*4   LINEAR, IA
C
C     SUBROUTINE FOR COMPUTING VARIOUS SUMS, MEANS, AND STANDARD
C      DEVIATIONS OF THE INPUT DATA
C
      LINEAR = IEQ .LE. 1 .OR. IEQ .GE. 6
      IF (IEQ .NE. 2) GO TO 20
C     MODIFICATIONS FOR QUADRATIC CASE
      NIV = 2
      NIVP1 = 3
      NVAR = 3 + ID
      DO 10 I = 1, N
      IF (ID .GT. 0) V(I,4) = V(I,3)
   10 V(I,3) = V(I,2) * V(I,2)
C     INITIALIZE SUMS.
   20 DO 30 J = 1, NIVP1
      S(J) = 0.D0
      DO 30 K = J, NIVP1
   30 SYX(J,K) = 0.D0
      IF (IEQ .LE. 2) GO TO 50
      J1 = 1
      J2 = NIVP1
      IF (IEQ .EQ. 5 .OR. IEQ .EQ. 7) J2 = 1
      IF (IEQ .EQ. 4 .OR. IEQ .EQ. 8) J1 = 2
      DO 40 J = J1, J2
      DO 40 I = 1, N
   40 V(I,J+NYDEV) = DLOG(V(I,J))
   50 DO 80 I = 1, N
      YI = V(I,NYCOL)
      S(1) = S(1) + YI
      SYX(1,1) = SYX(1,1) + YI * YI
      DO 70 J = 2, NIVP1
      XIJ = V(I,J+NXSET)
      S(J) = S(J) + XIJ
      SYX(1,J) = SYX(1,J) + YI * XIJ
      DO 60 K = J, NIVP1
      SYX(J,K) = SYX(J,K) + XIJ * V(I,K+NXSET)
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
C     TOTAL SUM OF SQUARES
      SST = SYX(1,1) - S(1) * S(1) / AN
C     MEANS AND STANDARD DEVIATIONS OF THE INPUT DATA
      DO 90 J = 1, NIVP1
      VMEAN(J) = S(J) / AN
      VMSQ(J) = SYX(J,J) - S(J) * S(J) / AN
   90 SDEV(J) = DSQRT(VMSQ(J) / (AN - 1.D0))
      IF (IEQ .EQ. 4) RETURN
      IF (IA) GO TO 120
```

```
      DO 110 J = 2, NIVP1                                               SUMS0610
      T(J-1) = SYX(1,J) - S(1) * S(J) / AN                              SUMS0620
      DO 100 K = J, NIVP1                                               SUMS0630
  100 H(J-1,K-1) = SYX(J,K) - S(J) * S(K) / AN                          SUMS0640
  110 CONTINUE                                                          SUMS0650
      RETURN                                                            SUMS0660
  120 DO 140 J = 2, NIVP1                                               SUMS0670
      T(J-1) = SYX(1,J) - YCEPT * S(J)                                  SUMS0680
      DO 130 K = J, NIVP1                                               SUMS0690
  130 H(J-1,K-1) = SYX(J,K)                                             SUMS0700
  140 CONTINUE                                                          SUMS0710
      RETURN                                                            SUMS0720
      END                                                               SUMS0730
```

```
      SUBROUTINE LINE(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION   V(NMAX,KCLUMN)
      COMMON /C1/ N, NMAX, KOLUMN, NIV, NIVP1, M1, NP
      COMMON /C2/ IEQ, M2(9), IND
      COMMON /C3/ ABORT, M3, IA, M4, NXSET, M5, NYC
      COMMON /C5/ Z1, P(8)
      COMMON /C6/ VMEAN(8), T(8)
      COMMON /C7/ YCEPT, Z2(4), XV, YV
      LOGICAL*4   ABORT, IA
      DATA        IND1/'LINE'/
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C     FOR LINEAR FUNCTIONS OF FORM
C
C       Y = A + B*X1 + C*X2 + D*X3 + E*X4 + F*X5 + G*X6 + H*X7
C
C     WHERE A MAY BE SPECIFIED
C
      IND = IND1
C     SOLVE FOR PARAMETERS (P).
      CALL SOLVE(NIV)
      IF (ABORT) RETURN
      P(1) = VMEAN(1)
      DO 10 J = 2, NP
      P(J) = T(J-1)
   10 P(1) = P(1) - P(J) * VMEAN(J)
      IF (IA) P(1) = YCEPT
C     Y-COMPUTED AND Y-RESIDUAL VALUES
      DO 30 I = 1, N
      V(I,NYC) = P(1)
      DO 20 J = 2, NIVP1
   20 V(I,NYC) = V(I,NYC) + P(J) * V(I,J+NXSET)
   30 CONTINUE
      IF (IEQ .NE. 2) RETURN
      XV = -P(2) / (2.D0 * P(3) )
      YV = P(1) + P(2) * XV + P(3) * XV * XV
      NIV = 1
      NIVP1 = 2
      RETURN
      END
```

```
      SUBROUTINE EXPO(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION    V(NMAX,KOLUMN)
      COMMON /C1/ M1, NMAX, KOLUMN, NIV, M2(2), NP
      COMMON /C2/ IEQ, M3(9), IND
      COMMON /C3/ ABORT, M4(2), IGUESS
      COMMON /C6/ VMEAN(8), T(8), Z1(64), E1(8)
      LOGICAL*4    ABORT, IGUESS
      DATA         IND2/'EXPO'/
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C     FOR POWER FUNCTIONS OF FORM
C
C       Y = A * X1**B * X2**C * X3**D * X4**E * X5**F * X6**G * X7**H
C
C     OR FOR EXPONENTIAL FUNCTIONS OF FORM
C
C       Y = EXP(A + B*X1 + C*X2 + D*X3 + E*X4 + F*X5 + G*X6 + H*X7)
C
C     SET SUBROUTINE INDICATOR
      IND = IND2
      IF (IGUESS) GO TO 20
C     FIRST, DETERMINE LEAST-SQUARES SOLUTIONS OF PARAMTERS P1(J)
C     FOR LOG-LINEAR FORM (POWER)
C       LN(Y) = LN(A1) + B1 * LN(X1) + C1 * LN(X2) + . . . + H1 * LN(X7)
C     OR SEMILOG-LINEAR FORM (EXPONENTIAL)
C       LN(Y) = A1 + B1 * X1 + C1 * X2 + . . . + H1 * X7
      CALL SOLVE(NIV)
      IF (ABORT) RETURN
      P1(1) = VMEAN(1)
      DO 10 J = 2, NP
      P1(J) = T(J-1)
   10 P1(1) = P1(1) - P1(J) * VMEAN(J)
      IF (IEQ .EQ. 3) P1(1) = DEXP(P1(1))
C     DETERMINE LEAST-SQUARES SOLUTIONS OF PARAMETERS P(J)
   20 CALL ITER(V)
      RETURN
      END
```

AD-A032 579   RAND CORP SANTA MONICA CALIF   F/G 12/1
CURVES: A COST ANALYSIS CURVE-FITTING PROGRAM.(U)
SEP 76 H E BOREN, G W CORWIN   F44620-73-C-0011
UNCLASSIFIED   R-1753-1-PR   NL

2 OF 2
AD
A032579

END
DATE
FILMED
1-77

```
      SUBROUTINE ASYM(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION   V(NMAX,KCLUMN)
      COMMON /C1/ N, NMAX, KCLUMN, M1(2), NVAR
      COMMON /C3/ ABORT, M2, IA, M3(3), NYC
      COMMON /C4/ S(8), SYX11
      COMMON /C5/ AN, A, B, C
      COMMON /C6/ VMEAN(8), T(8), H(8,8)
      COMMON /C7/ YCEPT, DELTA
      COMMON /C8/ Z1(6), YDEVSQ, Z2, SST
      DIMENSION   SUM(9), SAVSUM(4)
      LOGICAL*4   ABORT, IA, SOLVED
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C     FOR ASYMPTOTIC-POWER FUNCTIONS OF FORM
C
C                             Y = A + B * X**C
C
C     WHERE A MAY BE SPECIFIED
C
      A = YCEPT
      SOLVED = .FALSE.
C     SET INITIAL VALUE OF C1
      C1 = -8.101D0
C     SET FIRST-ITERATION DESIGNATOR TO 1.
      ITERS = 1
C     SET C INCREMENT INITIALLY TO 0.1.
      DC = 0.1D0
      DO 200 K = 1, 162
C     STEP INITIAL C VALUE BY DC INCREMENT.
      C1 = C1 + DC
      IF (K .NE. 82) GO TO 10
      ITERS = 1
      C1 = 0.001D0
   10 C = C1
   20 DO 30 I = 1, 7
   30 SUM(I) = 0.D0
      DO 40 I = 1, N
      XP = V(I,2)**C
      XPSQ = XP * XP
C     V(I,NVAR+4) IS LN(X1).
      XP1 = XP * V(I,NVAR+4)
      SUM(1) = SUM(1) + XP
      SUM(2) = SUM(2) + XPSQ
      SUM(3) = SUM(3) + XP1
      SUM(4) = SUM(4) + XPSQ * V(I,NVAR+4)
      SUM(5) = SUM(5) + XP1 * V(I,1)
      SUM(6) = SUM(6) + XP * (V(I,1) - VMEAN(1))
      IF (IA) SUM(7) = SUM(7) + XP * V(I,1)
   40 CONTINUE
      IF (.NOT.IA) GO TO 50
      B = (SUM(7) - A * SUM(1)) / SUM(2)
      G = SUM(4)*SUM(7)-SUM(2)*SUM(5)-A*(SUM(1)*SUM(4)-SUM(2)*SUM(3))
      YDEVS1 = SYX11 - 2.D0*A*S(1) + AN*A*A - B*B*SUM(2)
      GO TO 60
   50 B = SUM(6)/(SUM(2) - (SUM(1) * SUM(1)/AN))
      A = (S(1) - (B * SUM(1)))/AN
      G = SUM(5) - (B * SUM(4)) - (A * SUM(3))
      YDEVS1 = SST - B * SUM(6)
   60 IF (K .EQ. 1) QSAVE1 = YDEVS1
```

```
      IF (K .EQ. 162) QSAVE3 = YDEVS1
      IF (ITERS .EQ. 2) GO TO 100
      IF (G) 70,160,80
   70 M = -1
      GO TO 90
   80 M = 1
   90 ITERS = 2
      GO TO 190
  100 IF (M .GT. 0) GO TO 110
      IF (G) 130,160,120
  110 IF (G) 120,160,130
  120 C = C - (DC * 0.5D0)
      GO TO 140
  130 IF (DC .GT. 0.07D0) GO TO 190
      C = C + (DC * 0.5D0)
  140 IF (IA) GO TO 142
      DDA = DABS(A/ASTORE - 1.D0)
      IF (DDA .GE. DELTA) GO TO 150
  142 DDB = DABS(B/BSTORE - 1.D0)
      IF (DDB .GE. DELTA) GO TO 150
      DDC = DABS(C/CSTORE - 1.D0)
      IF (DDC .GE. DELTA) GO TO 150
      GO TO 160
  150 ASTORE = A
      BSTORE = B
      CSTORE = C
      DC = DC * 0.5D0
      GO TO 20
C     USE NEW VARIABLES FOR TEMPORARY SOLUTION.
  160 YDEVSQ = YDEVS1
      IF (.NOT. SOLVED .OR. YDEVSQ .LE. QSAVE2) GO TO 170
      A = ASAVE
      B = BSAVE
      C = CSAVE
      SUM(1) = SAVSUM(1)
      SUM(2) = SAVSUM(2)
      SUM(4) = SAVSUM(4)
      YDEVSQ = QSAVE2
      GO TO 180
C     STORE PARAMETER VALUES AND SUM OF SQUARES OF Y RESIDUALS.
  170 ASAVE = A
      BSAVE = B
      CSAVE = C
      SAVSUM(1) = SUM(1)
      SAVSUM(2) = SUM(2)
      SAVSUM(4) = SUM(4)
      QSAVE2 = YDEVSQ
C     SET FIRST-ITERATION DESIGNATOR TO 1.
  180 ITERS = 1
C     SET SOLUTION DESIGNATOR TO TRUE
      SOLVED = .TRUE.
      DC = 0.1D0
      C1 = C1 - DC
      GO TO 200
  190 ASTORE = A
      BSTORE = B
      CSTORE = C
  200 CONTINUE
C     IF A UNIQUE SOLUTION FOR C DOES NOT EXIST IN THE SPECIFIED RANGE,
C     PRINT A MESSAGE RELATING TO THAT FACT.
      IF (SOLVED.AND.YDEVSQ.LT.QSAVE1.AND.YDEVSQ.LT.QSAVE3) GO TO 220
```

```
      WRITE (6, 210)
  210 FORMAT (//'0NO SOLUTION HAS BEEN FOUND FOR THIS PROBLEM IN ',
     1 'THE RANGE OF -8 TO +8 FOR C.')
      ABORT = .TRUE.
      RETURN
C     RESTORE VARIABLES TO ORIGINALS.
  220 A = ASAVE
      B = BSAVE
      C = CSAVE
      SUM(1) = SAVSUM(1)
      SUM(2) = SAVSUM(2)
      SUM(4) = SAVSUM(4)
      SUM(8) = 0.D0
      SUM(9) = 0.D0
      T(1) = 0.D0
      T(2) = 0.D0
      T(3) = 0.D0
      DO 230 I = 1, N
      XP = V(I,2)**C
      V(I,NYC) = A + B * XP
      XPL = V(I,NVAR+4) * XP
      SUM(8) = SUM(8) + XPL
      SUM(9) = SUM(9) + XPL * XPL
  230 CONTINUE
      IF (IA) GO TO 240
      H(1,1) = AN
      H(1,2) = SUM(1)
      H(1,3) = B * SUM(8)
      GO TO 250
  240 H(1,1) = 1.D0
      H(1,2) = 0.D0
      H(1,3) = 0.D0
  250 H(2,2) = SUM(2)
      H(2,3) = B * SUM(4)
      H(3,3) = B * B * SUM(9)
      CALL SOLVE(3)
      RETURN
      END
```

```
      SUBROUTINE SOLVE(NSIZE)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /C2/ M1(10), IND
      COMMON /C3/ ABORT
      COMMON /C6/ Z1(8), T(8), H(8,8)
      LOGICAL*4   ABORT
      DIMENSION   IPIVOT(8), INDEX(8,2)
C
C     SUBROUTINE FOR SOLVING SIMULTANEOUS EQUATIONS
C     H BECOMES INVERTED H; T BECOMES SOLUTION VECTOR.
C
      IF (NSIZE .GT. 1) GO TO 10
      IF (H(1,1) .EQ. 0.D0) GO TO 150
      H(1,1) = 1.D0 / H(1,1)
      T(1) = H(1,1) * T(1)
      RETURN
C     FILL OUT LOWER TRIANGLE OF H MATRIX.
   10 DO 20 J = 2, NSIZE
      JM1 = J - 1
      DO 20 K = 1, JM1
   20 H(J,K) = H(K,J)
C     INITIALIZATION
      DO 30 J = 1, NSIZE
   30 IPIVOT(J) = 0
C     PARTIAL MATRIX INVERSION ROUTINE
      DO 120 I = 1, NSIZE
C     SEARCH FOR PIVOT ELEMENT
      HMAX = 0.D0
      DO 60 J = 1, NSIZE
      IF (IPIVOT(J) .EQ. 1) GO TO 60
      DO 50 K = 1, NSIZE
      IF (IPIVOT(K) - 1) 40, 50, 150
   40 IF (DABS(HMAX) .GE. DABS(H(J,K))) GO TO 50
      IROW = J
      ICOL = K
      HMAX = H(J,K)
   50 CONTINUE
   60 CONTINUE
      IF (HMAX .EQ. 0.D0) GO TO 150
      IPIVOT(ICOL) = IPIVOT(ICOL) + 1
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
      IF (IROW .EQ. ICOL) GO TO 80
      DO 70 L = 1, NSIZE
      TEMP = H(IROW,L)
      H(IROW,L) = H(ICOL,L)
   70 H(ICOL,L) = TEMP
      TEMP = T(IROW)
      T(IROW) = T(ICOL)
      T(ICOL) = TEMP
   80 PIVOT = H(ICOL,ICOL)
      INDEX(I,1) = IROW
      INDEX(I,2) = ICOL
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
      H(ICOL,ICOL) = 1.D0
      DO 90 L = 1, NSIZE
   90 H(ICOL,L) = H(ICOL,L)/PIVOT
      T(ICOL) = T(ICOL)/PIVOT
C     REDUCE NON-PIVOT ROWS
      DO 110 L1 = 1, NSIZE
      IF (L1 .EQ. ICOL) GO TO 110
```

```
      TEMP = H(L1,ICOL)
      H(L1,ICOL) = 0.D0
      DO 100 L = 1, NSIZE
  100 H(L1,L) = H(L1,L) - H(ICOL,L) * TEMP
      T(L1) = T(L1) - T(ICOL) * TEMP
  110 CONTINUE
  120 CONTINUE
C     INTERCHANGE COLUMNS.
      DO 140 I = 1, NSIZE
      L = NSIZE + 1 - I
      IF (INDEX(L,1) .EQ. INDEX(L,2)) GO TO 140
      IROW = INDEX(L,1)
      ICOL = INDEX(L,2)
      DO 130 K = 1, NSIZE
      TEMP = H(K,IROW)
      H(K,IROW) = H(K,ICOL)
      H(K,ICOL) = TEMP
  130 CONTINUE
  140 CONTINUE
      RETURN
C     ERROR MESSAGE
  150 WRITE (6, 160) IND
  160 FORMAT ('0ZERO DETERMINANT IN SUBROUTINE SOLVE.  THIS RUN ',
     1 'HAS BEEN TERMINATED.'/ 1H0, 'SOLVE WAS LAST CALLED FROM ', A4)
      ABORT = .TRUE.
      RETURN
      END
```

```
      SUBROUTINE ITER(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION    V(NMAX,KOLUMN)
      COMMON /C1/ N, NMAX, KOLUMN, M1, NIVP1
      COMMON /C2/ IEQ, M2(8), LIM, IND
      COMMON /C3/ ABORT, M3(3), NXSET, NYCOL, NYC
      COMMON /C4/ S(8), SYX(8,8)
      COMMON /C5/ AN, P(8)
      COMMON /C6/ VMEAN(8), T(8), H(8,8), P1(8), SDEV(8)
      COMMON /C7/ Z1, DELTA, Z2, EA
      COMMON /C8/ DFT, Z3(7), SST
      LOGICAL*4   ABORT, SOLVED, IEQ3
      DIMENSION   FP(8), PTEMP(3,8), Q(3), DIFF(8)
      DATA        IND3/'ITER'/
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C     FOR NON-LINEAR FUNCTIONS WHERE AN ITERATIVE PROCEDURE IS
C     REQUIRED (VIZ, POWER (IEQ=3) AND EXPONENTIAL (IEQ=5) FUNCTIONS)
C
C     SET SUBROUTINE INDICATOR
      IND = IND3
      IEQ3 = IEQ .EQ. 3
C     SET INITIAL GUESSES
      DO 10 J = 1, NIVP1
   10 P(J) = P1(J)
C     RECALCULATE SUM, SUMSQ, MEAN AND STD DEV FOR ACTUAL (NONLOG) DATA.
      DO 20 J = 1, NIVP1
      S(J) = 0.D0
      DO 20 K = J, NIVP1
   20 SYX(J,K) = 0.D0
      DO 30 I = 1, N
      DO 30 J = 1, NIVP1
      S(J) = S(J) + V(I,J)
      DO 30 K = J, NIVP1
   30 SYX(J,K) = SYX(J,K) + V(I,J) * V(I,K)
      DO 40 J = 1, NIVP1
      VMEAN(J) = S(J) / AN
   40 SDEV(J) = DSQRT((SYX(J,J) - S(J) * S(J) / AN) / DFT)
      SST = SYX(1,1) - S(1) * S(1) / AN
C     SET SOLUTION INDICATOR TO FALSE
      SOLVED = .FALSE.
   50 DO 310 L = 1, LIM
C     COMPUTED Y VALUES AND PARTIAL OF Y WITH RESPECT TO PARAMETER A.
      IF (IEQ3) GO TO 80
      DO 70 I = 1, N
      TEMP = P(1)
      DO 60 J = 2, NIVP1
   60 TEMP = TEMP + P(J) * V(I,J)
   70 V(I,NYC) = DEXP(TEMP)
      GO TO 110
   80 DO 100 I = 1, N
      TEMP = P(1)
      DO 90 J = 2, NIVP1
   90 TEMP = TEMP * V(I,J)**P(J)
  100 V(I,NYC) = TEMP
C     IF A SOLUTION HAS BEEN OBTAINED (SOLVED=TRUE), GO TO ENDING.
  110 IF (SOLVED) GO TO 330
C     CLEAR H AND T MATRICES.
      DO 130 I = 1, NIVP1
      T(I) = 0.D0
```

```
      DO 120 J = I, NIVP1
  120 H(I,J) = 0.D0
  130 CONTINUE
      DO 170 I = 1, N
C     PARTIAL OF Y FUNCTION WITH RESPECT TO PARAMETERS
      FP(1) = V(I,NYC)
      IF (IEQ3) FP(1) = V(I,NYC) / P(1)
      DO 140 J = 2, NIVP1
  140 FP(J) = V(I,NYC) * V(I,J+NXSET)
C     DIFFERENCE BETWEEN OBSERVED Y AND CALCULATED Y (Y RESIDUAL)
      YDIFF = V(I,1) - V(I,NYC)
C     CALCULATE H AND T MATRICES.
      DO 160 II = 1, NIVP1
      DO 150 JJ = II, NIVP1
  150 H(II,JJ) = H(II,JJ) + (FP(II) * FP(JJ))
      T(II) = T(II) + YDIFF * FP(II)
  160 CONTINUE
  170 CONTINUE
C     SOLVE FOR CORRECTIONS TO PREVIOUS SOLUTIONS.
      CALL SOLVE(NIVP1)
      IF (ABORT) RETURN
C     FIND WHICH FRACTIONAL PART OF CORRECTION TERMS, WHEN ADDED TO
C     PARAMETER VALUES, GIVES LOWEST SUM OF SQUARES OF Y RESIDUALS.
      TEMP = 1.0D0
  180 TEMP = 0.5D0 * TEMP
      DO 250 J = 1, 3
      FI = TEMP * (J - 1)
      DO 190 K = 1, NIVP1
  190 PTEMP(J,K) = P(K) + T(K) * FI
      Q(J) = 0.D0
      DO 240 I = 1, N
      YTEMP = PTEMP(J,1)
      IF (IEQ3) GO TO 210
      DO 200 K = 2, NIVP1
  200 YTEMP = YTEMP + PTEMP(J,K) * V(I,K)
      YTEMP = DEXP(YTEMP)
      GO TO 230
  210 DO 220 K = 2, NIVP1
  220 YTEMP = YTEMP * V(I,K)**PTEMP(J,K)
  230 YDIFF = V(I,1) - YTEMP
      Q(J) = Q(J) + (YDIFF * YDIFF)
  240 CONTINUE
  250 CONTINUE
      LM = 1
      DO 260 J = 2, 3
      IF (Q(LM) .LT. Q(J)) GO TO 260
      LM = J
  260 CONTINUE
      IF (LM .GT. 1) GO TO 280
      DO 270 I = 1, NIVP1
      IF (DABS(T(I) * 2.D0 * TEMP) .GT. DELTA) GO TO 180
  270 CONTINUE
  280 DO 290 I = 1, NIVP1
      DIFF(I) = DABS(PTEMP(LM,I)/P(I) - 1.D0)
      P(I) = PTEMP(LM,I)
  290 CONTINUE
      DO 300 I = 1, NIVP1
C     SOLVED WHEN RELATIVE DIFFERENCE IS LESS THAN DELTA.
      IF (DIFF(I) .GE. DELTA) GO TO 310
  300 CONTINUE
      SOLVED = .TRUE.
```

```
C     GO BACK AND COMPUTE NEW YC AND YDEV FOR FINAL CORRECTIONS
  310 CONTINUE
      IF (SOLVED) GO TO 50
C     ERROR MESSAGE
      WRITE (6, 320) LIM, (J, P1(J), J, P(J), DIFF(J), J = 1, NIVP1)
  320 FORMAT (//'0NO SOLUTION HAS BEEN OBTAINED FOR THIS RUN ',
     1 'AFTER ', I3, ' ITERATIONS.  THIS RUN HAS BEEN TERMINATED.'/
     2 1H0, 7X, 'INITIAL GUESSES', 20X, 'SOLUTION AT TERMINATION',
     * 20X, 'RELATIVE DIFFERENCE' / (/' P1(', I1, ') =', D15.5,
     4 20X, 'P(', I1, ') =', D17.5, D40.5) )
      ABORT = .TRUE.
      RETURN
  330 NYCOL = 1
      NXSET = 0
      IF (.NOT.IEQ3) EA = DEXP(P(1))
      RETURN
      END
```

```
      SUBROUTINE STAT(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION   V(NMAX,KOLUMN)
      COMMON /C1/ N, NMAX, KOLUMN, NIV, NIVP1
      COMMON /C2/ M1(11), IFV
      COMMON /C3/ M2(2), IA, M3(2), NYCOL, NYC, NYDEV
      COMMON /C4/ S(8), SYX(8,8)
      COMMON /C5/ AN, A
      COMMON /C6/ VMEAN(8), Z1(80), SDEV(8), RM(8,8)
      COMMON /C7/ Z2(2), RDEVM, RR(4), SEYSQ
      COMMON /C8/ DFT, DF1, DF2, CD, SEY, CV, YDEVSQ, FVALUE, SST
      LOGICAL*4   IA
C
C     SUBROUTINE FOR CALCULATING STATISTICS
C
      RDEVM = 0.D0
      YZERO = 0.D0
      YDEVSQ = 0.D0
      DO 20 I = 1, N
      V(I,NYDEV) = V(I,NYCOL) - V(I,NYC)
C     SUM OF SQUARES OF Y RESIDUALS
      YDEVSQ = YDEVSQ + V(I,NYDEV) * V(I,NYDEV)
      IF (V(I,NYCOL) .NE. 0.D0) GO TO 10
      YZERO = YZERO + 1.D0
      GO TO 20
   10 RDEVM = RDEVM + DABS(V(I,NYDEV) / V(I,NYCOL))
   20 CONTINUE
      RDEVM = RDEVM / (AN - YZERO)
      SEYSQ = 0.D0
      IF (DF1 .GT. 0.D0) SEYSQ = YDEVSQ / DF1
C     STANDARD ERROR OF THE ESTIMATE OF Y
      SEY = DSQRT(SEYSQ)
C     COEFFICIENT OF VARIATION (DECIMAL)
      CV = SEY / VMEAN(1)
      IF (IA) SST = SYX(1,1) - 2.D0*A*S(1) + A*A*AN
      RATIO = 0.D0
      IF (SST .GT. 0.D0) RATIO = YDEVSQ / SST
C     COEFFICIENT OF DETERMINATION
      CD = 1.D0 - RATIO
      IFV = 0
      DENOM = DF2 * RATIO
      IF (DENOM .GT. 0.D0) GO TO 30
      IFV = 2
      GO TO 40
C     F VALUE
   30 FVALUE = DF1 * CD / DENOM
      IF (FVALUE .GT. 1.0D+08) IFV = 1
   40 DO 50 J = 1, NIV
      J1 = J + 1
      DO 50 K = J1, NIVP1
      RM(J,K) = 0.D0
      DENOM = SDEV(J) * SDEV(K) * DFT
      IF (DENOM .NE. 0.D0) RM(J,K) = (SYX(J,K)-S(J)*S(K)/AN)/DENOM
   50 RM(K,J) = RM(J,K)
      RETURN
      END
```

```
      SUBROUTINE TVAL(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION   V(NMAX,KOLUMN)
      COMMON /C1/ N, NMAX, KOLUMN, NIV, M1(2), NP
      COMMON /C3/ M2, LINEAR, IA, M3(4), NYDEV
      COMMON /C4/ VV(72), COV(8,8)
      COMMON /C5/ AN, P(8), SE(8), TR(8), SIGLEV(8), BETA(8), VMSQ(8)
      COMMON /C6/ VMEAN(8), Z1(8), H(8,8), Z2(8), SDEV(8)
      COMMON /C7/ Z3(4), DW, WW(2), SEYSQ
      COMMON /C8/ Z4, DF1, Z5(2), SEY, Z6, YDEVSQ
      LOGICAL*4   LINEAR
      DATA        PI/3.14159 26535 89793/
C
C     SUBROUTINE FOR CALCULATING T-RELATED STATISTICS
C
      NSTART = 1 + IA
      IF (.NOT. LINEAR) GO TO 40
      DO 10 J = 2, NP
      BETA(J) = 0.D0
      IF (SDEV(1) .NE. 0.D0) BETA(J) = P(J) * SDEV(J) / SDEV(1)
   10 CONTINUE
C     CALCULATE VARIANCE-COVARIANCE MATRIX FOR LINEAR CASE.
      DO 12 I = 2, NP
      COV(1,I) = 0.D0
      IF (VMSQ(I) .GT. 0.D0) COV(1,I) = -SEYSQ * VMEAN(I) / VMSQ(I)
      COV(I,1) = COV(1,I)
      DO 12 J = 2, NP
   12 COV(I,J) = SEYSQ * H(I-1,J-1)
      IF (IA .EQ. 1) GO TO 60
C     CALCULATE THE STANDARD ERROR OF P(1) IF P(1) IS NOT SPECIFIED.
      SUM = 0.D0
      DO 30 I = 1, NIV
      SUM1 = 0.D0
      DO 20 J = 1, NIV
   20 SUM1 = SUM1 + VMEAN(J+1) * H(J,I)
      SUM1 = SUM1 * VMEAN(I+1)
      SUM = SUM + SUM1
   30 CONTINUE
      COV(1,1) = SEYSQ * (1.D0 / AN + SUM)
      GO TO 60
C     CALCULATE VARIANCE-COVARIANCE MATRIX FOR NONLINEAR CASE.
   40 DO 50 I = NSTART, NP
      DO 50 J = NSTART, NP
   50 COV(I,J) = SEYSQ * H(I,J)
   60 IDF1 = DF1
C     CALCULATE STANDARD ERRORS, T-RATIOS, AND SIGNIFICANCE LEVELS.
      DO 170 L = NSTART, NP
      SE(L) = 0.D0
      IF (COV(L,L) .GE. 0.D0) SE(L) = DSQRT(COV(L,L))
      TR(L) = 1.D50
      IF (SE(L) .GT. 0.D0) TR(L) = P(I) / SE(L)
      AA = DATAN2(TR(L), DSQRT(DF1))
      IF (IDF1 - 2) 70, 80, 90
   70 SIGLEV(L) = 2.D0 * AA / PI
      GO TO 160
   80 SIGLEV(L) = DSIN(AA)
      GO TO 160
   90 IF (IDF1 .GT. 3) GO TO 100
      SIGLEV(L) = 2.D0 * (AA + DSIN(AA) * DCOS(AA)) / PI
      GO TO 160
```

```
  100 Z = DCOS(AA)
      ZM = Z * Z
      IF (IDF1/2*2 .EQ. IDF1) GO TO 130
      I1 = (IDF1 - 3) / 2
      SUM = Z
      PROD = SUM
      DO 110 I = 1, I1
      AJ = 2 * I - 2
      PROD = PROD * (AJ + 2.D0) * ZM / (AJ + 3.D0)
      IF (DABS(PROD) .LT. 1.D-10) GO TO 120
      SUM = SUM + PROD
  110 CONTINUE
  120 SIGLEV(L) = 2.D0 * (AA + DSIN(AA) * SUM) / PI
      GO TO 160
  130 I1 = IDF1/2 - 1
      SUM = 1.D0
      PROD = SUM
      DO 140 I = 1, I1
      AJ = 2 * I - 2
      PROD = PROD * (AJ + 1.D0) * ZM / (AJ + 2.D0)
      IF (DABS(PROD) .LT. 1.D-10) GO TO 150
      SUM = SUM + PROD
  140 CONTINUE
  150 SIGLEV(L) = DSIN(AA) * SUM
  160 SIGLEV(L) = DABS(1.D0 - DABS(SIGLEV(L)))
  170 CONTINUE
C     CALCULATE THE DURBIN-WATSON STATISTIC.
      DW = 0.D0
      IF (YDEVSQ .EQ. 0.D0) RETURN
      DO 180 I = 2, N
      DIFF = V(I,NYDEV) - V(I-1,NYDEV)
  180 DW = DW + DIFF * DIFF
      DW = DW / YDEVSQ
      RETURN
      END
```

```
      SUBROUTINE OUT1
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /C1/ N, M1(2), NIV, NIVP1, M2, NP, LABEL
      COMMON /C2/ IEQ, M3(10), IFV
      COMMON /C3/ ABORT, LINEAR, IA, IGUESS, NXSET, NYCOL
      COMMON /C4/ Z1(72), COV(8,8), Z2(30), HEAD(9), LNH(8)
      COMMON /C5/ Z3, STATS(8,5)
      COMMON /C6/ VMEAN(8), Z4(72), B1(8), SDEV(8), RM(8,8)
      COMMON /C7/ Z5(2), RDEVM, EA, DW, XV, YV
      COMMON /C8/ DFT, DF1, DF2, CD, SEY, CV, YDEVSQ, FVALUE, SST
      LOGICAL*4   ABORT, LINEAR, IA, IGUESS, NIV1, NIV234, NIV567
      LOGICAL*4   SMALL, IEQ35
      DIMENSION   YXLABL(8), BLABEL(8), P(8), IBCD(8), LYX(8), IDG(4)
      DIMENSION   FV(2)
      EQUIVALENCE (P(1), STATS(1,1)), (YXLABL(1), HEAD(2))
      DATA        LYX/'Y','X1','X2','X3','X4','X5','X6','X7'/
      DATA        LN/'LN'/, CON,STANT/' (CON','STANT)'/, LBLANK/' '/
      DATA        IBCD/'A','B','C','D','E','F','G','H'/, BLANK/' '/
      DATA        SPE,CIFIED/'  SPE','CIFIED'/, LA/'A'/, DVP/0.1D0/
      DATA        LNLRSV/'(LN)'/, FV/' > 10**8','INFINITE'/
C
C     SUBROUTINE FOR PRINTING SUMMARY TABLE
C
      WRITE (6, 10)
   10 FORMAT (1H0, 49X, 'SUMMARY TABLE')
      IF (IEQ .GE. 6) WRITE (6, 20)
   20 FORMAT (1H0, 34X, 'NOTE -- STATISTICS ARE BASED ON LOGARITHMS')
      IEQ35 = IEQ .EQ. 3 .OR. IEQ .EQ. 5
      DO 30 J = 1, NIVP1
      BLABEL(J) = BLANK
      LNH(J) = LBLANK
      IF (LABEL .NE. 0) BLABEL(J) = YXLABL(J)
   30 CONTINUE
      IF (NYCOL .GT. 1) LNH(1) = LN
      IF (NXSET .EQ. 0) GO TO 50
      DO 40 J = 2, NP
   40 LNH(J) = LN
   50 IF (LINEAR) GO TO 80
      WRITE (6, 60)
   60 FORMAT (1H0, 52X, 'STANDARD', 23X, 'SIGNIF' / 1H , 'PARAMETER',
     1 14X, 'VALUE', 25X, 'ERROR', 10X, 'T-RATIO', 9X, 'LEVEL')
      IF (IEQ35) WRITE (6, 70)
   70 FORMAT (1H+, 34X, 'INITIAL GUESS')
      MAX = 4
      GO TO 100
   80 WRITE (6, 90)
   90 FORMAT (1H0, 52X, 'STANDARD', 23X, 'SIGNIF', 9X, 'BETA' /
     1 1H , 'PARAMETER', 14X, 'VALUE', 25X, 'ERROR', 10X, 'T-RATIO',
     2 9X, 'LEVEL', 9X, 'COEFF')
      MAX = 5
  100 WRITE (6, 110)
  110 FORMAT (1H )
      LL = LBLANK
      IF (IEQ .EQ. 6) LL = LN
      IF (.NOT. IA) GO TO 140
  120 WRITE (6, 130) LL, LA, SPE,CIFIED, P(1)
  130 FORMAT (1H , A3, A2, A5, A6, F14.5, 15X, 3F15.5)
      GO TO 190
  140 IF (IEQ .NE. 6) GO TO 160
      DEXPA = DEXP(P(1))
```

```
      SMALL = DEXPA .LT. DVP .OR. DEXPA .GT. 1.D6
      IF (   SMALL   ) WRITE (6, 150) BLANK, LA, BLANK, BLANK, DEXPA
      IF (.NOT. SMALL) WRITE (6, 130) BLANK, LA, BLANK, BLANK, DEXPA
  150 FORMAT (1H , A3, A2, A5, A6, D18.5, 11X, 3F15.5)
      IF (IA) GO TO 190
  160 SMALL = DABS(P(1)) .LT. DVP .OR. DABS(P(1)) .GT. 1.D6
      IF (.NOT. SMALL) WRITE (6, 130) LL,LA,CON,STANT,(STATS(1,K),K=1,4)
      IF (   SMALL   ) WRITE (6, 150) LL,LA,CON,STANT,(STATS(1,K),K=1,4)
      IF (.NOT.IEQ35) GO TO 190
      SMALL = DABS(P1(1)) .LT. DVP .OR. DABS(P1(1)) .GT. 1.D6
      IF (.NOT.SMALL) WRITE (6, 170) P1(1)
      IF (  SMALL   ) WRITE (6, 180) P1(1)
  170 FORMAT (1H+, T32, F15.5)
  180 FORMAT (1H+, T32, D19.5)
  190 DO 220 J = 2, NP
      SMALL = DABS(P(J)) .LT. DVP .OR. DABS(P(J)) .GT. 1.D6
      IF (   SMALL   ) WRITE (6, 200) LBCD(J), YXLABL(J),
     1 (STATS(J,K), K = 1, MAX)
      IF (.NOT. SMALL) WRITE (6, 210) LBCD(J), YXLABL(J),
     1 (STATS(J,K), K = 1, MAX)
  200 FORMAT (4X, A4, A8, D19.5, 11X, 4F15.5)
  210 FORMAT (4X, A4, A8, F15.5, 15X, 4F15.5)
      IF (.NOT.IEQ35) GO TO 220
      SMALL = DABS(P1(J)) .LT. DVP .OR. DABS(P1(J)) .GT. 1.D6
      IF (.NOT.SMALL) WRITE (6, 170) P1(J)
      IF (  SMALL   ) WRITE (6, 180) P1(J)
  220 CONTINUE
      WRITE (6, 230)
  230 FORMAT (1H0)
      NIV1 = NIV .EQ. 1
      NIV234 = NIV .LE. 4 .AND. .NOT. NIV1
      NIV567 = NIV .GE. 5
      IF ( NIV1 ) WRITE (6, 240)
  240 FORMAT (1H0, 37X, 'STANDARD' /1H , 'VARIABLE', 15X, 'MEAN', 9X,
     1 'DEVIATION')
      IF (NIV234) WRITE (6, 250) (LNH(J), J = 1, NIVP1)
      IF (NIV234) WRITE (6, 260) (YXLABL(J), J = 1, NIVP1)
  250 FORMAT (1H , 62X, 'CORRELATION MATRIX' /1H , 37X, 'STANDARD', 10X,
     1 5(A2,13X))
  260 FORMAT (1H , 'VARIABLE', 15X, 'MEAN', 9X, 'DEVIATION',
     1 5(7X,A8))
      IF (NIV567) WRITE (6, 270) (LNH(J), J = 1, NIVP1)
      IF (NIV567) WRITE (6, 280) (YXLABL(J), J = 1, NIVP1)
  270 FORMAT (1H , 73X, 'CORRELATION MATRIX' /1H , 37X, 'STANDARD', 2X,
     1 8(8X,A2))
  280 FORMAT (1H , 'VARIABLE', 15X, 'MEAN', 9X, 'DEVIATION',
     1 5X, 8(2X,A8))
      WRITE (6, 110)
      DO 320 J = 1, NIVP1
      WRITE (6, 290) LNH(J), LYX(J), ELABEL(J), VMEAN(J), SDEV(J)
      IF (NIV234) WRITE (6, 300) (RM(J,K), K = 1, NIVP1)
      IF (NIV567) WRITE (6, 310) (RM(J,K), K = 1, NIVP1)
  290 FORMAT (1H , A3, A4, A8, 2F15.5)
  300 FORMAT (1H+, 45X, 5F15.5)
  310 FORMAT (1H+, 50X, 8F10.5)
  320 CONTINUE
      LNLR = LBLANK
      IF (LNH(1) .EQ. LN) LNLR = LNLRSV
      WRITE (6, 230)
      WRITE (6, 330) CD, LNLR, RDEVM
  330 FORMAT (1X, 'COEFFICIENT OF DETERMINATION (UNADJ), R SQ', F13.5,
```

```
     1 5X, 'MEAN OF ABSOLUTE RELATIVE DEVIATIONS ', A4, F14.5)
       WRITE (6, 340) SEY, CV, LNLR, YDEVSQ, LNIR, SST
   340 FORMAT (1X, 'STANDARD ERROR OF ESTIMATE', F29.5,
     1 5X, 'COEFF VARIATION (STD ERR EST / MEAN Y OBS)', F13.5 /
     2 1X, 'SUM OF SQUARES OF RESIDUALS ', A4, F23.5,
     3 5X, 'SUM OF SQUARES TOTAL ', A4, F30.5)
       IF (IFV .EQ. 0) WRITE (6, 350) FVALUE
       IF (IFV .GT. 0) WRITE (6, 360) FV(IFV)
   350 FORMAT (1X, 'F VALUE', 34X, F14.5)
   360 FORMAT (1X, 'F VALUE', 40X, A8)
       WRITE (6, 370) DW
   370 FORMAT (1H+, 60X, 'DURBIN-WATSON STATISTIC', F32.5)
       IF (IEQ .EQ. 5) WRITE (6, 380) EA
   380 FORMAT (1X, 'Y INTERCEPT', F44.5)
       IF (IEQ .EQ. 2) WRITE (6, 390) XV, YV
   390 FORMAT (1X, 'X COORDINATE OF VERTEX', F33.5,
     1 5X, 'Y COORDINATE OF VERTEX', F33.5)
       IDG(1) = DF1
       IDG(2) = DF2
       IDG(3) = DFT
       IDG(4) = N
       WRITE (6, 400) IDG
   400 FORMAT (1X, 'DEGREES OF FREEDOM FOR ERROR', I27,
     1 5X, 'DEGREES OF FREEDOM DUE TO REGRESSION', I19 /
     2 1X, 'TOTAL DEGREES OF FREEDOM', I31,
     3 5X, 'NUMBER OF DATA POINTS', I34)
       IF (NP .EQ. 1) RETURN
C
C      PRINT VARIANCE-COVARIANCE MATRIX
C
       NSTART = 1
       IF (IA) NSTART = 2
       IF (NIV567) WRITE (6, 410)
   410 FORMAT (1H0/ T42, 'VARIANCE-COVARIANCE MATRIX' / )
       IF (.NOT. NIV567) WRITE (6, 420)
   420 FORMAT (1H0/ T12, 'VARIANCE-COVARIANCE MATRIX' / )
       IF (IEQ .EQ. 6) GO TO 460
       WRITE (6, 430) (LBCD(J), J = NSTART, NP)
   430 FORMAT (T3, 8(13X, A1) )
       DO 450 I = NSTART, NP
       WRITE (6, 440) LBCD(I), (COV(I,J), J = NSTART, NP)
   440 FORMAT (T6, A1, 2X, 8(2X, D12.5) )
   450 CONTINUE
       RETURN
   460 IF (IA) WRITE (6, 430) (LBCD(J), J = 2, NP)
       IF (.NOT. IA) WRITE (6, 470) (LBCD(J), J = 2, NP)
   470 FORMAT (T13, 'LN A', 7(13X, A1) )
       DO 490 I = NSTART, NP
       IF (I .EQ. 1) WRITE (6, 480) (COV(I,J), J = 1, NP)
   480 FORMAT (T3, 'LN A', 2X, 8(2X, D12.5) )
       IF (I .GT. 1) WRITE (6, 440) LBCD(I), (COV(I,J), J = NSTART, NP)
   490 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE OUT2(V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION   V(NMAX,KOLUMN)
      COMMON /C1/ N, NMAX, KOLUMN, NIV, NIVP1, NVAR, NP, LABEL, ID
      COMMON /C2/ IEQ, NP1, NP2, NP3, LZZYES, M1, LNOUT, M2(16), NPAGE
      COMMON /C3/ ABORT, LINEAR, IA, M3, NXSET, NYCOL, NYC, NYDEV
      COMMON /C4/ Z1(166), HEAD(9), LNH(8)
      COMMON /C5/ AN
      COMMON /C7/ Z2(2), RDEVM
      COMMON /CX/ EXV(8), NEXV, NVT
      DIMENSION   ALTER(2), HEADS(8), TRAN(8), VHD(6), LETR(35), LNX(8)
      LOGICAL*4   ABORT, NP1, NP2, LNOUT, PLOTS, ID, SKIP
      EQUIVALENCE (YLN, LNH(1)), (YP, HEAD(2))
      DATA        IBLANK/' '/, BLANK/' '/
      DATA        OBSERV/'OBSERVED'/, COMP/'COMPUTED'/, BBD/'  D'/,
     1            RESID/'RESIDUAL'/, REL/'RELATIVE'/, BD/' D'/,
     2            EVIATN/'EVIATION'/
      DATA LETR/'A','B','C','D','E','F','G','H','I','J','K','L','M','N',
     1          'O','P','Q','R','S','T','U','V','W','X','Y','Z','1','2',
     2          '3','4','5','6','7','8','9'/
      DATA        TRAN/' Y: ', 'X1: ', 'X2: ', 'X3: ', 'X4: ', 'X5: ',
     1                 'X6: ', 'X7: '/
      DATA        ALTER/ 'MODIFIED', 'NOT USED'/
      DATA        VHD/ '  (X2)', '  (X3)', '  (X4)', '  (X5)',
     1                 '  (X6)', '  (X7)'/
C
C     SUBROUTINE FOR PRINTING INPUT DATA, COMPUTED Y VALUES,
C     Y RESIDUALS, AND PERCENT Y DEVIATIONS
C
      NIVPS = NIVP1
      IF (NEXV .EQ. 0) GO TO 5
      NIV = NIV + NEXV
      NIVPP = NIVP1 + 1
      NIVP1 = NIV + 1
    5 ITEMP = NP + IA
      IF (NP .EQ. 1) ITEMP = 0
      LINES = NP + NIVP1 + N + N/5 + LINEAR + IA + IA + ITEMP
      IF (NPAGE .EQ. 2) LINES = LINES + 2
      IF (IEQ .GT. 6) LINES = LINES + 2
      IF (IEQ .EQ. 6) LINES = LINES + 3
      IF (IEQ .EQ. 2) LINES = LINES + 1
      IF (LABEL .EQ. 2) LINES = LINES + 2
      PLOTS = NP1 .OR. NP2 .OR. NP3 .GT. 0
      SKIP = NYCOL .EQ. 1 .OR. LNOUT
      DEVMAX = -1.D10
      DEVMIN =  1.D10
      RDEVM = 0.D0
      NREPS = (N-1)/35 + 1
      NYPLOT = NYCOL
      NXPLOT = NXSET
      NP3PL = NP3
      NP3EQ = 0
      IF (IEQ .EQ. 0) GO TO 260
      DO 10 I = 1, NIVP1
      HEADS(I) = BLANK
   10 LNX(I) = LNH(I)
      HEADS(1) = OBSERV
      IF (NXSET+NYCOL .EQ. 1 .OR. LNOUT) GO TO 30
      DO 20 I = 1, NIVP1
   20 LNX(I) = IBLANK
```

```
      NYCOL = 1
      NXSET = 0
   30 IF (NVT .EQ. 0) GO TO 36
      DO 32 I = 1, NIVP1
      IF (EXV(I) .NE. 0.D0) HEADS(I) = ALTER(1)
   32 CONTINUE
      IF (NEXV .EQ. 0) GO TO 36
      DO 34 I = NIVPP, NIVP1
      EXV(I) = 0.D0
      HEAD(I+1) = VHD(I-2)
      HEADS(I) = ALTER(2)
      LNX(I) = IBLANK
   34 CONTINUE
   36 IF (LINES .LE. 21) GO TO 40
C     PRINT TITLE ON NEW PAGE
      CALL TITLE2
   40 WRITE (6, 50)
   50 FORMAT (/1H0, 49X, 'TABLE OF RESIDUALS')
      LINES = 0
      NTIMES = 1
   60 IF (NIV .GT. 4) GO TO 90
      WRITE (6, 70) (HEADS(J), J = 1, NIVP1), COMP, RESID, REL
      WRITE (6, 80) HEAD(1), (LNX(J), HEAD(J+1), J=1,NIVP1),
     1 LNX(1), YP, LNX(1), YP, EED, EVIATN
   70 FORMAT (1H0, 10X, 8(7X, A8) )
   80 FORMAT (1H , 2X, A8, 8(4X,A3,A8) )
      GO TO 120
   90 WRITE (6, 100) (HEADS(J), J = 1, NIVP1), COMP, RESID, REL
      WRITE (6, 110) HEAD(1), (LNX(J), HEAD(J+1), J=1,NIVP1),
     1 LNX(1), YP, LNX(1), YP, ED, EVIATN
  100 FORMAT (1H0, 10X, 11(3X, A8) )
  110 FORMAT (1H , 2X, A8, 11(1X,A2,A8) )
  120 WRITE (6, 200)
      IF (NTIMES .EQ. 2) GO TO 140
      YZERO = 0.D0
      DO 210 I = 1, N
      IF (LINES .LT. 40) GO TO 140
      CALL TITLE2
      LINES = 0
      WRITE (6, 130)
  130 FORMAT (/1H0, 44X, 'TABLE OF RESIDUALS (CONTINUED)')
      NTIMES = 2
      GO TO 60
  140 LET = IBLANK
      IF (PLOTS) LET = LETR((I-1)/NREPS + 1)
      VID = BLANK
      IF (ID) VID = V(I,NVAR)
      YC = V(I,NYC)
      YDEV = V(I,NYDEV)
      IF (SKIP) GO TO 150
      YC = DEXP (V(I,NYC))
      YDEV = V(I,1) - YC
  150 IF (V(I,NYCOL) .NE. 0.D0) GO TO 160
      YZERO = YZERO + 1.D0
      RDEV = 0.D0
      GO TO 170
  160 RDEV = YDEV / V(I,NYCOL)
  170 RDEVM = RDEVM + DABS(RDEV)
      IF (DEVMAX .LT. RDEV) DEVMAX = RDEV
      IF (DEVMIN .GT. RDEV) DEVMIN = RDEV
      IF (NIV .LE. 4) WRITE (6, 180) LET, VID, V(I,NYCOL),
```

```
     1 (V(I,J+NXSET), J = 2, NIVP1), YC, YDEV, RDEV
       IF (NIV .GT. 4) WRITE (6, 190) LET, VID, V(I,NYCOL),
     1 (V(I,J+NXSET), J = 2, NIVP1), YC, YDEV, RDEV
  180 FORMAT (1H , A2, A8, 8F15.5)
  190 FORMAT (1H , A2, A8, 11F11.3)
      LINES = LINES + 1
      IF (I/5*5 .EQ. I) WRITE (6, 200)
  200 FORMAT (1H )
  210 CONTINUE
      RDEVM = RDEVM / (AN - YZERO)
      WRITE (6, 220) DEVMIN, RDEVM, DEVMAX
  220 FORMAT ('0MINIMUM RELATIVE DEVIATION =', F10.5, ',    ',
     1    'MEAN ABSOLUTE RELATIVE DEVIATION =', F9.5, ',    ',
     2           'MAXIMUM RELATIVE DEVIATION =', F10.5)
      IF (NVT .EQ. 0) GO TO 226
      IF (NIVPS .LE. 6) WRITE (6, 222) (TRAN(I),EXV(I),I=1,NIVPS)
      IF (NIVPS .GE. 7) WRITE (6, 224) (TRAN(I),EXV(I),I=1,NIVPS)
  222 FORMAT (/' TRANSFORMATION FACTORS -- ', 6(A4, F10.5, 2X) )
  224 FORMAT (/' TRANSFORMATION FACTORS -- ', 6(A4, F10.5, 2X) / T28,
     1          2(A4, F10.5, 2X) )
  226 IF (.NOT.PLOTS) RETURN
      IF (.NOT.NP1) GO TO 240
      CALL TITLE2
      WRITE (6, 230) RESID, YLN, YP, COMP, YLN, YP
  230 FORMAT(1H0, 43X, A8, 1X, A3, A8,'     VERSUS     ', A8, 1X, A3, A8/)
      CALL PLOTYX ( N , V(1,NYDEV) , V(1,NYC) , 0 , 0 )
  240 IF (.NOT.NP2) GO TO 250
      CALL TITLE2
      WRITE (6, 230) OBSERV, YLN, YP, COMP, YLN, YP
      CALL PLOTYX ( N , V(1,NYPLOT), V(1,NYC) , 0 , 0 )
  250 IF (NP3PL .EQ. 0)  RETURN
      IF (NP3PL .LE. NIV) GO TO 310
      IF (NP3PL .LT. 8) RETURN
      NP3PL = 1
      IF (NIV .EQ. 1) NP3EQ = 1
      GO TO 310
C     FOLLOWING SECTION FOR PLOT-ONLY OPTION
  260 IF (NP3 .EQ. 0 .OR. NP3 .GE. 8) NP3PL = 1
      IF (NP3PL .GT. NIV) RETURN
      LINES = 0
      WRITE (6, 270) HEAD(1), HEAD(2), HEAD(NP3PL+2)
  270 FORMAT (1H0, 17X, 'OBSERVED'/ 1H , 2X, A8, 7X, A8, 7X, A8 / )
      LNH(NP3PL+1) = IBLANK
      LNH(1) = IBLANK
      DO 300 I = 1, N
      IF (LINES .LT. 40) GO TO 280
      CALL TITLE2
      WRITE (6, 270) HEAD(1), HEAD(2), HEAD(NP3PL+2)
      LINES = 0
  280 LET = LETR((I-1)/NREPS + 1)
      VID = BLANK
      IF (ID) VID = V(I,NVAR)
      WRITE (6, 290) LET, VID, V(I,1), V(I,NP3PL+1)
  290 FORMAT (1H , A2, A8, 2F15.5)
      LINES = LINES + 1
      IF (I/5*5 .EQ. I) WRITE (6, 200)
  300 CONTINUE
  310 CALL TITLE2
      WRITE (6, 230) OBSERV,LNH(1),YP, OBSERV,LNH(NP3PL+1),HEAD(NP3PL+2)
      LZERO = LZZYES
      CALL PLOTYX ( N, V(1,NYPLOT), V(1,NP3PL+1+NXPLOT), NP3EQ, LZERO)
```

```
      RETURN                                                            OUT21830
      END                                                               OUT21840
```

```
      SUBROUTINE PLOTYX ( N, Y, X, NEQ, LZERO )
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(N), X(N)
      COMMON /C2/ IEQ
      COMMON /C3/ M1, LINEAR
      COMMON /C5/ Z1, A, B, C
      COMMON /C9/ DUMMY(1535), OUTPUT(15,45), L1, LX(3)
      LOGICAL*4   LINEAR, LZERO
      LOGICAL*1   COLROW(120,45), L1, LX, NUMBER, ISYMBL, LETR(35)
      EQUIVALENCE (COLROW(1), OUTPUT(1)), (L1, I4)
      DATA ILINE/'-'/, ISYMBL/'.'/
      DATA NUMBER/'#'/, IBLANK/' '/, BLANK/' '/, ZEROL/'--------'/
      DATA LETR/'A','B','C','D','E','F','G','H','I','J','K','L','M','N',
     1          'O','P','Q','R','S','T','U','V','W','X','Y','Z','1','2',
     2          '3','4','5','6','7','8','9'/
      NREPS = (N-1) / 35 + 1
      YMAX = Y(1)
      YMIN = Y(1)
      XMAX = X(1)
      XMIN = X(1)
      DO 10 I = 2, N
      IF (X(I) .LT. XMIN) XMIN = X(I)
      IF (Y(I) .LT. YMIN) YMIN = Y(I)
      IF (X(I) .GT. XMAX) XMAX = X(I)
      IF (Y(I) .GT. YMAX) YMAX = Y(I)
   10 CONTINUE
C     PLOT FROM ORIGIN IF SELECTED
      IF (LZERO) XMIN = 0.D0
      IF (LZERO) YMIN = 0.D0
C     FIND SCALE FACTORS FOR 45 LINES AND 120 SPACES
      XSCAL = (XMAX - XMIN) / 120.D0
      YSCAL = (YMAX - YMIN) / 45.D0
      WRITE (6, 20) YMAX
   20 FORMAT ('0MAX VERT=',F14.5/)
      WRITE (6, 30)
   30 FORMAT (1H+,10X,1H ,12('_________.'))
C     FIND RECIPROCAL OF SCALE FOR MULTIPLICATION
      RXSCAL = 1.D0 / XSCAL
      RYSCAL = 1.D0 / YSCAL
C     BLANK OUT PLOT PAGE WITH EQUIVALENT 8-BYTE BLOCKS
      DO 40 K = 1, 15
      DO 40 L = 1, 45
   40 OUTPUT(K,L) = BLANK
      I4 = IBLANK
C     DETERMINE WHERE ZERO LINE IS, IF ANY
      NZERO = 0
      IF (YMIN * YMAX .LT. 0) NZERO = YMAX * RYSCAL + 1
      IF (NZERO .EQ. 0) GO TO 60
      DO 50 K = 1, 15
   50 OUTPUT(K,NZERO) = ZEROL
   60 DO 70 I = 1, N
C     DETERMINE HORIZONTAL AND VERTICAL CHARACTER POSITION
      K = (X(I)-XMIN) * RXSCAL + 1
      L = 45 - (Y(I)-YMIN) * RYSCAL
C     ACCEPT RIGHT-HAND PLOT BOUNDRY AS WITHIN LAST CHARACTER
C     POSITION AND UPPER BOUNDRY AS WITHIN FIRST LINE
      IF (K .GT. 120) K = 120
      IF (L .LT. 1) L = 1
C     MOVE PLOT CHARACTER TO BEGINNING OF INTEGER TEST WORD
      L1 = COLROW(K,L)
```

```
C     FOR OVERLAPPING POINTS, PLOT *
      IF (I4 .NE. IBLANK .AND. I4 .NE. ILINE) COLROW(K,L) = NUMBER
C     PLOT SINGLE PLOT POINT AS ALPHANUMERIC
      IF (I4 .EQ. IBLANK .OR. I4 .EQ. ILINE) COLROW(K,L) = LETR((I-1)/
     1 NREPS + 1)
   70 CONTINUE
      IF (NEQ .NE. 1) GO TO 100
      XX = XMIN - 0.5D0  *  XSCAL
      DO 90 K = 1, 120
      XX = XX + XSCAL
      YY = A + B * XX
      IF (LINEAR) GO TO 80
      IF (IEQ .EQ. 2) YY = YY + C * XX * XX
      IF (IEQ .EQ. 3) YY = A * XX**B
      IF (IEQ .EQ. 4) YY = A + B * XX**C
      IF (IEQ .EQ. 5) YY = DEXP(YY)
   80 L = 45 - (YY - YMIN) * RYSCAL
      IF (L .LT. 1 .OR. L .GT. 45) GO TO 90
      I4 = COLROW(K,L)
      IF (I4 .EQ. IBLANK .OR. I4 .EQ. ILINE) COLROW(K,L) = ISYMBL
   90 CONTINUE
  100 WRITE (6, 110) OUTPUT
  110 FORMAT (1H , 10X, '|', 15A8 , '|')
      WRITE (6, 30)
      WRITE (6, 120) YMIN, XMIN, XMAX, YSCAL, XSCAL
  120 FORMAT ('0MIN VERT=',F14.5/' MIN HORZ=',F14.5,86X,'MAX HORZ='
     1 F14.5/'0VERT INCREMENT=',F12.5/' HORZ INCREMENT=',F12.5)
      RETURN
      END
```

```
********************** A S S E M B L E R   R O U T I N E ****************
READMEMO START 0
         ENTRY MEMRE
         ENTRY DATER
         EXTRN IBCOM#
         EXTRN FIOCS#
*                                                                      *
*              FIRST ENTRY POINT.  THIS ROUTINE PICKS UP OPERANDS ONE *
*              AND TWO AND STORES THEM AT THE TWO FULL WORDS AT BUFLOC*
*                                                                      *
         USING *,15
MEMRE    B     *+12                BRANCH AROUND NAME
         DC    XL4'07000000'       ROUTINE NAME FOR CALL TRACE
         DC    CL4'CORE'           ROUTINE NAME FOR CALL TRACE
         STM   14,3,12(13)         SAVE REGISTERS
         LM    2,3,0(1)            FETCH OPERAND ADDRESSES
         L     3,0(3)              FETCH OPERAND 2 (LENGTH)
         STM   2,3,BUFADR          STORE BUFFER LOC AND LENGTH
         LA    1,CORE2             R1=A(SECOND ENTRY POINT)
         LA    3,CLOAD             SET BASE REGISTER FOR CLOAD
         BALR  2,3                 LINK TO MODIFY IBCOM ADCON
         LM    14,3,12(13)         RESTORE REGISTERS
         SR    15,15               SUPPRESS VARIABLE RETURN
         BR    14                  RETURN
         DROP  15
*              SECOND ENTRY POINT. IBCOM ENTERS AT CORE2 THINKING IT  *
*              WENT TO FIOCS. THIS ROUTINE SIMULATES FIOCS BY POINTING*
*              TO BUFFER ADDRESS AND LENGTH STORED BY FIRST ROUTINE.  *
*              IBCOM IS RESTORED TO NORMAL, FOLLOWED BY RETURN TO     *
*              IBCOM.  A WRITE BUFFER IS INITIALIZED TO BLANKS BEFORE *
*              IBCOM FILLS IT TO ALLOW T FORMAT TO WORK CORRECTLY.    *
*                                                                      *
         USING *,1
CORE2    ST    4,SAVE4             SAVE R4.
         LR    4,1                 R4=A(CORE2)
         USING CORE2,4
         DROP  1
         LR    1,0                 R1 POINTS TO FIOCS CALL PARAMETERS
         TM    1(1),X'0F'          TEST FOR OUTPUT, FIRST TIME
         BO    OUTPUT              BRANCH TO FIRST OUTPUT ROUTINE
         L     1,VFIOCS            R1=A(FIOCS) TO RESTORE IBCOM
         LA    3,CLOAD             SET BASE REGISTER FOR CLOAD
         BALR  2,3                 LINK TO MODIFY IBCOM ADCON
         LM    2,3,BUFADR          LOAD ARRAY ADR AND LENGTH
         B     RETURN              BRANCH TO RETURN TO IBCOM
OUTPUT   LM    2,3,BUFADR          LOAD ARRAY ADR AND LENGTH
         MVI   0(2),X'40'          BLANK FIRST BUFFER LOCATION
         BCTR  3,0                 -1 L-1 CHAR TO BE BLANKED
         BCTR  3,0                 -1 LENGTH CODE FOR MOVE=LENGTH-1
         EX    3,DMOVE             EXECUTE DUMMY MOVE TO CLEAR BUFFER
         LA    3,2(3)              R3=R3+2 RESTORE ORIGINAL LENGTH
RETURN   L     4,SAVE4             RESTORE R4
         LR    1,0                 R1=A(IBCOM ARGUMENTS)
         DROP  4
         B     6(1)                RETURN TO IBCOM
DMOVE    MVC   1(0,2),0(2)         EXECUTED. CLEARS UP TO 257 BYTE BUFFER
*              R1= AN ADDRESS THE CALLER WANTS STORED AT VFIOCS IN    *
*              R15 MUST BE A(IBCOM) TO SATISFY BASE REG  REQMNTS IN   *
*              IBCOM. CALLER LOADS R3=A(CLOAD) FOR ME                 *
*                                                                      *
```

```
         USING *,3
CLOAD    ST    15,SAVE          SAVE R15
         L     15,VIBCOM        R15=A(IBCOM) FOR IBCOM BASE REG
         MVI   74(15),X'50'     MAKE LOAD  A STORE INSTRUCTION
         EX    0,74(15)         STORES R1 AT VFIOCS IN IBCOM
         MVI   74(15),X'58'     RESTORE LOAD INSTRUCTION TO STORE
         L     15,SAVE          RESTORE R15
         BR    2                RETURN
BUFADR   DS    2F               STORAGE FOR A(BUFFER) AND ITS LENGTH
SAVE     DS    F                STORAGE FOR R15
SAVE4    DS    F                STORAGE FOR R4
VIBCOM   DC    A(IBCOM#)        A(L 1, VIBCOM INSTN IN IBCOM-74)
VFIOCS   DC    A(FIOCS#)        ADDRESS OF FIOCS ROUTINE
*                                                                      *
*              DATER ENTRY POINT.  THIS ROUTINE LOADS THE DATE AND     *
*              TIME INTO TWO REAL*4 VARIABLES, WHERE FORTRAN CALL IS   *
*                        CALL DATER(DATE)                              *
*              AND DATE IS DIMENSIONED BY TWO.                         *
*              DATE AND TIME ARE IN 'Z' FORMAT OF FORM 00YYDDDF AND    *
*              HHMMSSTH, RESPECTIVELY, AFTER 'TIME DEC' CALL           *
*                                                                      *
DATER    STM   14,3,12(13)      SAVE REGISTERS
         BALR  12,0             SET UP BASE REGISTERS
         USING *,12
         ST    13,SAVEAREA+4    LINK SAVE AREAS
         LA    13,SAVEAREA
         L     R3,0(,R1)        GET ADDR OF DATE AND TIME FROM CALL PGM
         TIME  DEC              GET DATE AND TIME IN PACKED DECIMAL
         SRL   1,4(0)           SHIFT RT. TO FORM 00YYDDD
         SRL   0,16(0)          SHIFT RT. TO FORM 0000HHMM
         ST    R1,0(,R3)        STORE DATE IN FIRST REAL*4 WORD
         ST    R0,4(,R3)        STORE TIME IN SECOND REAL*4 WORD
         L     13,SAVEAREA+4    GET CALLING PGM SAVE AREA
         LM    14,3,12(13)      RESTORE CALLING REGISTERS
         BR    14               RETURN
SAVEAREA DS    18F              REGISTER SAVE AREA FOR TIME
R0       EQU   0
R1       EQU   1
R3       EQU   3
         END
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: R-1753-1-PR

2. GOVT ACCESSION NO.:

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): CURVES: A Cost Analysis Curve-Fitting Program

5. TYPE OF REPORT & PERIOD COVERED: Interim rept.

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): H. E. Boren, Jr. G. W. Corwin

8. CONTRACT OR GRANT NUMBER(s): F44620-73-C-0011

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
The Rand Corporation
1700 Main Street
Santa Monica, Ca. 90406

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

11. CONTROLLING OFFICE NAME AND ADDRESS:
FCRC Office (AF/RDXTR)
Director of Planning, Programming & Analysis
Hw USAF, Washington, D.C. 20330

12. REPORT DATE: Sept[redacted]1976

13. NUMBER OF PAGES: 107

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

No restrictions

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
Cost Analysis
Computer Programs
Curve Fitting
Regression Analysis
Statistical Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

see reverse side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Describes a FORTRAN-IV curve-fitting computer program (CURVES) that makes least-squares determinations of the parameters of any of eight types of equations selected by the user, given a set of observations on the dependent and independent variables of interest. The types of equations that can be fitted are: linear, quadratic, power, asymptotic-power, exponential, logarithmic-linear, and two types of semilogarithmic-linear. Except for the quadratic and asymptotic-power equations, up to seven independent variables may be used. Y-intercepts may be specified for all equations except the power and exponential. Various types of variable transformations are allowed. A correlation matrix of the input data is provided for all fitted equations using more than one independent variable. Also included are standard errors and Student's t-ratios of the parameters, significance levels, beta coefficients, the Durbin-Watson statictic, and the variance-covariance matrix of the parameters. A plot routine is also incorporated. The program is fairly small (about 92,000 bytes of core when H compiled), fast in execution time, and hence cheap to operate. (Author)