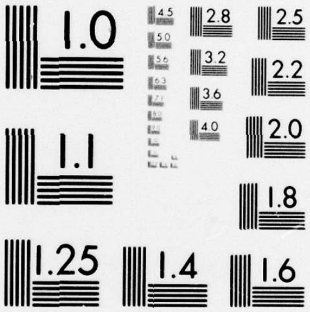
AD-A080 215 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/4
COMPUTER ASSISTED ANALYSIS FOR MILITARY MANAGERS. (U)
DEC 79 R D CONTE
UNCLASSIFIED AFIT/GOR/MA/79D-3 NL

1 OF 3
AD
A080215


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

AF INSTITUTE OF TECHNOLOGY

UNITED STATES AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/MA/79D-3

COMPUTER ASSISTED ANALYSIS
FOR
MILITARY MANAGERS

THESIS

AFIT/GOR/MA/79D-3

Robert D. Conte
Captain USA

Approved for public release; distribution unlimited

14 AFIT/GOR/MA/79D-3

6
COMPUTER ASSISTED ANALYSIS
FOR
MILITARY MANAGERS.

9 Master's THESIS,

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

12 220

by

10 Robert D. Conte B.S.
Captain USA

Graduate Operations Research

11 Dec[redacted]79

Approved for public release; distribution unlimited.

012225

Preface

This study was undertaken to provide user-oriented quantitative analysis software for managers and analysts using desk-top microcomputers. I sincerely hope that users will find the software helpful. I plan to maintain and expand the package and would appreciate any recommendations and suggestions. Any user who is interested in acquiring the software on diskette should contact me at my next assignment:

U.S. Army Concepts and Analysis Agency
ATTN: CPT Robert D. Conte
8120 Woodmont Avenue
Bethesda, Maryland 20014

Autovon 295-1605

I wish to thank CPT Roie Black and COL Bob Margenthaler for their thesis suggestions and guidance and Bill and Molly Bustard for their assistance throughout the AFIT program. And finally I express my boundless appreciation and gratitude for the understanding devotion of my wife, Maureen, and the loving assistance of my son, David, age three.

TABLE OF CONTENTS

List of Tables

ABSTRACT

With the increasing complexity of modern operational and logistical problems, military leaders increasingly demand informed decision making at all levels of management. Such decision making must be fully supported by timely and accurate quantitative analysis. The digital computer is ideal for such analysis, but large mainframe computers are not flexible and responsive enough for managers who must be prepared to make quantitative decisions in the field or operations center as well as the computer room.

The advent of the microprocessor is revolutionizing computer technology. Microcomputers are low-in-cost, portable, independent, responsive, and easy to use. They can provide much of the computer support managers and analysts need for time-sensitive problems, but they currently lack adequate quantitative software.

This study was undertaken to provide user-oriented analysis software that exploits the advantages of desk-top computers. Of the many useful quantitative techniques available, Regression Analysis, Linear Programming, and a Value Matrix Decision Aid were selected and implemented.

COMPUTER ASSISTED ANALYSIS
FOR
MILITARY MANAGERS

I INTRODUCTION

Command and management of military men and materiel becomes increasingly costly and complex year after year. Decision making is the essence of command and management, but historically popular "back-of-the-envelope" or "seat-of-the-pants" decision making is no longer sufficient for solving modern military problems.

Congress and the Department of Defense increasingly demand informed decision making at all levels of management and for all types of operations. But regardless the level of management or type of operation, informed decision making requires timely and accurate analysis that frequently must be quantitatively-based. Whether the analysis concerns the number of Soviet tanks opposite the Fulda Gap, the requirements of an armored division for fuel and ammunition, or the contingency for emergency rescue of Americans in the Middle East, timeliness and accuracy can translate to lives saved, missions accomplished, and dollars well-spent.

Managers must be prepared to make quantitatively-supported decisions at any time and any place, be it their office, a briefing room, or in the field. The timeliness and accuracy required for modern complex problems increasingly demand the use of digital computer support and the services are expected to pay about $49 billion for computer software alone during the next five years (Ref 5). Computer support must be reliable, responsive, and easily accessible. It should offer quick access to stored information, independence from special support, and flexibility of operation. Recent strides in computer technology have begun providing solutions to these needs.

One of the most impressive electronic developments of recent years is the microprocessor. The microprocessor is revolutionizing the fundamental concepts of data processing and is beginning to impact decision making and quantitative analysis. No longer must managers be hampered by the poor response time, inaccessibility, and inflexibility of centralized computer mainframe systems. Managers and their analysts can now use "desk-top" or "micro-" computer systems wherever and whenever they need computer support. They can quickly formulate and analyze problems, compare options, evaluate results, and then pose "what if" questions during a single interactive computer session--without concern for special data communication links or dedicated operators. Desk-top computers cannot replace current mainframe systems but can provide powerful augmentation--a modern desk-top computer the size of a briefcase has the capability of a room-sized computer of ten years ago and costs far less. As a result, distributed or non-centralized data processing is finally becoming reality.

But while engineers have provided the technology, programmers must furnish software that accentuates the advantages of desk-top computers. Quantitative analysis software requirements are broad and include not only a variety of traditional methods such as regression analysis, linear programming, forecasting, and statistics, but also include more recent forms of decision analysis and utility theory.

Problem and Scope

The problem addressed in this study was the development of a user-oriented, quantitative analysis software package that both exploits and demonstrates the advantages of desk-top computer systems. Full consideration was given to user needs for analysis techniques and software design, and to the restrictions imposed by limitations of desk-top computers. Of the wide variety of quantitative techniques currently used, regression analysis, linear programming, and a value matrix decision aid were selected for implementation because of their relative frequency of use and their particular utility for desk-top problem solving.

Sequence of Presentation

Chapter II provides background for the problem, discussing the present computer hardware and software environment. Chapter III further develops the problem and outlines the design considerations and actual software development, then Chapter IV presents the final conclusions and recommendations.

Finally come the Appendices, which include the detailed user instructions and program information. The Appendices are designed to provide independent user and programmer documentation that can be removed from the basic thesis.

II SYSTEMS STATE-OF-THE-ART

Computer State-of-the-Art

Large computer "mainframe" systems such as the CDC 6600 are unsurpassed in speed of execution, precision, memory storage, and language power. Such systems suffer many drawbacks, however, including high procurement and acquisition costs, the need for dedicated operators, fixed site operation, and generally inconvenient access. Because mainframes generally service many users simultaneously, response time for even simple problems can range from hours to days. Remote terminals allow more convenient access and better response, but terminals are still dependent upon the mainframe for data processing and storage. If the mainframe is "down" for maintenance or the operators are off duty, all its terminals are useless.

Desk-top computers, on the other hand, are small and light enough to carry by hand, are relatively low in cost, and have their own memory and data processors for totally independent operation. Because they serve as dedicated systems, there is no response lag due to multiple users. Most have video output displays (with printers optional) and allow quick-access, mass storage of data on low-cost disk storage devices. Memory is generally limited to a maximum of 64,000 bytes (approximately 64,000 characters), but this is sufficient for most moderate-sized problems. The most commonly used desk-top computer language is BASIC, but FORTRAN and PASCAL compilers are becoming available.

Military managers face problems of such variety that they must optimize the use of both mainframe and desk-top computer systems. Mainframes are ideal for extremely large and complex problems requiring great memory capacity, long running time, or the power of exotic programming languages. However, they restrict the manager by location, response time, and convenience--restrictions that are only partially alleviated by the use of remote terminals. Current desk-top computers are restricted by language selection, memory capacity, and execution speed, but they are unexcelled in convenience, access, responsiveness, and flexibility. Desk-top computers therefore are ideal if the manager needs real-time, "hip-pocket" quantitative analysis for problems of moderate size and complexity. Mainframe and desk-top systems should complement, not displace each other.

Software State-of-the-Art

Mainframe computers have been in use for many years and quantitative analysis software for them is abundant. Existing software is designed for batch processing, however, and requires the use of tediously punched or typed data files which must be entered before running the program. Current interactive software designed for printer or video mainframe terminals is somewhat more flexible but still generally requires tedious unformatted, line-oriented data input, manipulation, and output. Such software is not optimal for managers and analysts who are time-conscious and not "computer oriented."

The problem-solving core algorithms implemented in existing analysis software have generally been validated and proven useful to managers and analysts. These algorithms could be used to create far more flexible, user-oriented software if designed for and implemented on desk-top computers.

Unfortunately, though, a military literature search conducted through the Defense Documentation Center and the Air Force Institute of Technology during February-March 1979 revealed no non-proprietary analysis software designed specifically for desk-top computers. Some commercial software does exist for higher-priced "minicomputers" like the IBM 5100, but it is extremely costly and generally available only in progıamming languages such as APL and COBOL, which are not currently available for desk-top computers. Therefore independent software development is necessary.

Desk-Top Computer Capabilities

The special capabilities of most desk-top computers can enhance analysis and problem solving if properly exercised. The video output device used by most desk-top computers allows virtually instantaneous display of program data, rapid program transitions, complete screen control for logically formatting input/output, and can allow generation of special graphics. Furthermore, a mini-disk storage system the size of a shoe box can maintain the equivalent of 30-50 pages of typed, singlespaced text on a single diskette only five inches in diameter, allowing access to any data within seconds.

With software that exploits these capabilities, desk-top computers can allow the analyst to enter and edit data in logical, "paper and pencil" format, to quickly scan and skip unnecessary parts of the program being used, to "page through" output at a comfortable pace, and to support appropriate mathematical analysis with graphical depictions of results. Additionally, the small disk drives and individual diskettes can provide "personalized" storage of data to allow an analyst to physically segregate the files of different databases, projects, or individual problems.

A review of quantitative analysis techniques indicates that many analysis tools could be greatly enhanced if implemented on desk-top computers. It is the purpose of this study to verify this by development of an actual analysis software package for desk-top computers.

III CAAMM SOFTWARE PACKAGE DEVELOPMENT

User Needs

The modern military manager and analyst need computer support that is easy to use and provides both responsiveness and flexibility. Desk-top computers can satisfy the need if controlled by proper system software.

In order to provide ease of use, responsiveness, and flexibility, system software should be fully interactive to guide the user step by step through the complete process of problem formulation and solution. System software should specifically provide logically-formatted input for quick problem formulation, built-in editing for easy reformulation and error correction, built-in model display functions, full disk interaction for permanent storage of data, and output that is formatted in screen-size blocks for comfortable viewing. The user should be able to initialize the system, formulate a problem, display the model or data, edit or reformulate as necessary, solve the problem, and then store the model or data for later retrieval--all under program control.

The user should never have to list, add, or change actual program lines in order to solve problems, but the programmer should be able to quickly decipher program logic in order to make modifications and additions. Consequently the programming should be "invisible" for the user, should be modular for the programmer, and should stress structural commonality for both.

Once general user needs are determined, specific analysis techniques must be selected for implementation. Analysts currently use a great many different analysis tools including regression analysis, linear programming, forecasting, statistics, program and network evaluation, economic analysis, inventory control, queuing, and decision analysis. To constrain this study effort, only regression analysis, linear programming, and a value matrix decision aid were selected for implementation. Regression analysis and linear programming are currently two of the most often used analysis tools, and various forms of decision analysis are rapidly becoming equally as popular. Regression analysis is used to provide order and meaning to historical or gathered data, linear programming is used to optimize the allocation of limited resources, and the value matrix decision aid is used to quickly structure and evaluate highly subjective decision situations.

Programming Requirements

Regression analysis software should provide bivariate and multivariate models, both linear and non-linear. It should allow user-defined transformation models for ultimate flexibility, and should provide all the common, useful descriptive measures such as means and standard deviations, correlation coefficients, R^2, residuals, standard error, t-statistic, F-statistic, confidence limits, predicted values, and graphical curve and data plotting.

Linear programming software should provide for the naming of decision variables and constraints, the automatic handling of constraint ordering and non-negativity requirements, and should handle integer as well as real number allocation problems. It should also provide sensitivity analysis and the capability for complete model reformulation.

Decision software should permit models with all three major decision parameter--alternatives, judgement criteria, and risk. It should be especially conducive to the iterative formulation of subjective problems, and it should provide sensitivity analysis to demonstrate the effect of changing parameters.

Software Implementation

The Computer Assisted Analysis for Military Managers (CAAMM) software package was developed to satisfy these needs and requirements. While regression analysis, linear programming, and the decision aid represent only three of the many useful analysis tools, the basic programming methods for data input, output, editing, and display can be modified for use with virtually any other analysis algorithm such as forecasting, network analysis, or multivariate statistics.

The Apple II personal computer was selected for software development because of its combination of high quality special graphics, expandability, reliability, availability of peripheral equipment, and relatively low cost. Precise system configuration included the Apple II with 48,000 bytes of

Random Access Memory, one mini-disk storage system, and BASIC language in Read Only Memory. BASIC was selected because it is currently the most widely used desk-top computer programming language. The Apple video display provides a 40-column, 24-line screen format. Printer output options are provided in the programs, but a printer is not required.

The programs are somewhat system dependent, but Apple "peculiar" commands and procedures are explained in the Programmer's Guide. The programs should support translation for any system with 32,000 bytes of free memory, a disk drive, a video display screen, and floating point BASIC.

The limitations of BASIC as an interpreted programming language and the restrictions of free memory capacity forced many conscious tradeoffs during program development. "Foolproofing" of programs was generally limited to the immediate checking of inputs for range and mode errors, while many commonly used but less significant statistical measures and other features were initially omitted with hopes of addition later.

The target user of this package is the individual manager, analyst, or operations research/management science student. The user is expected to be familiar with basic quantitative techniques but is fully prompted for all program inputs.

Software Package Description

The CAAMM software package currently implements the techniques of regression analysis, linear programming, and the value matrix decision aid with a set of seven total programs. With the exception of the CAAMM MASTER program, which runs automatically when the system is intialized, all programs are normally "invisible" to the user. The CAAMM MASTER program exercises overall package control. Analysis techniques are selected directly from the CAAMM MASTER menu and return control to CAAMM MASTER upon termination.

All programs are fully interactive, i.e. all inputs are prompted from the video screen and entered under program control. Each program has its own data entry, disk storage/ retrieval, and editing functions. All data input is matrix oriented and uses the flexible nature of video screen control for logical format and "paper and pencil" style entry. All output is screen page oriented to allow perusal of data until the user is ready to continue. Multiple databases can be entered, edited, saved to disk, and retrieved for later use. While specific program routines are necessarily different, the basic construction, logic, style, and major features of all programs are parallel. The current programs of the CAAMM software package are outlined in Table I.

CAAMM MASTER is the control program which automatically runs when the system is initialized. It simply produces a selection menu for the three basic analysis techniques--regression analysis, linear programming, and the value matrix decision aid.

Regression Analysis is initialized by program REGR DATA. REGR DATA provides the data entry, storage, retrieval, and editing routines necessary for both the BIVAR REGR and MULVAR REGR programs. A menu at the end of the program allows selection of one of these programs for either bivariate regression or multivariate regression.

TABLE I

CAAMM Software Package

Technique or Purpose	Program
Overall Package Control	CAAMM MASTER
Regression Data Management	REGR DATA
Bivariate Regression	BIVAR REGR
Bivariate Data/Curve Fit Plot	BIVAR HIRES
Multivariate Regression	MULVAR REGR
Linear Programming	LINPROG
Value Matrix Decision Aid	DECISION

BIVAR REGR provides regression analysis for any two variables in a database entered or retrieved by REGR DATA, using the method of least-squares curve fitting. The program allows selection from among eight standard bivariate regression models, a user-defined model with transformations, and an automatic curve fit for seven of the standard models. It predicts model coefficients, calculates basic statistical measures, produces a table of residuals, and allows interpolation and extrapolation. It additionally offers a high resolution graphics portrayal of the raw data and regression line, implemented by the separate subprogram BIVAR HIRES. Useful features not currently available in BIVAR REGR include the t-statistic, the F-statistic, confidence limits on estimates, and a preliminary scattergram to aid model selection.

MULVAR REGR provides regression analysis for any number of variables in a database entered or retrieved by REGR DATA, and also uses the method of least-squares curve fitting. The program permits either a strict linear model or a user-defined, multi-transformation, non-linear model. It predicts model coefficients, calculates basic statistical measures, and produces a table of residuals. Useful features not currently available in MULVAR REGR include the F-statistic, confidence limits on estimates, and elasticity coefficients.

The linear programming technique for optimizing allocation of limited resources is provided by program LINPROG. It has various options for display of tableaux and intermediate problem solutions. It solves both maximization and minimization problems, but has not been tested for integer or mixed integer programming. The only significant feature not currently available in LINPROG is true parametric, sensitivity analysis, although the edit functions provide the same ultimate capability by allowing easy refinement of the model.

The value matrix decision aid is implemented by program DECISION. DECISION provides a structured approach to comparing the relative merit of various decision options, given different judgement criteria and uncertain states of nature. It is particularly well-suited to help the decision maker iteratively formulate, solve and refine models with highly subjective, but quantifiable parameters. The program permits maximization or minimization of value and subsequent sensitivity analysis.

Documentation Note

Because the User's Guide and Programmer's Guide are designed to provide independent user and programmer documentation, detailed information on the programs, their use, and their structure is deferred to the Appendices.

IV CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The analysis techniques implemented in the CAAMM software package are complementary and mutually supportive for well-rounded problem analysis and decision making. However, users are cautioned against trying to "force fit" problems in order to use a particular technique without considering other approaches. The CAAMM package is not exhaustive. It does not include every technique of value to the analyst, and the analyst should not neglect other approaches simply because they are inconvenient. Often the best methodology is to take several independent approaches to a problem. Similar results of analysis generate greater confidence; dissimilar results probably indicate that the problem has not been properly formulated. Sometimes a combination of techniques may be appropriate for analyzing various aspects of the problem. Classical "textbook" problems are rarely found outside academia.

The current desk-top computer limitations of memory size, processing speed, and the BASIC programming language are significant, but transient. Recent advances in mass data storage devices will soon exponentially increase both computer memory and storage capacity. Processing speed already increases with the debut of every new desk-top system. Furthermore, sophisticated programming languages such as FORTRAN and

PASCAL are already available for some desk-top computers, with other languages forthcoming. Desk-top computers may never totally replace mainframe systems, but their low cost, portability, flexibility, and other features increasingly promote their widespread use for time-sensitive decision making and analysis.

Finally, the CAAMM software package demonstrates that desk-top computers can provide the responsive, flexible, and easy-to-use computer support that military commanders and managers must have for timely and accurate problem analysis and decision making.

Recommendations

Despite current limitations, desk-top computers in general and the CAAMM package specifically can be highly useful analysis tools. But extensions and improvements can be made to further enhance their utility.

Of the multitude of hardware devices currently available for desk-top computers, there are several that would be of particular value to analysts. X-Y plotters can provide excellent, low-cost graphics "hard copy"; light sensitive devices that enable the user to enter or read data by touching the video screen would greatly speed program option selection and interactive data manipulation; and a standard 80-column video display would allow far more comprehensive screen output than the Apple computer provides.

CAAMM software would also be greatly enhanced by the addition of more extensive error recovery subroutines that would preclude the uncontrolled termination of programs due to software or user errors. Furthermore, since the major limitations on all desk-top computers are memory space and processing speed, the use of compiled, high-level programming languages and improved memory and data storage techniques would significantly increase the capability to handle complex problems. Finally, there are a great many useful analysis techniques that should be but have not yet been adequately implemented for desk-top computers. Such techniques include forecasting, network analysis, goal programming, queuing, inventory, cost analysis, multivariate statistics, and various types of simulation.

The CAAMM package should be a useful tool for analysts, but it is not a "cure all". With the additions and improvements suggested above--plus the software maturity gained only by countless hours of handling and modification--the CAAMM package could become the completely integrated analysis software package that military managers require for modern problem analysis.

Bibliography

1. Afifi, A. A. and S. P. Azen. Statistical Analysis a Computer Oriented Approach. New York: Academic Press, 1972.

2. Albanese, Robert. Managing: Toward Accountability for Performance (Revised Edition). Homewood, Illinois: Richard D. Irwin, Inc., 1978.

3. Apple Computer Inc. Applesoft II Basic Programming Reference Manual. Cupertino, California: Apple Computer Inc., 1978.

4. Cole, Phyllis and Brian Howard. A Manual for Using the Apple Disk II with DOS Version 3.2. Cupertino, California: Apple Computer Inc., 1979.

5. "Computer Software," Army Times, 11: 50 (October 22, 1979).

6. Garson, Dane B. Computer consultant, The Computer Stop Stop (personal correspondence). Lawndale, California, January 9, 1979.

7. Honeywell. Series 600/6000 Time-Sharing Applications Library Guide Volume II - Statistics. Wellesley Hills, Massachusetts: Honeywell Information Systems, 1971.

8. Honeywell. Series 600/6000 Time-Sharing Applications Library Guide Volume III - Management Science. Massachusetts: Honeywell Information Systems, 1972.

9. Kirschner, Frank D. Using the Apple DOS to Modify Programs Under Program Control, unpublished article. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, 1979.

10. Levin, Richard I. and Charles A. Kirkpatrick. Quantitative Approaches to Management (Third Edition). New York: Mc Graw-Hill Book Company, 1975.

11. Margenthaler, Charles R. Personal instructional material. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, 1979.

12. Mason, Robert D. Statistical Techniques in Business and Economics. Homewood, Illinois: Richard D. Irwin, Inc., 1967.

13. McNichols, Charles W. An Introduction to: Applied Multivariate Data Analysis. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, 1979.

14. Morris, William T. Decision Analysis. Columbus, Ohio: Grid, Inc., 1977.

15. Poole, Lon and Mary Borchers. Some Common BASIC Programs. Berkeley, California: Adam Osborne and Assoc. Inc., 1977.

16. Selvidge, Judith. Rapid Screening of Decision Options. McLean, Virginia: Decisions and Designs, Inc., 1976 (Defense Documentation Center #AD-A035386).

17. Spencer-Jones, Jack. "A General Purpose Decision-Making Model for Small Business,: Interface Age, 4 (7): 68 - 73 (July 1979).

18. Taha, Hamdy A. OPERATIONS RESEARCH AN INTRODUCTION. New York, NY: MacMillan Publishing Co., Inc.

19. Whaley, C. P. "Fuzzy Decision Making," Interface Age, 4 (11): 87 - 91 (November 1979).

APPENDIX A

CAAMM User's Guide and Instructional Notes

Introduction

The Computer Assisted Analysis for Military Managers (CAAMM) software package is designed to meet the needs of managers and analysts through easy-to-use, time-sensitive computer support. The package currently implements regression analysis, linear programming, and a value matrix decision aid. It is specifically designed for desk-top computers to exploit their responsiveness, convenient disk storage for data, and versatile video screen displays.

The programs are all fully interactive, i.e. all inputs are prompted from the video screen and entered under program control. Each program has its own data entry, disk storage/ retrieval, and editing functions. All data input is matrix oriented and uses the flexible nature of video screen control for logical format and "paper and pencil" style entry. All output is screen page oriented to allow perusal of data until the user is ready to continue. Multiple databases can be entered, edited, saved to disk, and retrieved for later use. While specific program routines are necessarily different, the basic construction, logic, style, and major features of all programs are parallel. For detailed information on program construction, see Appendix B, CAAMM Programmer's Guide. Note that all procedures outlined in the User's Guide apply specifically to implementation on the standard Apple II computer. Various procedures and capabilities may be different if the programs are converted for use on other systems. Appendix B includes information on Apple system-dependent language statements and features, to assist conversion efforts.

Users should note the following common procedures and warnings:

1) Please read the appropriate computer manuals. For the Apple II you should read the Applesoft II Basic Programming Manual (AS II Manual, Ref 3) and also the Disk Operating System Instructional and Reference Manaual (DOS Manual, Ref 4).

2) Be careful to enter data or responses in the form prompted; "foolproofing" routines are restricted because of the computer's limited memory capacity. Do not use commas, semicolons, or quotes in any data entry, alphanumeric name, or prompted response, e.g. type "10000", not "10,000".

3) You must follow every completed keyboard entry by depressing the {RETURN} key in order to register information in program memory. Until {RETURN} is depressed, you can edit the entry by left/right spacing with the cursor arrows and retyping. When the entry has been corrected, press {RETURN}. If you note entry errors after {RETURN}, continue with the program--special editing options are automatically available and can be invoked later.

4) If a program terminates early due to software or user error, re-enter the program by typing the command 'RUN', then press {RETURN}. This procedure restarts the entire program and destroys all current data. If the program "hangs," i.e. output stops for an unusually long time, or if you need to halt execution on an emergency basis, press the {RESET}

key. If you press {RESET} either intentionally or accidentally, you can attempt to re-enter the program by typing "3D0G" {RETURN}, then "RUN" {RETURN}.

5) Be sure to insert the software diskette in the disk drive, close the drive door, and initialize the system in accordance with the DOS Manual. Do not open the door to the disk drive until cycling of the disk has completely stopped. Opening the door prematurely would disrupt the current program and possibly destroy stored data.

6) If you want printer output from a program, you must not only select the printer output option when prompted by the program, but must physically turn on the printer. If you do not have a printer, do not select a printer output option--that could fatally "hang" the program.

7) Finally note that all "Yes/No" type prompts from the screen, e.g. "Do you want output routed to the printer?" can be answered affirmatively by "Yes" or "Y" and negatively by "No", "N", or a simple {RETURN}. This allows the user to quickly skip unnecessary parts of the program with minimum actual inputs.

Each of the three analysis techniques represented in the CAAMM package are discussed in detail in the sections that follow.

Section 1, Regression Analysis

Overview of Regression Analysis

Regression analysis is used to analyze the interrelationships among two or more data variables, e.g. weight, height, and age. Establishing such relationships enable the analyst to explain observed phenomena or data and to predict unobserved phenomena or data. Specifically the technique builds a linear or non-linear model by the method of least-squares curve fitting. The variable for which values are to be predicted, e.g. success in pilot training, is called the dependent variable or criterion variable. The variables used to do the prediction, e.g. age, time-in-service, and flight experience, are called independent variables or predictor variables (Ref 13:4-1).

Regression analysis is useful for examining such complex relationships as job satisfaction and soldier retention, prediction of success indicators in training, cost/volume and cost/profit, and predictions of spare part or ammunition requirements for combat.

While most good statistics textbooks devote some coverage to regression techniques, the following references are particularly readable: Mc Nichols (Ref 13: chap 4), Affifi (Ref 1: chap 3), and Mason (Ref 12: chap 9).

Overview of Programs REGR DATA, BIVAR REGR, and MULVAR REGR

Regression analysis in the CAAMM package is handled by three main programs: REGR DATA, BIVAR REGR, and MULVAR REGR.

REGR DATA automatically runs upon selection of "Regression Analysis" on the CAAMM MASTER menu. It provides the "housekeeping" functions of data entry, storage, retrieval, display, and editing for both BIVAR REGR and MULVAR REGR. A menu at the end of the program allows selection of either bivariate or multivariate analysis. The dimensioned data capacity is 20 variables with 80 observations. Multiple databases can be maintained on disk and periodically updated by REGR DATA.

BIVAR REGR accepts a database from REGR DATA, then allows selection from among nine different regression models, shown in Table II. A tenth option is an automatic curve fit for all of the first seven models, which may help the analyst with no idea which model to select. Transformations for the user-defined model are entered interactively, and once data has been transformed, a linear regression is run. BIVAR REGR allows the user to identify any variable as independent and any variable as dependent. The program first calculates and displays the means and unbiased standard deviations of the two variables. Then it calculates and displays the regression constant and coefficient for the model selected, followed by various statistical measures. Statistical measures include unadjusted R^2 (coefficient of determination), the correlation coefficient, the standard error and variance of the estimate, and the degrees of freedom. The user then has the option to

TABLE II

Bivariate Regression Analysis Options

Model	Form
Linear	$y = A + Bx$
Power	$y = A + x^B$
Exponential	$y = A + e^{Bx}$
Logarithmic	$y = A + B \log(x)$
Hyperbolic 1	$y = A + B/x$
Hyperbolic 2	$y = 1/(A + Bx)$
Hyperbolic 3	$y = x/(A + Bx)$
Nth Order Polynomial	$y = A + Bx + Cx^2 + Dx^3 + \ldots$
User-Defined	$f(y) = A + B\,f(x)$

list a table of residuals, to predict both 'x' and 'y' values, and to produce a high resolution graphics picture of the original data and curve fit model. The graphics option is implemented by a small but separate program called BIVAR HIRES. Finally the user is given the option to run another regression using the same database, to edit the database, enter a new database, or to quit the program.

MULVAR REGR accepts a database from REGR DATA, then allows selection of either a multivariate linear model or a user-defined model with interactive transformations. The user can identify any number of variables as being independent, and any remaining variable as dependent. The program first calculates and optionally displays means and unbiased standard deviations, then the simple correlation matrix. It then calculates and displays the regression constant, coefficients, and various statistical measures. Statistics include the standard error and t-ratio for the constant and each coefficient, adjusted R^2, the multiple correlation coefficient, the standard error and variance of the estimate, degrees of freedom, and the Durbin-Watson statistic. It then optionally displays the variance-covariance matrix and a table of residuals. Then the user is allowed to predict values of the dependent variable by specifying a value for each independent variable. Finally the user is given the option to run another regression using the same database, to edit the database, to enter a new database, or to quit the program.

Basic programming algorithms for BIVAR REGR linear, power, exponential, and logarithmic models were modified from Garson (Ref 6); the Nth order model was modified from Poole and Borchers (Ref 15:151-153); all other models are original and were derived analytically. The basic multivariate linear regression algorithm used in MULVAR REGR was modified from Honeywell (Ref 7:193-196). The user-defined transformation technique, program REGR DATA, program BIVAR HIRES, and the additional input and output routines of both regression programs are all original.

Section 2, Linear Programming

Overview of Linear Programming

Linear programming is a classical analysis tool used to optimize the allocation of limited resources for competing uses, e.g. the best procurement mix of strategic missiles. Specifically it seeks to maximize or minimize the value of a pre-defined linear function of decision alternatives, subject to the availability of pertinent resources. The decision alternatives are called decision variables or activities, while the limited resources are called constraints. The linear function used to evaluate actual payoff is called the objective function. The most popular linear programming method--"simplex"--is an iterative process in which various linear combinations of variables are tested for feasibility against the different constraints and for optimum total payoff.

Linear programming is most commonly used for readily quantified problems such as production models involving dollar profit or cost, but it can also be used for subjective value models such as cadre fill for newly constituted combat units, allocation of intelligence assets for various parts of the world, or research and development emphasis for new weapons.

Virtually all operations research textbooks include good summaries of linear programming techniques. Among the more readable references are Taha (Ref 18: chap 3-4) and Levin and Kirkpatrick (Ref 10: chap 10).

Overview of Program LINPROG

CAAMM program LINPROG implements the simplex linear programming method. It allows the user to create a linear programming model by one of two interactive query methods or by retrieving a model from storage on disk. Of the two interactive query methods, one allows the naming of variables and constraints while the other permits an abbreviated, strictly mathematical format. Once the model has been entered, it can be edited (to include additions and deletions as well as direct changes), it can be displayed for perusal, or it can be saved to disk storage for later use. The user has various solution output options, to include the initial tableau, the intermediate solutions, and the final tableau. Regardless of intermediate output, the optimal solution is displayed for both basis variables and dual variables.

True sensitivity analysis is not provided in LINPROG, but the model editing functions are extensive and provide even greater effective sensitivity capability--changing, adding, or deleting variables, constraints, and model coefficients.

Available computer memory restricts the linear programming model to 20 structural variables and 20 constraints. Recommendations for changing this limitation can be found in Appendix B, CAAMM Programmer's Guide.

The simplex algorithm programming code was modified from Honeywell (Ref 8: 86.1-86.5). All input/output and other data management functions are original.

Section 3, Value Matrix Decision Aid

Overview of Value Matrix Decision Aid

There are many forms of decision analysis, all designed to structure decision problems with multiple alternatives or options. The common objective is to determine the optimum decision option or to at least provide a basis for comparison. The matrix approach is specifically designed for problems that can be structured as one decision with multiple options, multiple attributes or judgement criteria, and one primary uncertainty with multiple states of nature. The matrix approach uses a linear "additive weighting" technique. Each decision option is weighted by both the relative importance of judgement criteria and by the relative likelihood of the uncertain states of nature, then summed for comparison with other options.

The matrix approach is useful for single decision problems involving risk, such as contingency plans, alert measures, and procurement packages.

The primary source for the three-dimensional matrix decision approach is Selvidge (Ref 16), although two-dimensional (non-risk) models are discussed by Spencer-Jones (Ref 17), Morris (Ref 14: chap 1), and Whaley (Ref 19).

Overview of Program DECISION

CAAMM program DECISION implements the matrix approach to decision analysis. It allows the user to retrieve a decision model from disk storage or to create one interactively.

To create a model, the user enters the names of decision options he is considering, then enters the names of judgement criteria. After assessing the relative importance or weighting of the various judgement criteria, the user enters the names and relative likelihoods or probabilities of the uncertain states of nature, if uncertainty exists. Finally the user assigns a subjective value to each option, relative to each state of nature and judgement criterion. Intermediate editing options throughout the process permit the user to correct entries and readjust assessed values until he is completely satisfied with the model. Judgement criterion weightings and state probabilities are normalized to 1.00 automatically. Once the model has been entered, it can be edited, displayed, or saved to disk storage for later use. Once evaluation of the model begins, the user can elect to find the optimum decision option by maximizing payoff or by minimizing payoff.

The user should note that since all model numerical inputs are subjective assessments, the "optimum" solution should be used only for comparative analysis. Sensitivity analysis is provided and demonstrates the effect of changing either criterion weights or state probabilities. Additional model editing can also be used for analyzing the effect of actually adding, deleting, or changing options, judgement criteria, or states of nature.

Available computer memory restricts the decision model to 15 decision options, 20 judgement criteria, and 5 uncertain states of nature. Recommendations for changing these constraints can be found in Appendix B, Programmer's Guide.

The programming code for DECISION is entirely original, but Selvidge (Ref 16) and Spencer-Jones (Ref 17) provided valuable formatting ideas.

Section 4, Sample Problems

Several sample problems have been selected to illustrate use of the various CAAMM package programs. The sample problems demonstrate most of the program options and procedures, but there is no substitute for actual computer time.

Study the problems sequentially, because the discussions build upon each other. The first problem discussion is especially comprehensive and should be studied carefully.

Preliminary

The first step is to power up and initialize the system. For the standard Apple II computer, you must:

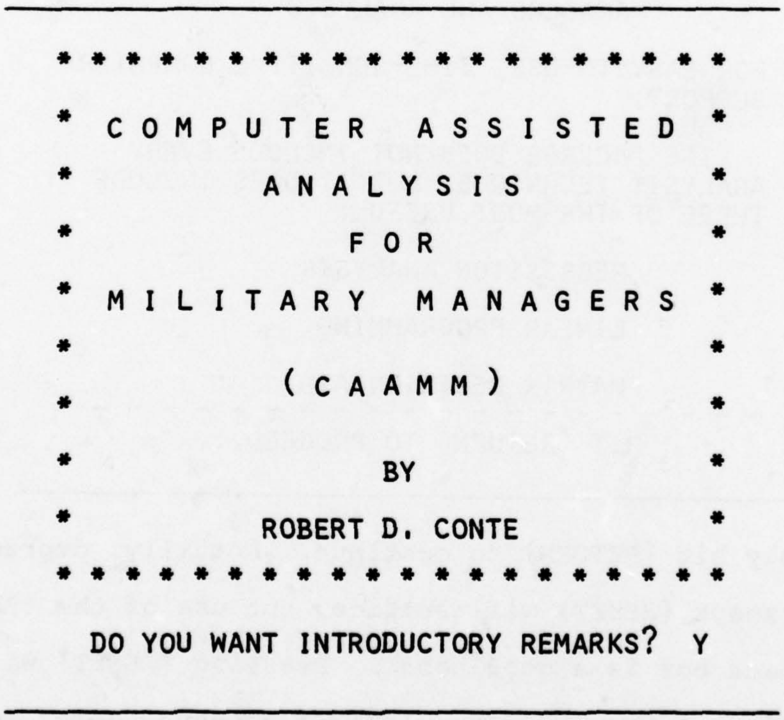
- Insert the CAAMM diskette into the disk drive

- Turn on the computer at the switch in the rear

- Turn on the video monitor

- Initialize the disk system from BASIC by typing "PR#7" {RETURN}, where 7 is the slot number of the disk control card inside the computer.

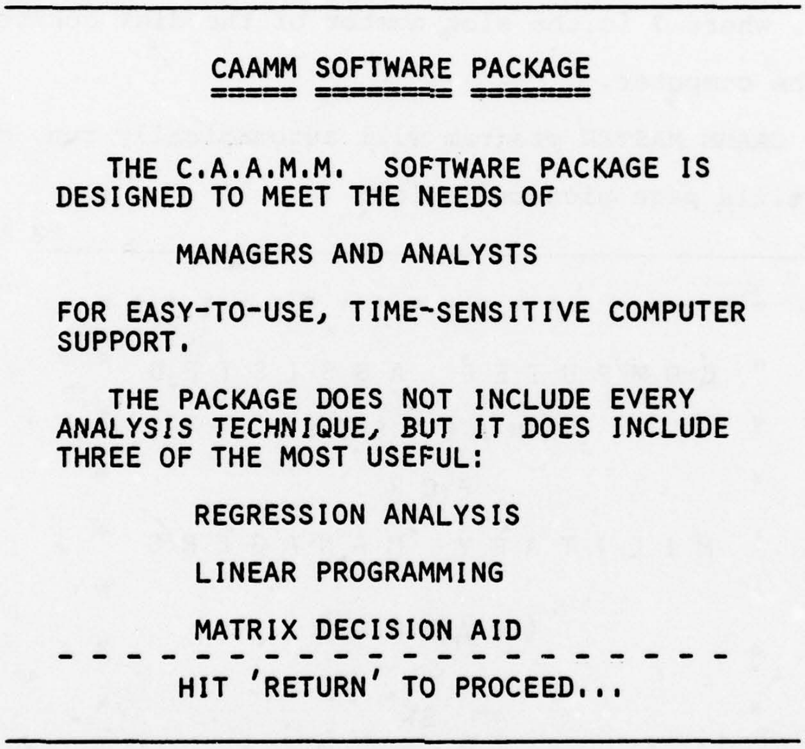
The CAAMM MASTER program will automatically run, displaying the title page pictured below:



```
* * * * * * * * * * * * * * * * * * * *
*  C O M P U T E R   A S S I S T E D  *
*            A N A L Y S I S          *
*                F O R                *
*  M I L I T A R Y   M A N A G E R S  *
*                                     *
*            ( C A A M M )            *
*                                     *
*                 BY                  *
*           ROBERT D. CONTE           *
* * * * * * * * * * * * * * * * * * * *
   DO YOU WANT INTRODUCTORY REMARKS?  Y
```

Type a "Y" or "YES," then {RETURN} for the introductory remarks. If you do not need the edification of introductory remarks, type an "N" or "NO". NOTE: all complete data entries must be followed by a {RETURN} in order to register the information in memory. From this point on, the {RETURN} will seldom be specifically mentioned.

If you respond affirmatively to the first query, the following will appear:



```
                 CAAMM SOFTWARE PACKAGE
                 ===== ======== =======

    THE C.A.A.M.M.  SOFTWARE PACKAGE IS
DESIGNED TO MEET THE NEEDS OF

               MANAGERS AND ANALYSTS

FOR EASY-TO-USE, TIME-SENSITIVE COMPUTER
SUPPORT.

    THE PACKAGE DOES NOT INCLUDE EVERY
ANALYSIS TECHNIQUE, BUT IT DOES INCLUDE
THREE OF THE MOST USEFUL:

                REGRESSION ANALYSIS

                LINEAR PROGRAMMING

                MATRIX DECISION AID
- - - - - - - - - - - - - - - - - - - -
             HIT 'RETURN' TO PROCEED...
```

Simply hit {RETURN} to continue. Actually, depressing any key except {RESET} will suffice, but use of the {RETURN} key or space bar is a good habit. Pressing {RESET} will fatally disrupt the program. If this happens, refer to the warnings at the beginning of the User's Guide for recovery procedures.

```
                       ** NOTE **

            FOR RAPID PROGRAM TRANSITIONS,

        ALL  'YES/NO'  TYPE QUESTIONS OR PROMPTS
        CAN BE ANSWERED BY  'Y'  FOR 'YES' OR
        BY 'N'  OR A SIMPLE  'RETURN'  FOR 'NO'...

        - - - - - - - - - - - - - - - - - - - - -

               HIT 'RETURN' TO PROCEED...
```

- NOTE: henceforth, all "YES/NO" type questions or option prompts can be answered by "Y" for "Yes" or by an "N" or simple {RETURN} for "No". In other words, the "default" is "No". Other defaults will be highlighted as we continue. Knowledge of the defaults will help you to quickly bypass unnecessary parts of the program.

- The final screen page of CAAMM MASTER is the Master Menu:

```
              CAAMM  MASTER  MENU
              =====  ======  ====

       YOU MAY SELECT ONE OF THE FOLLOWING:
       --- --- ------ --- -- --- ---------

            1.  REGRESSION ANALYSIS

            2.  LINEAR PROGRAMMING

            3.  MATRIX DECISION AID

            4.  ** QUIT  CAAMM  PACKAGE **

                WHICH  SELECTION?
```

- Enter the menu number of the analysis technique you plan to use. Note that the Regression Analysis programs are treated as a consolidated package. Select Regression Analysis to run REGR DATA, Linear Programming to run LINPROG, or Matrix Decision Aid to run DECISION.

Problem 1, Bivariate Regression

The fleagle valve in a turbine blade extender pump used on most commercial jet aircraft engines has exhibited the following history of failure frequency per 1000 units in operation (Ref 11):

TABLE III

Fleagle Valve Failure Rate

PERIOD	HRS. OF OPERATION	FAILURES	PERIOD	HRS. OF OPERATION	FAILURES
1	0-10	5	11	101-110	15
2	11-20	4	12	111-120	21
3	21-30	6	13	121-130	26
4	31-40	6	14	131-140	32
5	41-50	5	15	141-150	35
6	51-60	4	16	151-160	36
7	61-70	6	17	161-170	41
8	71-80	8	18	171-180	47
9	81-90	9	19	181-190	55
10	91-100	11	20	191-200	62

If you were responsible for budgeting funds and maintenance for these valves next year, you could use regression analysis to predict the number of failures that will likely occur over various durations of operation. To analyze the problem using CAAMM software:

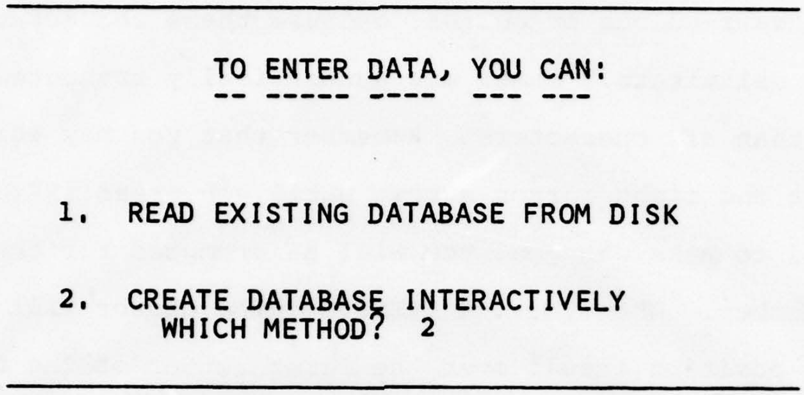
- Select Regression Analysis on the CAAMM MASTER Menu.

Program REGR DATA will run and soon present:

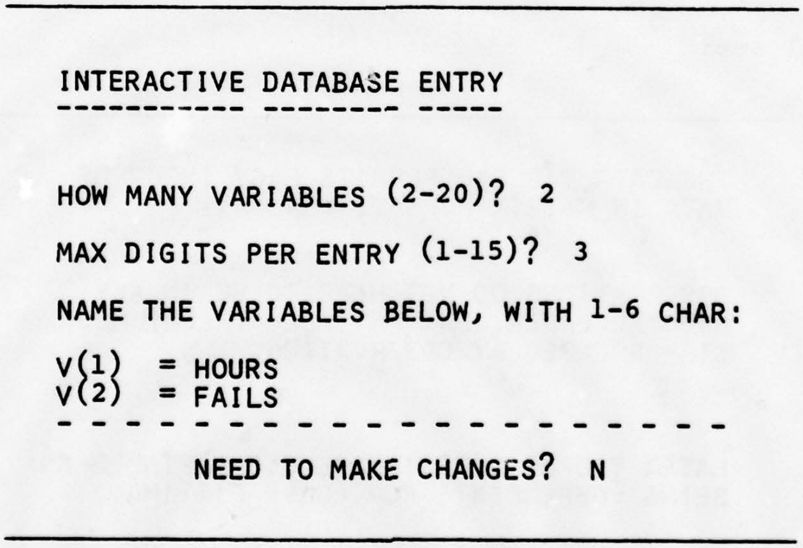
```
* * * * * * * * * * * * * * * * * * * *

             R E G R E S S I O N

               A N A L Y S I S

                     BY

               ROBERT D. CONTE

* * * * * * * * * * * * * * * * * * * *


    DO YOU WANT INTRODUCTORY REMARKS? Y
```

```
REGRESSION ANALYSIS IS USED TO EXAMINE
THE INTERRELATIONSHIPS AMONG TWO OR MORE
VARIABLES FOR WHICH DATA IS AVAILABLE.


THE PACKAGE INCLUDES 3 INTERCONNECTED
MAIN PROGRAMS TO PROVIDE:

    DATABASE MANAGEMENT

    BIVARIATE REGRESSION

    MULTIVARIATE REGRESSION
- - - - - - - - - - - - - - - - - - - -
HIT 'RETURN' TO GO ON, OR 'Q' TO QUIT
```

- You now have a choice of options for entering problem data. Since you currently have no files on disk, select option 2.



```
          TO ENTER DATA, YOU CAN:
          -- ----- ----  --- ---


1.  READ EXISTING DATABASE FROM DISK

2.  CREATE DATABASE INTERACTIVELY
       WHICH METHOD? 2
```

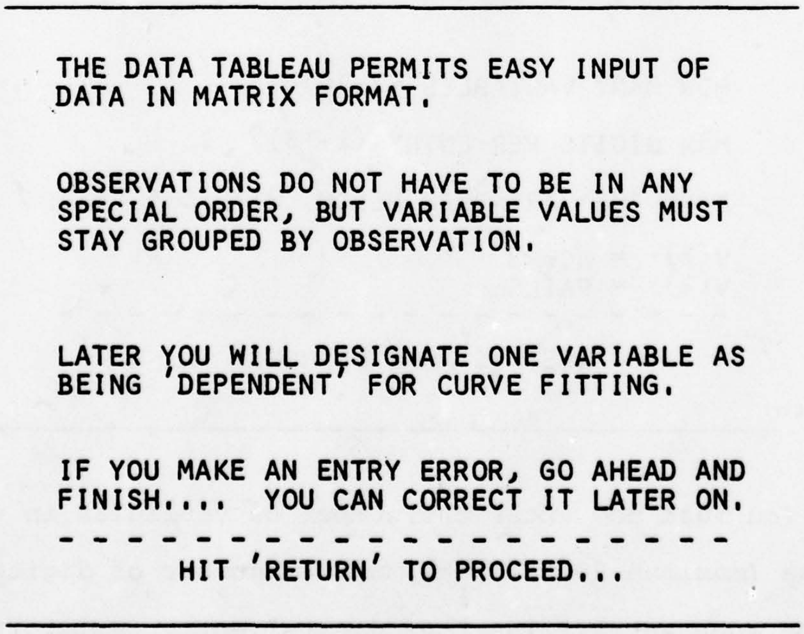


```
INTERACTIVE DATABASE ENTRY
----------- -------- -----


HOW MANY VARIABLES (2-20)?  2

MAX DIGITS PER ENTRY (1-15)?  3

NAME THE VARIABLES BELOW, WITH 1-6 CHAR:

V(1)  = HOURS
V(2)  = FAILS
- - - - - - - - - - - - - - - - - - - - -

        NEED TO MAKE CHANGES?  N
```

- You must now enter the number of variables in your database (maximum 20) and the maximum number of digits you need for each entry. The "max digits" entry is important because it becomes the field width for data entry and formatting. The default field width (max digits) is five.

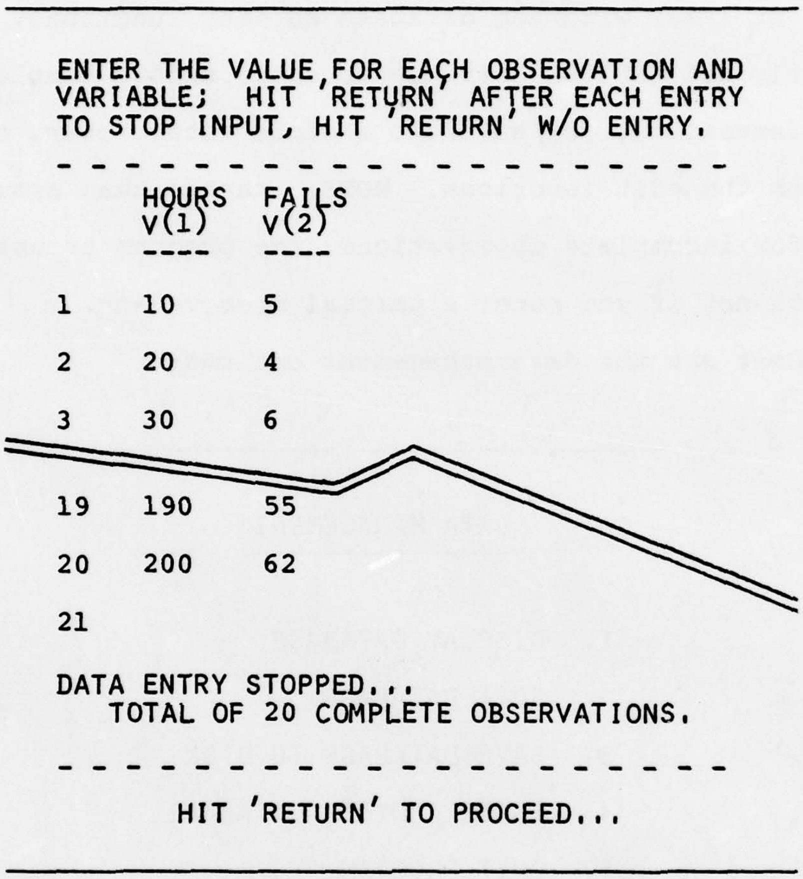
- Next enter names for your variables, using up to six alphanumeric characters, including spaces. Do not enter commas, semi-colons or quotes, because these characters are used as delimiters. Names are automatically truncated if longer than six characters. Remember that you may edit with the left and right cursor arrows until you press {RETURN}. If you need to make changes, you will be prompted for the variable index number. When you hit {RETURN}, the cursor will automatically position itself over the first letter of the name to be changed.

- If you answer "Y" to the prompt for "Instructions?" you will see:



```
THE DATA TABLEAU PERMITS EASY INPUT OF
DATA IN MATRIX FORMAT.

OBSERVATIONS DO NOT HAVE TO BE IN ANY
SPECIAL ORDER, BUT VARIABLE VALUES MUST
STAY GROUPED BY OBSERVATION.


LATER YOU WILL DESIGNATE ONE VARIABLE AS
BEING 'DEPENDENT' FOR CURVE FITTING.


IF YOU MAKE AN ENTRY ERROR, GO AHEAD AND
FINISH....  YOU CAN CORRECT IT LATER ON.
- - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO PROCEED...
```

- Next is the actual entry of data:

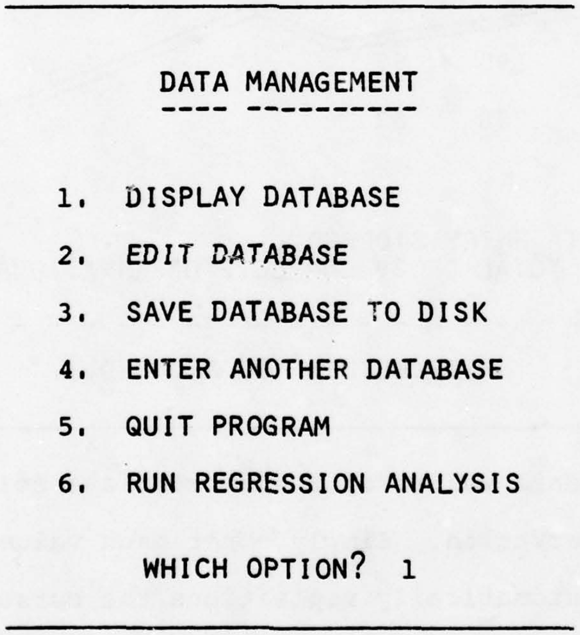


```
ENTER THE VALUE FOR EACH OBSERVATION AND
VARIABLE;  HIT 'RETURN' AFTER EACH ENTRY
TO STOP INPUT, HIT 'RETURN' W/O ENTRY.
- - - - - - - - - - - - - - - - - - - - -
      HOURS  FAILS
      V(1)   V(2)
      ----   ----

1     10     5

2     20     4

3     30     6


19    190    55

20    200    62

21

DATA ENTRY STOPPED...
   TOTAL OF 20 COMPLETE OBSERVATIONS.

- - - - - - - - - - - - - - - - - - - - -

       HIT 'RETURN' TO PROCEED...
```

- Enter data values in the appropriate columns, observation by observation. Simply enter each value, then {RETURN}. The program automatically repositions the cursor for input in the next column. Observation index numbers are displayed for convenience. Continue entering values until you have no more observations (maximum 80) then press {RETURN} with no entry. The display heading and variable names are protected and remain in view while the observation values scroll underneath. The screen accommodates a maximum of five variables across;

if there are more than five variables, data must be entered in sequential blocks. If you make an entry error, you can correct it later with the database editing functions. If you accidently hit {RETURN} without an entry before completing your observations, you have two choices--start over, or add data with the edit functions. NOTE: rather than assume zero values for incomplete observations, the program truncates your data set if you enter a partial observation.

- Next are the data management options:



```
                DATA MANAGEMENT
                ---- ----------

        1.  DISPLAY DATABASE
        2.  EDIT DATABASE
        3.  SAVE DATABASE TO DISK
        4.  ENTER ANOTHER DATABASE
        5.  QUIT PROGRAM
        6.  RUN REGRESSION ANALYSIS

              WHICH OPTION? 1
```

- This is the Master or "crossroads" Menu for REGR DATA. Each of the first four options returns to this menu after execution so a typical sequence might be to first display the database to ensure that it was correctly entered, then edit if necessary, save to disk storage, and finally to enter another database, leave the program, or start regression analysis.

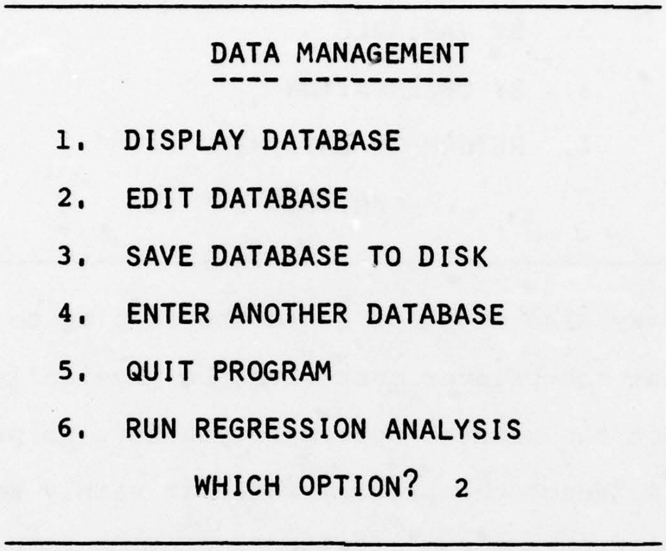
- There are three database display options:

```
                DATABASE DISPLAY OPTIONS
                -------- ------- -------

        1.  ENTIRE DATABASE

        2.  BY VARIABLE

        3.  BY OBSERVATION

        4.  RETURN TO LAST MENU

                    WHICH? 1
```

- You may also elect to route the display to a printer. Remember that the printer must first be physically powered up. Do not select the printer option if you have no printer--you will fatally "hang" the program while it vainly searches for a non-existant peripheral. Selecting option 1, you will see:

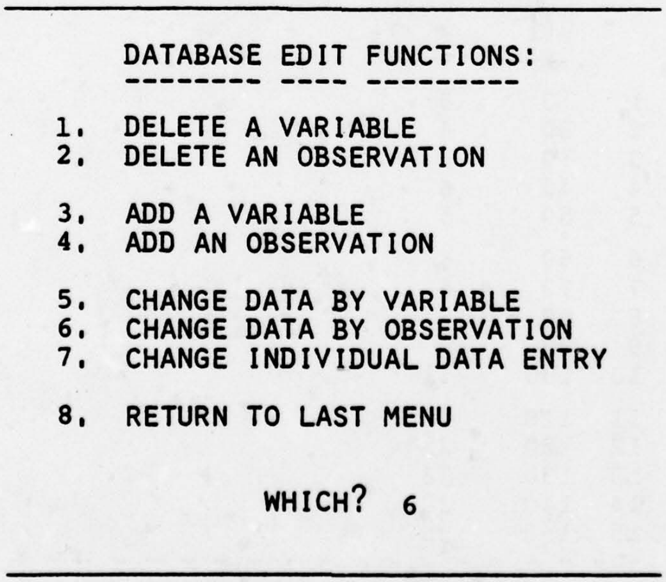
```
       HOURS  FAILS
       V(1)   V(2)
       ----   ----

   1   10     5
   2   20     4
   3   30     6
   4   40     6
   5   50     5

   6   60     4
   7   70     6
   8   80     8
   9   90     9
   10  100    11

   11  110    15
   12  120    21
   13  130    26
   14  140    32
   15  150    35
   - - - - - - - - - - - - - - - - - - -
        HIT 'RETURN' TO PROCEED...
```

- Observations are printed in blocks of five for easier reading. No more than 15 observations or 5 variables will appear at one time. Press {RETURN} for subsequent blocks of the database. Returning to the Master Menu:



```
          DATA MANAGEMENT
          ---- ----------

   1.  DISPLAY DATABASE

   2.  EDIT DATABASE

   3.  SAVE DATABASE TO DISK

   4.  ENTER ANOTHER DATABASE

   5.  QUIT PROGRAM

   6.  RUN REGRESSION ANALYSIS

         WHICH OPTION? 2
```

- The Edit function menu appears below:



```
       DATABASE EDIT FUNCTIONS:
       -------- ---- ----------

   1.  DELETE A VARIABLE
   2.  DELETE AN OBSERVATION

   3.  ADD A VARIABLE
   4.  ADD AN OBSERVATION

   5.  CHANGE DATA BY VARIABLE
   6.  CHANGE DATA BY OBSERVATION
   7.  CHANGE INDIVIDUAL DATA ENTRY

   8.  RETURN TO LAST MENU

             WHICH? 6
```

- Should you decide to change the values of one observation, for example, you would select option 6 and then see:

```
CHANGE DATA BY OBSERVATION
------ ---- -- -----------

WHICH OBSERVATION (1-20)?  7

                 OBSERVATION  7
                 -----------

VARIABLE         CURRENT        CHANGE TO

V(1)   HOURS     70             65
V(2)   FAILS     6              7

- - - - - - - - - - - - - - - - - - - - -
                 EDIT ANOTHER ?  N
```

- You can default to the current value simply by pressing {RETURN} and moving to the next input position.

- When you finish editing, you will return to the Master Menu. You might now want to save the database to a disk file for future use (and protect yourself from the ubiquitous power failure or fatal program error).

```
        SAVE DATABASE TO DISK
        ---- -------- -- ----

SAVE UNDER WHAT FILE NAME?  BIDEMO
```

- You will be queried for a file name, which must not exceed 30 characters in length and must start with a letter. Any character may be used (including spaces, dashes, and colons) except the comma. Try to insure that there is sufficient storage space on the disk you are using, and be careful not to use a filename already in use. If in doubt, CATALOG the diskette before starting the program. Typing "CATALOG" {RETURN} will cause a display like this:

```
DISK VOLUME 254

 A 011 CAAMM MASTER
 A 085 DECISION
 B 003 CHAIN
 A 050 REGR DATA
 A 067 BIVAR REGR
 A 011 BIVAR HIRES
 A 055 MULVAR REGR
 A 093 LINPROG
 T 002 BIDEMO
 T 002 MULDEMO
 T 002 LPDEMO
 T 003 DECDEMO
```

- Files denoted by an "A" are the CAAMM software programs, files denoted by a "B" are assembly language programs, and files denoted by a "T" are the data files or "text files". Since the data files for regression analysis, linear programming, and decision analysis are not compatible and disk storage space is limited, it is recommended to segregate the different programs. You must ensure, however, that the four regression analysis programs and the CHAIN program are located on the same diskette (see Programmer's Guide for additional details).

- After saving the database to disk, we now proceed to actual regression analysis by selecting option 6 or hitting {RETURN} at the Master Menu:

```
              DATA MANAGEMENT
              ---- ----------

          1.  DISPLAY DATABASE

          2.  EDIT DATABASE

          3.  SAVE DATABASE TO DISK

          4.  ENTER ANOTHER DATABASE

          5.  QUIT PROGRAM

          6.  RUN REGRESSION ANALYSIS

                    WHICH OPTION? 6
```

```
PLEASE SELECT ONE OF THE METHODS BELOW:
------ ------ --- -- --- ------- -----

1.  BIVARIATE  REGRESSION  ANALYSIS

      (FOR DATA SETS OF ONE DEPENDENT
      AND ONE INDEPENDENT VARIABLE)

2.  MULTIVARIATE  REGRESSION  ANALYSIS

      (FOR DATA SETS OF ONE DEPENDENT
      AND ONE OR MORE INDEP VARIABLES)

    WHICH METHOD? 1
```

- Bear in mind that while the procedure may appear complicated in writing, the program is fully automatic and prompts for all inputs. Once a database has been stored on disk, you can later retrieve it and proceed directly to Regression Analysis in just a few seconds.

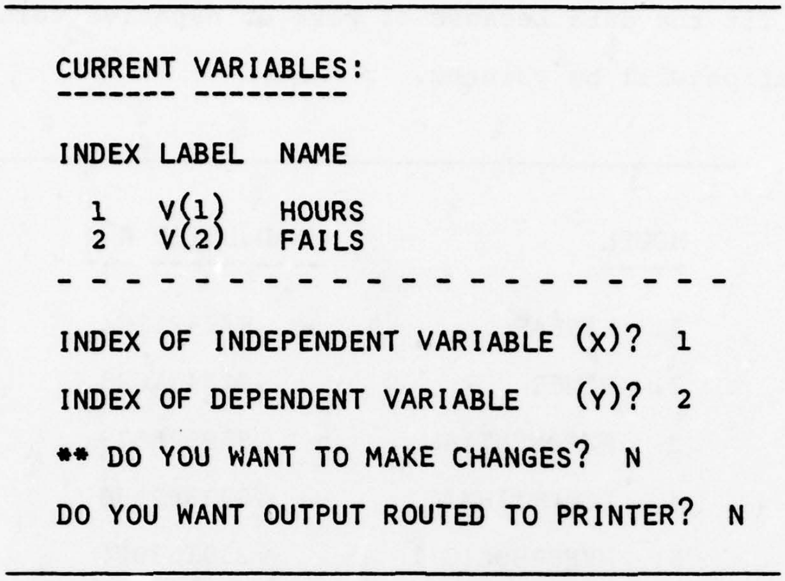
- Recalling our original fleagle valve failure problem with two variables, we should select Bivariate Regression Analysis. We could use Multivariate Analysis, but it is much slower and less flexible than the Bivariate program.

- The system will automatically load BIVAR REGR for bivariate analysis. In a few seconds, the following bivariate menu will appear:

```
          BIVARIATE CURVE FITTING OPTIONS:
          --------- ----- ------- -------

     1.   LINEAR           Y = A + (B*X)
     2.   POWER            Y = A * (X^B)
     3.   EXPONENTIAL      Y = A * EXP(B*X)
     4.   LOGARITHMIC      Y = A + B*LOG(X)
     5.   HYPERBOLIC 1     Y = A + (B/X)
     6.   HYPERBOLIC 2     Y = 1/(A + B*X)
     7.   HYPERBOLIC 3     Y = X/(A + B*X)
     8.   N'TH ORDER       Y = A + B*X + C*X^2 +...
     9.   USER-DEFINED MODEL
     10.  AUTOMATIC FIT USING MODELS 1-7 ABOVE

              WHICH REGRESSION MODEL?  10
```

- At this point you may have no idea which model might best fit the data, so select option 10 to get an automatic curve fit using the first seven models.

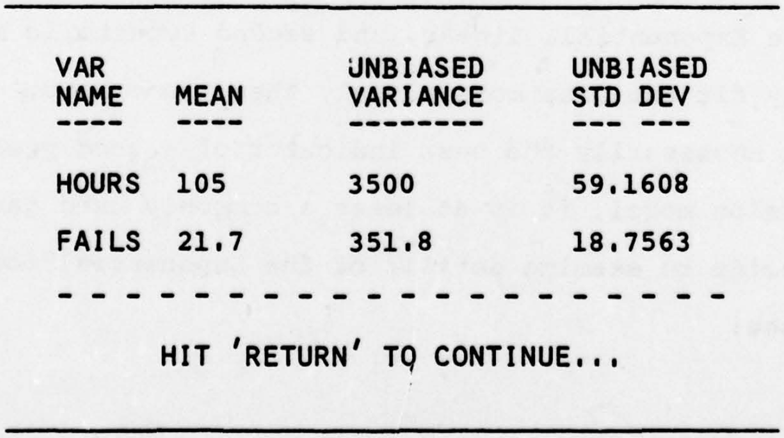
- Regardless which model you select, the display will list all variables in your database:



```
CURRENT VARIABLES:
------- ---------

INDEX LABEL  NAME

  1   V(1)   HOURS
  2   V(2)   FAILS
- - - - - - - - - - - - - - - - - - - - - - - - -

INDEX OF INDEPENDENT VARIABLE (X)?  1

INDEX OF DEPENDENT VARIABLE   (Y)?  2

** DO YOU WANT TO MAKE CHANGES?  N

DO YOU WANT OUTPUT ROUTED TO PRINTER?  N
```

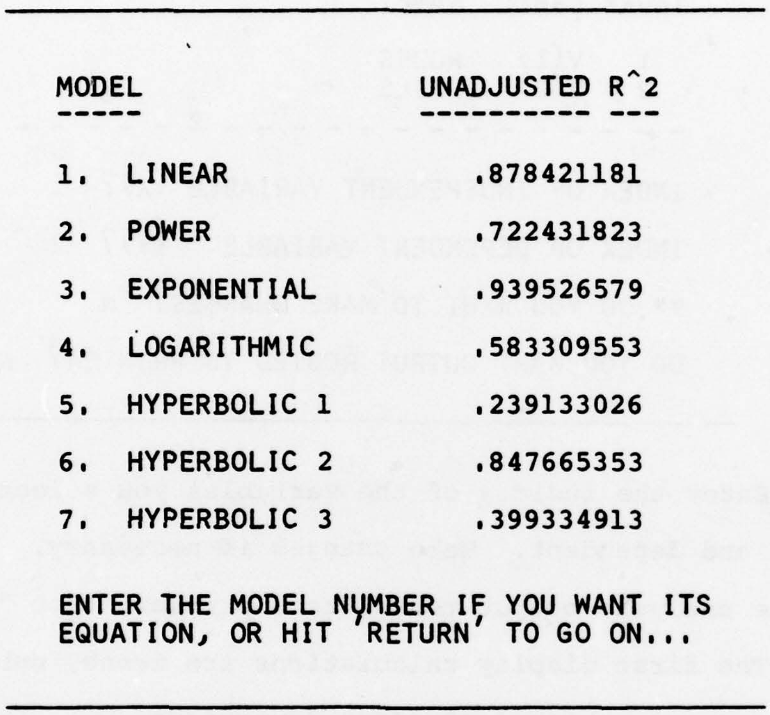
- Enter the indices of the variables you select as independent and dependent. Make changes if necessary. If you want the analysis output routed to a printer, type "Y".

- The first display calculations are means, unbiased variances, and unbiased standard deviations:



```
VAR                UNBIASED       UNBIASED
NAME   MEAN        VARIANCE       STD DEV
----   ----        --------       --- ---

HOURS  105         3500           59.1608

FAILS  21.7        351.8          18.7563
- - - - - - - - - - - - - - - - - - - - - - - - -

        HIT 'RETURN' TO CONTINUE...
```

- Once the data has been fitted to all seven models, the unadjusted R^2 values are displayed. If a particular model cannot fit the data because of zero or negative values, an explanation will be printed.



```
MODEL                    UNADJUSTED R^2
-----                    ---------- ---

1. LINEAR                 .878421181

2. POWER                  .722431823

3. EXPONENTIAL            .939526579

4. LOGARITHMIC            .583309553

5. HYPERBOLIC 1           .239133026

6. HYPERBOLIC 2           .847665353

7. HYPERBOLIC 3           .399334913

ENTER THE MODEL NUMBER IF YOU WANT ITS
EQUATION, OR HIT 'RETURN' TO GO ON...
```

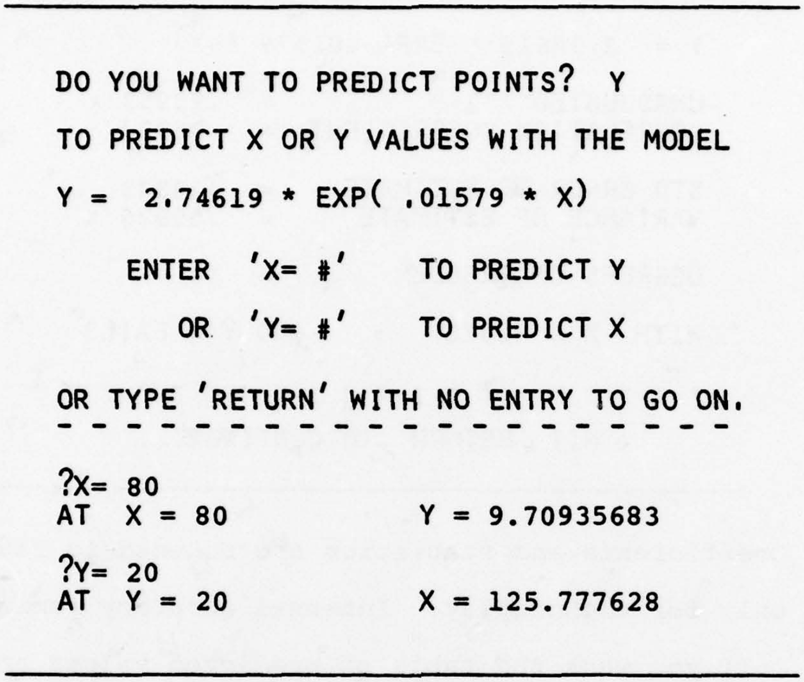
- For this problem, there is a wide spread of R^2 values, but the Exponential, Linear, and second Hyperbolic models clearly fit the data more closely than the others. While R^2 is not necessarily the best indicator of a good predictive regression model, it is at least a commonly used gauge. If you decide to examine details of the Exponential model, you will see:

```
EXPONENTIAL        Y = A * EXP(B*X)
- - - - - - - - - - - - - - - - - - - - -

Y =  2.74619 * EXP( .01579 * X)

UNADJUSTED R^2                  =  .93953
CORRELATION COEFFICIENT         =  .96929

STD ERROR OF ESTIMATE           =  .24343
VARIANCE OF ESTIMATE            =  .05926

DEGREES OF FREEDOM              =  18

WITH  X = HOURS           AND Y = FAILS

- - - - - - - - - - - - - - - - - - - - -

      HIT 'RETURN' TO CONTINUE...
```

- Coefficients and statistics are rounded to five decimal places only for the display. Internal accuracy remains nine digits. If you want the table of predicted values and residuals, the screen will list:

```
    ACTUAL ACTUAL PREDICTED   RESIDUAL
     'X'    'Y'     'Y'
    ------ ------ ---------   --------

 1  10     5      3.2158      1.7842
 2  20     4      3.7657      .2343
 3  30     6      4.4096      1.5904
 4  40     6      5.1637      .8363
 5  50     5      6.0467      -1.0467

16 160     36     34.3281     1.6719
17 170     41     40.1982     .8018
18 180     47     47.0722     -.0722
19 190     55     55.1216     -.1216
20 200     62     64.5475     -2.5475
- - - - - - - - - - - - - - - - - - - -

      HIT 'RETURN' TO CONTINUE...
```

- If you choose to interpolate or extrapolate for either variable, simply follow the screen prompts:

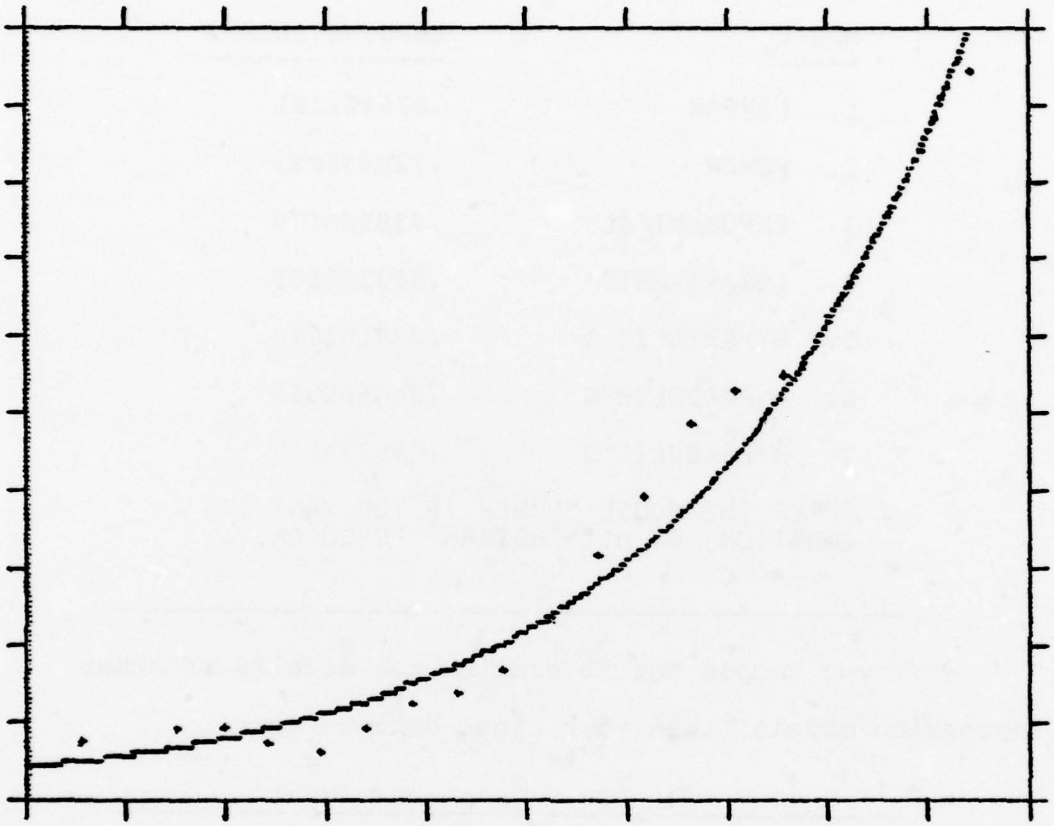


```
DO YOU WANT TO PREDICT POINTS?  Y

TO PREDICT X OR Y VALUES WITH THE MODEL

Y =  2.74619 * EXP( .01579 * X)

     ENTER  'X= #'    TO PREDICT Y

        OR  'Y= #'    TO PREDICT X

OR TYPE 'RETURN' WITH NO ENTRY TO GO ON.
- - - - - - - - - - - - - - - - - - - - -

?X= 80
AT  X = 80            Y = 9.70935683

?Y= 20
AT  Y = 20            X = 125.777628
```

- Regardless how many values you predict, the regression model and instructions remain protected at the top of the screen. Bear in mind, however, that the regression model may be valid only within the range of the observed data.

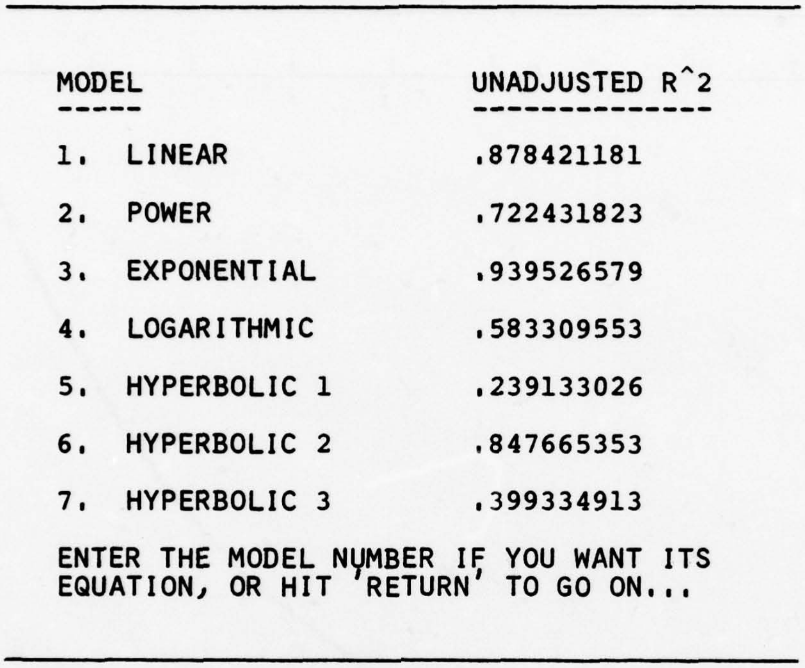
- Finally you are offered a high resolution color graphics plot which depicts the original data points as well as the regression curve. The graphics plot will help you visualize the nat'ure of the original data and provide a far better picture of "just how close" the regression curve fits the data. Data is plotted with the independent variable horizontally and the dependent variable vertically. The graph borders extend 15%

beyond the original data points. The origin is represented by cross-hairs if it falls within the data boundaries.



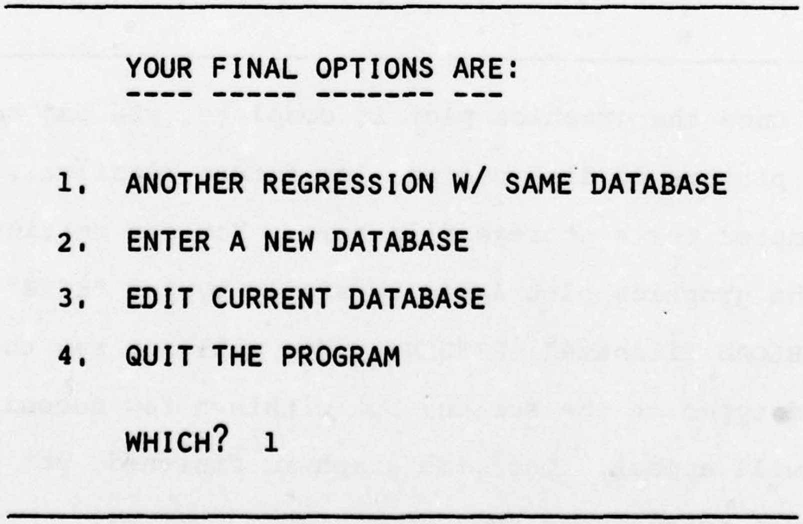
- Once the graphics plot is complete, you can save the actual picture to disk storage for future retrieval. You will be prompted for a storage file name. You can retrieve and display the graphics plot later by simply typing "HGR2" {RETURN}, then "BLOAD filename" {RETURN}. You will not see the BLOAD command typed on the screen, but within a few seconds, the graph will appear. Once the graph is finished, you can return to normal programming by typing "TEXT" {RETURN}.

- Whether you elect to save the graphics plot or not, you will soon return to the main program:



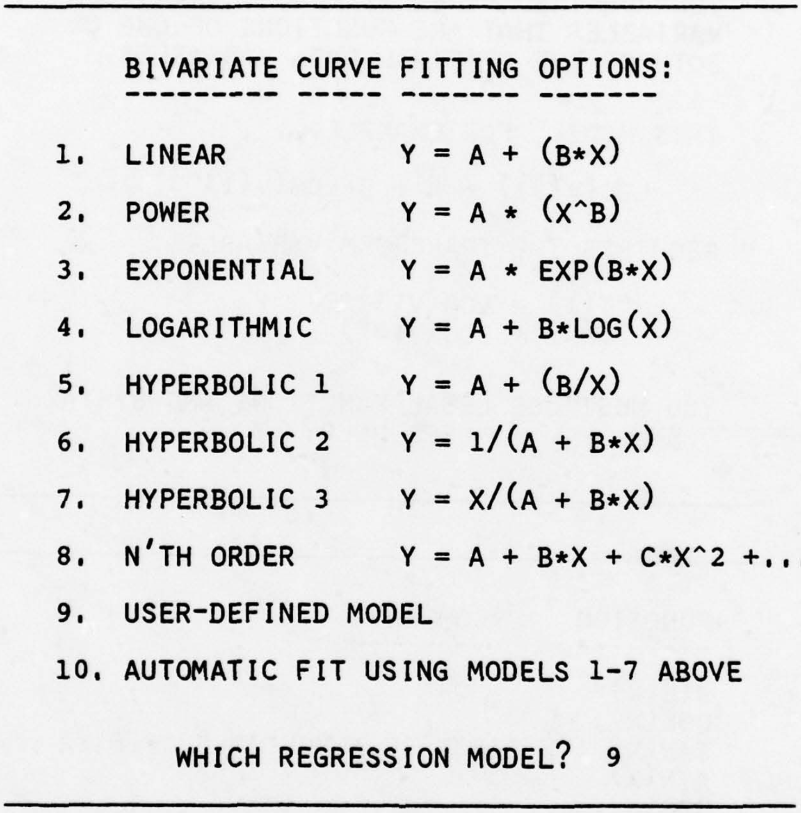
```
MODEL                       UNADJUSTED R^2
-----                       --------------

1.  LINEAR                  .878421181

2.  POWER                   .722431823

3.  EXPONENTIAL             .939526579

4.  LOGARITHMIC             .583309553

5.  HYPERBOLIC 1            .239133026

6.  HYPERBOLIC 2            .847665353

7.  HYPERBOLIC 3            .399334913

ENTER THE MODEL NUMBER IF YOU WANT ITS
EQUATION, OR HIT 'RETURN' TO GO ON...
```

- If you choose not to examine the details of other regression models, then your final options are:



```
    YOUR FINAL OPTIONS ARE:
    ---- ----- ------- ---

1.  ANOTHER REGRESSION W/ SAME DATABASE

2.  ENTER A NEW DATABASE

3.  EDIT CURRENT DATABASE

4.  QUIT THE PROGRAM

    WHICH? 1
```

- Options 2 and 3 both return to the REGR DATA program and its full range of data management options. Option 4 returns to the control program, CAAMM MASTER. Selecting the first option returns us to the ten original Bivariate Model options:



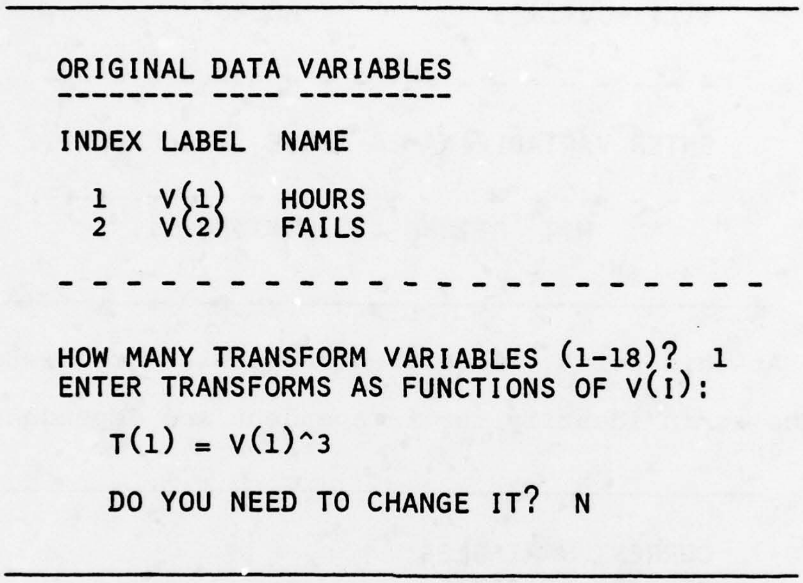
```
          BIVARIATE CURVE FITTING OPTIONS:
          --------- ----- ------- -------

    1.  LINEAR             Y = A + (B*X)
    2.  POWER              Y = A * (X^B)
    3.  EXPONENTIAL        Y = A * EXP(B*X)
    4.  LOGARITHMIC        Y = A + B*LOG(X)
    5.  HYPERBOLIC 1       Y = A + (B/X)
    6.  HYPERBOLIC 2       Y = 1/(A + B*X)
    7.  HYPERBOLIC 3       Y = X/(A + B*X)
    8.  N'TH ORDER         Y = A + B*X + C*X^2 +...
    9.  USER-DEFINED MODEL
    10. AUTOMATIC FIT USING MODELS 1-7 ABOVE

             WHICH REGRESSION MODEL? 9
```

- If you are not satisfied with any of the standard regression models, you can define your own model with a transformation of variables. If you request instructions, you will see:

```
DEFINING A MODEL ALLOWS YOU TO FIT DATA
TO VIRTUALLY ANY ALGEBRAIC CURVE THAT
CAN BE EXPRESSED AS A TRANSFORMATION OF
THE BASIC LINEAR MODEL, I.E.

Y = A + B*X     ===>     T(Y) = A + B*T(X)

YOUR DEFINED MODEL WILL USE 'TRANSFORM'
VARIABLES THAT ARE FUNCTIONS OF ONE OR
BOTH OF THE ORIGINAL DATA VARIABLES.

THIS MODEL, FOR EXAMPLE...

    LOG(V(2)) = A + B*LOG(V(1)^2)

REQUIRES THE TRANSFORM VARIABLES:

      T(1) = LOG(V(1)^2)
      T(2) = LOG(V(2))

YOU MUST USE LEGAL FUNCTIONS AND SYNTAX.
              NEED HELP?  Y
```

```
FUNCTION     COMMENTS
--------     --------

SIN(X)
COS(X)
TAN(X)    X CAN'T BE A MULTIPLE OF PI/2
ATN(X)
ABS(X)
SQR(X)    SQUARE ROOT; X MUST BE >= 0
EXP(X)    'E' RAISED TO THE POWER OF X
LOG(X)    NATURAL LOG; X MUST BE > 0

** PARENTHESES MUST BE CLOSED

** EXPONENTIATE USING '^',  E.G. 'X^2'

** THE VALUE 'PI' CAN BE ENTERED AS 'PI'
** TRIG FUNCTIONS EXPECT X IN RADIANS
- - - - - - - - - - - - - - - - - - - - -

       HIT 'RETURN' TO CONTINUE...
```

- Please note the restrictions on arguments for the various intrinsic functions. If you want to try a cubic model of the form "y = A + Bx^3" then enter the transformation as below:



```
ORIGINAL DATA VARIABLES
-------- ---- ---------

INDEX LABEL  NAME

  1   V(1)   HOURS
  2   V(2)   FAILS

- - - - - - - - - - - - - - - - - - - - -

HOW MANY TRANSFORM VARIABLES (1-18)?  1
ENTER TRANSFORMS AS FUNCTIONS OF V(I):

   T(1) = V(1)^3

   DO YOU NEED TO CHANGE IT?  N
```

- The transform itself may be as long as 255 characters, but be certain to observe the proper syntax for closing parentheses, spelling of functions, and use of "V(I)".

- The transform process requires considerable disk interaction as files automatically pass back and forth. Once the transformation of data is completed, you can name the new variable:

```
THESE TRANSFORMS HAVE BEEN CREATED:
----- ---------- ---- ---- -------

                         VAR NAME

T(1) = V(1)^3            T-HOURS

- - - - - - - - - - - - - - - - - - - - -

ENTER VARIABLE NAMES ABOVE (1-6 CHAR)...

- - - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO CONTINUE...
```

- At this point, all current variables are listed, and you once again identify the independent and dependent variables:

```
CURRENT VARIABLES:
------- ---------

INDEX LABEL  NAME

  1   V(1)   HOURS
  2   V(2)   FAILS
  3   T(1)   T-HOUR

- - - - - - - - - - - - - - - - - - - - -

INDEX OF INDEPENDENT VARIABLE (X)?  3

INDEX OF DEPENDENT VARIABLE   (Y)?  2

   DO YOU WANT TO MAKE CHANGES?  N
```

- The regression now conducted is linear, but it operates on the transformed data. The transform model statistics appear below:

```
USER-DEFINED   FAILS = A + B*T-HOUR
- - - - - - - - - - - - - - - - - - - -

Y = 5.34287 + 1E-05 * X

UNADJUSTED R^2          =  .98148
CORRELATION COEFFICIENT =  .99069

STD ERROR OF ESTIMATE   =  2.62275
VARIANCE OF ESTIMATE    =  6.8788

DEGREES OF FREEDOM      =  18

WITH  X = T HOUR     AND  Y = FAILS
```

- Again you must note that "X" in the model above is actually "V(1)^3". The high resolution graphics plot would of course reflect a straight line curve fit.

- If you are now satisfied with the quality of the curve fit, you can print out residuals and predict values to help in your budget analysis for fleagle valves.

```
        YOUR FINAL OPTIONS ARE:
        ---- ----- ------- ---

1.  ANOTHER REGRESSION W/ SAME DATABASE
2.  ENTER A NEW DATABASE
3.  EDIT CURRENT DATABASE
4.  QUIT THE PROGRAM
    WHICH? 4
```

Problem 2, Multivariate Regression

A new airborne weapons carrier is being considered for development and has the following design specifications:

Maximum Highway Speed	:	65 miles per hour
Load Capacity	:	7.0 ton-miles per gallon of fuel
Weapon System Weight	:	10,000 pounds

As cost analyst, you are asked to estimate the initial tooling costs for the proposed vehicle (Ref 11). Table IV summarizes related historical data for 14 similar weapon systems.

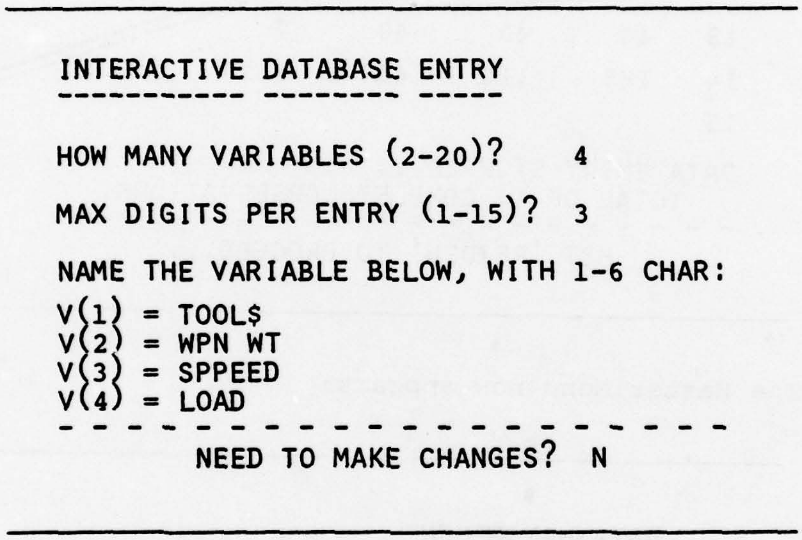
TABLE IV

Initial Tooling Cost and Various Weapon System Characteristic Data

Weapon System Type	Initial Tooling Cost (Millions $)	Weapon System Weight (100's of Lbs)	Max Speed Highway (MPH)	Load Capacity (Ton-miles/Gal)
1	28	27	43	1.8
2	35	28	48	2.0
3	40	29	50	1.5
4	60	35	65	2.0
5	50	32	70	5.0
6	55	40	100	2.6
7	90	45	110	4.5
8	70	60	43	17.0
9	285	135	45	29.0
10	130	70	100	21.0
11	105	90	43	9.0
12	80	70	23	9.0
13	40	40	40	7.0
14	185	110	45	6.8

To analyze the problem using CAAMM software, select Regression Analysis on the CAAMM MASTER Menu. Program REGR DATA will then run and offer the options to read data from disk or enter data interactively. Since we currently have no disk-stored files for this problem, select the interactive data entry option.

- As in Problem 1, we must now enter the basic variable information:



```
INTERACTIVE DATABASE ENTRY
----------- -------- -----

HOW MANY VARIABLES (2-20)?    4

MAX DIGITS PER ENTRY (1-15)?  3

NAME THE VARIABLE BELOW, WITH 1-6 CHAR:
V(1) = TOOL$
V(2) = WPN WT
V(3) = SPPEED
V(4) = LOAD
- - - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

- Enter data as we did in Problem 1. The data field width is small enough to permit all four variables on the same screen page. Simply enter each value in the appropriate variable solumn, treating each of the 14 historical weapon systems as a separate observation:

- Add data until the observation counter reaches 15, then hit {RETURN} to leave the entry mode.

```
ENTER THE VALUE FOR EACH OBSERVATION AND
VARIABLE;  HIT 'RETURN' AFTER EACH ENTRY
TO STOP INPUT, HIT 'RETURN' W/O ENTRY.
- - - - - - - - - - - - - - - - - - - - - - -

      TOOL$   WPN WT  SPEED   LOAD
      V(1)    V(2)    V(3)    V(4)
      ----    ----    ----    ----

1     28      27      43      1.8

2     35      28      48      2

3     40      29      50      1.5

13    40      40      40      7

14    185     110     45      6.8

15

DATA ENTRY STOPPED...
    TOTAL OF 14 COMPLETE OBSERVATIONS.
- - - - - - - - - - - - - - - - - - - - - - -
         HIT 'RETURN' TO PROCEED...
```

- The Master Menu now appears:

```
     DATA MANAGEMENT
     ---- ----------

1.   DISPLAY DATABASE

2.   EDIT DATABASE

3.   SAVE DATABASE TO DISK

4.   ENTER ANOTHER DATABASE

5.   QUIT PROGRAM

6.   RUN REGRESSION ANALYSIS

            WHICH OPTION?  1
```

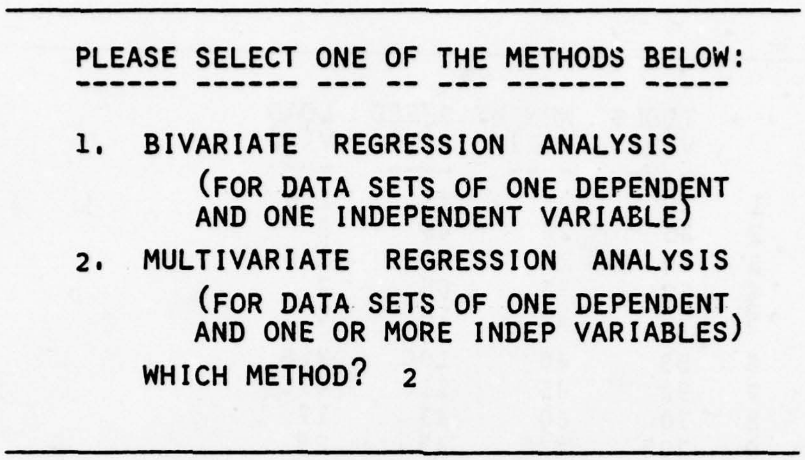
- Select option 1 to get a listing of the database:

```
        TOOL$   WPN WT  SPEED   LOAD
        V(1)    V(2)    V(3)    V(4)
        ----    ----    ----    ----
    1   28      27      43      1.8
    2   35      28      48      2
    3   40      29      50      1.5
    4   60      35      65      2
    5   50      32      70      5

    6   55      40      100     2.6
    7   90      45      110     4.5
    8   70      60      43      17
    9   285     135     45      29
    10  130     70      100     21

    11  105     90      43      9
    12  80      70      23      9
    13  40      40      40      7
    14  185     110     45      6.8
    - - - - - - - - - - - - - - - - - - - - - - - -
            HIT 'RETURN' TO PROCEED...
```

- There are no errors in the database, so upon return to the Master menu, we can elect to save the database to disk:

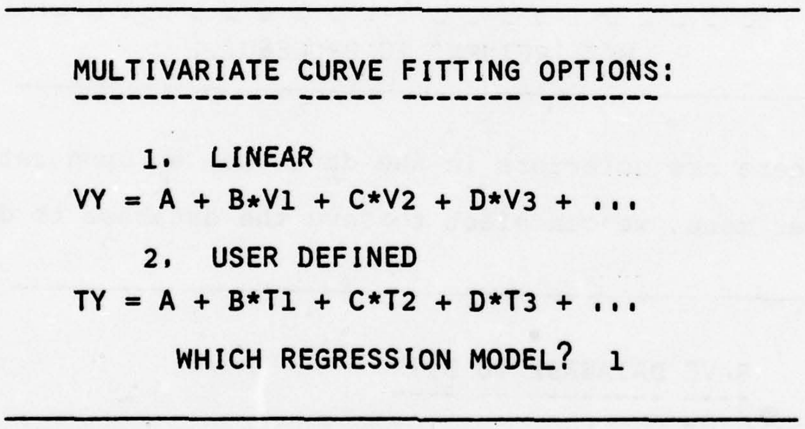
```
    SAVE DATABASE TO DISK
    ---- -------- -- ----

    SAVE UNDER WHAT FILE NAME?  MULDEMO
```

- Upon returning again to the Master Menu and selecting the Regression Analysis option, we choose Multivariate Regression:



```
PLEASE SELECT ONE OF THE METHODS BELOW:
------ ------ --- -- --- ------- -----

1. BIVARIATE  REGRESSION  ANALYSIS
      (FOR DATA SETS OF ONE DEPENDENT
      AND ONE INDEPENDENT VARIABLE)
2. MULTIVARIATE  REGRESSION  ANALYSIS
      (FOR DATA SETS OF ONE DEPENDENT
      AND ONE OR MORE INDEP VARIABLES)
   WHICH METHOD? 2
```

- Program MULVAR REGR now automatically runs and presents the following:



```
MULTIVARIATE CURVE FITTING OPTIONS:
------------ ----- ------- -------

    1. LINEAR
VY = A + B*V1 + C*V2 + D*V3 + ...
    2. USER DEFINED
TY = A + B*T1 + C*T2 + D*T3 + ...
      WHICH REGRESSION MODEL? 1
```

- Since we currently have no basis for using a non-linear model, we select option 1.

- The list of current variables soon appears and we must identify the independent and dependent variables.

- Note that we do not have to use all of the variables. We could choose to run different regressions using different combinations of variables.

```
CURRENT VARIABLES:
------- ---------

INDEX LABEL  NAME
  1   V(1)   TOOL$
  2   V(2)   WPN WT
  3   V(3)   SPEED
  4   V(4)   LOAD
- - - - - - - - - - - - - - - - - - - - -

YOU DON'T HAVE TO USE ALL THE VARIABLES.
SELECT ONLY THOSE NEEDED FOR THIS FIT...

HOW MANY INDEPENDENT VARIABLES?  3
  INDEX OF INDEP VARIABLE 1?  2
  INDEX OF INDEP VARIABLE 2?  3
  INDEX OF INDEP VARIABLE 3?  4
INDEX OF DEPENDENT VARIABLE?  1

** DO YOU WANT TO MAKE CHANGES?  N
```

- Next we are offered various output options:

```
    OUTPUT OPTIONS  (Y/N)
    ------ -------

1.  OUTPUT TO PRINTER?                 N

2.  TABLE OF MEANS/ STD DEVIATIONS?    Y

3.  SIMPLE CORRELATION MATRIX?         Y

4.  VARIANCE-CORRELATION MATRIX?       Y

5.  TABLE OF RESIDUALS?                Y

- - - - - - - - - - - - - - - - - - - -
          NEED TO CHANGE IT?  N
```

- All five options appear simultaneously, then the cursor automatically positions itself for a "Y/N" response at each one. Again, a simple {RETURN} defaults to "N".

- Since we did select Means and Standard Deviations, we will shortly see:

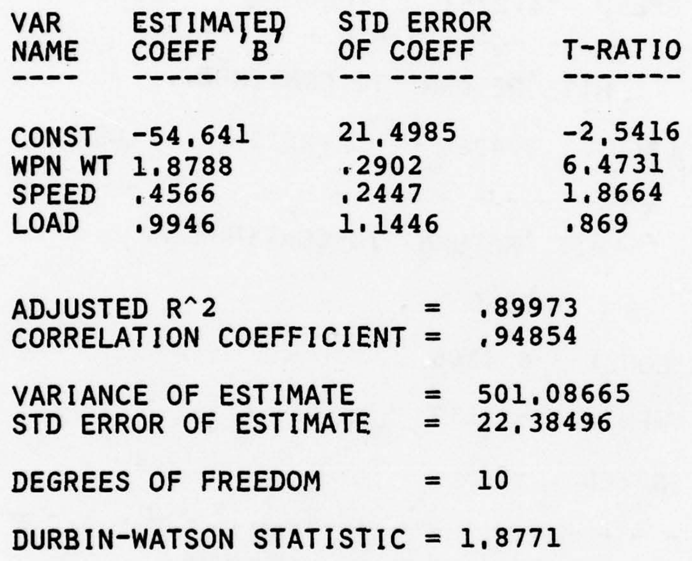
```
VAR            UNBIASED    UNBIASED
NAME    MEAN     STD DEV     VARIANCE
----    ----     --- ---     --------

WPN WT 57.9286  33.5959     1128.6868
SPEED  58.9286  26.514      702.9945
LOAD   8.4429   8.301       68.9073
TOOL$  89.5     70.6919     4997.3462
- - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO CONTINUE...
```

- We also requested the Simple Correlation Matrix, so we now get:

```
CORRELATION COEFFICIENTS
----------- ------------

       WPN WT   SPEED    LOAD    TOOL$
WPN WT 1        -.232    .7476   .9405
SPEED  -.232    1        -.058   -.0426
LOAD   .7476    -.058    1       .7744
TOOL$  .9405    -.0426   .7744   1
- - - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO CONTINUE...
```

- Note that four variables fill the screen; if we use more than four variables, the matrix is displayed in sequential blocks of four.

- Next the program calculates and automatically displays the regression coefficients and basic statistics:

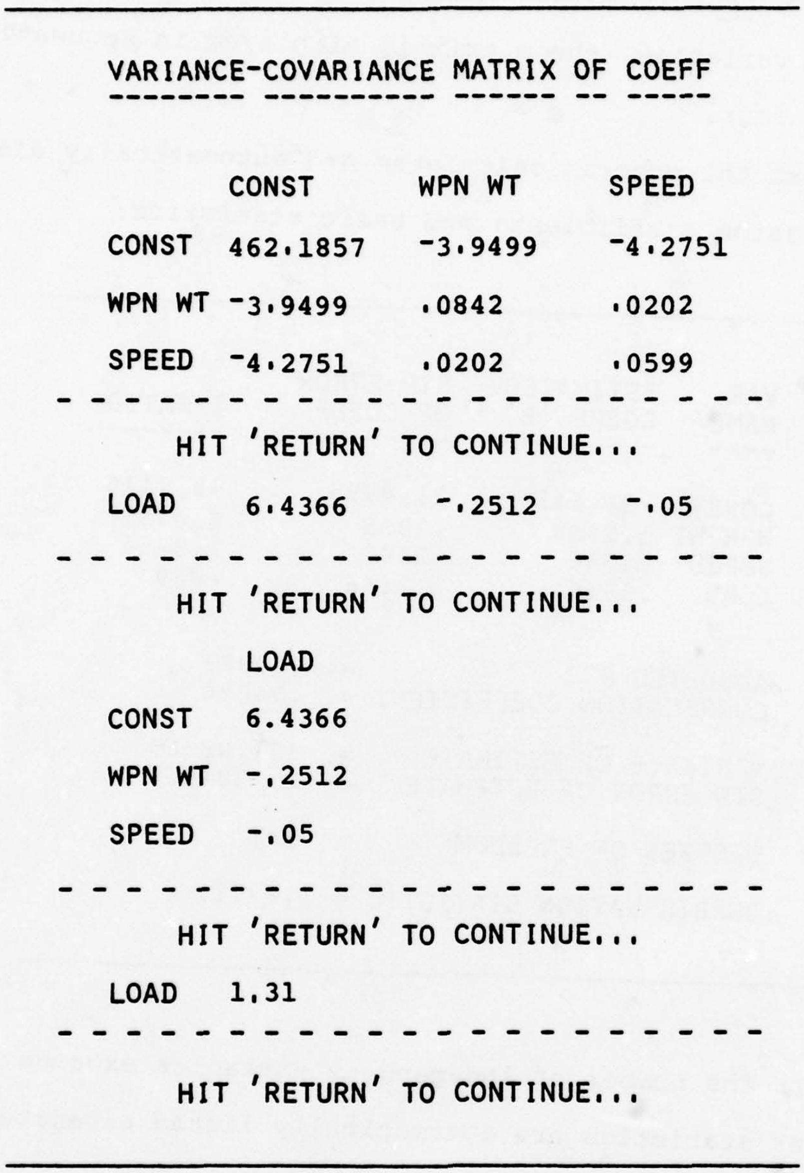


```
VAR     ESTIMATED    STD ERROR
NAME    COEFF 'B'    OF COEFF      T-RATIO
----    ----- ---    -- -----      -------

CONST   -54.641      21.4985       -2.5416
WPN WT  1.8788       .2902         6.4731
SPEED   .4566        .2447         1.8664
LOAD    .9946        1.1446        .869

ADJUSTED R^2              =  .89973
CORRELATION COEFFICIENT =  .94854

VARIANCE OF ESTIMATE      =  501.08665
STD ERROR OF ESTIMATE     =  22.38496

DEGREES OF FREEDOM        =  10

DURBIN-WATSON STATISTIC = 1.8771
```

- If the number of independent variables exceeds four, the lower statistics are automatically listed separately, following the model coefficients.

- We had originally requested the Variance-Covariance Matrix of Coefficients, so we now get the Var-Covar Matrix.

- The matrix is paged in sequential blocks because we have more than three total variables.



```
VARIANCE-COVARIANCE MATRIX OF COEFF
-------- ---------- ------ -- -----

          CONST        WPN WT       SPEED

CONST     462.1857     ¯3.9499      ¯4.2751

WPN WT    ¯3.9499      .0842        .0202

SPEED     ¯4.2751      .0202        .0599
- - - - - - - - - - - - - - - - - - - - - -
     HIT 'RETURN' TO CONTINUE...

LOAD        6.4366      ¯.2512        ¯.05
- - - - - - - - - - - - - - - - - - - - - -
     HIT 'RETURN' TO CONTINUE...

            LOAD

CONST       6.4366

WPN WT      -.2512

SPEED       -.05
- - - - - - - - - - - - - - - - - - - - - -
     HIT 'RETURN' TO CONTINUE...

LOAD     1.31
- - - - - - - - - - - - - - - - - - - - - -
     HIT 'RETURN' TO CONTINUE...
```

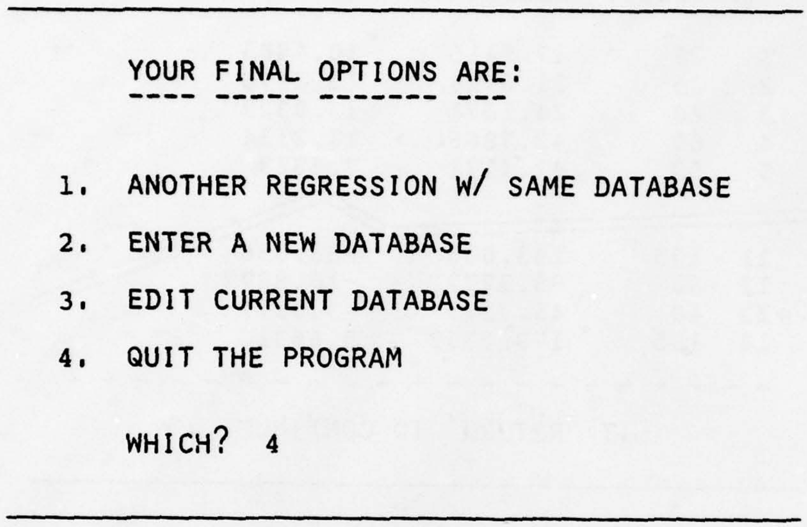
- We also had requested the Table of Residuals, which appears on the following page.

```
     ACTUAL  PREDICTED
      'Y'       'Y'       RESIDUAL
     ------  ---------    --------

 1   28      17.5115      10.4885
 2   35      21.8724      13.1276
 3   40      24.1671      15.8329
 4   60      42.7866      17.2134
 5   50      42.4171      7.5829

 11  105     143.036      -38.036
 12  80      96.3277      -16.3277
 13  40      45.7377      -5.7377
 14  185     179.3369     5.6631
 - - - - - - - - - - - - - - - - - - - - - -
           HIT 'RETURN' TO CONTINUE...
```

- At this point we have all the primary information we need, but if we want to predict the cost of initial tooling given our original design specifications, we must "predict values of TOOL$:"

```
 DO YOU WANT TO PREDICT VALUES FOR
     TOOL$ ?  Y


 ENTER VALUES FOR INDEPENDENT VARIABLES:

 V(2)  WPN WT  100
 V(3)  SPEED   65
 V(4)  LOAD    7

 PREDICTED VALUE OF TOOL$ = 169.8805

 - - - - - - - - - - - - - - - - - - - - - -
          ANOTHER PREDICTION?  N
```

- This concludes the analysis, and we now reach the final set of options:



```
        YOUR FINAL OPTIONS ARE:
        ---- ----- ------- ---

    1.  ANOTHER REGRESSION W/ SAME DATABASE

    2.  ENTER A NEW DATABASE

    3.  EDIT CURRENT DATABASE

    4.  QUIT THE PROGRAM

        WHICH? 4
```

- NOTE: the user-defined model option functions precisely the same in MULVAR REGR as it does in BIVAR REGR.

Problem 3, Linear Programming

An experimental Army infantry unit is to be equipped with a fleet of new personnel/cargo vehicles. The fleet must be capable of carrying at least 2000 troops and 200 tons of supplies. Additionally the fleet is limited to not more than 320 vehicles.

Three new vehicles are being considered, and the fleet can be mixed. Capability and cost figures appear in Table V below:

TABLE V

New Vehicle Capabilities and Costs

VEHICLE	TROOPS	SUPPLIES	COST
Vehicle 1	5	1 ton	$ 12,000
Vehicle 2	8	.5 ton	$ 9,000
Vehicle 3	4	1 ton	$ 10,000

As a procurement analyst, you are asked to determine the mix of vehicles that will satisfy transport requirements at the minimum cost (Ref 11).

We can formulate the problem as follows:

Decision Variables:

$X(1) = VEH1$ = number of vehicle 1's to procure

$X(2) = VEH2$ = number of vehicle 2's to procure

$X(3) = VEH3$ = number of vehicle 3's to procure

Objective Function:

Minimize $Cost = 12000VEH1 + 9000VEH2 + 10000VEH3$

Subject To: The Constraints:

$5VEH1 + 8VEH2 + 4VEH3 \geq 2000$ Troops

$1VEH1 + .5VEH2 + 1VEH3 \geq 200$ Tons of Cargo

$1VEH1 + 1VEH2 + 1VEH3 \leq 320$ Total Fleet

To analyze the problem using CAAMM software, select Linear Programming on the CAAMM MASTER Menu. Program LINPROG will subsequently run and present:

```
      * * * * * * * * * * * * * * * * * * * * *
                      L I N E A R
                P R O G R A M M I N G
                          BY
                   ROBERT D. CONTE
      * * * * * * * * * * * * * * * * * * * * *
         DO YOU WANT INTRODUCTORY REMARKS?  Y
```

```
      LINEAR PROGRAMMING IS USED TO DETERMINE
      THE BEST ALLOCATION FOR SCARCE RESOURCES
      IN ORDER TO
                MAXIMIZE OR MINIMIZE
      A LINEAR FUNCTION DEFINING AN OBJECTIVE
      SUCH AS PROFIT, COST, TONNAGE, ETC.
      - - - - - - - - - - - - - - - - - - - -
      HIT 'RETURN' TO GO ON, OR 'Q' TO QUITE
```

```
         TO ENTER AN LP MODEL, YOU CAN:
         -- ----- -- -- -----  --- ---

      1.  READ EXISTING MODEL FROM DISK
      2.  CREATE MODEL INTERACTIVELY
             (USING NAMED VARIABLES)
      3.  CREATE MODEL INTERACTIVELY
             (USING NUMBERS ONLY)

                  WHICH METHOD?  2
```

Select option 2 to enter the model interactively with variable names, then select the type of optimization and identify the objective to be optimized:

```
        TO SOLVE THE PROBLEM, YOU CAN:
        -- ----- --- -------  --- ---

        1.  MAXIMIZE THE OBJ FUNCTION
    OR  2.  MINIMIZE THE OBJ FUNCTION

          WHICH? 2
```

```
    WHAT OBJECTIVE DO YOU WANT TO MINIMIZE?
    (E.G.  CASUALTIES, COST, MATERIEL)
          COST
```

- Now enter the names of the decision variables, remembering to simply hit {RETURN} when you finish. Use the "Changes?" option if necessary.

```
    LIST THE VARIABLES (MAX 20) THAT AFFECT
    COST, USING 1-6 CHAR DESCRIPTORS:

    TO STOP INPUT, HIT 'RETURN' W/O ENTRY.
    - - - - - - - - - - - - - - - - - - - -
    X(1) = VEH 1
    X(2) = VEH 2
    X(3) = VEH 3
    X(4) =

    - - - - - - - - - - - - - - - - - - - -
            NEED TO MAKE CHANGES?  N
```

- Next enter the names of the constraints in the same manner:

```
LIST CONSTRAINTS (MAX 20) THAT AFFECT
COST, USING 1-6 CHAR DESCRIPTORS

TO STOP INPUT, HIT 'RETURN' W/O ENTRY.
- - - - - - - - - - - - - - - - - - - - - -
C(1)  = MEN
C(2)  = TONS
C(3)  = FLEET
C(4)  =

- - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?
```

- Now enter the Objective Function coefficients, directly under the variable names as they appear, row-by-row:

```
OBJECTIVE FUNCTION - MINIMIZE COST  ...

VEH 1  VEH 2  VEH 3
12000  9000   10000

- - - - - - - - - - - - - - - - - - - - - -
       NEED TO MAKE CHANGES?  N
```

- Then enter the Constraint coefficients in similar fashion. Note that non-negativity constraints are assumed, and the sequence of entering constraints makes no difference, i.e. "less-than" constraints dc not necessarily have to precede "greater-than" or "equality" constraints. The inequality symbol you enter can be:

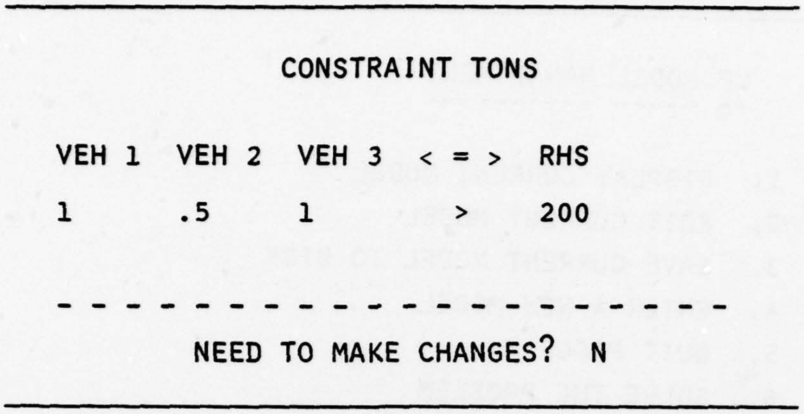
<, <=, LT, or LE	acceptable for "less than"
= or EQ	acceptable for "equality"
>, >=, GT, or GE	acceptable for "greater than"

Constraints with a negative Right Hand Side value are automatically reversed.

- Constraint entries:

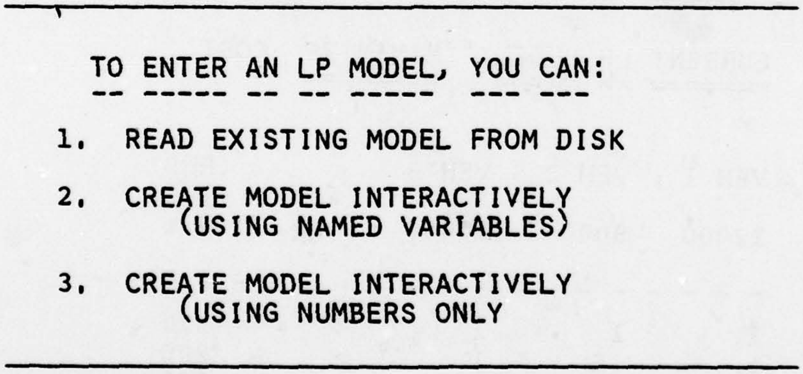
```
                CONSTRAINT TROOPS

VEH 1   VEH 2   VEH 3   < = >   RHS
5       8       4         >     2000

- - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

```
                CONSTRAINT FLEET

VEH 1   VEH 2   VEH 3   < = >   RHS
1       1       1         <     320

- - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```



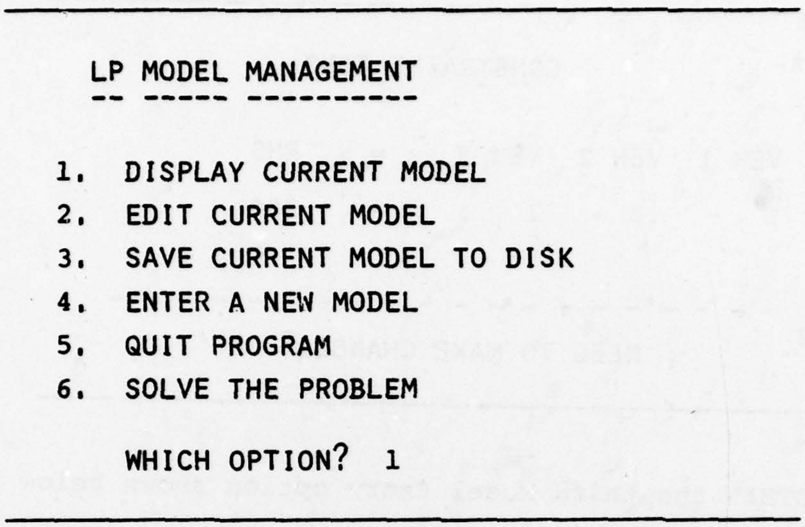
```
              CONSTRAINT TONS

    VEH 1  VEH 2  VEH 3  < = >  RHS

    1      .5     1        >    200

    - - - - - - - - - - - - - - - - - - - - - -

          NEED TO MAKE CHANGES? N
```

- NOTE: the third Model Entry option shown below is almost identical in form to the procedure just described for option 2, but it is faster because variable and constraint names are not required.



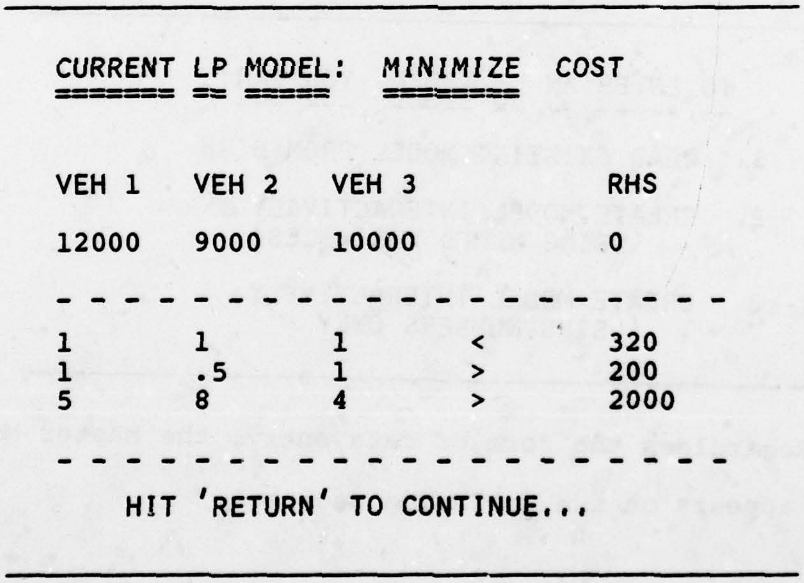
```
     TO ENTER AN LP MODEL, YOU CAN:
     -- ----- -- -- -----  --- ---
  1.  READ EXISTING MODEL FROM DISK
  2.  CREATE MODEL INTERACTIVELY
         (USING NAMED VARIABLES)
  3.  CREATE MODEL INTERACTIVELY
         (USING NUMBERS ONLY
```

- Regardless the form of data entry, the Master Menu finally appears on the following page.



```
        LP MODEL MANAGEMENT
        -- ----- ----------

     1.  DISPLAY CURRENT MODEL
     2.  EDIT CURRENT MODEL
     3.  SAVE CURRENT MODEL TO DISK
     4.  ENTER A NEW MODEL
     5.  QUIT PROGRAM
     6.  SOLVE THE PROBLEM

         WHICH OPTION? 1
```

- The Display option lets you check the model. Output to the printer is optional.



```
     CURRENT LP MODEL:  MINIMIZE  COST
     ======= == =====   ========

     VEH 1   VEH 2   VEH 3           RHS

     12000   9000    10000           0

     - - - - - - - - - - - - - - - - - - - - -

     1       1       1       <       320
     1       .5      1       >       200
     5       8       4       >       2000

     - - - - - - - - - - - - - - - - - - - - -

         HIT 'RETURN' TO CONTINUE...
```

- The Model Editing functions are extensive:

```
     LP MODEL EDIT FUNCTIONS:
     -- ----- ---- ---------


   1.  DELETE A VARIABLE
   2.  DELETE A CONSTRAINT

   3.  ADD A VARIABLE
   4.  ADD A CONSTRAINT

   5.  CHANGE COEFFICIENTS BY VARIABLE
   6.  CHANGE COEFFICIENTS BY CONSTRAINT
   7.  CHANGE INDIVIDUAL COEFFICIENT

   8.  CHANGE RHS VALUES
   9.  CHANGE OBJECTIVE FUNCTION

   10. RETURN TO LAST MENU

       WHICH FUNCTION?
```

- If you were to select option 6, for example, you could selectively change any element in any constraint:

```
   CHANGE COEFFICIENTS BY CONSTRAINT
   ------ ------------ -- ----------

   1  FLEET
   2  TONS
   3  TROOPS
   - - - - - - - - - - - - - - - - - - - - - - -

   WHICH CONSTRAINT (1-3)?  3
```

- As the cursor positions itself at each element, enter a new value or hit {RETURN} to default to the current value.

```
                  TROOPS
                  ------

VARIABLE      CURRENT    CHANGE TO
--------      -------    ------ --

1   VEH 1     5          8
2   VEH 2     8
3   VEH 3     4

INEQUALITY    >

RHS           2000
- - - - - - - - - - - - - - - - - - - -

          NEED TO MAKE CHANGES?  N
```

- Once editing is completed, you can re-display the model, save it to disk, enter a new model, or proceed to solve the problem. If you elect to solve the problem now, you will have various output options. As in program MULVAR REGR, they appear simultaneouly:

```
   OUTPUT OPTIONS (Y/N)
   ------ -------

1.  OUTPUT TO PRINTER?                N

2.  INITIAL TABLEAU?                  Y

3.  INTERMEDIATE BASIC SOLUTIONS?     Y

4.  FINAL TABLEAU?                    Y

- - - - - - - - - - - - - - - - - - - -

          NEED TO MAKE CHANGES?  N
```

- Next the simplex variable table is displayed:

```
                 VARIABLE TABLE:
                 ======== ======

DECISION VARIABLES ARE       1-3

1  VEH 1
2  VEH 2
3  VEH 3
SURPLUS VARIABLES ARE        3-5
SLACK VARIABLES ARE          6-6
ARTIFICIAL VARIABLES ARE     7-8
- - - - - - - - - - - - - - - - - - - - - - - - -
          HIT 'RETURN' TO CONTINUE...
```

- We originally requested the Intial Simplex Tableau, so it appears below, displaying in sequential pages:

```
                 INITIAL TABLEAU
                 ======= =======

X(1)      X(2)      X(3)      X(4)      X(5)
-6        -8.5      -5        1         1
- - - - - - - - - - - - - - - - - - - - - - - - -
1         1         1         0         0
1         .5        1         -1        0
5         8         4         0         -1
- - - - - - - - - - - - - - - - - - - - - - - - -
          HIT 'RETURN' TO CONTINUE...
```

```
X(6)     X(7)     X(8)     RHS

0        0        0        -2200
- - - - - - - - - - - - - - - - - - - - - -
1        0        0        320
0        1        0        200
0        0        1        2000
- - - - - - - - - - - - - - - - - - - - - -
         HIT 'RETURN' TO CONTINUE...
```

- We requested Intermediate Solutions, so they appear sequentially, from iteration to iteration, until the optimal solution is found (which automatically displays whether you select Intermediate Solutions or not):

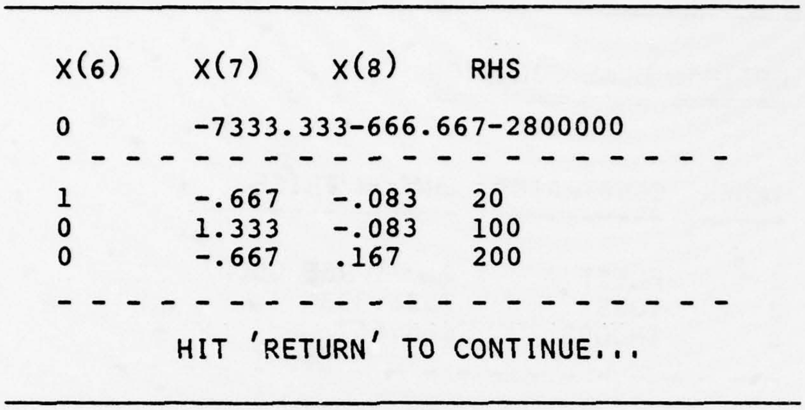
```
INITIAL BASIC FEASIBLE SOLUTION
======= ===== ======== ========

INDEX  VARIABLE  VALUE
-----  --------  -----

  2    VEH 2     200
  3    VEH 3     100
  6              20

** COST =  2800000

*** THIS SOLUTION IS MINIMAL ***
- - - - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO CONTINUE...
```

- That is followed by the Optimal Dual Solution:

```
OPTIMAL DUAL SOLUTION
======= ==== ========

INDEX   CONSTRAINT   SHADOW PRICE
-----   ----------   ------ -----

1       FLEET        0    (FREE GOOD)
2       TONS         7333.333
3       TROOPS       666.667
- - - - - - - - - - - - - - - - - - - - - - -
        HIT 'RETURN' TO CONTINUE...
```

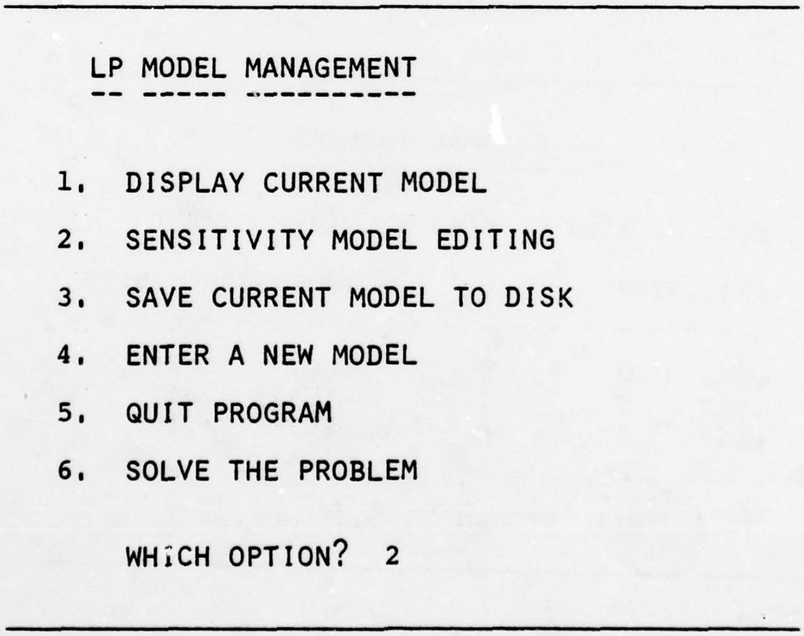
- Following the solutions, the actual Optimal Tableau appears, displaying in sequential blocks:

```
             OPTIMAL TABLEAU
             ------- -------

X(1)     X(2)     X(3)     X(4)     X(5)

1333.3330         0        7333.333666.667
- - - - - - - - - - - - - - - - - - - - - - -

.083     0        0        .667     .083
.917     0        1        -1.333   .083
.167     1        0        .667     -.167
- - - - - - - - - - - - - - - - - - - - - - -
        HIT 'RETURN TO CONTINUE...
```



```
X(6)    X(7)    X(8)    RHS

0       -7333.333-666.667-2800000
- - - - - - - - - - - - - - - - - - - -
1       -.667   -.083   20
0       1.333   -.083   100
0       -.667   .167    200
- - - - - - - - - - - - - - - - - - - -
        HIT 'RETURN' TO CONTINUE...
```

- Finally the program returns to the Master Menu where it offers:

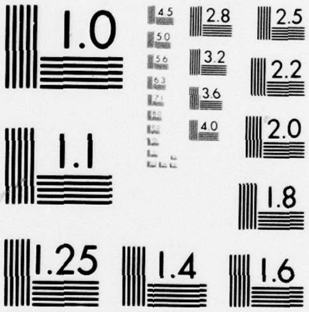


```
    LP MODEL MANAGEMENT
    -- ----- ----------

1.  DISPLAY CURRENT MODEL

2.  SENSITIVITY MODEL EDITING

3.  SAVE CURRENT MODEL TO DISK

4.  ENTER A NEW MODEL

5.  QUIT PROGRAM

6.  SOLVE THE PROBLEM

    WHICH OPTION?  2
```

- The Sensitivity Model Editing option does not provide true sensitivity analysis, but it accomplishes the same objective by providing the full range of normal Edit functions to

AD-A080 215 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/4
COMPUTER ASSISTED ANALYSIS FOR MILITARY MANAGERS. (U)
DEC 79 R D CONTE
UNCLASSIFIED AFIT/GOR/MA/79D-3 NL

2 OF 3
AD
A080215



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

evaluate the effect of changing, adding, or deleting parameters:

```
     LP MODEL EDIT FUNCTIONS:
     -- ----- ---- ---------

  1.  DELETE A VARIABLE
  2.  DELETE A CONSTRAINT

  3.  ADD A VARIABLE
  4.  ADD A CONSTRAINT

  5.  CHANGE COEFFICIENTS BY VARIABLE
  6.  CHANGE COEFFICIENTS BY CONSTRAINT
  7.  CHANGE INDIVIDUAL COEFFICIENT

  8.  CHANGE RHS VALUES
  9.  CHANGE OBJECTIVE FUNCTION

 10.  RETURN TO LAST MENU

      WHICH FUNCTION?
```

Problem 4, Value Matrix Decision Aid

American forces in Europe are constantly faced by the threat of a Warsaw Pact attack. Of continual concern are decisions pertaining to alert status. Placing troops at any level of alert costs money, whether the Pact attacks or not. Certainly there is political cost as well--both nationally and internationally. But obviously the military risk is of paramount importance--is it not?

Unfortunately, intelligence cannot state with certainty that the Pact is planning an attack--or is not planning an attack. The result is a highly subjective decision environment which we must try to structure and quantify in order to logically evaluate and compáre alternatives.

There are many sophisticated forms of decision analysis and complex simulation, but often the decision maker simply needs a quick aid to help "sort out" the alternatives and other considerations. With this objective, perhaps the Value Matrix Decision Aid as implemented in CAAMM program DECISION could be of use. Since the Decision Aid accommodates only the three major decision components--alternatives, judgement criteria, and risk--the first step must be to simplify the decision environment to a set of these three discrete parameters. Table VI (Ref 16) provides a reasonable, but certainly not the only approach to structuring the problem.

TABLE VI

Subjective Decision Situation

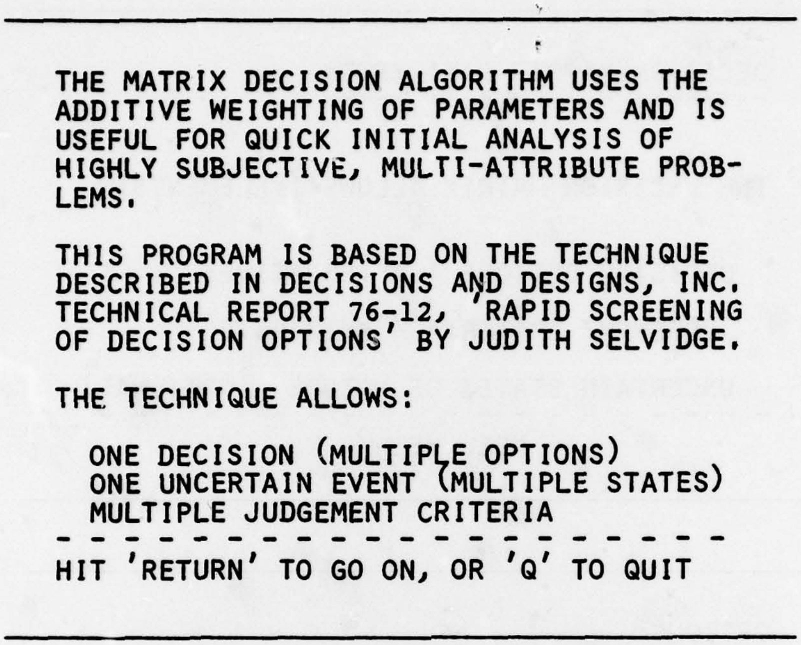
DECISION OPTIONS:	Maintain Status Quo
	Military Vigilance
	Simple Alert
	Reinforced Alert
JUDGEMENT CRITERIA:	Alert Cost
	Political Cost
	Military Risk
UNCERTAIN STATES OF NATURE:	Pact Planning Attack?
	Pact Not Planning Attack?

To analyze the problem using CAAMM software, select Matrix Decision Aid on the CAAMM MASTER Menu. Program DECISION will subsequently run and present:

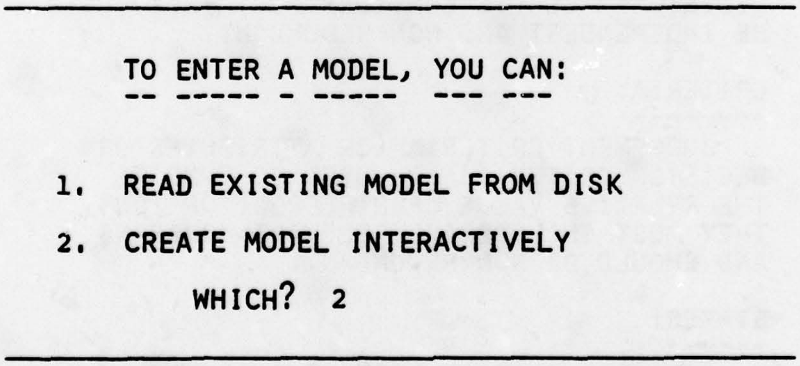
```
* * * * * * * * * * * * * * * * * * * *

          V A L U E   M A T R I X

              D E C I S I O N

              A N A L Y S I S

                    BY

              ROBERT D. CONTE

* * * * * * * * * * * * * * * * * * * *

  DO YOU WANT INTRODUCTORY REMARKS?  Y
```

```
         MATRIX DECISION ANALYSIS
         ------ -------- --------

DECISION ANALYSIS PROVIDES A STRUCTURED
APPROACH TO COMPARING THE RELATIVE MERIT
OF VARIOUS ALTERNATIVES OR OPTIONS.

REGARDLESS WHETHER AN ACTUAL DECISION IS
MADE, THE ANALYSIS CAN ILLUMINATE PARTS
OF THE PROBLEM PREVIOUSLY UNCONSIDERED
AND CAN PROVIDE THE BASIS FOR ELABORATE
SUBSEQUENT STUDIES.

AMONG THE VARIOUS FORMS OF QUANTITATIVE
DECISION ANALYSIS ARE:

   DECISION TREES
   COST/BENEFIT ANALYSIS
   UTILITY THEORY
   PROCESS DIAGRAMS
- - - - - - - - - - - - - - - - - - - -

HIT 'RETURN' TO GO ON, OR 'Q' TO QUIT
```

And continues:



```
THE MATRIX DECISION ALGORITHM USES THE
ADDITIVE WEIGHTING OF PARAMETERS AND IS
USEFUL FOR QUICK INITIAL ANALYSIS OF
HIGHLY SUBJECTIVE, MULTI-ATTRIBUTE PROB-
LEMS.

THIS PROGRAM IS BASED ON THE TECHNIQUE
DESCRIBED IN DECISIONS AND DESIGNS, INC.
TECHNICAL REPORT 76-12, 'RAPID SCREENING
OF DECISION OPTIONS' BY JUDITH SELVIDGE.

THE TECHNIQUE ALLOWS:

  ONE DECISION (MULTIPLE OPTIONS)
  ONE UNCERTAIN EVENT (MULTIPLE STATES)
  MULTIPLE JUDGEMENT CRITERIA
- - - - - - - - - - - - - - - - - - - - -
HIT 'RETURN' TO GO ON, OR 'Q' TO QUIT
```



```
    TO ENTER A MODEL, YOU CAN:
    -- ----- - -----  --- ---

1.  READ EXISTING MODEL FROM DISK

2.  CREATE MODEL INTERACTIVELY

        WHICH?  2
```

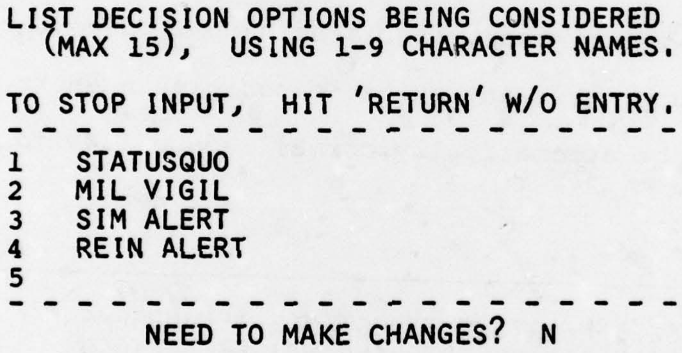
- Select option 2 for interactive entry.

- The section is introduced by:

```
DECISION MATRIX DATA ENTRY
-------- ------ ---- -----

THE DECISION MATRIX ALLOWS 3 ELEMENTS:

  DECISION OPTIONS / ALTERNATIVES

  JUDGEMENT CRITERIA / ATTRIBUTES

  UNCERTAIN STATES OF NATURE  (OPTIONAL)
- - - - - - - - - - - - - - - - - - - -
               NEED HELP?  Y
```

```
OPTIONS:
-------
   THE OPTIONS OR ALTERNATIVES ARE THE
DIFFERENT POSSIBLE COURSES OF ACTION THE
DECISION-MAKER IS CONSIDERING.  THEY MUST
BE INDEPENDENT AND NON-REDUNDANT.

CRITERIA:
--------
   JUDGEMENT CRITERIA (OR ATTRIBUTES OR
DECISION CRITERIA) ARE USED TO COMPARE
THE RELATIVE VALUE OF DIFFERENT OPTIONS,
THEY MUST INCLUDE ALL RELEVANT CONCERNS
AND SHOULD BE NON-REDUNDANT.

STATES:
------
   STATES OF NATURE REFER TO THE SET OF
UNCERTAIN OUTCOMES THAT CAN RESULT FROM
AN EVENT OVER WHICH WE HAVE LITTLE OR
NO CONTROL.
- - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO PROCEED...
```

- Next enter the names of our Decision Options. As in the other programs, press {RETURN} without entry to leave the input mode. Remember that for results to be meaningful, the various Decision Options must be distinctly separate and should encompass all possibile alternatives. In more complex problems, several Options could be classified together, thereby reducing the total. If editing is necessary, it is easily accomplished by entering the index of the item to be changed.



```
LIST DECISION OPTIONS BEING CONSIDERED
  (MAX 15),  USING 1-9 CHARACTER NAMES.

TO STOP INPUT,  HIT 'RETURN' W/O ENTRY.
- - - - - - - - - - - - - - - - - - - - -
1   STATUSQUO
2   MIL VIGIL
3   SIM ALERT
4   REIN ALERT
5
- - - - - - - - - - - - - - - - - - - - -
        NEED TO MAKE CHANGES?  N
```

- Now enter the Judgement Criteria in similar fashion. Remember that the Judgement Criteria, or "attributes" as they are frequently called, are the primary means of discriminating among options. Consequently they must be comprehensive, non-redundant, and they must be capable of discriminating among Options.

```
LIST JUDGEMENT CRITERIA BEING CONSIDERED
 (MAX 20),  USING 1-8 CHARACTER NAMES.

TO STOP INPUT,  HIT 'RETURN' W/O ENTRY.
- - - - - - - - - - - - - - - - - - - - - - - -
1   ALERTCST
2   POL COST
3   MIL RISK
- - - - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

- Now we must assess the relative importance of the various Judgement Criteria. Do not simply provide an ordinal ranking but actually weight the criteria using paired-comparison or any other procedure. Use any suitable positive numbers, as the weights will be automatically normalized to 1.00 for the evaluation later.

```
ASSESS RELATIVE IMPORTANCE OF JUDGEMENT
 CRITERIA BY ENTERING WEIGHT FACTORS.
WEIGHTS WILL BE NORMALIZED AUTOMATICALLY
- - - - - - - - - - - - - - - - - - - - - - - -
1   ALERTCST 10
2   POL COST 30
3   MIL RISK 60

- - - - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

```
    ** NORMALIZED CRITERION WEIGHTS **
- - - - - - - - - - - - - - - - - - - - - - - -
1  ALERTCST .1
2  POL COST .3
3  MIL RIST .6
- - - - - - - - - - - - - - - - - - - - - - - -
         NEED TO READJUST VALUES?  N
```

- Once you see the normalized weights, you might decide to readjust the original weightings. The preceding text page will then reappear, complete with the current weights. To readjust figures, simply type over the old values. Once you are satisfied with the final weightings, press {RETURN} to continue.

- Next enter the Uncertain States of Nature which affect the problem:

```
IS YOUR DECISION AFFECTED BY UNCERTAIN
             STATES OF NATURE?  Y

LIST STATES OF NATURE BEING CONSIDERED
  (MAX 5),  USING 1-8 CHARACTER NAMES.

TO STOP INPUT,  HIT 'RETURN' W/O ENTRY.
- - - - - - - - - - - - - - - - - - - - - -
1   PACTATTCK
2   NO ATTACK
3
- - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

- Then enter the estimated likelihood or probability of each state in the same fashion as before:

```
ESTIMATE THE PROBABILITY OR LIKELIHOOD
  OF EACH OF THE UNCERTAIN OUTCOMES.

NUMBERS WILL BE NORMALIZED AUTOMATICALLY
- - - - - - - - - - - - - - - - - - - - - -
1   PACTATTC 10
2   NO ATTAC 90
- - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

```
    ** NORMALIZED STATE PROBABILITIES **
- - - - - - - - - - - - - - - - - - - - - -
1   PACTATTC .1
2   NO ATTAC .9
- - - - - - - - - - - - - - - - - - - - - -
       NEED TO READJUST VALUES?  N
```

- After the likelihoods are normalized and adjustments made, we must now assess values for the actual value matrices:

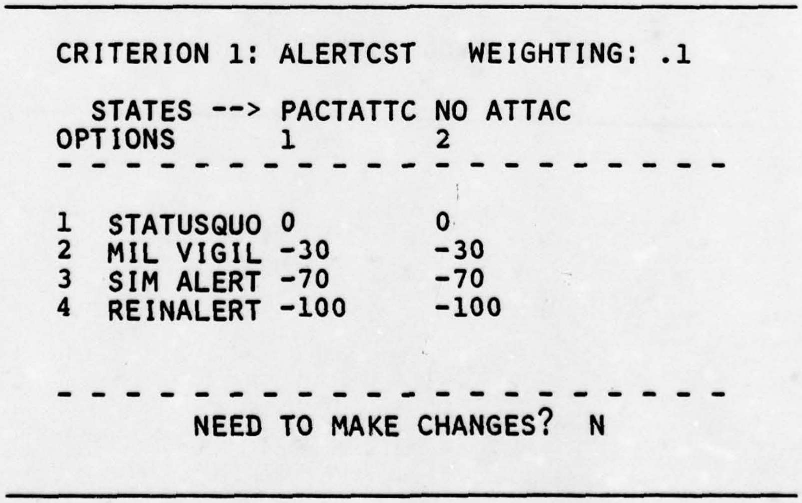
```
         ** VALUE MATRIX DATA ENTRY **

   TO USE THE VALUE MATRIX, YOU MUST
QUANTIFY YOUR SUBJECTIVE ESTIMATE OF THE
VALUE OF EACH OPTION, RELATIVE TO EACH
JUDGEMENT CRITERION AND STATE OF NATURE.

   VALUES MUST BE CORRELATED AMONG THE
OPTIONS, CRITERIA, AND STATES OF NATURE
FOR RESULTS TO BE MEANINGFUL, SO TAKE
ADVANTAGE OF THE OPTIONS FOR CHANGING
INPUT DATA.

   VALUES MAY BE POSITIVE OR NEGATIVE,
THE RECOMMENDED SCALES BEING  -100 TO 0
OR  0 TO 100.  LATER YOU WILL HAVE THE
OPTION TO MAXIMIZE OR MINIMIZE.
- - - - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO PROCEED...
```

- Note that we must assume total independence among the various options, states, and judgement critera.

- The composition of value matrices must be deliberate and iterative. Take full advantage of local editing options, which--as before--are screen position oriented to preclude re-entering the entire matrix. The object is to subjectively evaluate each option in the context of a particular State of

Nature and a specific Judgement Criterion. The type question that should stimulate evaluation is "what relative value can we place on the military risk if we remain at STATUS QUO and the Pact does plan to attack?" A reverse approach is the "regret" technique associated with "how much is the regret if we remain at STATUS QUO and the Pact plans to attack?" The regret approach is fully treated in Selvidge (Ref 16) and is illustrated in this problem.

- The evaluation must be consistent from State to State and Criterion to Criterion. Use any range of values, e.g. 0 to 100 or -100 to 0, but be consistent. The entire matrix framework is displayed on the screen prior to positioning the cursor for the first evaluation. Note that the criterion weighting is depicted on each page, along with indices of all parameters. Only three States of Nature will fit on the screen at one time, so paging is necessary. Now let us assess and enter the values for each matrix:



```
CRITERION 1: ALERTCST   WEIGHTING: .1

  STATES --> PACTATTC NO ATTAC
OPTIONS      1        2
- - - - - - - - - - - - - - - - - - - -

1  STATUSQUO 0        0
2  MIL VIGIL -30      -30
3  SIM ALERT -70      -70
4  REINALERT -100     -100

- - - - - - - - - - - - - - - - - - - -
        NEED TO MAKE CHANGES?  N
```

```
CRITERION 3: MIL RISK   WEIGHTING:  .6

  STATES --> PACTATTC NO ATTAC
OPTIONS      1        2
- - - - - - - - - - - - - - - - - - - - - -

1  STATUSQUO -100     0
2  MIL VIGIL -45      0
3  SIM ALERT -15      0
4  REINALERT 0        0

- - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

```
CRITERION 2: POL COST   WEIGHTING: .3

  STATES --> PACTATTC NO ATTAC
OPTIONS      1        2
- - - - - - - - - - - - - - - - - - - - - -

1  STATUSQUO -100     0
2  MIL VIGIL -35      -20
3  SIM ALERT -10      -50
4  REINALERT 0        -100

- - - - - - - - - - - - - - - - - - - - - -
          NEED TO MAKE CHANGES?  N
```

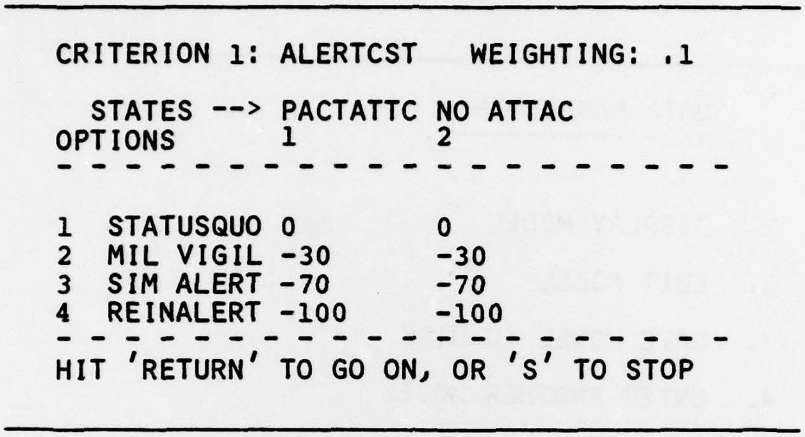
- Following completion of the value matrices, we arrive at the Master Menu (reached directly if we had loaded a model from disk):

```
DATA MANAGEMENT
---- ----------


1.  DISPLAY MODEL
2.  EDIT MODEL
3.  SAVE MODEL TO DISK
4.  ENTER ANOTHER MODEL
5.  CONTINUE PROGRAM
6.  QUIT PROGRAM

    WHICH OPTION? 1
```

- The Display Option offers:

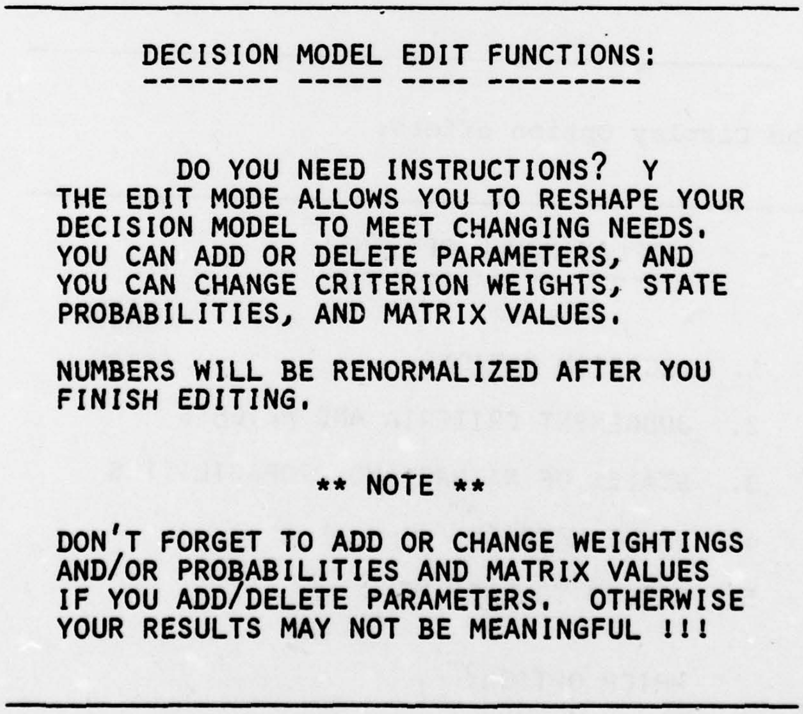
```
DISPLAY MODEL OPTIONS:
------- ----- -------


1.  DECISION OPTIONS
2.  JUDGEMENT CRITERIA AND WEIGHTS
3.  STATES OF NATURE AND PROBABILITIES
4.  VALUE MATRICS
5.  RETURN TO LAST MENU

    WHICH OPTION?
```

- Output can be routed to the printer, and Display option 4 results in:



```
CRITERION 1: ALERTCST   WEIGHTING: .1

  STATES --> PACTATTC NO ATTAC
OPTIONS      1        2
- - - - - - - - - - - - - - - - - - - -

1  STATUSQUO 0        0
2  MIL VIGIL -30      -30
3  SIM ALERT -70      -70
4  REINALERT -100     -100
- - - - - - - - - - - - - - - - - - - -
HIT 'RETURN' TO GO ON, OR 'S' TO STOP
```

- Returning to the Master Menu, we may select the Edit option:



```
     DECISION MODEL EDIT FUNCTIONS:
     -------- ----- ---- ---------


        DO YOU NEED INSTRUCTIONS?  Y
THE EDIT MODE ALLOWS YOU TO RESHAPE YOUR
DECISION MODEL TO MEET CHANGING NEEDS.
YOU CAN ADD OR DELETE PARAMETERS, AND
YOU CAN CHANGE CRITERION WEIGHTS, STATE
PROBABILITIES, AND MATRIX VALUES.

NUMBERS WILL BE RENORMALIZED AFTER YOU
FINISH EDITING.

              ** NOTE **

DON'T FORGET TO ADD OR CHANGE WEIGHTINGS
AND/OR PROBABILITIES AND MATRIX VALUES
IF YOU ADD/DELETE PARAMETERS.  OTHERWISE
YOUR RESULTS MAY NOT BE MEANINGFUL !!!
```

```
            DECISION MODEL EDIT FUNCTIONS:
            -------- ----- ---- ---------

    1.  DELETE ANY PARAMETER

    2.  ADD ANY PARAMETER

    3.  CHANGE AN OPTION

    4.  CHANGE CRITERION AND/OR WEIGHTING

    5.  CHANGE STATE AND/OR PROBABILITY

    6.  CHANGE VALUE MATRIX

    7.  RETURN TO LAST MENU

        WHICH?
```

- The Edit procedures are similar to the original data entry and local editing technique.

- If we elect to proceed to evaluation of the Decision Options, we soon get the option for optimization:

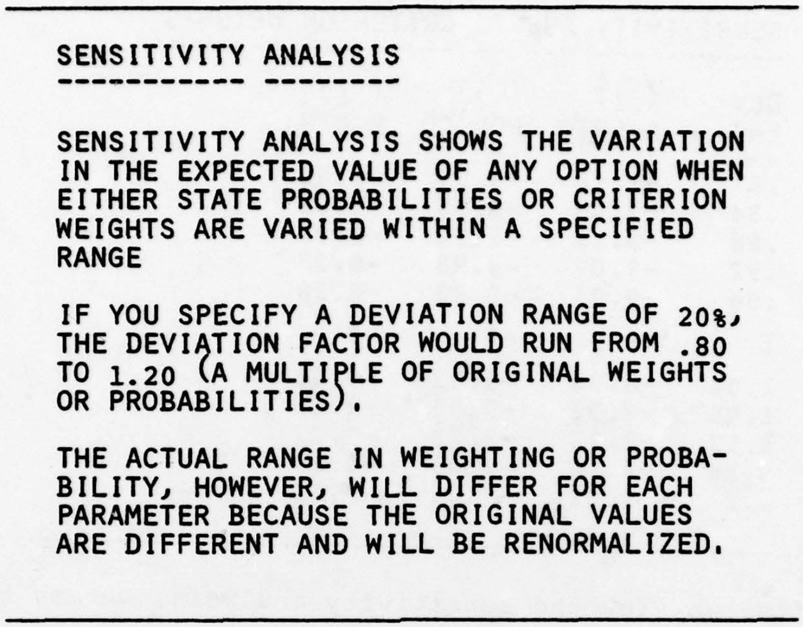
```
    TO DETERMINE THE OPTIMUM DECISION OPTION
    -- --------- --- ------- -------- ------

                   YOU CAN:
                   --- ---

        1.  MAXIMIZE THE DECISION VALUE

        2.  MINIMIZE THE DECISION VALUE

            WHICH? 1
```

- We should choose to maximize the evaluation, which is equivalent to minimizing "regret". The expected value of the "optimum" option is displayed, followed by the entire Expected Value Table:

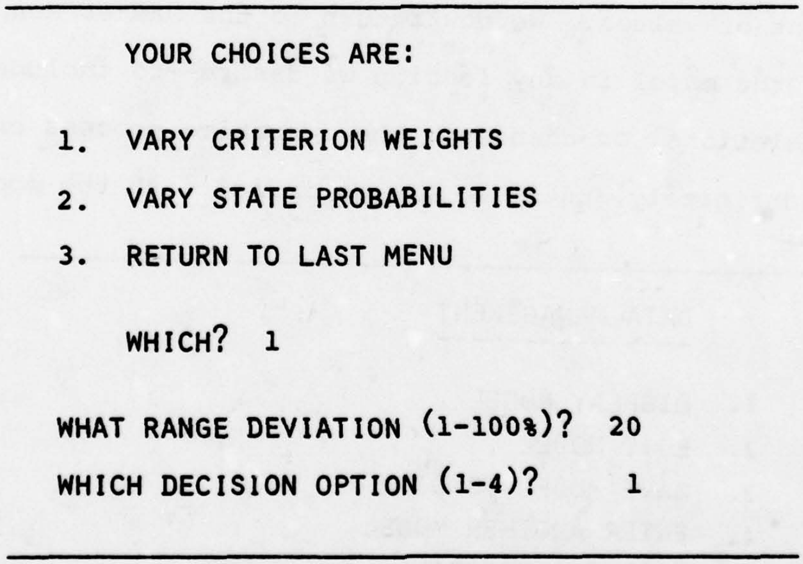
```
***  WHEN DECISION VALUE IS MAXIMIZED...

THE OPTIMUM OPTION IS  ** STATUSQUO **

   WITH AN EXPECTED VALUE OF  -9
- - - - - - - - - - - - - - - - - - - - - - -
       HIT 'RETURN' TO PROCEED...
```

```
EXPECTED VALUE TABLE
-------- ----- -----

OPTION         VALUE
------         -----
1  STATUSQUO   -9
2  MIL VIGIL   -12.15
3  SIM ALERT   -21.7
4  REINALERT   -37
```

- Sensitivity Analysis options are now offered and should be executed:

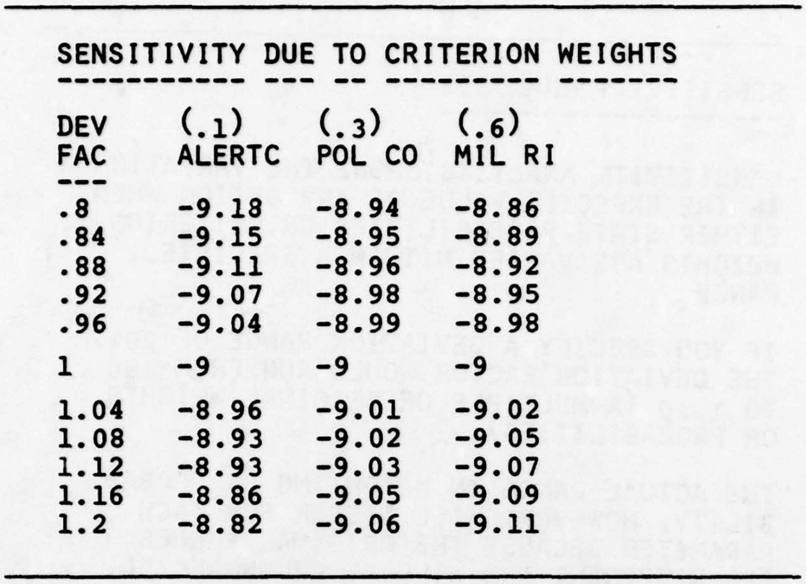


```
SENSITIVITY ANALYSIS
----------- --------

SENSITIVITY ANALYSIS SHOWS THE VARIATION
IN THE EXPECTED VALUE OF ANY OPTION WHEN
EITHER STATE PROBABILITIES OR CRITERION
WEIGHTS ARE VARIED WITHIN A SPECIFIED
RANGE

IF YOU SPECIFY A DEVIATION RANGE OF 20%,
THE DEVIATION FACTOR WOULD RUN FROM .80
TO 1.20 (A MULTIPLE OF ORIGINAL WEIGHTS
OR PROBABILITIES).

THE ACTUAL RANGE IN WEIGHTING OR PROBA-
BILITY, HOWEVER, WILL DIFFER FOR EACH
PARAMETER BECAUSE THE ORIGINAL VALUES
ARE DIFFERENT AND WILL BE RENORMALIZED.
```



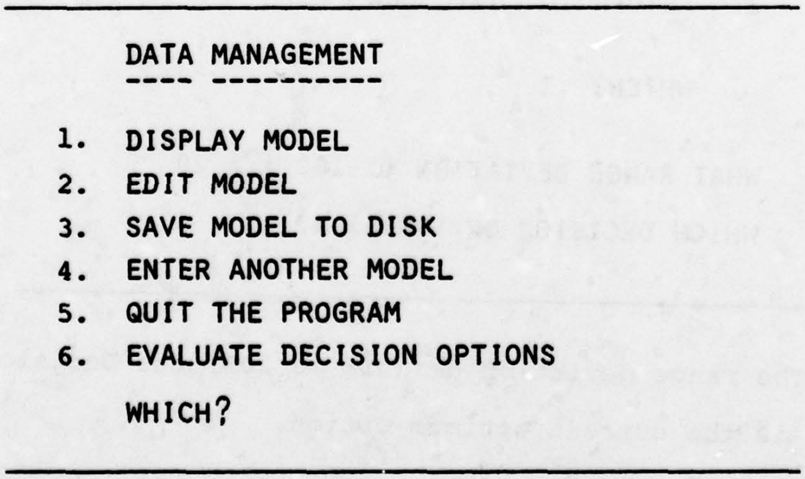
```
        YOUR CHOICES ARE:

1.      VARY CRITERION WEIGHTS

2.      VARY STATE PROBABILITIES

3.      RETURN TO LAST MENU

        WHICH?  1

WHAT RANGE DEVIATION (1-100%)?  20

WHICH DECISION OPTION (1-4)?     1
```

- The range deviation default is 20%; the Decision Option default is the current optimum Option.



```
SENSITIVITY DUE TO CRITERION WEIGHTS
----------- --- -- --------- -------

DEV     (.1)     (.3)     (.6)
FAC     ALERTC   POL CO   MIL RI
---
.8      -9.18    -8.94    -8.86
.84     -9.15    -8.95    -8.89
.88     -9.11    -8.96    -8.92
.92     -9.07    -8.98    -8.95
.96     -9.04    -8.99    -8.98
1       -9       -9       -9
1.04    -8.96    -9.01    -9.02
1.08    -8.93    -9.02    -9.05
1.12    -8.93    -9.03    -9.07
1.16    -8.86    -9.05    -9.09
1.2     -8.82    -9.06    -9.11
```

- After we view the sensitivity analysis, we may have greater or lesser confidence in our problem formulation or assessment of values. We now return to the Master Menu and can Edit the model in any fashion we desire--to include additions, deletions, or changes. The iterative process can continue indefinitely, until we are satisfied with the model:



```
          DATA MANAGEMENT
          ---- ----------

     1.   DISPLAY MODEL
     2.   EDIT MODEL
     3.   SAVE MODEL TO DISK
     4.   ENTER ANOTHER MODEL
     5.   QUIT THE PROGRAM
     6.   EVALUATE DECISION OPTIONS

          WHICH?
```

APPENDIX B

CAAMM Programmer's Guide

Introduction

The Programmer's Guide is intended to provide programmers the basic information they need to quickly locate and analyze sections of program code for modification or translation. One section is devoted to each individual program, and the final section is devoted to explaining the functions of commands and procedures that are not "standard" BASIC.

Programmers should consult the appropriate operator manuals for their own computer as well as the Applesoft and Disk Operating System manuals for the Apple computer.

All of the CAAMM programs are structured in parallel fashion. All have similar data input, editing, and output capabilities and procedures. The most important general techniques are outlined below:

1) The data input routines for all programs are matrix formatted and totally interactive. The routines use variable positioning of the screen cursor to permit logical, easy-to-follow input of data. The user need only remember to press {RETURN} after each entry because the program automatically repositions the cursor for the next input. The same basic technique applies to the edit functions of each program, permitting the interactive addition, deletion, or change of any

database element. The method is enhanced in most of the edit routines by the addition of variable sizing and positioning of the text scrolling window, permitting position-oriented editing as opposed to data-oriented editing.

2) The database display functions and other output routines are structured for "paging" of output. Each logical segment of output is displayed and remains in view until the user is ready to continue output. If a model or database is too large for a single screen, it is displayed in logically sequential blocks until completed. The Apple II permits only a 40x24 position text format, so programmers with 80-column screen displays should modify the output routines to take advantage of the greater coverage.

3) Sequential disk files are used for permanent data storage and for creating the user-defined function transformations used in both BIVAR REGR and MULVAR REGR. The function transformations are generated by using disk text files to create a transformation subroutine while the program is running. The technique is an extension of Kirschner (Ref 9). Disk files could be used to provide "virtual" memory, thereby considerably expanding potential data handling capacity, but current disk drives are too slow for extensive interaction.

4) All the programs primarily use fixed dimensioning of data arrays. While not as economical as variable dimensions, fixed dimensions permit easy expansion of databases during edit sessions. Expansion would otherwise be difficult because

BASIC does not permit redimensioning of arrays while the program is in operation. Dimensions can be adjusted depending upon actual memory available. The programs require approximately 32 K bytes of free memory, including both program and array space. Input and output formats are already written to accommodate significantly larger databases. Eliminating remark statements and compressing more program code onto multiple statement lines would free considerable memory space.

5) To maximize effective use of available memory, the Apple CHAIN command was used in the set of regression analysis programs. Basically the command permits one program to overlay another but preserves all data currently in memory. This procedure allows REGR DATA to help the user create a database and then run either BIVAR REGR or MULVAR REGR for the appropriate regression, thus eliminating the need for separate data management routines. The same principle applies to the small BIVAR HIRES program which provides the graphics capability for BIVAR REGR. Note that all the Regression programs must be maintained on the same diskette, together with the machine language CHAIN program.

Section 1, CAAMM MASTER

CAAMM MASTER is the control program for the CAAMM software diskette. It is approximately 2000 bytes in size and requires no array space.

TABLE VII

Logical Programming Blocks, CAAMM MASTER

BLOCK	LOCATION
Introduction	10000-10230
Master/Terminal Menu	10240-10510
Utility Routines	10550-10600
Intro Remarks	10620-10880

Section 2, REGR DATA

REGR DATA handles data management for the Regression Analysis programs BIVAR REGR and MULVAR REGR. It is currently 12300 bytes in size and requires 8700 bytes array space.

TABLE VIII

Logical Programming Blocks, REGR DATA

BLOCK	LOCATION
Introduction	10000-10220
Data Entry Menu	10230-10340
Master Menu	10360-10580
Terminal Menu	10600-10850
Utility Subroutines	10890-11050
Interactive Data Entry	11070-11780
Display Database	11790-12700
Print Variable Heading	12720-12950
Database Editing	12970-14760
Read/Write Disk Files	14780-15150
Instructions	15170-15430

TABLE IX

Key Variables Guide, REGR DATA

VARIABLE	PURPOSE
D(*,*)	Main Database Array
N	Total Number of Observations
NC	Number of Columns for Formats
NV	Number of Variables in Database
SN(*)	Sorting Array for Number of Observations
VAR$(*)	Array of Variable Names

Section 3, BIVAR REGR

BIVAR REGR provides the bivariate regression routines for the Regression Analysis collection of programs. It is currently 16700 bytes in size and requires up to 10700 bytes array space.

TABLE X

Logical Programming Blocks, BIVAR REGR

BLOCK	LOCATION
Introduction	10000-10140
Bivariate Model Selection	10160-10410
User-Defined Model	10430-11110
Selection of Variables	11130-11330
Curve Fit Control	11410-11890
Termination	11910-12150
Utility Subroutines	12190-12340
Variable Listing Routine	12360-12510
Regression Models	12530-14690
Means and Std Deviations	14710-14860
Curve Fit Statistics	14880-15200
Residuals	15220-15460
Prediction of Values	15480-15770
Read/Save Disk Files	15790-16090
Instructions	16110-16520
User-Defined Transforms	16540-16650

TABLE XI

Key Variables Guide, BIVAR REGR

VARIABLE	PURPOSE
A(*)	Array of Regression intercepts
B(*)	Array of Regression coefficients
BC	Index of Current bivariate model
D(*,*)	From REGR DATA
IB	Sum of 1/X(I) • Y(I)
IV	Index of Independent Variable
IX	Sum of 1/X(I)
IY	Sum of 1/Y(I)
L1	Sum of $(\text{Log}(X(I)))^2$
LX	Sum of Log (X(I))
N	Total number of observations
NV	Total number of variables
R(*)	Array of R^2 values
SERR(*)	Array of Standard Error values
SXIY	Sum of X(I)/Y(I)
SYIX	Sum of Y(I)/X(I)
TV	Number of transform variables
VAR$(*)	From REGR DATA
X(*)	Independent Variable Data Array
X1	Sum of X(I)
X2	Sum of $X(I)^2$
XI2	Sum of $1/X(I)^2$
XMEAN	Mean of X(*)
XSD	Standard Deviation of X(*)
Y(*)	Dependent Variable Data Array
Y1	Sum of Y(I)
Y2	Sum of $Y(I)^2$
YI2	Sum of $1/Y(I)^2$
YLX	Sum of Y(I) • LOG(X(I))
YMEAN	Mean of Y(*)
YSD	Standard Deviation of Y(*)

Section 4, BIVAR HIRES

BIVAR HIRES is a subprogram called by BIVAR REGR and produces a high resolution graphics display of the raw data and curve model. It is currently 2200 bytes in size and requires 8000 bytes of memory for the graphics screen plus up to 400 bytes array space.

TABLE XII

Logical Programming Blocks, BIVAR HIRES

BLOCK	LOCATION
Introduction	10000-10100
Plot Boundary of Graph	10150-10230
Determine Extremities	10250-10320
Plot Cross Hairs for Origin	10340-10370
Plot Data Points	10390-10440
Plot Regression Curve	10460-10540
Termination	10560-10650
Sort Routine	10690-10760
Curve Fit Subroutines	10780-10860

TABLE XIII

Key Variables Guide, BIVAR HIRES

VARIABLE	PURPOSE
A(*), B(*), BC	From BIVAR REGR
BX, UX	Left and right edges of graph
BY, UY	Bottom and top edges of graph
H(*)	Temporary sort array
H0, V0	HIRES coordinates of origin
HS, VS	Horizontal and vertical scale factors
X(*), Y(*)	From BIVAR REGR
XP, YP	HIRES coordinates of data points

Section 5, MULVAR REGR

MULVAR REGR provides the multivariate regression routines for the Regression Analysis collection of programs. It currently is 13500 bytes in size and requires 19700 bytes array space.

TABLE XIV

Logical Programming Blocks, MULVAR REGR

BLOCK	LOCATION
Introduction	10000-10120
Multivariate Model Selection	10140-10300
User-Defined Model	10320-11000
Selection of Variables	11020-11260
Output Options	11280-11460
Means and Std Deviations	11490-11690
	13530-13700
Correlation Coefficients	11710-11840
	13720-14050
Regression Coefficients/Statistics	11860-12710
Prediction of Values	12730-12770
	14590-14770
Termination	12790-13110
Utility Subroutines	13160-13340
Variable Listing	13360-13510
Variance-Covariance	14070-14390
Residuals	14410-14570
Read/Write Disk Files	14790-15090
Instructions	15110-15530
User-Defined Transforms	15550-15667

TABLE XV

Key Variables Guide, MULVAR REGR

VARIABLE	PURPOSE
A(*,*)	Reduction matrix
B(*)	Regression coefficients
D(*,*)	From REGR DATA
ESS	Error Sum of Squares
IV	Number of independent variables
M	Number of regression variables
M(*)	Regression variable means
N, NV	From REGR DATA
P$(*)	Output options
Q(*)	RESIDUALS
R(*,*)	All correlation coefficients
R2	R^2 value
RV(*)	Regression variable indices
S(*)	Regression variable std deviations
SD(*)	All std deviations
T(*)	All means
U(*,*)	Regression variable corr coefficients
V(*)	Transform temporary array
VAR$(*)	From REGR DATA
VE	Variance of the Estimate

Section 6, LINPROG

LINPROG implements the simplex method of Linear Programming. It is currently 23400 bytes in size and requires 9200 bytes array space.

TABLE XVI

Logical Programming Blocks, LINPROG

BLOCK	LOCATION
Introduction	10000-10210
Data Entry Menu	10230-10390
Master Menu	10410-10640
Output Options	10670-10830
Set Up Slack, Surplus, Artificial Var	10850-11380
Phase I Simplex	11510-13010
Phase II Simplex	13040-14230
Optimal Solution	14250-14660
Termination	14680-14720
Utility Subroutines	14760-14960
LP Tableau	14980-15460
Intermediate Solutions	15480-15760
Interactive Model Entry	15780-17120
Display Function	17140-17230
Edit Functions	17250-19690
Dismantle Original Array	19710-20230
Read/Save Disk Files	20250-20640
Instructions	20660-20740

TABLE XVII

Key Variables Guide, LINPROG

VARIABLE	PURPOSE
A(*,*)	Constraint coefficients
B(*)	Right Hand Side values
C(*)	Obj Function coefficients
CST$(*)	Constraint names
CT$(*)	Constraint inequalities
D(*)	Basic variable subscripts
E	Number of equality constraints
F(*)	Phase I Obj Function vector
G	Number of greater than constraints
L	Number of less than constraints
NC	Number of constraints
NV	Number of decision variables
OBJ$	Objective being optimized
P$	Printer option
P$(*)	Output options
T$(*,*)	Complete model in string array
TV	Total number of simplex variables
V(*)	Incoming Variable Index
VAR$(*)	Names of variables
Z	Z-value
Z(*)	Z-row vector
Z1	+1 for max problems or -1 for min problems

Section 7, DECISION

DECISION implements the Value Matrix Decision Aid. It is currently 21400 bytes in size and requires 8700 bytes array space.

TABLE XVIII

Logical Programming Blocks, DECISION

BLOCK	LOCATION
Introduction	10000-10270
Data Entry Menu	10290-10400
Master Menu	10420-10640
Evaluation Control/Termination	10660-10720
Utility Subroutines	10760-11050
Interactive Model Formulation	11070-13580
Evaluation of Matrices	13600-14260
Sensitivity Analysis	14280-15510
Read/Write Disk Files	15530-15970
Edit Functions	15990-17330
Display Functions	17350-17810
Instructions	17830-18600

TABLE XIX

Key Variable Table, DECISION

VARIABLE	PURPOSE
CP(*)	Value matrix combined on states, per criterion
CR$(*)	Criterion names
CV(*,*)	Value matrices, combined across criteria
EV(*)	Expected Values
FAC	Sensitivity scale factor
M$	Max or Min
NC	Number of Criteria
NOP	Number of Options
NP(*)	Normalized State probabilities
NS	Number of States
NW(*)	Normalized Criterion weights
OP$(*)	Option names
P$	Printer option
P(*)	State likelihoods
RP(*)	Revised probability/weight for sensitivity
SN$(*)	State names
V(*,*,*)	Value matrices
W(*)	Criterion weights

Section 8, Programming Language Guide

This section outlines the major Apple computer system-dependent procedures and programming language statements that are used in the CAAMM software package.

This section is intended only to help users understand some of the basic technical procedures and to help programmers interpret Apple statements for possible translation. Programmers and users both should refer to the Apple computer reference manuals (Ref 3) and (Ref 4) for greater detail.

System Initialization: First insert the CAAMM software diskette into the disk drive and turn on the computer. From BASIC type "PR#7" where 7 is the slot number of the disk drive controller card inside the computer. The control program CAAMM MASTER will automatically run and present the different analysis options.

Disk Operating System

CATALOG - displays the names of all files on a diskette.

PR#7 - initializes the disk drive when its controller card is in slot 7.

CHAIN - permits one program to load into memory over the top of the first program, without destroying data. This command is only available if the assembly language routine CHAIN is on the diskette in the disk drive.

OPEN, CLOSE, WRITE, READ - used to open, close, write data to, and read data from disk text files.

EXEC - executes successive lines in a test file as if they had just been typed from the keyboard; can be used to create new program lines.

<u>Printer Control</u>

PR#7 - initializes the printer when its interface card is in slot 7.

POKE 1148,32 - initializes the printer for 600 baud data transmission rate, using Apple serial interface.

POKE 1788,40 - sets up 40 column printed output, using Apple serial interface.

<u>Screen Display Control</u>

HOME - clears the current scrolling window and positions the cursor at the top.

VTAB, HTAB - move cursor vertically, horizontally.

CALL -998 - moves cursor up one line.

CALL -958 - clears screen from point of cursor to bottom of page.

POS (0) - returns current cursor position.

POKE 34,T - protects the top T lines on the screen.

POKE 33,W - sets width of new scrolling area.

POKE 32,L - protects the left L columns on the screen.

TEXT - resets the scrolling area to full 40 x 24 format.

<u>Data Input</u>

INPUT "XYZ"; A$ - prints "XYZ" on the screen and waits for user input for A$; requires {RETURN} following entry.

GET A$ - waits for a one-character input for A$; does not need {RETURN}

High Resolution Graphics

HGR2 - initializes 2d HIRES screen for multicolor, 280x192 point high resolution graphics display.

HCOLOR - sets color for next plotting.

HPLOT X,Y - plots one dot as screen coordinates (X,Y), measured from upper left corner of screen.

Variables

A - real variable; for numeric data only.

A$ - string variable for alphanumeric data; maximum length is 255 characters.

Variable Names - may be up to 255 characters long, but only the first two characters are significant.

Final Notes

Software development on the Apple computer is currently confined by the lack of a standard 80-column display, the lack of matrix operators, and the lack of formatted printing statements. Such restrictions have been largely overcome in the CAAMM software package, but programmers using computers with such capabilities should exploit them by program modification.

APPENDIX C

CAAMM Program Listings

This appendix includes actual CAAMM software listings. Remark lines help identify major routines and logical blocks.

Section 1, CAAMM MASTER

```
10000  IF CFLAG <  > 0 THEN 10240
10010  CLEAR : TEXT : HOME : VTAB 2
10020  PRINT "* * * * * * * * * * * * * * * * * * * *"
10030  PRINT
10040  PRINT "*  C O M P U T E R   A S S I S T E D  *"
10050  PRINT
10060  PRINT "*           A N A L Y S I S           *"
10070  PRINT
10080  PRINT "*                F O R                *"
10090  PRINT
10100  PRINT "*  M I L I T A R Y   M A N A G E R S  *"
10110  PRINT
10120  PRINT "*                                     *"
10130  PRINT "            ( C A A M M )"
10140  PRINT "*                                     *"
10150  PRINT
10160  PRINT "*                 BY                  *"
10170  PRINT
10180  PRINT "*           ROBERT D. CONTE           *"
10190  PRINT
10200  PRINT "* * * * * * * * * * * * * * * * * * * *"
10210 BELL$ = "": REM - BELL
10220 D$ =  CHR$ (13) +  CHR$ (4)
10230  VTAB 24: INPUT "  DO YOU WANT INTRODUCTORY REMARKS?  ";A$: IF  LE
 FT$ (A$,1) = "Y" THEN  GOSUB 10630
10240  PRINT BELL$: TEXT : HOME : VTAB 2
10250  PRINT  TAB( 9);"CAAMM  MASTER  MENU"
10260  PRINT  TAB( 9);"=====  ======  ===="
10270  PRINT
10280  PRINT "YOU MAY SELECT ONE OF THE FOLLOWING:"
10290  PRINT "--- --- ------ --- -- --- ---------"
10300  POKE 33,36: POKE 32,4
10310  PRINT : PRINT
10320  PRINT "1.  REGRESSION  ANALYSIS"
10330  PRINT : PRINT
10340  PRINT "2.  LINEAR  PROGRAMMING"
10350  PRINT : PRINT
10360  PRINT "3.  MATRIX  DECISION  AID"
10370  PRINT : PRINT : PRINT
10380  PRINT "4.  ** QUIT  CAAMM  PACKAGE **"
10390  TEXT
10400  VTAB 23: HTAB 9: INPUT "WHICH  SELECTION?  ";A$:S =  INT ( VAL (A
 $))
10410  IF S < 1 OR S > 4 OR A$ = "" THEN  CALL  - 998: GOTO 10400
10420  IF S = 4 THEN 10500
10430
10440  REM  ** CHAIN ANALYSIS PROGRAMS **
```

```
10450 CFLAG = 1
10460  PRINT D$;"BLOAD CHAIN, A520"
10470  IF S = 1 THEN  CALL 520"REGR DATA"
10480  IF S = 2 THEN  CALL 520"LINPROG"
10490  IF S = 3 THEN  CALL 520"DECISION"
10500  TEXT : HOME : VTAB 5: PRINT "THANK  YOU...": PRINT : PRINT "   HO
 PE YOU GOT WHAT YOU NEEDED !!!": VTAB 12
10510  END
10520
10530  REM   ************************************************************

10540
10550  REM  ** UTILITY SUBROUTINES **
10560  VTAB 23: GOSUB 10590: PRINT  TAB( 6);"HIT 'RETURN' TO PROCEED...
  ";: GET A$: PRINT
10570  RETURN
10580
10590  PRINT "- - - - - - - - - - - - - - - - - - - - -"
10600  RETURN
10610
10620  REM  ** INTRODUCTORY REMARKS **
10630  HOME
10640  PRINT  TAB( 8);"CAAMM  SOFTWARE  PACKAGE"
10650  PRINT  TAB( 8);"=====  ========  ======="
10660  PRINT : PRINT
10670  PRINT "   THE  C.A.A.M.M.  SOFTWARE PACKAGE IS DESIGNED TO MEET T
 HE NEEDS OF"
10680  PRINT
10690  PRINT  TAB( 9);"MANAGERS AND ANALYSTS"
10700  PRINT
10710  PRINT "FOR EASY-TO-USE, TIME-SENSITIVE COMPUTERSUPPORT."
10720  PRINT : PRINT
10730  PRINT "    THE PACKAGE DOES NOT INCLUDE EVERY  ANALYSIS TECHNIQUE
 , BUT IT DOES INCLUDE THREE OF THE MOST USEFUL:"
10740  PRINT
10750  PRINT  TAB( 10);"REGRESSION ANALYSIS"
10760  PRINT
10770  PRINT  TAB( 10);"LINEAR PROGRAMMING"
10780  PRINT
10790  PRINT  TAB( 10);"MATRIX DECISION AID"
10800  GOSUB 10560
10810  HOME : VTAB 5
10820  PRINT  TAB( 15);BELL$;"** NOTE **"
10830  PRINT : PRINT
10840  PRINT "     FOR RAPID PROGRAM TRANSITIONS,"
10850  PRINT
10860  PRINT "ALL  'YES/NO'  TYPE QUESTIONS OR PROMPTSCAN BE ANSWERED BY
   'Y'  FOR 'YES' OR BY'N'  OR A SIMPLE  'RETURN'  FOR 'NO'...."
10870  GOSUB 10560
10880  RETURN
```

Section 2, REGR DATA

```
10000  IF RDFLAG = 1 THEN 10180
10010  IF RDFLAG = 2 THEN 10370
10020 BELL$ = "": REM - BELL
10030  PRINT BELL$
10040  TEXT : HOME : VTAB 5
10050  PRINT "* * * * * * * * * * * * * * * * * * * * * *"
10060  PRINT
10070  PRINT "          R E G R E S S I O N"
10080  PRINT
10090  PRINT "            A N A L Y S I S"
10100  PRINT : PRINT
10110  PRINT "                  BY"
10120  PRINT
10130  PRINT "            ROBERT D. CONTE"
10140  PRINT
10150  PRINT "* * * * * * * * * * * * * * * * * * * * * *"
10160  VTAB 24: INPUT "  DO YOU WANT INTRODUCTORY REMARKS?  ";A$: IF  LE
 FT$ (A$,1) = "Y" THEN  GOSUB 15170
10170
10180  CLEAR
10190  REM - DIMENSIONED FOR 20 VARIABLES, 80 OBSERVATIONS
10200  DIM D(80,20),SN(10),VAR$(20)
10210 D$ =  CHR$ (13) +  CHR$ (4)
10220
10230  TEXT : HOME : POKE 33,38: POKE 32,2
10240  VTAB 5
10250  PRINT "     TO ENTER DATA, YOU CAN:"
10260  PRINT "     -- ----- ----  --- ---"
10270  PRINT : PRINT : PRINT
10280  PRINT "1.  READ EXISTING DATA BASE FROM DISK"
10290  PRINT : PRINT
10300  PRINT "2.  CREATE DATA BASE INTERACTIVELY"
10310  TEXT
10320  VTAB 18: HTAB 7: INPUT "WHICH METHOD?  ";A$:DE =  VAL (A$): IF A$
  = "" THEN  GOSUB 14780: GOTO 10370
10330  IF DE < 1 OR DE > 2 THEN  CALL  - 998: GOTO 10320
10340  ON DE GOSUB 14780,11070: GOTO 10370
10350
10360  REM - MAIN MENU
10370  HOME : POKE 33,34: POKE 32,6
10380  VTAB 3
10390 X =  FRE (0): REM   - CLEANS UP UNUSED STRING VALUES
10400  PRINT "     DATA MANAGEMENT"
10410  PRINT "     ---- ----------"
10420  PRINT : PRINT
10430  PRINT "1.  DISPLAY DATA BASE"
10440  PRINT
```

```
10450  PRINT "2.  EDIT DATA BASE"
10460  PRINT
10470  PRINT "3.  SAVE DATA BASE TO DISK"
10480  PRINT
10490  PRINT "4.  ENTER ANOTHER DATA BASE"
10500  PRINT
10510  PRINT "5.  QUIT PROGRAM"
10520  PRINT : PRINT
10530  PRINT "6.  RUN REGRESSION ANALYSIS"
10540  TEXT
10550  VTAB 22: HTAB 11: INPUT "WHICH OPTION?  ";A$:DO =  VAL (A$): IF A
 $ = "" THEN 10600
10560  IF DO < 1 OR DO > 6 THEN  CALL  - 998: GOTO 10550
10570  ON DO GOSUB 11790,12970,14990: IF DO < 4 THEN 10370
10580  ON DO - 3 GOTO 10180,10830,10600
10590
10600  TEXT : HOME : VTAB 3
10610  PRINT "PLEASE SELECT ONE OF THE METHODS BELOW:"
10620  PRINT "------ ------ --- -- --- ------- -----"
10630  VTAB 8
10640  PRINT "1.  BIVARIATE  REGRESSION  ANALYSIS"
10650  PRINT
10660  PRINT "       (FOR DATA SETS OF ONE DEPENDENT"
10670  PRINT "        AND ONE INDEPENDENT VARIABLE)"
10680  VTAB 15
10690  PRINT "2.  MULTIVARIATE  REGRESSION  ANALYSIS"
10700  PRINT
10710  PRINT "       (FOR DATA SETS OF ONE DEPENDENT"
10720  PRINT "        AND ONE OR MORE INDEP VARIABLES)"
10730  PRINT
10740  VTAB 22
10750  INPUT "    WHICH METHOD?  ";A$:CM =  VAL (A$)
10760  IF CM < 1 OR CM > 2 THEN  CALL  - 998: GOTO 10750
10770
10780  REM  ** CHAIN CURVE FIT PROGRAMS **
10790  PRINT D$;"BLOAD CHAIN, A520"
10800  IF CM = 1 THEN  CALL 520"BIVAR REGR"
10810  IF CM = 2 THEN  CALL 520"MULVAR REGR"
10820
10830  PRINT D$;"BLOAD CHAIN, A520"
10840  CALL 520"CAAMM MASTER"
10850  END
10860
10870  REM     *********************************************************

10880
10890  REM  ** UTILITY SUBROUTINES **
10900  VTAB 23: GOSUB 10930: PRINT "     HIT 'RETURN' TO PROCEED...   ";:
 GET A$: PRINT
10910  RETURN
10920
10930  PRINT "- - - - - - - - - - - - - - - - - - - - - -"
10940  RETURN
```

```
10950
10960  TEXT : POKE 34,23: VTAB 23: GOSUB 10930: INPUT "        NEED TO MA
 KE CHANGES?  ";A$:A$ =  LEFT$ (A$,1)
10970  RETURN
10980
10990  VTAB 23: GOSUB 10930: PRINT "HIT 'RETURN' TO GO ON, OR 'Q' TO QUI
 T ";
11000  GET A$: IF  ASC (A$) = 81 THEN 10830
11010  PRINT
11020  RETURN
11030
11040  PRINT : VTAB 23: GOSUB 10930: INPUT "        EDIT ANOTHER?  ";A$:A
 $ =  LEFT$ (A$,1): PRINT
11050  RETURN
11060
11070  REM   ** INTERACTIVE DATA ENTRY **
11080  TEXT : HOME
11090  PRINT "INTERACTIVE DATA BASE ENTRY"
11100  PRINT "----------- ---- ---- -----"
11110  PRINT : PRINT
11120  POKE 34,4
11130  INPUT "HOW MANY VARIABLES (2-20)?     ";A$:NV =  VAL (A$): IF NV
 < 2 OR NV > 20 THEN  CALL  - 998: GOTO 11130
11140  PRINT
11150  INPUT "MAX DIGITS PER ENTRY (1-15)?   ";A$:FW =  VAL (A$) + 2: IF
  FW < 7 THEN FW = 7
11160  IF FW > 15 THEN  CALL  - 998: GOTO 11150
11170 NC =  INT (35 / FW): REM    - NUMBER OF COLUMNS
11180  PRINT
11190  PRINT "NAME THE VARIABLES BELOW, WITH 1-6 CHAR:"
11200  VTAB 12
11210  FOR J = 1 TO NV
11220  IF J > 10 THEN  POKE 33,20: POKE 32,20: VTAB J + 1
11230    PRINT "V(";J;")"; TAB( 7);: INPUT "= ";A$:VAR$(J) =  LEFT$ (A$,
 6)
11240  NEXT J
11250  TEXT
11260  REM - CHANGES?
11270  GOSUB 10960: IF A$ >  < "Y" THEN 11320
11280  GOSUB 11670: IF CE = 0 THEN 11320
11290 VAR$(CE) =  LEFT$ (A$,6)
11300  GOTO 11280
11310
11320  TEXT : HOME : VTAB 5
11330  INPUT "  NEED INSTRUCTIONS FOR DATA ENTRY?  ";A$
11340  IF  LEFT$ (A$,1) = "Y" THEN  GOSUB 15310
11350  HOME : VTAB 2
11360 SN(0) = 9999
11370  PRINT "ENTER THE VALUE FOR EACH OBSERVATION ANDVARIABLE;  HIT 'RE
 TURN' AFTER EACH ENTRY"
11380  PRINT "TO STOP INPUT, HIT 'RETURN' W/O ENTRY."
11390  GOSUB 10930: POKE 34,11
```

```
11400 AV = 0: IF NV / NC >  INT (NV / NC) THEN AV = 1
11410 K = 1: REM  - 1ST BLOCK OF NC VARIABLES
11420  GOSUB 12720
11430   I = 1: REM - 1ST OBSERVATION
11440    PRINT I; TAB( 6);
11450    FOR J = K * NC - NC + 1 TO K * NC
11460      INPUT "";A$:D(I,J) =  VAL (A$): IF A$ = "" THEN 11530
11470      IF J = NV THEN 11500
11480      CALL  - 998: HTAB FW * (J - (K - 1) * NC) + 6
11490    NEXT J
11500  PRINT
11510 I = I + 1: IF I > 80 THEN 11530
11520  GOTO 11440
11530 SN(K) = I - 1: IF SN(K) < SN(K - 1) THEN N = SN(K)
11540  CALL  - 998: CALL  - 958: PRINT : PRINT "DATA ENTRY STOPPED..."
11550  PRINT "   TOTAL OF ";N;" COMPLETE OBSERVATIONS."
11560  IF N > 2 THEN 11610
11570  PRINT : PRINT BELL$;"** SORRY, BUT THAT'S NOT ENOUGH FOR ANY TYPE
   OF REGRESSION ANALYSIS !!"
11580  PRINT : PRINT "ENTER MORE DATA OR QUIT NOW !!"
11590  PRINT : PRINT : GOSUB 10990
11600  GOTO 10230
11610  PRINT : PRINT : GOSUB 10900
11620 K = K + 1: IF K >  INT (NV / NC) + AV THEN 11640
11630  GOTO 11420
11640  TEXT :DE = 0
11650  RETURN
11660 
11670  REM  ** VARIABLE NAME CORRECTION ROUTINE **
11680  VTAB 24: INPUT "INDEX OF ITEM TO CHANGE (0 TO STOP)? ";A$:CE =  I
 NT ( VAL (A$)): IF A$ = "" OR CE = 0 THEN  RETURN
11690  IF CE > NV OR CE < 0 THEN 11680
11700  IF CE > 10 THEN 11740
11710  REM - SET WINDOW TO PROTECT PARTS OF SCREEN
11720  POKE 33,19: POKE 32,1: VTAB CE + 11: HTAB 8
11730  GOTO 11750
11740  POKE 33,20: POKE 32,20: VTAB CE + 1: HTAB 9
11750  INPUT "";A$
11760  POKE 32,0: POKE 33,40
11770  RETURN
11780 
11790  REM   ** DISPLAY DATA BASE **
11800  IF P$ = "Y" THEN  PRINT D$;"PR#0"
11810  TEXT : HOME : VTAB 5
11820  PRINT  TAB( 8);"DATA BASE DISPLAY OPTIONS"
11830  PRINT  TAB( 8);"---- ---- ------- -------"
11840  PRINT : PRINT : PRINT
11850  POKE 33,32: POKE 32,8
11860  PRINT
11870  PRINT "1.  ENTIRE DATA BASE"
11880  PRINT
11890  PRINT "2.  BY VARIABLE"
```

```
11900  PRINT
11910  PRINT "3.  BY OBSERVATION"
11920  PRINT : PRINT
11930  PRINT "4.  RETURN TO LAST MENU"
11940  TEXT
11950  VTAB 22: HTAB 13: INPUT "WHICH?  ";A$:DP =  INT ( VAL (A$))
11960  IF A$ = "" OR DP = 4 THEN AO = 0:AV = 0: RETURN
11970  IF DP < 1 OR DP > 4 THEN  CALL  - 998: GOTO 11950
11980  CALL  - 998: CALL  - 958: INPUT " DO YOU WANT IT ROUTED TO THE PR
 INTER? ";P$:P$ =  LEFT$ (P$,1)
11990
12000  IF P$ = "Y" THEN  PRINT D$;"PR#4": PRINT : POKE 1148,32: POKE 178
 8,40: PRINT : GOSUB 10930: PRINT : PRINT
12010 AV = 0: IF NV / NC >  INT (NV / NC) THEN AV = 1
12020 AO = 0: IF N / 15 >  INT (N / 15) THEN AO = 1
12030
12040  HOME
12050  ON DP GOTO 12480,12070,12300
12060
12070  REM - BY VARIABLE
12080  GOSUB 13340
12090  PRINT "WHICH VARIABLE (1-";NV;")?  ";: INPUT "";VI
12100  IF VI < 1 OR VI > NV THEN  CALL  - 998: GOTO 12090
12110  HOME
12120  PRINT  TAB( 6);VAR$(VI)
12130  PRINT
12140  PRINT  TAB( 6);"V(";VI;")"
12150  PRINT  TAB( 6);"----"
12160  POKE 34,4
12170  FOR Q = 1 TO  INT (N / 15) + AO
12180    IF Q / 2 =  INT (Q / 2) THEN  POKE 33,20: POKE 32,20: HOME : VT
 AB 5
12190    FOR I = Q * 15 - 14 TO Q * 15
12200      PRINT I; TAB( 6);D(I,VI)
12210      IF I / 5 =  INT (I / 5) THEN  PRINT
12220      IF I = N THEN  TEXT : GOTO 12270
12230    NEXT I
12240  IF Q / 2 =  INT (Q / 2) AND DO = 1 THEN  TEXT : GOSUB 10900: HOME

12250  NEXT Q
12260 VI = 0
12270  GOSUB 10900
12280  GOTO 11800
12290
12300  REM - BY OBSERVATION
12310  VTAB 5
12320  PRINT "WHICH OBSERVATION (1-";N;")?  ";: INPUT "";OI
12330  IF OI < 1 OR OI > N THEN  CALL  - 998: GOTO 12320
12340  HOME : VTAB 3: POKE 34,5
12350  PRINT "OBSERVATION  ";OI
12360  PRINT "-----------"
12370  PRINT
```

```
12380  FOR J = 1 TO NV
12390    PRINT "V(";J;")"; TAB( 7);VAR$(J); TAB( FW + 7);D(OI,J)
12400    IF J = NV THEN 12430
12410    IF J / 5 =  INT (J / 5) THEN  PRINT
12420    IF J / 15 =  INT (J / 15) THEN  GOSUB 10900: HOME
12430  NEXT J
12440 OI = 0
12450  GOSUB 10900
12460  GOTO 11800
12470
12480  REM  - ENTIRE ARRAY
12490 K = 1: REM - 1ST BLOCK OF NC VARIABLES
12500  GOSUB 12720
12510  POKE 34,4
12520    FOR Q = 1 TO  INT (N / 15) + AO
12530      FOR I = Q * 15 - 14 TO Q * 15
12540      PRINT I; TAB( 6);
12550        FOR J = K * NC - NC + 1 TO K * NC
12560          PRINT D(I,J); TAB( FW * (J - (K - 1) * NC) + 6);
12570          IF  POS (H) = 0 THEN  CALL  - 998
12580          IF J = NV THEN 12600
12590        NEXT J
12600        PRINT
12610        IF I / 5 =  INT (I / 5) THEN  PRINT
12620        IF I = N THEN 12670
12630      NEXT I
12640      PRINT
12650      IF DO = 1 THEN  GOSUB 10900: HOME
12660    NEXT Q
12670    IF DO = 1 THEN  PRINT : GOSUB 10900: TEXT : HOME
12680 K = K + 1: IF K >  INT (NV / NC) + AV THEN 12700
12690  GOTO 12500
12700 NC = NC / DO: GOTO 11800
12710
12720  REM  ** PRINT VARIABLE LABELS **
12730  REM - PRINTS NC VARIABLES ACROSS
12740  IF DE = 2 THEN  VTAB 8
12750  HTAB 6: CALL  - 958
12760 J = K * NC - NC + 1
12770    PRINT VAR$(J); TAB( FW * (J - (K - 1) * NC) + 6)
12780    IF J = NV THEN 12800
12790 J = J + 1: IF J <  = K * NC THEN 12770
12800  VTAB 3
12810  IF DE = 2 THEN  VTAB 10
12820  HTAB 6
12830 J = K * NC - NC + 1
12840    PRINT "V(";J;")"; TAB( FW * (J - (K - 1) * NC) + 6)
12850    IF J = NV THEN 12870
12860 J = J + 1: IF J <  = K * NC THEN 12840
12870  VTAB 4
12880  IF DE = 2 THEN  VTAB 11
12890  HTAB 6
```

```
12900 J = K * NC - NC + 1
12910    PRINT "----"; TAB( FW * (J - (K - 1) * NC) + 6)
12920    IF J = NV THEN 12940
12930 J = J + 1: IF J <  = K * NC THEN 12910
12940  PRINT
12950  RETURN
12960
12970  REM  ** DATA BASE EDITING **
12980  TEXT : HOME : POKE 33,35: POKE 32,5
12990  VTAB 3
13000  PRINT "  DATA BASE EDIT FUNCTIONS:"
13010  PRINT "  ---- ---- ---- ---------"
13020  PRINT : PRINT
13030  PRINT "1.  DELETE A VARIABLE"
13040  PRINT "2.  DELETE AN OBSERVATION"
13050  PRINT
13060  PRINT "3.  ADD A VARIABLE"
13070  PRINT "4.  ADD AN OBSERVATION"
13080  PRINT
13090  PRINT "5.  CHANGE DATA BY VARIABLE"
13100  PRINT "6.  CHANGE DATA BY OBSERVATION"
13110  PRINT "7.  CHANGE INDIVIDUAL DATA ENTRY"
13120  PRINT : PRINT
13130  PRINT "8.  RETURN TO LAST MENU"
13140  TEXT
13150  VTAB 21: HTAB 10: INPUT "WHICH?  ";A$:DC =  VAL (A$): IF A$ = ""
 OR DC = 8 THEN  RETURN
13160  IF DC < 1 OR DC > 8 THEN  CALL  - 998: GOTO 13150
13170  ON DC GOTO 13470,13610,13760,13910,14070,14330,14550: GOTO 13190
13180
13190  REM - COMPRESS DATA BASE
13200 Q = 0
13210  FOR J = 1 TO NV
13220    IF J = VI THEN 13310
13230   Q = Q + 1
13240   VAR$(Q) = VAR$(J)
13250 R = 0
13260    FOR I = 1 TO N
13270      IF I = OI THEN 13300
13280     R = R + 1
13290     D(R,Q) = D(I,J)
13300    NEXT I
13310  NEXT J
13320  RETURN
13330
13340  REM - LIST VARIABLES
13350  PRINT : PRINT : PRINT
13360  FOR J = 1 TO NV
13370  IF J > 10 THEN  POKE 33,20: POKE 32,20: VTAB J - 3
13380   H = 3: IF J = 10 THEN H = 2
13390    PRINT "V(";J;")"; SPC( H);VAR$(J)
13400    IF NV < 6 THEN  PRINT
```

```
13410  NEXT J
13420  TEXT
13430  IF DC = 7 THEN  VTAB 21: GOSUB 10930: RETURN
13440  VTAB 23: GOSUB 10930: VTAB 24
13450  RETURN
13460
13470  REM  - DELETE VARIABLE
13480  HOME : VTAB 3
13490  PRINT "DELETE VARIABLE"
13500  PRINT "------ --------"
13510  GOSUB 13340
13520  PRINT "WHICH VARIABLE (1-";NV;")?  ";: INPUT "";A$: IF A$ = "" TH
 EN 12980
13530 VI =  VAL (A$): IF VI < 1 OR VI > NV THEN  CALL  - 998: GOTO 13520

13540  GOSUB 13190
13550  FOR I = 1 TO N:D(I,NV) = 0: NEXT I
13560  HOME : PRINT BELL$: GOSUB 13340
13570 NV = NV - 1:VI = 0
13580  POKE 34,23: GOSUB 11040: IF A$ = "Y" THEN 13480
13590  GOTO 12980
13600
13610  REM - DELETE OBSERVATION
13620  HOME : VTAB 3
13630  PRINT "DELETE OBSERVATION"
13640  PRINT "------ -----------"
13650  PRINT : PRINT
13660  PRINT "WHICH OBSERVATION (1-";N;")?  ";: INPUT "";A$: IF A$ = ""
 THEN 12980
13670 OI =  VAL (A$): IF OI < 1 OR OI > N THEN  CALL  - 998: GOTO 13660
13680  GOSUB 13190
13690  FOR J = 1 TO NV:D(N,J) = 0: NEXT J
13700  HOME : VTAB 10
13710  PRINT BELL$;"**  OBSERVATION ";OI;" HAS BEEN DELETED"
13720 N = N - 1:OI = 0
13730  GOSUB 11040: IF A$ = "Y" THEN 13620
13740  GOTO 12980
13750
13760  REM - ADD VARIABLE
13770  TEXT : HOME : VTAB 3
13780  PRINT "ADD A VARIABLE"
13790  PRINT "--- - --------"
13800  GOSUB 13340
13810  IF NV = 20 THEN  PRINT "SORRY, BUT THAT'S THE MAXIMUM!": GOSUB 10
 900: GOTO 12980
13820 AVAR = 1: REM - FLAG FOR SUBROUTINE
13830  INPUT "NAME OF NEW VARIABLE?  ";A$: IF A$ = "" THEN 12980
13840 NV = NV + 1:VI = NV
13850 VAR$(VI) =  LEFT$ (A$,6)
13860  GOTO 14140
13870 AVAR = 0
13880  GOSUB 11040: IF A$ = "Y" THEN 13770
```

```
13890  GOTO 12980
13900
13910  REM -ADD OBSERVATION
13920  IF N < 80 THEN 14000
13930  TEXT : HOME : VTAB 3
13940  PRINT "ADD AN OBSERVATION"
13950  PRINT "--- -- -----------"
13960  PRINT : PRINT
13970  PRINT BELL$;"YOU CURRENTLY HAVE 80 OBSERVATIONS."
13980  PRINT : PRINT
13990  PRINT "SORRY, BUT THAT'S THE MAXIMUM!": GOSUB 10900: GOTO 12980
14000 N = N + 1:OI = N
14010 AOBS = 1: REM - SET FLAG FOR SUBROUTINE
14020  GOTO 14400
14030 AOBS = 0
14040  GOSUB 11040: IF A$ = "Y" THEN 13930
14050  GOTO 12980
14060
14070  REM - CHANGE A VARIABLE
14080  TEXT : HOME   VTAB 3
14090  PRINT "CHANGE DATA BY VARIABLE"
14100  PRINT "------ ---- -- --------"
14110  GOSUB 13340
14120  PRINT "WHICH VARIABLE (1-";NV;")?  ";: INPUT "";A$: IF A$ = "" TH
 EN 12980
14130 VI =  VAL (A$): IF VI < 1 OR VI > NV THEN  CALL  - 998: GOTO 14120

14140  HOME : VTAB 3
14150  PRINT  TAB( 6);VAR$(VI)
14160  PRINT
14170  PRINT  TAB( 6);"V(";VI;")"
14180  PRINT  TAB( 6);"----"
14190  PRINT
14200  PRINT  TAB( 6);"CURRENT     CHANGE TO"
14210  PRINT
14220  POKE 34,9
14230  PRINT
14240  FOR I = 1 TO N
14250     PRINT I; TAB( 6);D(I,VI); TAB( 17);: INPUT "";A$: IF A$ = "" TH
 EN 14270
14260   D(I,VI) =  VAL (A$)
14270  NEXT I
14280  PRINT : PRINT
14290  IF AVAR = 1 THEN 13870
14300  GOSUB 11040: IF A$ = "Y" THEN 14080
14310  GOTO 12980
14320
14330  REM -CHANGE OBSERVATION
14340  TEXT : HOME : VTAB 3
14350  PRINT "CHANGE DATA BY OBSERVATION"
14360  PRINT "------ ---- -- -----------"
14370  PRINT : PRINT
```

```
14380  PRINT "WHICH OBSERVATION (1-";N;")?  ";: INPUT "";A$: IF A$ = ""
 THEN 12980
14390 OI =  VAL (A$): IF OI < 1 OR OI > N THEN  CALL  - 998: GOTO 14380
14400  HOME : VTAB 3
14410  PRINT  TAB( 14);"OBSERVATION  ";OI
14420  PRINT  TAB( 14);"-----------"
14430  PRINT
14440  PRINT "VARIABLE     CURRENT    CHANGE TO"
14450  PRINT
14460  POKE 34,7
14470  FOR J = 1 TO NV
14480     PRINT "V(";J;")"; TAB( 7);VAR$(J); TAB( 14);D(OI,J); TAB( 25);:
 INPUT "";A$: IF A$ = "" THEN 14500
14490    D(OI,J) =  VAL (A$)
14500  NEXT J
14510  IF AOBS = 1 THEN 14030
14520  GOSUB 11040: IF A$ = "Y" THEN 14340
14530  GOTO 12980
14540
14550  REM - CHANGE INDIVIDUAL ENTRY
14560  HOME : VTAB 3
14570  PRINT "CHANGE INDIVIDUAL DATA ENTRY"
14580  PRINT "------ ---------- ---- -----"
14590  GOSUB 13340
14600  PRINT "WHICH VARIABLE (1-";NV;")?    ";: INPUT "";A$: IF A$ = ""
 THEN 12980
14610 VI =  VAL (A$): IF VI < 1 OR VI > NV THEN  CALL  - 998: GOTO 14600

14620  PRINT
14630  PRINT "WHICH OBSERVATION (1-";N;")?  ";: INPUT "";A$: IF A$ = ""
 THEN 12980
14640 OI =  VAL (A$): IF OI < 1 OR OI > N THEN  CALL  - 998: GOTO 14630
14650  HOME : VTAB 3
14660  PRINT  TAB( 13);VAR$(VI)
14670  PRINT
14680  PRINT  TAB( 13);"V(";VI;")"
14690  PRINT  TAB( 13);"----"
14700  PRINT
14710  PRINT  TAB( 6);"CURRENT    CHANGE TO"
14720  PRINT
14730  PRINT OI; TAB( 6);D(OI,VI); TAB( 17);: INPUT "";A$: IF A$ = "" TH
 EN 14750
14740 D(OI,VI) =  VAL (A$)
14750  GOSUB 11040: IF A$ = "Y" THEN 14560
14760  GOTO 12980
14770
14780  REM  ** READ DATA FROM DISK FILE **
14790  HOME : VTAB 5
14800  INPUT "DO YOU WANT TO SEE THE DISK CATALOG?  ";A$
14810  IF  LEFT$ (A$,1) = "Y" THEN  HOME : PRINT D$;"CATALOG": VTAB 23:
 GOSUB 10930: GOTO 14860
14820  HOME : VTAB 3
```

```
14830  PRINT  TAB( 8);"READ DATA BASE FROM DISK"
14840  PRINT  TAB( 8);"---- ---- ---- ---- ----"
14850  PRINT : PRINT : PRINT
14860  HTAB 8: INPUT "WHAT DATA FILE NAME?  ";RFILE$: IF RFILE$ = "" THE
 N 10230
14870  PRINT D$;"OPEN ";RFILE$
14880  PRINT D$;"READ ";RFILE$
14890  INPUT N,NV,FW,NC
14900  FOR J = 1 TO NV
14910    INPUT VAR$(J)
14920    FOR I = 1 TO N
14930      INPUT D(I,J)
14940    NEXT I
14950  NEXT J
14960  PRINT D$;"CLOSE ";RFILE$
14970  RETURN
14980
14990  REM  ** SAVE DATA TO DISK FILE **
15000  HOME : VTAB 3
15010  PRINT  TAB( 8);"SAVE DATA BASE TO DISK"
15020  PRINT  TAB( 8);"---- ---- ---- -- ----"
15030  PRINT : PRINT : PRINT
15040  HTAB 6: INPUT "SAVE UNDER WHAT FILE NAME?  ";SFILE$: IF SFILE$ =
 "" THEN  RETURN
15050  PRINT D$;"OPEN ";SFILE$;D$;"DELETE ";SFILE$;D$;"OPEN ";SFILE$
15060  PRINT D$;"WRITE ";SFILE$
15070  PRINT N: PRINT NV: PRINT FW: PRINT NC
15080  FOR J = 1 TO NV
15090    PRINT VAR$(J)
15100    FOR I = 1 TO N
15110      PRINT D(I,J)
15120    NEXT I
15130  NEXT J
15140  PRINT D$;"CLOSE ";SFILE$
15150  RETURN
15160
15170  REM  ** INTRODUCTORY REMARKS **
15180  HOME : VTAB 4
15190  PRINT "REGRESSION ANALYSIS IS USED TO EXAMINE  THE INTERRELATIONS
 HIPS AMONG TWO OR MOREVARIABLES FOR WHICH DATA IS AVAILABLE."
15200  PRINT : PRINT : PRINT
15210  PRINT "THE PACKAGE INCLUDES 3 INTERCONNECTED   MAIN PROGRAMS TO P
 ROVIDE:"
15220  PRINT : PRINT
15230  PRINT "    DATA BASE MANAGEMENT"
15240  PRINT
15250  PRINT "    BIVARIATE REGRESSION"
15260  PRINT
15270  PRINT "    MULTIVARIATE REGRESSION"
15280  GOSUB 10990
15290  RETURN
15300
```

```
15310  REM  ** DATA ENTRY INSTRUCTIONS **
15320  HOME : VTAB 3
15330  PRINT "THE DATA TABLEAU PERMITS EASY INPUT OF  DATA IN MATRIX FOR
 MAT."
15340  PRINT : PRINT
15350  PRINT "OBSERVATIONS DO NOT HAVE TO BE IN ANY   SPECIAL ORDER, BUT
  VARIABLE VALUES MUST STAY GROUPED BY OBSERVATION."
15360  PRINT : PRINT
15370  IF NC < NV THEN  PRINT "ENTER DATA ";NC;" VARIABLES AT A TIME."
15380  PRINT : PRINT
15390  PRINT "LATER YOU WILL DESIGNATE ONE VARIABLE ASBEING 'DEPENDENT'
 FOR CURVE FITTING."
15400  PRINT : PRINT
15410  PRINT "IF YOU MAKE AN ENTRY ERROR, GO AHEAD ANDFINISH....  YOU CA
 N CORRECT IT LATER ON."
15420  GOSUB 10900
15430  RETURN
```

```
10000  REM    *** B I V A R I A T E    R E G R E S S I O N ***
10010  REM   ** REGRESSION DATA PROGRAM MUST BE RUN FIRST **
10020
10030  TEXT
10040  IF BC > 0 THEN 11890: REM   - RE-ENTRY FOR HIRES GRAPH ROUTINE
10050 D$ =  CHR$ (13) +  CHR$ (4)
10060 BELL$ = "": REM - BELL
10070  IF CM > 0 THEN 10110
10080  PRINT D$;"BLOAD CHAIN, A520"
10090  CALL 520"REGR DATA"
10100
10110  DIM X(N),Y(N),A(7),B(7),R(8),BD$(8),M$(8),EQ$(8),T(2)
10120  DIM E(17),S(9,10),U(10),H(N)
10130  DEF  FN S(X) =  INT (X * 100000 + .5) / 100000: REM  - 5 PLACE RO
 UND OFF FUNCTION
10140  DEF  FN R(X) =  INT (X * 10000 + .5) / 10000: REM - 4 PLACE ROUND
  OFF FUNCTION
10150
10160  PRINT BELL$
10170  HOME
10180 X =  FRE (0): REM  -CLEAN UP UNUSED STRING SPACE
10190  PRINT "   BIVARIATE REGRESSION OPTIONS:"
10200  PRINT "   --------- ---------- -------"
10210  PRINT
10220  PRINT "1.  LINEAR         Y = A + (B*X)"
10230  PRINT
10240  PRINT "2.  POWER          Y = A * (X↑B)"
10250  PRINT
10260  PRINT "3.  EXPONENTIAL    Y = A * EXP(B*X)"
10270  PRINT
10280  PRINT "4.  LOGARITHMIC    Y = A + B*LOG(X)"
10290  PRINT
10300  PRINT "5.  HYPERBOLIC 1   Y = A + (B/X)"
10310  PRINT
10320  PRINT "6.  HYPERBOLIC 2   Y = 1/(A + B*X)"
10330  PRINT
10340  PRINT "7.  HYPERBOLIC 3   Y = X/(A + B*X)"
10350  PRINT
10360  PRINT "8.  N'TH ORDER     Y = A +B*X +C*X↑2+..."
10370  PRINT "9.  USER-DEFINED MODEL"
10380  PRINT
10390  PRINT "10. AUTOMATIC FIT USING MODELS 1-7 ABOVE"
10400  INPUT "      WHICH REGRESSION MODEL?  ";A$:BC =  VAL (A$): IF BC
 < 1 OR BC > 10 THEN  CALL  - 998: GOTO 10400
10410  IF BC >  < 9 THEN 11130
10420
10430  REM  - CREATE THE USER-DEFINED MODEL
```

```
10440  HOME : VTAB 7
10450  INPUT "NEED INSTRUCTIONS FOR DEFINING MODEL?  ";A$
10460  IF  LEFT$ (A$,1) = "Y"  THEN  GOSUB 16110
10470
10480  HOME
10490  PRINT "ORIGINAL DATA VARIABLES"
10500  PRINT "-------- ---- ---------"
10510  GOSUB 12360
10520 T = 20 - NV
10530  PRINT "HOW MANY TRANSFORM VARIABLES (1-";T;")?  ";: INPUT "";TV
10540  HOME : POKE 34,20
10550  PRINT "ENTER TRANSFORMS AS FUNCTIONS OF V(I):"
10560  PRINT
10570  FOR J = 1 TO TV
10580    VTAB 21: CALL  - 958
10590    PRINT "    T(";J;") = ";: INPUT "";VAR$(J + NV)
10600  PRINT : INPUT "** DO YOU NEED TO CHANGE IT?  ";A$: IF  LEFT$ (A$,
 1) = "Y" THEN  VTAB 21: CALL  - 958: GOTO 10590
10610  NEXT J
10620
10630 SFILE$ = "": GOSUB 15790: REM  - SAVE TEMPORARY STORAGE FILE
10640  GOTO 16540: REM   - WRITE TRANSFORM FILE
10650
10660  REM  - RE-ENTRY POINT AFTER TRANSFORMS ARE 'EXEC'D
10670  INPUT "";A$: REM - DUMMY INPUT TO PREVENT THE FIRST TRUE INPUT FR
 OM TRYING TO READ THE EXEC FILE
10680  CALL 43089: REM - RESETS DOS (48K MEMORY)
10690  REM      - NEXT 8 LINES REPLACE VALUES LOST AFTER 'EXEC'
10700 D$ =  CHR$ (13) +  CHR$ (4)
10710 BELL$ = ""
10720  DIM D(80,20),VAR$(20),X(80),Y(80),A(7),B(7),R(8),BD$(8)
10730  DIM M$(8),EQ$(8),T(2),E(17),S(9,10),U(10)
10740 BC = 9:PI = 3.14159265
10750  DEF  FN R(X) =  INT (X * 10000 + .5) / 10000: REM - 4 PLACE ROUND
  OFF FUNCTION
10760  DEF  FN S(X) =  INT (X * 100000 + .5) / 100000: REM   - 5 PLACE R
 OUND OFF FUNCTION
10770  GOSUB 15980: REM   - READ TEMPORARY STORAGE FILE SAVED EARLIER
10780  DIM H(N): REM - USED FOR HIRES GRAPHICS OPTION
10790
10800  REM - ENTER TRANSFORMED DATA INTO ENLARGED D(I,J) MATRIX
10810  FOR I = 1 TO N
10820    FOR J = 1 TO NV
10830     V(J) = D(I,J)
10840    NEXT J
10850    GOSUB 16640
10860    FOR J = NV + 1 TO NV + TV
10870     D(I,J) = T(J - NV)
10880    NEXT J
10890  NEXT I
10900
10910  PRINT BELL$ + BELL$
```

```
10920  TEXT : HOME : VTAB 2
10930  PRINT "THESE TRANSFORMS HAVE BEEN CREATED:"
10940  PRINT "----- ---------- ---- ---- --------"
10950  VTAB 7
10960  FOR J = 1 TO TV
10970    PRINT "  T(";J;") = ";VAR$(J + NV)
10980    IF TV < 7 THEN  PRINT
10990  NEXT J
11000  PRINT : GOSUB 12230: PRINT : PRINT
11010  PRINT "ENTER VARIABLE NAMES ABOVE (1-6 CHAR)..."
11020  POKE 33,10: POKE 32,30
11030  VTAB 5: PRINT "VAR NAME"
11040  VTAB 7
11050  FOR J = 1 TO TV
11060    INPUT "";A$:VAR$(J + NV) =  LEFT$ (A$,6)
11070    IF TV < 7 THEN  PRINT
11080  NEXT J
11090  POKE 32,0: POKE 33,40
11100  CALL  - 958
11110  GOSUB 12200
11120
11130  HOME
11140  PRINT "CURRENT VARIABLES:"
11150  PRINT "------- ---------"
11160  GOSUB 12360
11170 SFILE$ = ""
11180  IF TV = 0 THEN 11220
11190  INPUT "WANT TO SAVE THE COMPLETE DATA BASE TO  DISK, INCLUDING TR
 ANSFORM VARIABLES?  ";A$: IF  LEFT$ (A$,1) >  < "Y" THEN 11220
11200  PRINT : INPUT "SAVE UNDER WHAT NAME?  ";SFILE$: GOSUB 15790
11210
11220  HOME : INPUT "INDEX OF INDEPENDENT VARIABLE (X)?  ";A$:IV =  INT
 ( VAL (A$)): IF IV < 1 OR IV > NV + TV THEN  CALL  - 998: GOTO 11220
11230  PRINT
11240  INPUT "INDEX OF DEPENDENT VARIABLE   (Y)?  ";A$:DV =  INT ( VAL (
 A$)): IF DV < 1 OR DV > NV + TV OR DV = IV THEN  CALL  - 998: GOTO 1124
 0
11250  PRINT : INPUT "** DO YOU WANT TO MAKE CHANGES?  ";A$: IF  LEFT$ (
 A$,1) = "Y" THEN 11220
11260
11270  REM  - ENTER THE SELECTED VARIABLES INTO X AND Y ARRAYS AND COMPU
 TE SUMS AND SUMS OF SQUARES OF X AND Y
11280 X1 = 0:Y1 = 0:X2 = 0:Y2 = 0
11290  FOR I = 1 TO N
11300   X(I) = D(I,IV):Y(I) = D(I,DV)
11310   X1 = X1 + X(I):Y1 = Y1 + Y(I)
11320   X2 = X2 + X(I) * X(I):Y2 = Y2 + Y(I) * Y(I)
11330  NEXT I
11340
11350  REM - READ MODEL NAMES AND EQUATIONS
11360  FOR CF = 1 TO 7: READ M$(CF),EQ$(CF): NEXT CF
11370  DATA      LINEAR, Y = A + B*X, POWER, Y = A * X↑B, EXPONENTIAL,
  Y = A * EXP(B*X), LOGRITHMIC, Y = A + B*LOG(X), HYPERBOLIC 1, Y = A +
 (B/X), HYPERBOLIC 2, Y = 1/(A + B*X), HYPERBOLIC 3, Y = X/(A + B*X)
```

```
11380
11390  TEXT : HOME : VTAB 5
11400  INPUT "DO YOU WANT OUTPUT ROUTED TO PRINTER? ";P$:P$ =  LEFT$ (P$
 ,1)
11410  GOSUB 14710: REM - MEANS AND STANDARD DEVIATIONS
11420
11430  IF BC >  < 10 THEN 11680
11440  REM  - AUTO CURVE FIT, MODELS 1-7
11450 AUTFLAG = 1
11460  HOME : VTAB 3
11470  FOR BC = 1 TO 7
11480  ON BC GOSUB 12530,12720,12940,13160,13380,13600,13820
11490  NEXT BC
11500  IF P$ = "Y" THEN  GOSUB 12310
11510  PRINT BELL$ + BELL$
11520  HOME : VTAB 3
11530  PRINT "MODEL"; TAB( 22);"UNADJUSTED R↑2"
11540  PRINT "-----"; TAB( 22);"---------- ---"
11550  PRINT
11560  FOR BC = 1 TO 7
11570    PRINT BC;".  ";M$(BC); TAB( 22);
11580    IF BD$(BC) >  < "" THEN  PRINT "CANNOT FIT DATA": GOTO 11600
11590    PRINT  TAB( 25);R(BC)
11600    PRINT
11610  NEXT BC
11620  IF P$ = "Y" THEN  PRINT D$;"PR#0"
11630  VTAB 22
11640  INPUT "ENTER THE MODEL NUMBER IF YOU WANT ITS  EQUATION, OR HIT '
 RETURN' TO GO ON...";A$:BC =  VAL (A$): IF A$ = "" THEN 11910
11650  IF BC < 1 OR BC > 7 THEN  CALL  - 998: GOTO 11640
11660  GOTO 11700
11670
11680  HOME : VTAB 3
11690  ON BC GOSUB 12530,12720,12940,13160,13380,13600,13820,14040,12530

11700  GOSUB 14830
11710  REM - SKIP OPTIONS IF BAD DATA
11720  IF BD$(BC) >  < "" THEN 11890
11730
11740  HOME : VTAB 3: INPUT "DO YOU WANT A TABLE OF RESIDUALS?  ";A$
11750  IF  LEFT$ (A$,1) = "Y" THEN  GOSUB 15220
11760
11770  PRINT : PRINT : INPUT "DO YOU WANT TO PREDICT POINTS?  ";A$
11780  IF  LEFT$ (A$,1) = "Y" THEN  GOSUB 15480
11790
11800  PRINT : PRINT : INPUT "DO YOU WANT A HIRES GRAPH OF THE CURVE?";A
 $
11810  IF  LEFT$ (A$,1) >  < "Y" THEN 11890
11820  PRINT : PRINT : PRINT
11830  PRINT "** WHEN THE GRAPH IS COMPLETED, THE BELL";
11840  PRINT  TAB( 16);"WILL SOUND..."
11850  PRINT : PRINT "THEN HIT 'RETURN' FOR THE MAIN PROGRAM."
11860  FOR PAUSE = 1 TO 1000: NEXT PAUSE
```

```
11870  PRINT D$;"BLOAD CHAIN, A520"
11880  CALL 520"BIVAR HIRES"
11890  IF AUTFLAG = 1 THEN 11520
11900
11910  HOME : VTAB 5
11920  PRINT "     YOUR FINAL OPTIONS ARE:"
11930  PRINT "     ---- ----- ------- ---"
11940  PRINT : PRINT : PRINT
11950  PRINT "1.  ANOTHER REGRESSION W/ SAME DATA BASE"
11960  PRINT
11970  PRINT "2.  ENTER A NEW DATA BASE"
11980  PRINT
11990  PRINT "3.  EDIT CURRENT DATA BASE"
12000  PRINT : PRINT
12010  PRINT "4.  QUIT THE PROGRAM"
12020  VTAB 22: INPUT "    WHICH?  ";A$:FC =  VAL (A$): IF FC < 1 OR FC
 > 4 THEN  CALL  - 998: GOTO 12020
12030  IF FC = 2 THEN RDFLAG = 1
12040  IF FC = 3 THEN RDFLAG = 2
12050  IF FC = 2 OR FC = 3 THEN 10080
12060  IF FC = 4 THEN 12130
12070  REM   - ENSURE THAT THE LAST SET OF TRANSFORMS ARE NOW TREATED AS
  'ORIGINAL' VARIABLES
12080 NV = NV + TV:TV = 0
12090  RESTORE :AUTFLAG = 0
12100  FOR CF = 1 TO 7:BD$(CF) = "": NEXT CF
12110  GOTO 10170
12120
12130  PRINT D$;"BLOAD CHAIN, A520"
12140  CALL 520"CAAMM MASTER"
12150  END
12160
12170  REM      ******************************************************

12180
12190  REM   ** UTILITY SUBROUTINES **
12200  VTAB 23: GOSUB 12230: PRINT "      HIT 'RETURN' TO CONTINUE...";:
 GET A$: PRINT
12210  RETURN
12220
12230  PRINT "- - - - - - - - - - - - - - - - - - -   - -"
12240  RETURN
12250
12260  VTAB 23: GOSUB 12230: PRINT "HIT 'RETURN' TO GO ON, OR 'Q' TO QUI
 T ";
12270  GET A$: IF  ASC (A$) = 81 THEN 12130
12280  PRINT
12290  RETURN
12300
12310  REM  ** TURN PRINTER ON **
12320  PRINT D$;"PR#4": PRINT : POKE 1148,32: POKE 1788,40
12330  PRINT : GOSUB 12230: PRINT : PRINT
12340  RETURN
```

```
12350
12360  REM  ** VARIABLE LISTING **
12370  PRINT
12380  PRINT "INDEX"; TAB( 7);"LABEL"; TAB( 14);"NAME"
12390  PRINT
12400  FOR J = 1 TO NV + TV
12410    IF J > 10 THEN  POKE 33,20: POKE 32,20: VTAB J - 5
12420    PRINT  TAB( 3);J; TAB( 7);
12430    IF J > NV THEN  PRINT "T(";J - NV;")";: GOTO 12450
12440    PRINT "V(";J;")";
12450    PRINT  TAB( 14);VAR$(J)
12460  NEXT J
12470  POKE 32,0: POKE 33,40
12480  VTAB 16
12490  PRINT : GOSUB 12230: PRINT : PRINT
12500  POKE 34,18
12510  RETURN
12520
12530  REM  ** LINEAR MODEL **
12540  PRINT "RUNNING LINEAR"
12550 XY = 0
12560  FOR I = 1 TO N
12570   XY = XY + (X(I) * Y(I))
12580  NEXT I
12590 Q = XY - X1 * Y1 / N
12600 Q1 = X2 - X1 * X1 / N
12610 Q2 = Y2 - Y1 * Y1 / N
12620 R(1) = (Q * Q) / (Q1 * Q2)
12630 B(1) = Q / Q1
12640 A(1) = Y1 / N - B(1) * (X1 / N)
12650 ERR = Q2 - Q * Q / Q1
12660 SERR(1) =  SQR (ERR / (N - 2)): RETURN
12670 Y = A(1) + B(1) * X: RETURN
12680 X = (Y - A(1)) / B(1): RETURN
12690  PRINT "Y =  "; FN S(A(1));" + "; FN S(B(1));" * X"
12700  RETURN
12710
12720  REM   ** POWER MODEL **
12730  PRINT "RUNNING POWER"
12740 LX = 0:LY = 0:L1 = 0:L2 = 0:LB = 0
12750  FOR I = 1 TO N
12760    IF X(I) <  = 0 OR Y(I) <  = 0 THEN BD$(2) = "X OR Y": RETURN
12770   LX = LX +  LOG (X(I)):LY = LY +  LOG (Y(I))
12780   L1 = L1 + ( LOG (X(I)) *  LOG (X(I))):L2 = L2 + ( LOG (Y(I)) *
 LOG (Y(I)))
12790   LB = LB + ( LOG (X(I)) *  LOG (Y(I)))
12800  NEXT I
12810 Q = LB - LX * LY / N
12820 Q1 = L1 - LX * LX / N
12830 Q2 = L2 - LY * LY / N
12840 R(2) = (Q * Q) / (Q1 * Q2)
```

```
12850 B(2) = Q / Q1
12860 A(2) =  EXP ((LY / N) - (B(2) * (LX / N)))
12870 ERR = Q2 - Q * Q / Q1
12880 SERR(2) =  SQR (ERR / (N - 2)): RETURN
12890 Y = A(2) * X ↑ B(2): RETURN
12900 X = (Y / A(2)) ↑ (1 / B(2)): RETURN
12910  PRINT "Y =  "; FN S(A(2));" * X↑"; FN S(B(2))
12920  RETURN
12930
12940  REM  ** EXPONENTIAL MODEL **
12950  PRINT "RUNNING EXPONENTIAL"
12960 LY = 0:L2 = 0:XLY = 0
12970  FOR I = 1 TO N
12980    IF Y(I) <  = 0 THEN BD$(3) = "Y": RETURN
12990   LY = LY +  LOG (Y(I))
13000   L2 = L2 + ( LOG (Y(I)) *  LOG (Y(I)))
13010   XLY = XLY + (X(I) *  LOG (Y(I)))
13020  NEXT I
13030 Q = XLY - X1 * LY / N
13040 Q1 = X2 - X1 * X1 / N
13050 Q2 = L2 - LY * LY / N
13060 R(3) = (Q * Q) / (Q1 * Q2)
13070 B(3) = Q / Q1
13080 A(3) =  EXP ((LY / N) - (B(3) * (X1 / N)))
13090 ERR = Q2 - Q * Q / Q1
13100 SERR(3) =  SQR (ERR / (N - 2)): RETURN
13110 Y = A(3) *  EXP (B(3) * X): RETURN
13120 X =  LOG (Y / A(3)) / B(3): RETURN
13130  PRINT "Y =  "; FN S(A(3));" * EXP("; FN S(B(3));" * X)"
13140  RETURN
13150
13160  REM  ** LOGARITHMIC MODEL **
13170  PRINT "RUNNING LOGARITHMIC"
13180 LX = 0:L1 = 0:YLX = 0
13190  FOR I = 1 TO N
13200    IF X(I) <  = 0 THEN BD$(4) = "X": RETURN
13210   LX = LX +  LOG (X(I))
13220   L1 = L1 + ( LOG (X(I)) *  LOG (X(I)))
13230   YLX = YLX + (Y(I) *  LOG (X(I)))
13240  NEXT I
13250 Q = YLX - LX * Y1 / N
13260 Q1 = L1 - LX * LX / N
13270 Q2 = Y2 - Y1 * Y1 / N
13280 R(4) = (Q * Q) / (Q1 * Q2)
13290 B(4) = Q / Q1
13300 A(4) = (Y1 - (B(4) * LX)) / N
13310 ERR = Q2 - Q * Q / Q1
13320 SERR(4) =  SQR (ERR / (N - 2)): RETURN
13330 Y = A(4) + B(4) *  LOG (X): RETURN
13340 X =  EXP ((Y - A(4)) / B(4)): RETURN
13350  PRINT "Y =  "; FN S(A(4));" + "; FN S(B(4));" * LOG(X)"
13360  RETURN
```

```
13370
13380  REM  ** HYPERBOLIC 1  MODEL **
13390  PRINT "RUNNING HYPERBOLIC 1"
13400 IX = 0:XI2 = 0:SYIX = 0
13410  FOR I = 1 TO N
13420    IF X(I) = 0 THEN BD$(5) = "X": RETURN
13430   IX = IX + 1 / X(I)
13440   XI2 = XI2 + (1 / X(I)) * (1 / X(I))
13450   SYIX = SYIX + Y(I) / X(I)
13460  NEXT I
13470 Q = SYIX - IX * Y1 / N
13480 Q1 = XI2 - IX * IX / N
13490 Q2 = Y2 - Y1 * Y1 / N
13500 R(5) = (Q * Q) / (Q1 * Q2)
13510 B(5) = Q / Q1
13520 A(5) = Y1 / N - B(5) * IX / N
13530 ERR = Q2 - Q * Q / Q1
13540 SERR(5) =  SQR (ERR / (N - 2)): RETURN
13550 Y = A(5) + B(5) / X: RETURN
13560 X = (A(5) + B(5)) / Y: RETURN
13570  PRINT "Y =  "; FN S(A(5));" + "; FN S(B(5));" / X"
13580  RETURN
13590
13600  REM  ** HYPERBOLIC 2  MODEL **
13610  PRINT "RUNNING HYPERBOLIC 2"
13620 IY = 0:YI2 = 0:SXIY = 0
13630  FOR I = 1 TO N
13640    IF Y(I) = 0 THEN BD$(6) = "Y": RETURN
13650   IY = IY + 1 / Y(I)
13660   YI2 = YI2 + (1 / Y(I)) * (1 / Y(I))
13670   SXIY = SXIY + X(I) / Y(I)
13680  NEXT I
13690 Q = SXIY - IY * X1 / N
13700 Q1 = X2 - X1 * X1 / N
13710 Q2 = YI2 - IY * IY / N
13720 R(6) = (Q * Q) / (Q1 * Q2)
13730 B(6) = Q / Q1
13740 A(6) = IY / N - B(6) * X1 / N
13750 ERR = Q2 - Q * Q / Q1
13760 SERR(6) =  SQR (ERR / (N - 2)): RETURN
13770 Y = 1 / (A(6) + B(6) * X): RETURN
13780 X = B(6) / Y - A(6) / B(6): RETURN
13790  PRINT "Y =  1 / ("; FN S(A(6));" + "; FN S(B(6));" * X)"
13800  RETURN
13810
13820  REM  ** HYPERBOLIC 3  MODEL **
13830  PRINT "RUNNING HYPERBOLIC 3"
13840 IX = 0:IY = 0:XI2 = 0:YI2 = 0:IB = 0
13850  FOR I = 1 TO N
13860    IF X(I) = 0 OR Y(I) = 0 THEN BD$(7) = "X OR Y": RETURN
13870   IX = IX + 1 / X(I):IY = IY + 1 / Y(I)
13880   XI2 = XI2 + (1 / X(I)) * (1 / X(I)):YI2 = YI2 + (1 / Y(I)) * (1
 / Y(I))
```

```
13890   IB = IB + 1 / (X(I) * Y(I))
13900  NEXT I
13910 Q = IB - IX * IY / N
13920 Q1 = XI2 - IX * IX / N
13930 Q2 = YI2 - IY * IY / N
13940 R(7) = (Q * Q) / (Q1 * Q2)
13950 A(7) = Q / Q1
13960 B(7) = IY / N - A(7) * IX / N
13970 ERR = Q2 - Q * Q / Q1
13980 SERR(7) =  SQR (ERR / (N - 2)): RETURN
13990 Y = X / (A(7) + B(7) * X): RETURN
14000 X = A(7) / (1 / Y - B(7)): RETURN
14010  PRINT "Y =  X / ("; FN S(A(7));" + "; FN S(B(7));" * X)"
14020  RETURN
14030
14040  REM  ** N'TH ORDER MODEL **
14050  HOME : VTAB 2
14060  PRINT "N'TH ORDER"
14070  PRINT "---- -----"
14080  VTAB 8
14090 HD = N - 1: IF HD > 8 THEN HD = 8
14100  PRINT "DEGREE OF EQUATION (1-";HD;")?  ";: INPUT "";A$:D =  INT (
  VAL (A$))
14110  IF D < N AND D <  = 8 THEN 14130
14120  CALL  - 998: CALL  - 998: CALL  - 958: GOTO 14100
14130  VTAB 14: PRINT "** RUNNING ORDER ";D;" REGRESSION..."
14140  FOR I = 1 TO 2 * D + 1:E(I) = 0: NEXT I
14150  FOR J = 1 TO D + 2:U(J) = 0: FOR I = 1 TO D + 1:S(I,J) = 0: NEXT
 I: NEXT J
14160 E(1) = N
14170  FOR I = 1 TO N
14180    FOR J = 2 TO 2 * D + 1
14190     E(J) = E(J) + X(I) ↑ (J - 1)
14200    NEXT J
14210    FOR K = 1 TO D + 1
14220     S(K,D + 2) = U(K) + Y(I) * X(I) ↑ (K - 1)
14230     U(K) = U(K) + Y(I) * X(I) ↑ (K - 1)
14240    NEXT K
14250   U(D + 2) = U(D + 2) + Y(I) ↑ 2
14260  NEXT I
14270  FOR J = 1 TO D + 1
14280    FOR K = 1 TO D + 1
14290     S(J,K) = E(J + K - 1)
14300    NEXT K
14310  NEX1 J
14320  GOSUB 14500: REM - SOLVE SIMULTANEOUS EQUATIONS
14330  HOME : VTAB 2: PRINT BELL$ + BELL$
14340  PRINT "N'TH ORDER    Y = A +BX +CX↑2 +DX↑3 +..."
14350  GOSUB 12230
14360  PRINT : PRINT
14370  PRINT "              CONSTANT = "; TAB( 25); FN S(S(1,D + 2))
14380  FOR J = 1 TO D
14390  PRINT J;" DEGREE COEFFICIENT = "; TAB( 25); FN S(S(J + 1,D + 2))
14400  NEXT J
```

```
14410 Q = 0
14420  FOR J = 2 TO D + 1:Q = Q + S(J,D + 2) * (U(J) - E(J) * U(1) / N):
 NEXT J
14430 Q2 = U(D + 2) - U(1) ↑ 2 / N
14440 ERR = Q2 - Q
14450 R(8) = Q / Q2
14460 SERR(8) =  SQR (ERR / (N - D - 1))
14470  RETURN
14480 Y = S(1,D + 2): FOR J = 1 TO D:::Y = Y + S(J + 1,D + 2) * X ↑ J: N
 EXT J: RETURN
14490
14500  REM  ** GAUSS-JORDAN EQUATION SOLVER W/ PARTIAL PIVOTING **
14510  FOR J = 1 TO D + 1
14520   T =  - 1
14530    FOR K = J TO D + 1
14540      IF T >  ABS (S(K,J)) THEN 14570
14550     T =  ABS (S(K,J))
14560     I1 = K
14570    NEXT K
14580    FOR I = 1 TO D + 2:SR = S(I1,I):S(I1,I) = S(J,I):S(J,I) = SR: N
 EXT I
14590 Z = S(J,J)
14600    FOR I = 1 TO D + 2:S(J,I) = S(J,I) / Z: NEXT I
14610    FOR K = 1 TO D + 1
14620      IF K = J THEN 14670
14630     Z = S(K,J)
14640      FOR I = 1 TO D + 2
14650       S(K,I) = S(K,I) - Z * S(J,I)
14660      NEXT I
14670    NEXT K
14680  NEXT J
14690  RETURN
14700
14710  REM   ** TABLE OF MEANS/ STD DEVIATIONS **
14720 XMEAN = X1 / N:YMEAN = Y1 / N
14730 XSD =  SQR ((X2 - X1 * X1 / N) / (N - 1))
14740 YSD =  SQR ((Y2 - Y1 * Y1 / N) / (N - 1))
14750  HOME : VTAB 5
14760  IF P$ = "Y" THEN  GOSUB 12310
14770  PRINT "VAR"; TAB( 18);"UNBIASED"; TAB( 30);"UNBIASED"
14780  PRINT "NAME"; TAB( 8);"MEAN"; TAB( 18);"VARIANCE"; TAB( 30);"STD
 DEV"
14790  PRINT "----"; TAB( 8);"----"; TAB( 18);"--------"; TAB( 30);"---
 ---"
14800  PRINT
14810  PRINT VAR$(IV); TAB( 8); FN R(XMEAN); TAB( 18); FN R(XSD * XSD);
 TAB( 30); FN R(XSD)
14820  PRINT
14830  PRINT VAR$(DV); TAB( 8); FN R(YMEAN); TAB( 18); FN R(YSD * YSD);
 TAB( 30); FN R(YSD)
14840  IF P$ = "Y" THEN  PRINT D$;"PR#0"
14850  GOSUB 12200
14860  RETURN
```

```
14870
14880  REM   ** CURVE FIT STATISTICS **
14890  IF P$ = "Y" THEN  GOSUB 12310
14900  IF BC = 8 THEN 15040
14910  HOME : VTAB 2: IF AUTFLAG = 0 THEN  PRINT BELL$ + BELL$
14920  IF BC = 9 THEN BC = 1: PRINT "USER DEFINED"; TAB( 16);VAR$(DV);"
 = A + B*";VAR$(IV): GOTO 14940
14930  PRINT M$(BC); TAB( 18);EQ$(BC)
14940  PRINT
14950  GOSUB 12230
14960  PRINT : PRINT
14970  IF BD$(BC) = "" THEN 15020
14980  VTAB 10
14990  PRINT "** THIS MODEL CANNOT FIT DATA WHERE THE ";BD$(BC);" VALUES
  ARE ZERO";
15000  IF BC < 5 THEN  PRINT " OR NEGATIVE.": GOTO 15190
15010  IF BC <  = 7 THEN  PRINT ".": GOTO 15190
15020  ON BC GOSUB 12690,12910,13130,13350,13570,13790,14010
15030  POKE 33,38: POKE 32,2
15040  PRINT : PRINT
15050  PRINT "UNADJUSTED R↑2            ="; TAB( 28 ); FN S( R( BC ))
15060  PRINT "CORRELATION COEFFICIENT ="; TAB( 28 ); FN S(  SQR (R( BC)))
15070  PRINT
15080  PRINT "STD ERROR OF ESTIMATE   ="; TAB( 28 ); FN S( SERR( BC ))
15090  PRINT "VARIANCE OF ESTIMATE    ="; TAB( 28 ); FN S( SERR( BC ) * SERR
 ( BC ))
15100  PRINT
15110  PRINT "DEGREES OF FREEDOM      =" TAB( 28 );: IF BC = 8 THEN  PRIN
 T N - D - 1: GOTO 15140
15120  PRINT N - 2
15130  POKE 32,0: POKE 33,40
15140  PRINT : PRINT
15150  PRINT "WITH  X = ";VAR$( IV ); TAB( 22);" AND Y = ";VAR$( DV)
15160  PRINT
15170  IF P$ = "Y" THEN  PRINT D$;"PR#0"
15180  IF BC = 8 THEN  INPUT "WANT ANOTHER RUN?  ";A$:A$ =  LEFT$ (A$,1)
 : IF A$ = "Y" THEN 11680
15190  GOSUB 12200
15200  RETURN
15210
15220  REM   ** TABLE OF PREDICTED VALUES AND RESIDUALS **
15230  HOME : VTAB 2: POKE 34,5
15240  IF P$ = "Y" THEN  GOSUB 12310
15250  PRINT  TAB( 4);"ACTUAL"; TAB( 11);"ACTUAL"; TAB( 18 );"PREDICTED";
  TAB( 30 );"RESIDUAL"
15260  PRINT  TAB( 4);" 'X'"; TAB( 11 );" 'Y'"; TAB( 18 );"   'Y'"
15270  PRINT  TAB( 4);"------"; TAB( 11);"------"; TAB( 18 );"---------";
  TAB( 30 );"--------"
15280  PRINT
15290  FOR I = 1 TO N
15300    PRINT I; TAB( 4); FN R(X( I)); TAB( 11 ); FN R( Y( I ));
15310   X = X( I )
```

```
15320     ON BC GOSUB 12670,12890,13110,13330,13550,13770,13990,14480
15330     PRINT  TAB( 18); FN R(Y); TAB( 30 ); FN R( Y(I) - Y )
15340     IF I = N THEN 15370
15350     IF I / 5 =  INT ( I / 5 ) THEN  PRINT
15360     IF I / 15 =  INT ( I / 15 ) THEN  GOSUB 12200: HOME
15370  NEXT I
15380  IF P$ = "Y" THEN  PRINT D$;"PR#0"
15390  GOSUB 12200
15400  HOME : VTAB 6
15410  PRINT  TAB( 4);"MEAN X"
15420  PRINT
15430 X = XMEAN: ON BC GOSUB 12670,12890,13110,13330,13550,13770,13990,1
 4480
15440  PRINT  TAB( 4); FN S(XMEAN); TAB( 18); FN S(Y)
15450  GOSUB 12200
15460  TEXT : HOME : RETURN
15470
15480  REM  ** INTERPOLATION/EXTRAPOLATION **
15490  HOME : VTAB 2
15500  IF BC = 8 THEN  PRINT "TO PREDICT Y VALUES WITH THE N'TH ORDER MO
 DEL...": GOTO 15540
15510  PRINT "TO PREDICT X OR Y VALUES WITH THE MODEL"
15520  PRINT
15530  ON BC GOSUB 12690,12910,13130,13350,13570,13790,14010
15540  PRINT : PRINT
15550  PRINT "     ENTER  'X= #'   TO PREDICT Y": IF BC = 8 THEN 15580
15560  PRINT
15570  PRINT "        OR  'Y= #'   TO PREDICT X"
15580  PRINT : PRINT
15590  PRINT "OR TYPE 'RETURN' WITH NO ENTRY TO GO ON."
15600  GOSUB 12230: IF BC = 8 THEN  POKE 34,10: GOTO 15620
15610  POKE 34,14
15620  PRINT
15630  INPUT A$
15640  IF P$ = "Y" THEN  GOSUB 12310
15650  IF A$ = "" THEN  PRINT D$;"PR#0": TEXT : HOME : RETURN
15660  IF BC = 8 THEN 15680
15670  IF  LEFT$ (A$,2) = "Y=" THEN 15720
15680  IF  LEFT$ (A$,2) >  < "X=" THEN 15750
15690 X =  VAL ( MID$ (A$,3)): ON BC GOSUB 12670,12890,13110,13330,13550
 ,13770,13990,7420
15700  CALL  - 998: PRINT "AT  X = ";X; TAB( 22);"Y = ";Y
15710  GOTO 15620
15720 Y =  VAL ( MID$ (A$,3)): ON BC GOSUB 12680,12900,13120,13340,13560
 ,13780,14000
15730  CALL  - 998: PRINT "AT  Y = ";Y; TAB( 22);"X = ";X
15740  GOTO 15620
15750  PRINT : PRINT "** IMPROPER FORMAT.  PLEASE RE-ENTER..."
15760  FOR PAUSE = 1 TO 1000: NEXT PAUSE
15770  CALL  - 998: CALL  - 958: GOTO 15630
15780
```

```
15790  REM    ** SAVE DISK FILE **
15800  IF SFILE$ >  < "" THEN NV = NV + TV: GOTO 15820
15810 SFILE$ = "TEMPFILE"
15820  PRINT D$;"OPEN ";SFILE$;D$;"DELETE ";SFILE$;D$;"OPEN ";SFILE$
15830  PRINT D$;"WRITE ";SFILE$
15840  PRINT N: PRINT NV
15850  FOR J = 1 TO NV
15860    PRINT VAR$(J)
15870    FOR I = 1 TO N: PRINT D( I,J): NEXT I
15880  NEXT J
15890  IF SFILE$ >  < "TEMPFILE" THEN 15930
15900  PRINT TV
15910  FOR J = 1 TO TV: PRINT VAR$( NV + J): NEXT J
15920  GOTO 15940
15930 NV = NV - TV
15940  PRINT D$;"CLOSE ";SFILE$
15950 SFILE$ = ""
15960  RETURN
15970
15980  REM    ** READ DISK FILE **
15990  PRINT D$;"OPEN TEMPFILE"
16000  PRINT D$;"READ TEMPFILE"
16010  INPUT N,NV
16020  FOR J = 1 TO NV
16030    INPUT VAR$(J)
16040    FOR I = 1 TO N: INPUT D( I,J): NEXT I
16050  NEXT J
16060  INPUT TV
16070  FOR J = 1 TO TV: INPUT VAR$( NV + J): NEXT J
16080  PRINT D$;"CLOSE TEMPFILE"
16090  RETURN
16100
16110  REM  ** USER MODEL INSTRUCTIONS **
16120  HOME
16130  PRINT "DEFINING A MODEL ALLOWS YOU TO FIT DATA TO VIRTUALLY ANY A
 LGEBRAIC CURVE THAT   CAN BE EXPRESSED AS A TRANSFORMATION OF THE BASIC
  LINEAR MODEL, I.E."
16140  PRINT
16150  PRINT "Y = A + B*X    ===>     T(Y) = A + B*T(X)"
16160  PRINT "YOUR DEFINED MODEL WILL USE 'TRANSFORM' VARIABLES THAT ARE
  FUNCTIONS OF ONE OR  BOTH OF THE ORIGINAL DATA VARIABLES."
16170  PRINT
16180  PRINT
16190  PRINT "THIS MODEL, FOR EXAMPLE..."
16200  PRINT
16210  PRINT  TAB( 5);"LOG(V(2)) = A + B*LOG(V(1)↑2)"
16220  PRINT
16230  PRINT "REQUIRES THE TRANSFORM VARIABLES:"
16240  PRINT
16250  PRINT "       T(1) = LOG(V(1)↑2)"
16260  PRINT "       T(2) = LOG(V(2))"
16270  PRINT : PRINT
```

```
16280  INPUT "YOU MUST USE LEGAL FUNCTIONS AND SYNTAX.              NEED
  HELP?  ";A$
16290  IF  LEFT$ (A$,1) <  > "Y" THEN  RETURN
16300
16310  REM   ** LEGAL FUNCTIONS AND SYNTAX **
16320  HOME : VTAB 2
16330  PRINT "FUNCTION"; TAB( 14);"COMMENTS"
16340  PRINT "--------"; TAB( 14);"--------"
16350  PRINT
16360  PRINT "SIN(X)"
16370  PRINT "COS(X)"
16380  PRINT "TAN(X)"; TAB( 10 );"X CAN'T BE A MULTIPLE OF PI/2"
16390  PRINT "ATN(X)"
16400  PRINT "ABS(X)"
16410  PRINT "SQR(X)"; TAB( 10 );"SQUARE ROOT; X MUST BE >= 0"
16420  PRINT "EXP(X)"; TAB( 10 );"'E' RAISED TO THE POWER OF X"
16430  PRINT "LOG(X)"; TAB( 10 );"NATURAL LOG; X MUST BE > 0"
16440  PRINT : PRINT
16450  PRINT "** PARENTHESES MUST BE CLOSED"
16460  PRINT
16470  PRINT "** EXPONENTIATE USING '↑',  E.G. 'X↑2'"
16480  PRINT
16490  PRINT "** THE VALUE 'PI' CAN BE ENTERED AS 'PI' "
16500  PRINT "** TRIG FUNCTIONS EXPECT X IN RADIANS"
16510  GOSUB 12200
16520  RETURN
16530
16540  REM   ** WRITE TRANSFORM FILES **
16550    PRINT D$;"OPEN TF";D$;"DELETE TF";D$;"OPEN TF"
16560    PRINT D$;"WRITE TF"
16570  FOR J = 1 TO TV
16580    PRINT 16650 + J * 10;" T           (";J;") = ";VAR$(J + NV)
16590  NEXT J
16600  PRINT 16650 + J * 10;" RETURN"
16610  PRINT "GOTO 10660"
16620  PRINT D$;"EXEC TF"
16630  END
16640  REM    ** TRANSFORMS **
16650  REM  - USER-DEFINED TRANSFORMATION LINES CREATED TO START HERE
```

Section 4, BIVAR HIRES

```
10000  REM     *** HIRES GRAPH OF BIVARIATE  CURVE ***
10010  REM  ** REGRESSION DATA AND BIVARIATE REGRESSION MUST BE RUN FIRS
 T **
10020
10030 D$ =  CHR$ (13) +  CHR$ (4)
10040 B$ = "": REM - BELL
10050  IF BC > 0 THEN 10080
10060  PRINT D$;"BLOAD CHAIN, A520"
10070  CALL 520"REGR DATA"
10080  TEXT : HOME
10090 PLFLAG = 0
10100 X =  FRE (0): REM  -CLEAN UP UNUSED STRING SPACE
10110
10120  REM   - TURN ON HIRES PAGE 2 FULL SCREEN
10130  HGR2
10140
10150  REM  - PLOT GRAPH BOUNDARY AND TICK MARKS
10160  HCOLOR= 6
10170  HPLOT 40,10 TO 260,10 TO 260,170 TO 40,170 TO 40,10
10180  FOR I = 0 TO 10
10190    HPLOT 36,10 + I * 16 TO 40,10 + I * 16
10200    HPLOT 260,10 + I * 16 TO 264,10 + I * 16
10210    HPLOT 40 + I * 22,6 TO 40 + I * 22,10
10220    HPLOT 40 + I * 22,170 TO 40 + I * 22,174
10230  NEXT I
10240
10250  REM -      SORT X & Y ARRAYS TO FIND LEAST AND GREATEST X & Y
10260  FOR I = 1 TO N:H(I) = X(I): NEXT
10270  GOSUB 10690
10280 EX = (H(N) - H(1)) / 15:BX = H(1) - EX:UX = H(N) + EX
10290  FOR I = 1 TO N:H(I) = Y(I): NEXT
10300  GOSUB 10690
10310 EX = (H(N) - H(1)) / 15:BY = H(1) - EX:UY = H(N) + EX: REM - PROVI
 DE A 15% BORDER AROUND DATA POINTS
10320 HS = 220 / (UX - BX):VS = 160 / (UY - BY)
10330
10340  REM -     PLOT CROSS HAIRS FOR ORIGIN, IF WITHIN BOUNDARIES
10350  HCOLOR= 5
10360  IF 0 > BX THEN H0 = 40 +  ABS (BX) * HS:H0 =  INT (H0 / 2 - .5) *
  2 + 1: HPLOT H0,10 TO H0,170
10370  IF 0 > BY THEN V0 = 10 + UY * VS: HPLOT 40,V0 TO 260,V0
10380
10390  REM -     PLOT DATA POINTS
10400  HCOLOR= 7
10410  FOR I = 1 TO N
10420 XP = (X(I) - BX) * HS + 40:YP = (UY - Y(I)) * VS + 10
10430    HPLOT XP,YP TO XP + 1,YP
10440  NEXT I
```

```
10450
10460  REM -     PLOT CURVE
10470 INCR = (UX - BX) / 220
10480  FOR I =  INT (BX) TO  INT (UX + .5) STEP INCR
10490 X = I: ON BC GOSUB 10790,10800,10810,10820,10830,10840,10850,10860

10500 XP = (X - BX) * HS + 40:YP = (UY - Y) * VS + 10
10510  IF XP < 40 OR XP > 260 OR YP < 10 OR YP > 170 THEN PLFLAG = 0: GO
 TO 10540
10520  IF PLFLAG = 0 THEN  HPLOT XP,YP:PLFLAG = 1: GOTO 10540
10530    HPLOT  TO XP,YP
10540  NEXT I
10550
10560  PRINT BELL$ + BELL$: GET A$
10570  TEXT : HOME : VTAB 7
10580  PRINT "DO YOU WANT TO SAVE THE GRAPH FOR FUTURE"
10590  PRINT  TAB( 5);"REFERENCE ?   ";: INPUT "";A$: IF  LEFT$ (A$,1) >
   < "Y" THEN 10620
10600  PRINT : PRINT : INPUT "SAVE UNDER WHAT NAME ?  ";HG$
10610  PRINT D$;"BSAVE ";HG$;",A$4000, L$2000": REM - SAVES ON HIRES PAG
 E 2
10620
10630  PRINT D$;"BLOAD CHAIN, A520"
10640  CALL 520"BIVAR REGR"
10650  END
10660
10670  REM  **********************************************************

10680
10690  REM -     SORTING ROUTINE
10700  FOR I = 1 TO N - 1
10710    FOR J = I + 1 TO N
10720      IF H(I) <  = H(J) THEN 10740
10730     RS = H(I):H(I) = H(J):H(J) = RS
10740    NEXT J
10750  NEXT I
10760  RETURN
10770
10780  REM CURVE FIT MODEL SUBROUTINES
10790 Y = A(1) + B(1) * X: RETURN
10800 Y = A(2) * X ↑ B(2): RETURN
10810 Y = A(3) *  EXP (B(3) * X): RETURN
10820 Y = A(4) + B(4) *  LOG (X): RETURN
10830 Y = A(5) + B(5) / X: RETURN
10840 Y = 1 / (A(6) + B(6) * X): RETURN
10850 Y = X / (A(7) + B(7) * X): RETURN
10860 Y = S(1,D + 2): FOR J = 1 TO D:Y = Y + S(J + 1,D + 2) * X ↑ J: NEX
 T J: RETURN
```

Section 5, MULVAR REGR

```
10000  REM   *** M U L T I V A R I A T E   R E G R E S S I O N ***
10010  REM  ** REGRESSION DATA MUST BE RUN FIRST **
10020
10030 D$ =  CHR$ (13) +  CHR$ (4)
10040 BELL$ = "": REM -BELL
10050  IF CM > 0 THEN 10090
10060  PRINT D$;"BLOAD CHAIN, A520"
10070  CALL 520"REGR DATA"
10080
10090  DIM A(20,20),C(20,20),R(20,20),U(20,20),SD(20),M(20),Q(80)
10100  DIM RV(20),X(20,20),B(20),P$(6),S(20),T(10)
10110  DEF  FN R(X) =  INT (X * 10000 + .5) / 10000: REM         - 4 PLAC
 E ROUND OFF FUNCTION
10120  DEF  FN S(X) =  INT (X * 100000 + .5) / 100000: REM - 5 PLACE ROU
 ND OFF FUNCTION
10130
10140  PRINT BELL$
10150  TEXT : HOME : POKE 33,38: POKE 32,2
10160  VTAB 3
10170 X =  FRE (0): REM - CLEAN UP UNUSED STRING SPACE
10180  PRINT "MULTIVARIATE CURVE FITTING OPTIONS:"
10190  PRINT "------------ ----- ------- -------"
10200  PRINT : PRINT : PRINT
10210  PRINT  TAB( 7);"1.  LINEAR"
10220  PRINT
10230  PRINT "VY = A + B*V1 + C*V2 + D*V3 + ..."
10240  PRINT : PRINT   PRINT
10250  PRINT  TAB( 7);"2.  USER-DEFINED"
10260  PRINT
10270  PRINT "TY = A + B*T1 + C*T2 + D*T3 + ..."
10280  TEXT
10290  VTAB 21: HTAB 7: INPUT "WHICH REGRESSION MODEL?  ";A$:MC =  VAL (
 A$): IF A$ = "" OR MC = 1 THEN 11020
10300  IF MC >  < 2 THEN  CALL  - 998: GOTO 10290
10310
10320  REM - CREATE THE USER DEFINED MODEL
10330  HOME : VTAB 7
10340  INPUT "NEED INSTRUCTIONS FOR DEFINING MODEL?  ";A$
10350  IF  LEFT$ (A$,1) = "Y" THEN  GOSUB 15110
10360
10370  HOME
10380  PRINT "ORIGINAL DATA VARIABLES"
10390  PRINT "-------- ---- ---------"
10400  GOSUB 13360
10410 T = 20 - NV
10420  PRINT "HOW MANY TRANSFORM VARIABLES (1-";T;")?  ";: INPUT "";A$:T
 V =  VAL (A$): IF TV < 1 OR TV > T THEN  CALL  - 998: GOTO 10420
```

```
10430  HOME : POKE 34,20
10440  PRINT "ENTER TRANSFORMS AS FUNCTIONS OF V(I):"
10450  PRINT
10460  FOR J = 1 TO TV
10470    VTAB 21: CALL  - 958
10480    PRINT "   T(";J;") = ";: INPUT "";VAR$(J + NV)
10490  PRINT : INPUT "** DO YOU NEED TO CHANGE IT?  ";A$: IF  LEFT$ (A$,
 1) = "Y" THEN  VTAB 21: CALL  - 958: GOTO 10480
10500  NEXT J
10510  PRINT
10520
10530 SFILE$ = "": GOSUB 14790: REM  - SAVE TEMPORARY STORAGE FILE
10540  GOTO 15550: REM  - WRITE TRANSFORM FILE
10550
10560  REM  - RE-ENTRY POINT AFTER TRANSFORMS ARE 'EXEC'D
10570  INPUT "";A$: REM - DUMMY INPUT TO PREVENT THE FIRST TRUE INPUT FR
 OM TRYING TO READ THE EXEC FILE
10580  CALL 43089: REM - RESETS DOS (48K MEMORY)
10590  REM   - NEXT 8 LINES REPLACE VALUES LOST AFTER 'EXEC'
10600 D$ =  CHR$ (13) +  CHR$ (4)
10610 BELL$ = ""
10620  DIM A(20,20),C(20,20),R(20,20),U(20,20),SD(20),M(20),Q(80),S(20)
10630  DIM RV(20),X(20,20),B(20),P$(5),VAR$(20),D(80,20),V(20),T(20)
10640 MC = 2:PI = 3.14159265
10650  DEF  FN R(X) =  INT (X * 10000 + .5) / 10000: REM        - 4 PLAC
 E ROUND OFF FUNCTION
10660  DEF  FN S(X) =  INT (X * 100000 + .5) / 100000: REM  - 5 PLACE RO
 UND OFF FUNCTION
10670  GOSUB 14980: REM - READ TEMPORARY STORAGE FILE SAVED EARLIER
10680
10690  REM  - ENTER TRANSFORMED DATA INTO ENLARGED D(I,J) MATRIX
10700  FOR I = 1 TO N
10710    FOR J = 1 TO NV
10720     V(J) = D(I,J)
10730    NEXT J
10740    GOSUB 15650
10750    FOR J = NV + 1 TO NV + TV
10760     D(I,J) = T(J - NV)
10770    NEXT J
10780  NEXT I
10790
10800  PRINT BELL$
10810  TEXT : HOME : VTAB 2
10820  PRINT "THESE TRANSFORMS HAVE BEEN CREATED!"
10830  PRINT "----- ---------- ---- ---- --------"
10840  VTAB 7
10850  FOR J = 1 TO TV
10860    PRINT "  T(";J;") = ";VAR$(J + NV)
10870    IF TV < 7 THEN  PRINT
10880  NEXT J
10890  PRINT : GOSUB 13200: PRINT : PRINT
10900  PRINT "ENTER VARIABLE NAMES ABOVE (1-6 CHAR)..."
```

```
10910  POKE 33,10: POKE 32,30
10920  VTAB 5: PRINT "VAR NAME"
10930  VTAB 7
10940  FOR J = 1 TO TV
10950    INPUT "";A$:VAR$( J + NV) =  LEFT$ (A$,6)
10960    IF TV < 7 THEN  PRINT
10970  NEXT J
10980  POKE 32,0: POKE 33,40
10990  CALL  - 958
11000  GOSUB 13170
11010
11020  HOME
11030  PRINT "CURRENT VARIABLES:"
11040  PRINT "------- ---------"
11050  GOSUB 13360
11060  IF TV = 0 THEN 11090
11070  INPUT "WANT TO SAVE THE COMPLETE DATA BASE TO  DISK, INCLUDING TR
 ANSFORM VARIABLES?  ";A$: IF  LEFT$ (A$,1) >  < "Y" THEN 11090
11080  PRINT : INPUT "SAVE UNDER WHAT NAME?  ";SFILE$: GOSUB 14790
11090  IF NV < 3 THEN IV = 1:M = 2: GOTO 11190
11100  HOME : PRINT "YOU DON'T HAVE TO USE ALL THE VARIABLES.SELECT ONLY
  THOSE NEEDED FOR THIS FIT..."
11110  PRINT
11120  INPUT "HOW MANY INDEPENDENT VARIABLES?  ";A$:IV =  INT ( VAL (A$)
 ): IF IV < 1 OR IV > NV - 1 THEN  CALL  - 998: GOTO 11120
11130 M = IV + 1
11140  IF N > M THEN 11180
11150  PRINT : PRINT : PRINT BELL$;"** SORRY, BUT YOU ONLY HAVE ENOUGH D
 ATA FOR ";N - 2;" INDEPENDENT VARIABLES !!"
11160  PRINT : INPUT "DO YOU WANT TO ENTER NEW DATA?  ";A$: IF  LEFT$ (A
 $,1) = "Y" THEN 10060
11170  GOTO 11090
11180  HOME
11190  FOR J = 1 TO IV
11200    PRINT "  INDEX OF INDEP VARIABLE ";J;: INPUT "?   ";A$:RV(J) =
  INT ( VAL (A$)): IF RV(J) < 1 OR RV(J) > 20 THEN  CALL  - 998: GOTO 11
 200
11210  NEXT J
11220  PRINT
11230  INPUT "INDEX OF DEPENDENT VARIABLE?   ";A$:RV(M) =  INT ( VAL (A$
 )): IF RV(M) < 1 OR RV(M) > NV + TV THEN  CALL  - 998: GOTO 11230
11240  PRINT
11250  INPUT "** DO YOU WANT TO MAKE CHANGES?  ";A$: IF  LEFT$ (A$,1) =
 "Y" THEN  HOME : GOTO 11090
11260 RV(0) = 0:VAR$(0) = "CONST"
11270
11280  TEXT : HOME : VTAB 5
11290  PRINT "    OUTPUT OPTIONS  (Y/N)"
11300  PRINT "    ------ -------"
11310  PRINT : PRINT
11320  PRINT "1.  OUTPUT TO PRINTER?"
11330  PRINT
```

```
11340  PRINT "2.  TABLE OF MEANS/ STD DEVIATIONS?"
11350  PRINT
11360  PRINT "3.  SIMPLE CORRELATION MATRIX?"
11370  PRINT
11380  PRINT "4.  VARIANCE-COVARIANCE MATRIX?"
11390  PRINT
11400  PRINT "5.  TABLE OF RESIDUALS?"
11410  VTAB 9
11420  FOR J = 1 TO 5
11430    HTAB 38: INPUT "";A$:P$(J) =  LEFT$ (A$,1)
11440    PRINT
11450  NEXT J
11460  GOSUB 13220: IF A$ = "Y" THEN 11280
11470
11480
11490  REM -COMPUTE MEANS, STD DEVIATIONS
11500  FOR I = 1 TO N:D(I,0) = 1: NEXT I: REM  - PUT ONES IN 1ST COLUMN
11510  IF REG = 1 THEN 11690: REM - SKIP RECALCULATION IF ALREADY DONE
11520  TEXT : HOME   VTAB 5: PRINT "CALCULATING MEANS AND STD DEVIATIONS
 ..."
11530  FOR I = 0 TO NV + TV
11540    FOR J = 0 TO NV + TV
11550     X = 0
11560      FOR L = 1 TO N
11570       X = X + D(L,I) * D(L,J)
11580      NEXT L
11590     X(I,J) = X:C(I,J) = X
11600    NEXT J
11610   T(I) = X(0,I) / X(0,0)
11620   SD(I) = 0
11630    IF I = 0 THEN 11650
11640   SD(I) =  SQR (X(I,I) / (N - 1) - X(0,I) * X(0,I) / (N * (N - 1))
 )
11650  NEXT I
11660  FOR I = 1 TO NV + TV
11670   M(I) = T(I):S(I) = SD(I)
11680  NEXT I
11690  IF P$(2) = "Y" THEN  GOSUB 13530
11700
11710  REM - COMPUTE CORRELATION COEFFICIENTS
11720  IF REG = 1 THEN 11790: REM - SKIP RECALCULATION IF ALREADY DONE
11730  TEXT : HOME : VTAB 5: PRINT "CALCULATING CORRELATION COEFFICIENTS
 ..."
11740  FOR I = 1 TO NV + TV
11750    FOR J = 1 TO NV + TV
11760     R(I,J) = (N * X(I,J) - X(0,I) * X(0,J)) / (N * (N - 1) * SD(I)
 * SD(J))
11770    NEXT J
11780  NEXT I
11790  IF P$(3) = "Y" THEN  GOSUB 13720
11800  FOR I = 0 TO M - 1
11810    FOR J = I + 1 TO M
```

```
11820      IF 1 - R(RV(I),RV(J)) < 1E - 9 THEN 12830: REM - WEED OUT LIN
 EAR DEPENDENCIES
11830    NEXT J
11840  NEXT I
11850
11860  REM  - COMPUTE CURVE FIT
11870  TEXT : HOME : VTAB 5: PRINT "CALCULATING REGRESSION COEFFICIENTS.
 .."
11880  FOR I = 0 TO IV
11890   B(I) = C(RV(I),RV(M))
11900    FOR J = 0 TO IV
11910     X(I,J) = C(RV(I),RV(J))
11920    NEXT J
11930  NEXT I
11940  PRINT
11950  FOR I = 0 TO IV
11960    FOR J = 0 TO IV
11970  IF I >  < J THEN 12000
11980   A(I,J) = 1
11990    GOTO 12010
12000   A(I,J) = 0
12010    NEXT J
12020  NEXT I
12030  FOR I = 0 TO IV
12040     IF X(I,I) < 1E - 6 THEN 12790
12050   B(I) = B(I) / X(I,I)
12060    FOR J = 0 TO IV
12070     A(I,J) = A(I,J) / X(I,I)
12080      IF J = I THEN 12100
12090     X(I,J) = X(I,J) / X(I,I)
12100    NEXT J
12110   X(I,I) = 1
12120    FOR L = 0 TO IV
12130      IF L = I THEN 12210
12140     B(L) = B(L) - X(L,I) * B(I)
12150      FOR J = 0 TO IV
12160       A(L,J) = A(L,J) - X(L,I) * A(I,J)
12170        IF J = I THEN 12190
12180       X(L,J) = X(L,J) - X(L,I) * X(I,J)
12190      NEXT J
12200     X(L,I) = 0
12210    NEXT L
12220  NEXT I
12230 ESS = C(RV(M),RV(M))
12240  FOR I = 0 TO IV
12250   ESS = ESS - B(I) * C(RV(I),RV(M))
12260  NEXT I
12270 VE = ESS / (N - M)
12280 R2 = 1 - VE / (S(M) * S(M))
12290  IF P$(1) = "Y" THEN  GOSUB 13310
12300  PRINT BELL$ + BELL$
12310  TEXT : HOME : VTAB 2
```

```
12320  PRINT "VAR"; TAB( 8);"ESTIMATED"; TAB( 20);"STD ERROR"
12330  PRINT "NAME"; TAB( 8);"COEFF 'B'"; TAB( 20 );"OF COEFF"; TAB( 32);
 "T-RATIO"
12340  PRINT "----"; TAB( 8);"----- ---"; TAB( 20 );"-- -----"; TAB( 32);
 "-------"
12350  PRINT
12360  FOR I = 0 TO IV
12370  PRINT VAR$( RV( I )); TAB( 8); FN R(B( I)); TAB( 20 ); FN R( SQR (A( I,
 I) * VE)); TAB( 32); FN R(B( I ) /  SQR (A( I,I ) * VE))
12380  NEXT I
12390  IF IV > 3 THEN  GOSUB 13170: HOME
12400  POKE 33,38: POKE 32,2
12410  PRINT : PRINT : PRINT : PRINT
12420  PRINT "ADJUSTED R↑2              ="; TAB( 28 ); FN S(R2)
12430  IF R2 < 0 THEN 12450
12440  PRINT "CORRELATION COEFFICIENT ="; TAB( 28 ); FN S( SQR (R2 ))
12450  PRINT
12460  PRINT "VARIANCE OF ESTIMATE    ="; TAB( 28 ); FN S(VE)
12470  PRINT "STD ERROR OF ESTIMATE   ="; TAB( 28 ); FN S( SQR (VE ))
12480  PRINT
12490  PRINT "DEGREES OF FREEDOM      ="; TAB( 28 );N - M
12500  POKE 32,0: POKE 33,40
12510  PRINT
12520
12530  REM - CALCULATE DURBIN WATSON STATISTIC
12540  FOR I = 1 TO N
12550   Z = D( I,RV(M)) - B( 0)
12560    FOR J = 1 TO IV
12570     Z = Z - B(J) * D( I,RV(J))
12580    NEXT J
12590   Q( I ) = Z
12600  NEXT I
12610 W = 0
12620  FOR I = 2 TO N
12630   W = W + (Q(I) - Q( I - 1)) * (Q(I) - Q( I - 1))
12640  NEXT I
12650  PRINT "DURBIN-WATSON STATISTIC = "; FN S(W / ESS)
12660  IF P$(1 ) = "Y" THEN  PRINT D$;"PR#0"
12670  GOSUB 13170
12680
12690  IF P$( 4 ) = "Y" THEN  GOSUB 14070: REM - VAR-COVAR MATRIX
12700
12710  IF P$( 5 ) = "Y" THEN  GOSUB 14410: REM  - RESIDUALS
12720
12730  TEXT : HOME : VTAB 7
12740  PRINT "DO YOU WANT TO PREDICT VALUES FOR"
12750  PRINT
12760  PRINT  TAB( 5);VAR$(RV( IV + 1));: INPUT " ?  ";A$: IF  LEFT$ (A$,
 1) = "Y" THEN  GOSUB 14590
12770  GOTO 12880
12780
12790  HOME : VTAB 7
```

```
12800  PRINT "CANNOT CONTINUE THE REGRESSION..."
12810  PRINT : PRINT
12820  PRINT BELL$;"** CORRELATION MATRIX BECOMING SINGULAR": GOTO 12870

12830  HOME : VTAB 7
12840  PRINT "CANNOT CONTINUE THE REGRESSION..."
12850  PRINT : PRINT
12860  PRINT BELL$;"** RECHECK YOUR DATA FOR DEPENDENCIES..."
12870  GOSUB 13170
12880  TEXT : HOME : VTAB 5
12890  PRINT "    YOUR FINAL OPTIONS ARE:"
12900  PRINT "    ---- ----- ------- ---"
12910  PRINT : PRINT : PRINT
12920  PRINT "1.  ANOTHER REGRESSION W/ SAME DATA BASE"
12930  PRINT
12940  PRINT "2.  ENTER A NEW DATA BASE"
12950  PRINT
12960  PRINT "3.  EDIT CURRENT DATA BASE"
12970  PRINT : PRINT
12980  PRINT "4.  QUIT THE PROGRAM"
12990  VTAB 22: INPUT "    WHICH?  ";A$:FC =  VAL (A$): IF FC < 1 OR FC
 > 4 THEN  CALL  - 998: GOTO 12990
13000  IF FC = 2 THEN RDFLAG = 1
13010  IF FC = 3 THEN RDFLAG = 2
13020  IF FC = 2 OR FC = 3 THEN 10060
13030  IF FC = 4 THEN 13090
13040  REM  - ENSURES LAST TRANSFORMS ARE NOW TREATED AS 'ORIGINAL' VARI
 ABLES
13050 NV = NV + TV:TV = 0
13060 REG = 1: REM - FLAG TO SKIP RECALCULATION OF MEANS, STD DEV, AND C
 ORR COEFF
13070  GOTO 10150
13080
13090  PRINT D$;"BLOAD CHAIN, A520"
13100  CALL 520"CAAMM MASTER"
13110  END
13120
13130  REM     *******************************************************

13140
13150
13160  REM  ** UTILITY SUBROUTINES **
13170  VTAB 23: GOSUB 13200: PRINT  TAB( 6);"HIT 'RETURN' TO CONTINUE...
 ";: GET A$: PRINT
13180  RETURN
13190
13200  PRINT "- - - - - - - - - - - - - - - - - - - - -"
13210  RETURN
13220  POKE 34,23: VTAB 23: GOSUB 13200: HTAB 7: INPUT "NEED TO CHANGE I
 T?  ";A$:A$ =  LEFT$ (A$,1): PRINT
13230  RETURN
13240
```

```
13250
13260  VTAB 23: GOSUB 13200: PRINT "HIT 'RETURN' TO GO ON, OR 'Q' TO QUI
 T ";
13270  GET A$: IF  ASC (A$ ) = 81 THEN 13440
13280  PRINT
13290  RETURN
13300
13310  REM  ** TURN PRINTER ON **
13320  PRINT D$;"PR#4": PRINT  : POKE 1148,32: POKE 1788,40
13330  PRINT : GOSUB 13200: PRINT : PRINT
13340  RETURN
13350
13360  REM  ** VARIABLE LISTING **
13370  PRINT
13380  PRINT "INDEX"; TAB( 7);"LABEL"; TAB( 14 );"NAME"
13390  PRINT
13400  FOR J = 1 TO NV + TV
13410    IF J > 10 THEN  POKE 33,20: POKE 32,20: VTAB J - 5
13420    PRINT  TAB( 3);J; TAB( 7);
13430    IF J > NV THEN  PRINT "T(";J - NV;")";: GOTO 13450
13440    PRINT "V(";J;" )";
13450    PRINT  TAB( 14 );VAR$( J )
13460  NEXT J
13470  POKE 32,0: POKE 33,40
13480  VTAB 16
13490  PRINT : GOSUB 13200: PRINT : PRINT
13500  POKE 34,18
13510  RETURN
13520
13530  REM  ** MEANS / STD DEVIATIONS PRINT-OUT **
13540  IF P$( 1 ) = "Y" THEN  GOSUB 13310
13550  PRINT BELL$
13560  TEXT : HOME : VTAB 2
13570  PRINT "VAR"; TAB( 18);"UNBIASED"; TAB(  30 );"UNBIASED"
13580  PRINT "NAME"; TAB(  8);"MEAN"; TAB(  18 );"STD DEV"; TAB(  30 );"VARIA
 NCE"
13590  PRINT "----"; TAB(  8);"----"; TAB(  18 );"--- ---"; TAB(  30 );"-----
 ---"
13600  POKE 34,4
13610  FOR I = 1 TO M
13620  PRINT
13630   M( I ) = T(RV( I )):S( I ) = SD( RV( I ))
13640    PRINT VAR$(RV( I )); TAB( 8 ); FN R( M( I )); TAB(  18); FN R(S( I )); T
 AB(  30 ); FN R(S(I) * S(I))
13650    IF I = M THEN 13670
13660    IF I / 8 =  INT (I / 8) AND P$(1 ) >  < "Y" THEN  GOSUB 13170: H
 OME
13670  NEXT I
13680  IF P$( 1 ) = "Y" THEN  PRINT D$;"PR#0"
13690  GOSUB 13170
13700  RETURN
13710
```

```
13720  REM  ** CORRELATION COEFFICIENT MATRIX **
13730 AV = 0: IF (M) / 4 >  INT ((M) / 4) THEN AV = 1
13740 U = 1
13750  IF P$(1) = "Y" THEN  GOSUB 13310
13760  PRINT BELL$
13770  TEXT : HOME : VTAB 3
13780  PRINT "CORRELATION COEFFICIENTS"
13790  PRINT "----------- ------------"
13800  PRINT : PRINT
13810    HTAB 8
13820    FOR J = U * 4 - 3 TO U * 4
13830      PRINT VAR$(RV(J)); TAB( 8 * (J - (U - 1) * 4) + 8);
13840      IF J = M THEN 13860
13850    NEXT J
13860  PRINT : PRINT : PRINT
13870  POKE 34,9
13880  FOR Q = 1 TO  INT ((M) / 4) + AV
13890    FOR I = Q * 4 - 3 TO Q * 4
13900      PRINT VAR$(RV(I)); TAB( 8);
13910      FOR L = U * 4 - 3 TO U * 4
13920       U(I,L) = R(RV(I),RV(L))
13930        PRINT  FN R(U(I,L)); TAB( 8 * (L - (U - 1) * 4) + 8);
13940        IF L = M THEN 13960
13950      NEXT L
13960      PRINT : PRINT
13970  IF I = M THEN 14010
13980    NEXT I
13990  IF P$(1) >  < "Y" THEN  GOSUB 13170: HOME
14000  NEXT Q
14010  IF P$(1) >  < "Y" THEN  GOSUB 13170
14020 U = U + 1: IF U >  INT ((M) / 4) + AV THEN 14040
14030  GOTO 13770
14040  IF P$(1) = "Y" THEN  PRINT D$;"PR#0": GOSUB 13170
14050  RETURN
14060
14070  REM  ** VARIANCE-COVARIANCE PRINT-OUT **
14080 AV = 0: IF M / 3 >  INT (M / 3) THEN AV = 1
14090 U = 1
14100  IF P$(1) = "Y" THEN  GOSUB 13310
14110  TEXT : HOME : VTAB 3
14120  PRINT "VARIANCE-COVARIANCE MATRIX OF COEFF"
14130  PRINT "-------- ---------- ------ -- -----"
14140  PRINT : PRINT
14150    HTAB 8
14160    FOR J = U * 3 - 3 TO U * 3 - 1
14170      PRINT VAR$(RV(J)); TAB( 11 * (J + 1 - (U - 1) * 3) + 8);
14180    IF J = IV THEN 14200
14190    NEXT J
14200  PRINT : PRINT : PRINT
14210  POKE 34,9
14220  FOR Q = 1 TO  INT (M / 3) + AV
14230    FOR I = Q * 3 - 3 TO Q * 3 - 1
```

```
14240       PRINT VAR$(RV(I)); TAB( 8);
14250     FOR L = U * 3 - 3 TO U * 3 - 1
14260       A(I,L) = A(I,L) * VE
14270        PRINT  FN R(A(I,L)); TAB( 11 * (L + 1 - (U - 1) * 3) + 8);
14280        IF L = IV THEN 14300
14290       NEXT L
14300   PRINT : PRINT
14310   IF I = IV THEN 14350
14320     NEXT I
14330   IF P$(1) >  < "Y" THEN  GOSUB 13170: HOME
14340   NEXT Q
14350   IF P$(1) >  < "Y" THEN  GOSUB 13170
14360 U = U + 1: IF U >  INT (M / 3) + AV THEN 14380
14370   GOTO 14110
14380   IF P$(1) = "Y" THEN  PRINT D$;"PR#0": GOSUB 13170
14390   RETURN
14400
14410   REM  ** TABLE OF PREDICTED VALUES AND RESIDUALS **
14420   IF P$(1) = "Y" THEN  GOSUB 13310
14430   TEXT : HOME : VTAB 2
14440   PRINT  TAB( 5)"ACTUAL"; TAB( 14);"PREDICTED"
14450   PRINT  TAB( 6);"'Y'"; TAB( 17);"'Y'"; TAB( 27);"RESIDUAL"
14460   PRINT  TAB( 5);"------"; TAB( 14);"---------"; TAB( 27);"--------
 "

14470   PRINT
14480   POKE 34,5
14490   FOR I = 1 TO N
14500   PRINT I; TAB( 5);D(I,RV(M)); TAB( 14); FN R(D(I,RV(M)) - Q(I)); T
 AB( 27); FN R(Q(I))
14510   IF I = N THEN 14540
14520   IF I / 5 =  INT (I / 5) THEN  PRINT
14530   IF I / 15 =  INT (I / 15) AND P$(1) >  < "Y" THEN  GOSUB 13170: H
 OME
14540   NEXT I
14550   IF P$(1) = "Y" THEN  PRINT D$;"PR#0"
14560   GOSUB 13170
14570   RETURN
14580
14590   REM   ** DEPENDENT VARIABLE PREDICTIONS **
14600   HOME : VTAB 3
14610   PRINT "ENTER VALUES FOR INDEPENDENT VARIABLES:"
14620   PRINT : PRINT
14630   FOR J = 1 TO IV
14640     IF J > 10 THEN  POKE 33,20: POKE 32,20: VTAB J - 4
14650     PRINT "V( ";RV(J);" )"; TAB( 7);VAR$(RV(J)); TAB( 15);: INPUT "";
 XP(J)
14660   NEXT J
14670   POKE 32,0: POKE 33,40
14680 YP = B(0)
14690   FOR I = 1 TO IV
14700    YP = YP + XP(I) * B(I)
14710   NEXT I
```

```
14720  PRINT : PRINT
14730  PRINT BELL$;"PREDICTED VALUE OF ";VAR$(RV(IV + 1));" =  "; FN R(Y
 P)
14740  PRINT : PRINT
14750  GOSUB 13200
14760  HTAB 10: INPUT "ANOTHER PREDICTION?  ";A$: IF  LEFT$ (A$,1) = "Y"
  THEN 14590
14770  RETURN
14780
14790  REM   ** SAVE TEMPORARY STORAGE FILE **
14800  IF SFILE$ >  < "" THEN NV = NV + TV: GOTO 14820
14810 SFILE$ = "TEMPFILE"
14820  PRINT D$;"OPEN ";SFILE$;D$;"DELETE ";SFILE$;D$;"OPEN ";SFILE$
14830  PRINT D$;"WRITE ";SFILE$
14840  PRINT N: PRINT NV
14850  FOR J = 1 TO NV
14860    PRINT VAR$(J)
14870    FOR I = 1 TO N: PRINT D(I,J): NEXT I
14880  NEXT J
14890  IF SFILE$ >  < "TEMPFILE" THEN 14930
14900  PRINT TV
14910  FOR J = 1 TO TV: PRINT VAR$( NV + J): NEXT J
14920  GOTO 14940
14930 NV = NV - TV
14940  PRINT D$;"CLOSE ";SFILE$
14950 SFILE$ = ""
14960  RETURN
14970
14980  REM    ** READ DISK FILE **
14990  PRINT D$;"OPEN TEMPFILE"
15000  PRINT D$;"READ TEMPFILE"
15010  INPUT N,NV
15020  FOR J = 1 TO NV
15030    INPUT VAR$(J)
15040    FOR I = 1 TO N: INPUT D(I,J): NEXT I
15050  NEXT J
15060  INPUT TV
15070  FOR J = 1 TO TV: INPUT VAR$( NV + J): NEXT J
15080  PRINT D$;"CLOSE TEMPFILE"
15090  RETURN
15100
15110  REM   ** USER MODEL INSTRUCTIONS **
15120  HOME
15130  PRINT "DEFINING A MODEL ALLOWS YOU TO FIT DATA TO VIRTUALLY ANY A
 LGEBRAIC CURVE THAT   CAN BE EXPRESSED AS A TRANSFORMATION OF THE BASIC
  LINEAR MODEL."
15140  PRINT
15150  PRINT "YOUR DEFINED MODEL WILL USE 'TRANSFORM' VARIABLES THAT ARE
  FUNCTIONS OF ANY OR  ALL OF THE ORIGINAL DATA VARIABLES."
15160  PRINT
15170  PRINT "** TOTAL VARIABLES CANNOT EXCEED 20."
15180  PRINT
```

```
15190  PRINT "THIS MODEL, FOR EXAMPLE..."
15200  PRINT
15210  PRINT "V(7) = A +B*LOG( V(1)↑2) +C*( V(4)+V(6))/2"
15220  PRINT
15230  PRINT "REQUIRES 2 TRANSFORM VARIABLES:"
15240  PRINT
15250  PRINT "      T(1) = LOG( V(1)↑2)"
15260  PRINT "      T(2) = ( V(4)+V(6))/2"
15270  PRINT : PRINT
15280  INPUT "YOU MUST USE LEGAL FUNCTIONS AND SYNTAX.          NEED H
 ELP? ";A$
15290  IF  LEFT$ (A$,1) = "Y" THEN  GOSUB 15320
15300  RETURN
15310
15320  REM  ** LEGAL FUNCTIONS AND SYNTAX **
15330  HOME : VTAB 2
15340  PRINT "FUNCTION"; TAB( 14);"COMMENTS"
15350  PRINT "--------"; TAB( 14);"--------"
15360  PRINT
15370  PRINT "SIN(X)"
15380  PRINT "COS(X)"
15390  PRINT "TAN(X)"; TAB( 10 );"X CAN'T BE A MULTIPLE OF PI/2"
15400  PRINT "ATN(X)"
15410  PRINT "ABS(X)"
15420  PRINT "SQR(X)"; TAB( 10 );"SQUARE ROOT; X MUST BE >= 0"
15430  PRINT "EXP(X)"; TAB( 10 );"'E' RAISED TO THE POWER OF X"
15440  PRINT "LOG(X)"; TAB( 10 );"NATURAL LOG; X MUST BE > 0"
15450  PRINT : PRINT : PRINT
15460  PRINT "** PARENTHESES MUST BE CLOSED"
15470  PRINT
15480  PRINT "** EXPONENTIATE USING '↑', E.G. 'X↑2'"
15490  PRINT
15500  PRINT "** THE VALUE 'PI' CAN BE ENTERED AS 'PI'"
15510  PRINT "** TRIG FUNCTIONS EXPECT X IN RADIANS"
15520  GOSUB 13170
15530  RETURN
15540
15550  REM  ** WRITE TRANSFORM FILES **
15560    PRINT D$;"OPEN TF";D$;"DELETE TF";D$;"OPEN TF"
15570    PRINT D$;"WRITE TF"
15580  FOR J = 1 TO TV
15590    PRINT 15650 + J * 10;" T(";J;") = ";VAR$( J + NV)
15600  NEXT J
15610  PRINT 15650 + J * 10;" RETURN"
15620  PRINT "GOTO 10560"
15630  PRINT D$;"EXEC TF"
15640  END
15650  REM   ** TRANSFORMS **
15660  REM - USER-DEFINED TRANSFORMATION LINES CREATED TO START HERE
```

```
10000 BELL$ = "": REM - BELL
10010  PRINT BELL$
10020  TEXT : HOME : VTAB 3 .
10030  PRINT "* * * * * * * * * * * * * * * * * * * *"
10040  PRINT
10050  PRINT  TAB( 15);"L I N E A R"
10060  PRINT
10070  PRINT  TAB( 10);"P R O G R A M M I N G"
10080  PRINT : PRINT
10090  PRINT  TAB( 19);"BY"
10100  PRINT
10110  PRINT  TAB( 13);"ROBERT D. CONTE"
10120  PRINT
10130  PRINT "* * * * * * * * * * * * * * * * * * * *"
10140  VTAB 24: INPUT "  DO YOU WANT INTRODUCTORY REMARKS?  ";A$: IF  LE
 FT$ (A$,1) = "Y" THEN  GOSUB 20660
10150
10160  CLEAR
10170  REM   - DIMENSIONED FOR 20 VARIABLES, 20 CONSTRAINTS
10180  DIM A(20,60),B(20),C(60),T$(20,22),CST$(20),VAR$(20),CT$(20)
10190  DIM D(20),F(60),H(2,20),V(60),Z(60)
10200  DEF  FN R(X) =  INT (X * 1000 + .5) / 1000: REM  - 3 PLACE ROUND
 OFF FUNCTION
10210 D$ =  CHR$ (13) +  CHR$ (4)
10220
10230  HOME : POKE 33,37: POKE 32,3
10240  VTAB 2
10250  PRINT "  TO ENTER AN LP MODEL, YOU CAN:"
10260  PRINT "  -- ----- -- -- -----  --- ---"
10270  VTAB 9
10280  PRINT "1.  READ EXISTING MODEL FROM DISK"
10290  PRINT : PRINT
10300  PRINT "2.  CREATE MODEL INTERACTIVELY"
10310  PRINT "       (USING NAMED VARIABLES)"
10320  PRINT : PRINT
10330  PRINT "3.  CREATE MODEL INTERACTIVELY"
10340  PRINT "       (USING NUMBERS ONLY)"
10350  PRINT : PRINT : PRINT
10360  TEXT
10370  HTAB 8: INPUT "WHICH METHOD?  ";A$:DE =  VAL (A$): IF A$ = "" THE
 N DE = 1
10380  IF DE = 1 THEN  GOSUB 20440: GOTO 10410
10390  ON DE - 1 GOSUB 15780,16270
10400
10410  TEXT : HOME : POKE 33,35: POKE 32,5
10420  VTAB 2
10430 X =  FRE (0): REM  - CLEAN UP UNUSED STRING SPACE
```

```
10440  PRINT "  LP MODEL MANAGEMENT"
10450  PRINT "  -- ----- ----------"
10460  PRINT : PRINT
10470  PRINT "1.  DISPLAY CURRENT MODEL"
10480  PRINT
10490  IF OPFLAG = 1 THEN  PRINT "2.  SENSITIVITY MODEL EDITING":OPFLAG
 = 0: GOTO 10510
10500  PRINT "2.  EDIT CURRENT MODEL"
10510  PRINT
10520  PRINT "3.  SAVE CURRENT MODEL TO DISK"
10530  PRINT
10540  PRINT "4.  ENTER A NEW MODEL"
10550  PRINT
10560  PRINT "5.  QUIT PROGRAM"
10570  PRINT : PRINT
10580  PRINT "6.  SOLVE THE PROBLEM"
10590  PRINT : PRINT : PRINT
10600  TEXT
10610  HTAB 10: INPUT "WHICH OPTION?  ";A$:DO =  VAL (A$): IF A$ = "" TH
EN 10660
10620  IF DO < 1 OR DO > 6 THEN  CALL  - 998: GOTO 10610
10630  ON DO GOSUB 17140,17250,20250: IF DO < 4 THEN 10410
10640  ON DO - 3 GOTO 10160,14700,10660
10650
10660  GOSUB 19710
10670  TEXT : HOME : VTAB 5
10680  PRINT "  OUTPUT OPTIONS (Y/N)"
10690  PRINT "  ------ -------"
10700  PRINT : PRINT
10710  PRINT "1.  OUTPUT TO PRINTER?"
10720  PRINT
10730  PRINT "2.  INITIAL TABLEAU?"
10740  PRINT
10750  PRINT "3.  INTERMEDIATE BASIC SOLUTIONS?"
10760  PRINT
10770  PRINT "4.  FINAL TABLEAU?"
10780  VTAB 9
10790  FOR J = 1 TO 4
10800    HTAB 36: INPUT "";A$:P$(J) =  LEFT$ (A$,1)
10810  PRINT
10820  NEXT J
10830  GOSUB 14830: IF A$ = "Y" THEN 10670
10840
10850  REM    ** SET UP SLACK, SURPLUS, AND ARTIFICIAL VARIABLES **
10860
10870  REM - SUPLUSES FOR > CONSTRAINTS
10880 J = NV + 1
10890    FOR I = 1 TO NC
10900      IF I >  < J - NV + L + E THEN 10920
10910     A(I,J) =  - 1
10920    NEXT I
10930 J = J + 1: IF J > NV + G THEN 10960
```

```
10940  GOTO 10890
10950
10960  REM - SLACKS FOR < CONSTRAINTS
10970  FOR J = NV + G + 1 TO NV + G + L
10980    FOR I = 1 TO NC
10990      IF I >  < J - NV - G THEN 11010
11000     A(I,J) = 1
11010    NEXT I
11020  NEXT J
11030
11040  REM - ARTIFICIALS FOR = CONSTRAINTS
11050  FOR J = NV + G + L + 1 TO NV + G + L + E
11060  FOR I = 1 TO NC
11070      IF I >  < J - NV - G THEN 11090
11080     A(I,J) = 1
11090    NEXT I
11100  NEXT J
11110
11120  REM - ARTIFICIALS FOR > AND = CONSTRAINTS
11130  FOR J = NV + G + L + E + 1 TO NV + NC + G
11140    FOR I = 1 TO NC
11150      IF I >  < J - NV - G THEN 11170
11160     A(I,J) = 1
11170    NEXT I
11180  NEXT J
11190
11200 TV = NV + NC + G: REM   - TOTAL NUMBER OF ALL VARIABLES
11210
11220  IF P$(1) = "Y" THEN  GOSUB 14940
11230  TEXT : HOME : IF DE = 3 THEN  VTAB 5
11240  PRINT  TAB( 10 );"VARIABLE TABLE:"
11250  PRINT  TAB( 10 );"======== ====="
11260  PRINT : IF DE = 3 THEN  PRINT
11270  PRINT "DECISION VARIABLES ARE"; TAB( 28 );"1-";NV
11280  IF DE = 2 THEN N = NV:AFLAG = 1: GOSUB 17730:AFLAG = 0
11290  PRINT
11300  IF G = 0 THEN 11330
11310  PRINT "SURPLUS VARIABLES ARE"; TAB( 28 );NV;"-";NV + G
11320  PRINT
11330  IF L = 0 THEN 11360
11340  PRINT "SLACK VARIABLES ARE"; TAB( 28);NV + G + 1;"-";NV + G + L
11350  PRINT
11360  IF G + E = 0 THEN 11380
11370  PRINT "ARTIFICIAL VARIABLES ARE"; TAB( 28);NV + G + L + 1;"-";NV
 + NC + G
11380  GOSUB 14770
11390
11400  REM  ******** START PHASE I ********
11410 IT = 0: REM  - NUMBER OF ITERATIONS
11420  IF G + E = 0 THEN 13060
11430  FOR I = L + 1 TO NC
11440    IF B(I) > 0 THEN 11520: REM - START AT PHASE I
11450  NEXT I
```

```
11460
11470  FOR I = 1 TO NC
11480   D(I) = NV + G + I
11490  NEXT I
11500  GOTO 13060: REM -START AT PHASE II
11510  REM  ******** PHASE I ********
11520 PH = 1:PFLAG = 1
11530  REM - F IS PHASE I OBJ FUNC VECTOR
11540  FOR J = NV + G + L + 1 TO TV
11550   F(J) =  - 1
11560  NEXT J
11570
11580  REM -D IS VECTOR OF BASIC VARIABLE SUBSCRIPT
11590  FOR I = 1 TO NC
11600   D(I) = NV + G + I
11610  NEXT I
11620
11630  REM  - Z(J) IS JTH Z-ROW ENTRY  Z(J)-C(J)
11640  FOR J = 1 TO TV
11650   Z(J) = 0
11660    FOR I = 1 TO NC
11670     Z(J) = Z(J) + F(D(I)) * A(I,J)
11680    NEXT I
11690   Z(J) = Z(J) - F(J)
11700  NEXT J
11710
11720  REM  - Z IS PHASE I OBJ FUNC VALUE
11730 Z = 0
11740  FOR I = 1 TO NC
11750   Z = Z + F(D(I)) * B(I)
11760  NEXT I
11770
11780  IF P$(2) >  < "Y" THEN 11860
11790  TEXT : HOME : VTAB 2
11800  HTAB 12: PRINT "INITIAL TABLEAU"
11810  HTAB 12: PRINT "======= ======="
11820  PRINT : PRINT
11830  GOSUB 14980
11840  REM -   THE PHASE I OBJ FUNC IS OBTAINED BY LETTING  Z(J)=0  FOR
 LEGITIMATE VARIABLES AND  Z(J)=-1 FOR ARTIFICIAL VARIABLES
11850
11860  IF P$(3) >  < "Y" THEN 11940
11870  HOME : VTAB 2
11880  PRINT "INITIAL BASIC SOLUTION"
11890  PRINT "======= ===== ========"
11900  PRINT
11910  GOSUB 15480: GOSUB 14770
11920
11930  REM - GET RID OF ARTIFICIAL VARIABLES
11940 BM = 99999
11950  REM -FIND MOST NEGATIVE Z(J)
```

```
11960 X = 1
11970  FOR J = 1 TO TV
11980    IF Z(J) >  = BM THEN 12060
11990    FOR I = 1 TO TV
12000      IF J = V(I) THEN 12060
12010    NEXT I
12020
12030  REM - W/ ALL NEGATIVE ENTRIES, K IS COL OF INCOMING VARIABLE
12040   BM = Z(J)
12050   K = J
12060  NEXT J
12070
12080  IF BM >  = 0 THEN 12740
12090 BM = 9.9E + 34
12100
12110  REM - FIND MIN B(I)/A(I,K) FOR A(I,K)>0
12120  FOR I = 1 TO NC
12130    IF A(I,K) <  = 0 THEN 12180
12140    IF B(I) / A(I,K) > BM THEN 12180
12150   BM = B(I) / A(I,K)
12160  REM - R IS ROW OF LEAVING VARIABLE
12170   R = I
12180  NEXT I
12190  IF BM < 9.9E + 34 THEN 12270
12200 V(X) = K
12210 X = X + 1
12220  IF X <  = 120 THEN 11970
12230  PRINT : PRINT : PRINT "** UNBOUNDED SOLUTION DURING PHASE I"
12240  PRINT : GOSUB 14770: GOTO 10410
12250
12260  REM    ** PIVOT **
12270 P = A(R,K)
12280  REM - TRANSFORM ELEMENTS OFF PIVOT ROW AND COLUMN
12290  FOR I = 1 TO NC
12300    IF I = R THEN 12370
12310    FOR J = 1 TO TV
12320      IF J = K THEN 12360
12330     A(I,J) = A(I,J) - A(I,K) * A(R,J) / P
12340      IF  ABS (A(I,J)) > 1E - 5 THEN 12360
12350     A(I,J) = 0
12360    NEXT J
12370  NEXT I
12380
12390  REM - OBTAIN NEW OBJ FUNC VALUE
12400 Z = Z - Z(K) * B(R) / P
12410  IF  ABS (Z) > 1E - 5 THEN 12450
12420 Z = 0
12430
12440  REM - TRANSFORM OBJ FUNC ROW
12450  FOR J = 1 TO TV
12460    IF J = K THEN 12500
12470   Z(J) = Z(J) - Z(K) * A(R,J) / P
```

```
12480    IF  ABS (Z(J)) > 1E - 5 THEN 12500
12490   Z( J ) = 0
12500  NEXT J
12510
12520  REM -TRANSFORM B-VECTOR AND PIVOT COLUMN
12530  FOR I = 1 TO NC
12540    IF I = R THEN 12590
12550   B( I ) = B( I ) - B( R ) * A( I,K ) / P
12560   A( I,K ) = 0
12570    IF  ABS (B( I )) > 1E - 5 THEN 12590
12580   B( I ) = 0
12590  NEXT I
12600 B( R ) = B( R ) / P
12610 Z( K ) = 0
12620
12630  REM - TRANSFORM THE PIVOT ROW
12640  FOR J = 1 TO TV
12650   A( R,J ) = A( R,J ) / P
12660  NEXT J
12670
12680 IT = IT + 1: REM - INCREMENT ITERATION COUNTER
12690
12700  REM  - CHANGE BASIC VARIABLE INDEX VECTOR
12710 D( R ) = K
12720
12730  REM - CHECK FOR TERMINATION
12740  IF Z >  = 0 THEN 12960
12750  REM - CHECK OPTIMALITY CRITERION
12760  FOR J = 1 TO TV
12770    IF Z( J ) < 0 THEN 12870
12780  NEXT J
12790  FOR I = 1 TO NC
12800    IF D( I ) > NV + G + L THEN 12840
12810  NEXT I
12820
12830  GOTO 12960
12840  PRINT : PRINT : PRINT "** THERE IS NO FEASIBLE SOLUTION"
12850  PRINT : GOSUB 14770: GOTO 10410
12860
12870  IF P$( 3 ) >  < "Y" THEN 11940
12880  HOME : VTAB 2
12890  PRINT "BASIC SOLN AFTER ITERATION ";IT
12900  PRINT "===== ==== ===== ========="
12910  PRINT
12920  GOSUB 15480: GOSUB 14770
12930  REM - CONTINUE W/ NEXT ITERATION
12940  GOTO 11940
12950
12960  IF P$( 3 ) >  < "Y" THEN 13060
12970  HOME : VTAB 2
12980  PRINT "BASIC SOLN AFTER ITERATION ";IT
12990  PRINT "===== ==== ===== ========="
```

```
13000  PRINT
13010  GOSUB 15480: GOSUB 14770
13020
13030
13040  REM  ******* PHASE II *******
13050
13060 PH = 2
13070  REM -PRICE OUT VARIABLES AGAIN
13080  IF D(1) >  < 0 THEN 13120
13090  FOR I = 1 TO NC
13100   D(I) = NV + I
13110  NEXT I
13120  FOR J = 1 TO TV
13130   Z(J) = 0
13140    FOR I = 1 TO NC
13150   Z(J) = Z(J) + C(D(I)) * A(I,J)
13160  NEXT I
13170
13180  REM - Z(J) IS THE JTH Z-ROW ENTRY  Z(J)-C(J)
13190 Z(J) = Z(J) - C(J)
13200  NEXT J
13210 Z = 0
13220  REM  - Z IS 1ST PHASE II OBJ FUNC VALUE
13230  FOR I = 1 TO NC
13240   Z = Z + C(D(I)) * B(I)
13250  NEXT I
13260
13270  IF PFLAG = 1 THEN PFLAG = 0: GOTO 13340
13280  IF P$(2) >  < "Y" THEN 13340
13290  TEXT : HOME : VTAB 2
13300  HTAB 12: PRINT "INITIAL TABLEAU"
13310  HTAB 12: PRINT "======= ======="
13320  PRINT : PRINT
13330  GOSUB 14980
13340  IF P$(3) >  < "Y" THEN 13410
13350  TEXT : HOME : VTAB 2
13360  PRINT "INITIAL BASIC FEASIBLE SOLUTION"
13370  PRINT "======= ===== ======== ========"
13380  PRINT : GOSUB 15480
13390
13400  REM - FIND MOST NEGATIVE Z(J)
13410 BM = 99999
13420  REM -NEVER CONSIDER ENTERING AN ARTIFICIAL
13430  FOR J = 1 TO NV + G + L
13440    IF Z(J) >  = BM THEN 13470
13450   BM = Z(J)
13460   K = J
13470  NEXT J
13480
13490  REM - OPTIMAL SOLUTION FOUND
13500  IF BM >  = 0 THEN 14260
13510  IF P$(3) = "Y" THEN  GOSUB 14770
```

```
13520 BM = 9.9E + 34
13530
13540  REM -FIND MINIMUM B(I)/A(I,K) FOR A(I,K)>0 (UNLESS ARTIFICIAL VAR
IABLE IN BASIS W/ A(I,K)<0)
13550  FOR I = 1 TO NC
13560    IF A(I,K) < = 0 THEN 13610
13570    IF B(I) / A(I,K) > BM THEN 13660
13580   BM = B(I) / A(I,K)
13590   R = I
13600    GOTO 13660
13610    IF A(I,K) = 0 THEN 13660
13620    IF D(I) < = NV + G + L THEN 13660
13630    REM - ARTIFICIAL VARIABLE, THEN PIVOT ON THAT ROW
13640   R = I
13650    GOTO 13720
13660  NEXT I
13670  IF BM < 9.9E + 34 THEN 13720
13680  PRINT : PRINT : PRINT "** THE SOLUTION IS UNBOUNDED"
13690  PRINT : GOSUB 14770: GOTO 10410
13700
13710  REM - PIVOT
13720 P = A(R,K)
13730  REM -TRANSFORM ELEMENTS OFF PIVOT ROW AND COLUMN
13740  FOR I = 1 TO NC
13750    IF I = R THEN 13820
13760    FOR J = 1 TO TV
13770      IF J = K THEN 13810
13780     A(I,J) = A(I,J) - A(I,K) * A(R,J) / P
13790      IF  ABS (A(I,J)) > 1E - 5 THEN 13810
13800     A(I,J) = 0
13810    NEXT J
13820  NEXT I
13830
13840  REM - DETERMINE NEW OBJ FUNC VALUE
13850 Z = Z - Z(K) * B(R) / P
13860  IF  ABS (Z) > 1E - 5 THEN 13900
13870 Z = 0
13880
13890  REM - TRANSFORM REST OF Z-ROW
13900  FOR J = 1 TO TV
13910    IF J = K THEN 13950
13920   Z(J) = Z(J) - Z(K) * A(R,J) / P
13930    IF  ABS (Z(J)) > 1E - 5 THEN 13950
13940   Z(J) = 0
13950  NEXT J
13960
13970  REM -TRANSFORM B-VECTOR AND PIVOT COLUMN
13980  FOR I = 1 TO NC
13990    IF I = R THEN 14040
14000   B(I) = B(I) - B(R) * A(I,K) / P
14010   A(I,K) = 0
14020    IF  ABS (B(I)) > 1E - 5 THEN 14040
```

```
14030   B( I ) = 0
14040  NEXT I
14050 B( R ) = B( R ) / P
14060 Z( K ) = 0
14070
14080  REM - TRANSFORM PIVOT ROW
14090  FOR J = 1 TO TV
14100   A( R,J ) = A( R,J ) / P
14110  NEXT J
14120
14130 IT = IT + 1: REM - INCREMENT ITERATIONS
14140
14150  REM - CHANGE BASIC VARIABLE INDEX VECTOR
14160 D( R ) = K
14170
14180  IF P$( 3 ) >  < "Y" THEN 13410
14190  TEXT : HOME : VTAB 2
14200  PRINT "BASIC FEASIBLE SOLN AFTER ITERATION ";IT
14210  PRINT "===== ======== ==== ===== ========="
14220  PRINT : GOSUB 15480
14230  GOTO 13410: REM -CONTINUE PHASE II
14240
14250  REM - OPTIMAL SOLUTION
14260  PRINT
14270  IF P$( 3 ) >  < "Y" THEN 14310
14280  PRINT : PRINT : PRINT BELL$;"*** THIS SOLUTION IS ";M$;"IMAL ***"

14290  GOSUB 14770
14300  GOTO 14370
14310  TEXT : HOME : VTAB 3
14320  PRINT BELL$; TAB( 3);M$;"IMAL BASIC SOLN AFTER ";IT;" IT'S"
14330  PRINT  TAB( 3);"======= ===== ==== ====="
14340  PRINT : PRINT
14350  GOSUB 15480: GOSUB 14770
14360
14370  TEXT : HOME : VTAB 2
14380  PRINT "OPTIMAL DUAL SOLUTION"
14390  PRINT "======= ==== ========"
14400  PRINT
14410  PRINT "INDEX  CONSTRAINT  SHADOW PRICE"
14420  PRINT "-----  ----------  ------ -----"
14430  PRINT : POKE 34,7
14440 Q = NV + G + 1
14450  FOR J = Q TO TV
14460    PRINT J - (Q - 1); TAB( 8);CST$(J - (Q - 1)); TAB( 20 ); FN R(Z1
  * Z(J));: IF Z(J) = 0 THEN  PRINT "   (FREE GOOD)": GOTO 14480
14470  PRINT
14480    IF (J - Q + 1) / 15 =  INT ((J - Q + 1) / 15) THEN  GOSUB 14770
 : HOME
14490  NEXT J
14500
```

```
14510  FOR I = 1 TO NC
14520    IF B(I) = 0 THEN 14590
14530  NEXT I
14540 OPFLAG = 1
14550  GOSUB 14770
14560  GOTO 14620
14570
14580  REM - DEGENERATE SOLUTION
14590  PRINT : PRINT : PRINT "** OPTIMAL BASIC SOLUTION IS DEGENERATE"
14600  PRINT : GOSUB 14770: GOTO 10410
14610
14620  IF P$(4) >  < "Y" THEN 14680
14630  TEXT : HOME : VTAB 2
14640  HTAB 12: PRINT "OPTIMAL TABLEAU"
14650  HTAB 12: PRINT "------- -------"
14660  PRINT : GOSUB 14980
14670
14680  IF P$(1) = "Y" THEN  PRINT D$;"PR#0"
14690  GOTO 10410
14700  PRINT D$;"BLOAD CHAIN, A520"
14710  CALL 520"CAAMM MASTER"
14720  END
14730
14740  REM   ***********************************************************

14750
14760  REM  ** UTILITY ROUTINES **
14770  VTAB 23: GOSUB 14860: HTAB 5: PRINT "HIT 'RETURN' TO CONTINUE...
 ";: GET A$: PRINT
14780  RETURN
14790
14800  VTAB 23: GOSUB 14860: HTAB 7: INPUT "EDIT ANOTHER?  ";A$:A$ =  LE
 FT$ (A$,1): PRINT
14810  RETURN
14820
14830  VTAB 23: GOSUB 14860: HTAB 7: INPUT "NEED TO MAKE CHANGES?  ";A$:
 A$ =  LEFT$ (A$,1): PRINT
14840  RETURN
14850
14860  PRINT "- - - - - - - - - - - - - - - - - - - - -"
14870  RETURN
14880
14890  VTAB 23: GOSUB 14860: PRINT "HIT 'RETURN' TO GO ON, OR 'Q' TO QUI
 T ";
14900  GET A$: IF  ASC (A$) = 81 THEN 14700
14910  PRINT
14920  RETURN
14930  REM  ** TURN ON PRINTER **
14940  PRINT D$;"PR#4": PRINT : POKE 1148,32: POKE 1788,40
14950  PRINT : GOSUB 14860: PRINT : PRINT
14960  RETURN
14970
```

```
14980  REM  ** PRINT LP TABLEAU **
14990  POKE 34,4
15000 AV = 0: IF (TV + 1) / 5 >  INT ((TV + 1) / 5) THEN AV = 1
15010 J = 1
15020  HOME
15030    FOR I = J * 5 - 4 TO J * 5
15040      IF I = NV + 1 AND DFLAG = 1 THEN  PRINT "";: GOTO 15090
15050      IF I = NV + 2 AND DFLAG = 1 THEN  PRINT "RHS";: GOTO 15120
15060      IF DFLAG = 1 THEN  PRINT VAR$(I);: GOTO 15090
15070      IF I = TV + 1 THEN  PRINT "RHS";: GOTO 15120
15080      PRINT "X(";I;")";
15090      HTAB 8 * (I - (J - 1) * 5) + 1
15100      IF  POS (H) = 0 THEN  CALL  - 998
15110    NEXT I
15120  PRINT : PRINT
15130    REM -   OBJ FUNC ROW
15140    FOR I = J * 5 - 4 TO J * 5
15150      IF I = NV + 1 AND DFLAG = 1 THEN  PRINT "";: GOTO 15200
15160      IF I = NV + 2 AND DFLAG = 1 THEN  PRINT "0": GOTO 15220
15170      IF I = TV + 1 THEN  PRINT  FN R(Z);: GOTO 15220
15180      IF DFLAG = 1 THEN  PRINT Z1 * C(I);: GUTO 15200
15190      PRINT  FN R(Z(I));
15200      HTAB 8 * (I - (J - 1) * 5) + 1
15210    NEXT I
15220    IF  POS (H) = 0 THEN  CALL  - 998
15230    PRINT
15240    FOR I = J * 5 - 4 TO J * 5
15250      IF I = NV + 1 AND DFLAG = 1 THEN  PRINT "- - - - ";: GOTO 152
 90
15260      IF I = NV + 2 AND DFLAG = 1 THEN  PRINT "- - -": GOTO 15300
15270      PRINT "- - - - ";
15280      IF I = TV + 1 THEN  PRINT : GOTO 15300
15290    NEXT I
15300    FOR ROW = 1 TO NC
15310      FOR COL = J * 5 - 4 TO J * 5
15320        IF COL = NV + 1 AND DFLAG = 1 THEN  PRINT CT$(ROW);: GOTO 1
 5360
15330        IF COL = NV + 2 AND DFLAG = 1 THEN  PRINT B(ROW);: GOTO 153
 90
15340        IF COL = TV + 1 THEN  PRINT  FN R(B(ROW));: GOTO 15390
15350        PRINT  FN R(A(ROW,COL));
15360        HTAB 8 * (COL - (J - 1) * 5) + 1
15370        IF  POS (H) = 0 THEN  CALL  - 998
15380      NEXT COL
15390      PRINT
15400  NEXT ROW
15410  PRINT
15420 J = J + 1: IF J >  INT ((TV + 1) / 5) + AV THEN 15440
15430  GOSUB 14770: GOTO 15020
15440  GOSUB 14770
15450  TEXT
15460  RETURN
```

```
15470
15480  REM   ** PRINT INTERMEDIATE BASIC SOLUTIONS **
15490  REM - ORDER BASIC VARIABLES BY INDEX
15500  FOR I = 1 TO NC
15510   H( 1,I ) = D( I )
15520   H( 2,I ) = B( I )
15530  NEXT I
15540
15550  FOR I = 1 TO NC - 1
15560    FOR J = NC TO I + 1 STEP  - 1
15570      IF H(1,J) >  = H(1,J - 1) THEN 15630
15580      FOR K = 1 TO 2
15590       SAV = H(K,J):SAV$ = VAR$(J)
15600       H(K,J) = H(K,J - 1):VAR$(J) = VAR$(J - 1)
15610       H(K,J - 1) = SAV:VAR$( J - 1) = SAV$
15620      NEXT K
15630    NEXT J
15640  NEXT I
15650  REM  -PRINT BASIC VARIABLE INDICES AND VALUES
15660  POKE 33,34: POKE 32,6: PRINT
15670  PRINT "INDEX  VARIABLE"; TAB( 18);"VALUE"
15680  PRINT "-----  ---------"; TAB( 18);"-----"
15690  PRINT : POKE 34,7
15700  FOR I = 1 TO NC
15710    PRINT  TAB( 3);H(1,I); TAB( 8);VAR$(H(1,I)); TAB( 18); FN R(H(2
 ,I))
15720    IF I / 15 =  INT (I / 15) THEN  GOSUB 14770: HOME
15730  NEXT I
15740  PRINT : PRINT "** ";OBJ$;" =  "; FN R(Z1 * Z)
15750  POKE 32,0: POKE 33,40
15760  RETURN
15770
15780  REM      ** INTERACTIVE MODEL ENTRY, NAMED VARIABLES **
15790  GOSUB 17000
15800  TEXT : HOME : VTAB 5
15810  PRINT "WHAT OBJECTIVE DO YOU WANT TO ";M$;"IMIZE?"
15820  PRINT
15830  IF Q = 1 THEN  PRINT "(E.G.  TARGETS DESTROYED, UNITS FIELDED)":
 GOTO 15850
15840  PRINT "(E.G.  CASUALTIES, COST, MATERIEL)"
15850  PRINT : PRINT : HTAB 7: INPUT "";OBJ$
15860  HOME
15870  PRINT "LIST THE VARIABLES ( MAX 20 ) THAT AFFECT"
15880  PRINT OBJ$;", USING 1-6 CHAR DESCRIPTORS:"
15890  PRINT
15900  PRINT "TO STOP INPUT, HIT 'RETURN' W/O ENTRY."
15910  GOSUB 14860
15920 J = 1
15930  VTAB 7
15940  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
15950  PRINT "X(";J;" )"; TAB( 7);: INPUT "= ";A$: IF A$ = "" THEN  CALL
  - 998: CALL  - 958:NV = J - 1: GOTO 16000
```

```
15960 VAR$(J) =  LEFT$ (A$,6)
15970 J = J + 1: IF J > 20 THEN NV = 20: GOTO 16000
15980  GOTO 15940
15990  REM - CHANGES?
16000  PRINT : TEXT : POKE 34,23: GOSUB 14830: IF A$ >  < "Y" THEN 16060

16010 LIM = NV
16020  GOSUB 16880: IF CE = 0 THEN 16060
16030 VAR$(CE) =  LEFT$ (A$,6)
16040  GOTO 16020
16050
16060  TEXT : HOME
16070  PRINT "LIST CONSTRAINTS (MAX 20) THAT AFFECT"
16080  PRINT OBJ$;", USING 1-6 CHAR DESCRIPTORS"
16090  PRINT
16100  PRINT "TO STOP INPUT, HIT 'RETURN' W/O ENTRY."
16110  GOSUB 14860
16120 J = 1
16130  VTAB 7
16140  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
16150  PRINT "C(";J;")"; TAB( 7);: INPUT "= ";A$: IF A$ = "" THEN  CALL
  - 998: CALL  - 958:NC = J - 1: GOTO 16200
16160 CST$(J) =  LEFT$ (A$,6)
16170 J = J + 1: IF J > 20 THEN NC = 20: GOTO 16200
16180  GOTO 16140
16190  REM - CHANGES?
16200  PRINT : TEXT : POKE 34,23: GOSUB 14830: IF A$ >  < "Y" THEN 16250

16210 LIM = NC
16220  GOSUB 16880: IF CE = 0 THEN 16250
16230 CST$(CE) =  LEFT$ (A$,6)
16240  GOTO 16220
16250  GOTO 16370
16260
16270  REM      ** INTERACTIVE MODEL ENTRY, NUMBERS ONLY **
16280  GOSUB 17000
16290 OBJ$ = "Z"
16300  TEXT : HOME : VTAB 5
16310  INPUT "HOW MANY VARIABLES (MAX 20)?    ";NV: IF NV < 2 OR NV > 20
  THEN  CALL  - 998: GOTO 16310
16320  FOR I = 1 TO NV:VAR$(I) = "X(" +  STR$ (I) + ")": NEXT I
16330  PRINT
16340  INPUT "HOW MANY CONSTRAINTS (MAX 20)? ";NC: IF NC < 2 OR NC > 20
  THEN  CALL  - 998: GOTO 16340
16350  FOR I = 1 TO NC:CST$(I) = "C(" +  STR$ (I) + ")": NEXT I
16360  PRINT
16370  IF DE = 2 THEN  VTAB 23: CALL  - 958
16380  INPUT "MAX DIGITS PER COEFFICIENT (1-15)?   ";A$:FW =  VAL (A$) +
  2: IF FW < 7 THEN FW = 7
16390 CN =  INT (40 / (FW - 1)): REM  - NUMBER OF COLUMNS
16400
16410  TEXT : HOME : VTAB 2
```

```
16420  PRINT "ENTER THE COEFFICIENT FOR EACH VARIABLE;HIT 'RETURN' AFTER
  EACH COEFFICIENT"
16430  PRINT
16440  GOSUB 14860: PRINT : POKE 34,6
16450 AV = 0: IF (NV + 2) / CN >  INT ((NV + 2) / CN) THEN AV = 1
16460 I = 0
16470  PRINT "OBJECTIVE FUNCTION: ";M$;"IMIZE  ";OBJ$;" = ...": GOTO 164
 90
16480  VTAB 7: CALL  - 958: HTAB 14: PRINT "CONSTRAINT ";CST$(I)
16490  PRINT : PRINT
16500 K = 1
16510  FOR J = K * CN - CN + 1 TO K * CN
16520    IF J <  = NV THEN 16560
16530    IF I = 0 THEN 16600
16540  IF J = NV + 1 THEN  PRINT "< = >";: GOTO 16570
16550    IF J = NV + 2 THEN  PRINT "RHS";: GOTO 16600
16560    PRINT VAR$(J);
16570    HTAB FW * (J - (K - 1) * CN) + 1
16580  NEXT J
16590  IF  POS (H) = 1 THEN  CALL  - 998
16600  PRINT : PRINT
16610    FOR J = K * CN - CN + 1 TO K * CN
16620    IF J <  = NV THEN 16660
16630    IF I = 0 THEN 16710
16640      IF J = NV + 1 THEN  INPUT "  ";T$(I,NV + 1): GOSUB 16810: GOT
 O 16680
16650      IF J = NV + 2 THEN  INPUT "";T$(I,NV + 2): GOTO 16710
16660      INPUT "";A$: IF I = 0 THEN T$(0,J) =  STR$ ( VAL (A$) * Z1):
 GOTO 16680
16670     T$(I,J) = A$
16680      CALL  - 998: HTAB FW * (J - (K - 1) * CN) + 1
16690    NEXT J
16700  PRINT : PRINT : PRINT
16710 K = K + 1: IF K >  INT ((NV + 2) / CN) + AV THEN 16730
16720  GOTO 16510
16730  PRINT : GOSUB 14830: IF A$ >  < "Y" THEN 16760
16740  IF I = 0 THEN  VTAB 7: CALL  - 958: GOTO 16470
16750  GOTO 16480
16760 I = I + 1: IF I > NC THEN 16780
16770  GOTO 16480
16780  TEXT
16790  RETURN
16800
16810  REM  ** TRANSLATE INEQUALITY **
16820 A$ = T$(I,NV + 1)
16830  IF A$ = "<" OR A$ = "<=" OR A$ = "LT" OR A$ = "LE" THEN T$(I,NV +
  1) = "<": RETURN
16840  IF A$ = "=" OR A$ = "E" OR A$ = "EQ" THEN T$(I,NV + 1) = "=": RET
 URN
16850  IF A$ = ">" OR A$ = ">=" OR A$ = "GT" OR A$ = "GE" THEN T$(I,NV +
  1) = ">": RETURN
16860  RETURN
```

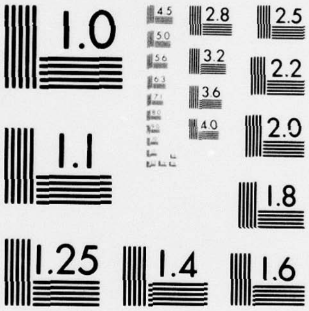
```
16870
16880  REM  ** PARAMETER ENTRY CORRECTION ROUTINE **
16890  VTAB 24: INPUT "INDEX OF ITEM TO CHANGE (0 TO STOP)? ";A$:CE =  I
 NT ( VAL (A$)): IF A$ = "" OR CE = 0 THEN  RETURN
16900  IF CE > LIM OR CE < 0 THEN 16890
16910  IF CE > 15 THEN 16950
16920  REM - SET WINDOW TO PROTECT PARTS OF SCREEN
16930  POKE 33,19: POKE 32,1: VTAB CE + 6: HTAB 8
16940  GOTO 16960
16950  POKE 33,20: POKE 32,20: VTAB CE - 9: HTAB 9
16960  INPUT "";A$
16970  POKE 32,0: POKE 33,40
16980  RETURN
16990
17000  REM   ** MAXIMIZE / MINIMIZE CHOICE **
17010  HOME : VTAB 7
17020  PRINT "     TO SOLVE THE PROBLEM, YOU CAN:"
17030  PRINT "     -- ----- --- -------  --- ---"
17040  PRINT : PRINT : PRINT
17050  PRINT "     1.  MAXIMIZE THE OBJ FUNCTION"
17060  PRINT : PRINT
17070  PRINT "OR  2.  MINIMIZE THE OBJ FUNCTION"
17080  VTAB 19: INPUT "       WHICH?  ";A$:Q =  VAL (A$): IF Q < 1 OR Q >
  2 THEN  CALL  - 998: GOTO 17080
17090  IF Q = 1 THEN Z1 = 1:M$ = "MAX": RETURN
17100  IF Q = 2 THEN Z1 =  - 1:M$ = "MIN": RETURN
17110  CALL  - 998: GOTO 17080
17120  RETURN
17130
17140  REM  ** DISPLAY CURRENT MODEL **
17150  TEXT : HOME : VTAB 5
17160  INPUT "DO YOU WANT IT ROUTED TO THE PRINTER? ";P$:P$ =  LEFT$ (P$
 ,1): IF P$ = "Y" THEN  GOSUB 14930
17170  HOME
17180  GOSUB 19710
17190  PRINT "CURRENT LP MODEL:  ";M$;"IMIZE  ";OBJ$
17200  PRINT "======= == =====   ========"
17210 TV = NV:DFLAG = 1: GOSUB 14980:DFLAG = 0
17220  IF P$ = "Y" THEN  PRINT D$;"PR#0"
17230  RETURN
17240
17250  REM   ** LP MODEL EDITING **
17260  GOSUB 19710
17270  TEXT : HOME : VTAB 2
17280  PRINT "  LP MODEL EDIT FUNCTIONS:"
17290  PRINT "  -- ----- ---- ----------"
17300  PRINT : PRINT
17310  PRINT "1.  DELETE A VARIABLE"
17320  PRINT "2.  DELETE A CONSTRAINT"
17330  PRINT
17340  PRINT "3.  ADD A VARIABLE"
17350  PRINT "4.  ADD A CONSTRAINT"
```

```
17360  PRINT
17370  PRINT "5.  CHANGE COEFFICIENTS BY VARIABLE"
17380  PRINT "6.  CHANGE COEFFICIENTS BY CONSTRAINT"
17390  PRINT "7.  CHANGE INDIVIDUAL COEFFICIENT"
17400  PRINT
17410  PRINT "8.  CHANGE RHS VALUES"
17420  PRINT "9.  CHANGE OBJECTIVE FUNCTION"
17430  PRINT : PRINT
17440  PRINT "10. RETURN TO LAST MENU"
17450  PRINT : PRINT : PRINT
17460  HTAB 5: INPUT "WHICH FUNCTION?  ";A$:DC =  INT ( VAL (A$))
17470  IF A$ = "" OR DC = 10 THEN  RETURN
17480  ON DC GOTO 17900,18030,18170,18330,18480,18770,19080,19320,19500
17490
17500  REM - COMPRESS DATA BASE
17510 Q = 0
17520  FOR J = 1 TO NV + 2
17530    IF J = VI THEN 17590
17540   Q = Q + 1:VAR$(Q) = VAR$(J)
17550 R = 0
17560    FOR I = 0 TO NC
17570     T$(R,Q) = T$(I,J):R = R + 1
17580    NEXT I
17590  NEXT J
17600 VI = 0: GOTO 17710
17610 Q = 0
17620  FOR I = 1 TO NC
17630    IF I = CI THEN 17690
17640   Q = Q + 1:CST$(Q) = CST$(I)
17650   R = 0
17660    FOR J = 1 TO NV + 2
17670     R = R + 1:T$(Q,R) = T$(I,J)
17680    NEXT J
17690  NEXT I
17700 CI = 0
17710  RETURN
17720
17730  REM   - LIST VARIABLES OR CONSTRAINTS
17740  VTAB 6: CALL  - 958
17750  FOR J = 1 TO N
17760    IF J > 20 THEN  POKE 33,11: POKE 32,29: VTAB J - 14: GOTO 17780

17770    IF J > 10 THEN  POKE 33,26: POKE 32,14: VTAB J - 4
17780   H = 2: IF J = 10 THEN H = 1
17790    PRINT J; SPC( H);
17800    IF VFLAG = 1 OR AFLAG = 1 THEN  PRINT VAR$(J): GOTO 17820
17810  PRINT CST$(J)
17820  NEXT J
17830  POKE 32,0: POKE 33,40
17840  IF DC = 7 THEN  VTAB 20: GOSUB 14860: RETURN
17850  IF AFLAG = 1 THEN  RETURN
17860  TEXT : POKE 34,23
```

AD-A080 215 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/4
COMPUTER ASSISTED ANALYSIS FOR MILITARY MANAGERS. (U)
DEC 79 R D CONTE
UNCLASSIFIED AFIT/GOR/MA/79D-3 NL

3 OF 3
AD
A080215

END
DATE
FILMED
3 - 80
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
17870  VTAB 23: GOSUB 14860
17880  RETURN
17890
17900  REM  - DELETE VARIABLE
17910  HOME : VTAB 3
17920  PRINT "DELETE VARIABLE"
17930  PRINT "------ --------"
17940 N = NV:VFLAG = 1: GOSUB 17730
17950  PRINT "WHICH VARIABLE ( 1-";NV;" )?  ";: INPUT "";A$: IF A$ = "" TH
 EN 17260
17960 VI =  VAL (A$)
17970  GOSUB 17500
17980 NV = NV - 1
17990  PRINT BELL$:N = NV: GOSUB 17730:VFLAG = 0
18000  GOSUB 14800: IF A$ = "Y" THEN 17910
18010  GOTO 17260
18020
18030  REM   - DELETE CONSTRAINT
18040  HOME : VTAB 3
18050  PRINT "DELETE CONSTRAINT"
18060  PRINT "------ ----------"
18070 N = NC: GOSUB 17730
18080  PRINT "WHICH CONSTRAINT (1-";NC;")?  ";: INPUT "";A$: IF A$ = ""
 THEN 17260
18090 CI =  VAL (A$)
18100  GOSUB 17610
18110  FOR J = 1 TO NV + 2:T$(NC,J ) = "0": NEXT J
18120 NC = NC - 1
18130  PRINT BELL$:N = NC: GOSUB 17730
18140  GOSUB 14800: IF A$ = "Y" THEN 18040
18150  GOTO 17260
18160
18170  REM - ADD VARIABLE
18180  TEXT : HOME : VTAB 3
18190  PRINT "ADD A VARIABLE"
18200  PRINT "--- - --------"
18210 N = NV:VFLAG = 1: GOSUB 17730:VFLAG = 0
18220  IF NV = 20 THEN  PRINT "SORRY, BUT THAT'S THE MAXIMUM!": GOSUB 14
 770: GOTO 17260
18230 AVFLAG = 1: REM  - FLAG FOR SUBROUTINE
18240  INPUT "NAME OF NEW VARIABLE?  ";A$: IF A$ = "" THEN 17260
18250  FOR I = 1 TO NC:T$( I,NV + 3 ) = T$(I,NV + 2):T$( I,NV + 2 ) = T$( I,N
 V + 1): NEXT I
18260 NV = NV + 1:VI = NV
18270 VAR$( VI ) =  LEFT$ (A$,6 )
18280  GOTO 18550
18290 AVFLAG = 0
18300  GOSUB 14800: IF A$ = "Y" THEN 18180
18310  GOTO 17260
18320
18330  REM   -ADD CONSTRAINT
18340  TEXT : HOME : VTAB 3
```

```
18350  PRINT "ADD A CONSTRAINT"
18360  PRINT "--- - ----------"
18370 N = NC: GOSUB 17730
18380  IF NC = 20 THEN  PRINT "SORRY, BUT THAT'S THE MAXIMUM!": GOSUB 14
 770: GOTO 17260
18390 AOBS = 1: REM - SET FLAG FOR SUBROUTINE
18400  INPUT "NAME OF NEW CONSTRAINT?  ";A$: IF A$ = "" THEN 17260
18410 NC = NC + 1:CI = NC
18420 CST$(CI) =  LEFT$ (A$,6)
18430  GOTO 18840
18440 AOBS = 0
18450  GOSUB 14800: IF A$ = "Y" THEN 18340
18460  GOTO 17260
18470
18480  REM  - CHANGE BY VARIABLE
18490  TEXT : HOME : VTAB 3
18500  PRINT "CHANGE COEFFICIENTS BY VARIABLE"
18510  PRINT "------ ------------ -- --------"
18520 N = NV:VFLAG = 1: GOSUB 17730:VFLAG = 0
18530  PRINT "WHICH VARIABLE (1-";NV;")?  ";: INPUT "";A$: IF A$ = "" TH
 EN 17260
18540 VI =  VAL (A$)
18550  TEXT : HOME : VTAB 2
18560  PRINT
18570  PRINT  TAB( 14);VAR$(VI)
18580  PRINT  TAB( 14);"-------"
18590  PRINT
18600  PRINT "CONSTRAINT   CURRENT    CHANGE TO"
18610  PRINT "----------   -------    ------ --"
18620  PRINT
18630  POKE 34,9
18640  PRINT "OBJ FUNC"; TAB( 14); VAL (T$(0,VI)) * Z1; TAB( 25);: INPUT
  "";A$: IF A$ = "" THEN  PRINT : GOTO 18670
18650 T$(0,VI) =  STR$ ( VAL (A$) * Z1)
18660  PRINT
18670  FOR I = 1 TO NC
18680    PRINT I; TAB( 5);CST$(I); TAB( 14);A(I,VI); TAB( 25);: INPUT ""
 ;A$: IF A$ = "" THEN 18700
18690   T$(I,VI) = A$
18700  NEXT I
18710  PRINT : PRINT
18720  IF AVFLAG = 1 THEN 18290
18730  POKE 34,23: GOSUB 14830: IF A$ = "Y" THEN 18550
18740  GOSUB 14800: IF A$ = "Y" THEN 18490
18750  GOTO 17260
18760
18770  REM   -CHANGE BY CONSTRAINT
18780  TEXT : HOME : VTAB 3
18790  PRINT "CHANGE COEFFICIENTS BY CONSTRAINT"
18800  PRINT "------ ------------ -- ----------"
18810 N = NC: GOSUB 17730
18820  PRINT "WHICH CONSTRAINT (1-";NC;")?  ";: INPUT "";A$: IF A$ = ""
 THEN 17260
```

```
18830 CI =  VAL (A$)
18840  TEXT : HOME : VTAB 2
18850  PRINT
18860  PRINT  TAB( 14);CST$(CI)
18870  PRINT  TAB( 14);"-------"
18880  PRINT
18890  PRINT "VARIABLE     CURRENT    CHANGE TO"
18900  PRINT "--------     -------    ------ --"
18910  PRINT
18920  POKE 34,9
18930  FOR J = 1 TO NV
18940    PRINT J; TAB( 5);VAR$(J); TAB( 14);T$(CI,J); TAB( 25);: INPUT "
 ";A$: IF A$ = "" THEN 18960
18950   T$(CI,J) = A$
18960  NEXT J
18970  PRINT
18980  PRINT "INEQUALITY"; TAB( 14);T$(CI,NV + 1); TAB( 25);: INPUT "";A
 $: IF A$ = "" THEN 19000
18990 T$(CI,NV + 1) = A$
19000  PRINT
19010  PRINT "RHS"; TAB( 14);T$(CI,NV + 2); TAB( 25);: INPUT "";A$: IF A
 $ = "" THEN 19030
19020 T$(CI,NV + 2) = A$
19030  IF AOBS = 1 THEN 18440
19040  POKE 34,23: GOSUB 14830: IF A$ = "Y" THEN 18840
19050  GOSUB 14800: IF A$ = "Y" THEN 18780
19060  GOTO 17260
19070
19080  REM  - CHANGE INDIVIDUAL COEFFICIENT
19090  TEXT : HOME : VTAB 3
19100  PRINT "CHANGE INDIVIDUAL COEFFICIENT"
19110  PRINT "------ ---------- -----------"
19120  POKE 34,4
19130 N = NV:VFLAG = 1: GOSUB 17730:VFLAG = 0
19140  PRINT "WHICH VARIABLE ( 1-";NV;" )?    ";: INPUT "";A$: IF A$ = ""
 THEN 17260
19150 VI =  VAL (A$)
19160  HOME :N = NC: GOSUB 17730
19170  PRINT "WHICH CONSTRAINT (1-";NC;")?  ";: INPUT "";A$: IF A$ = ""
 THEN 17260
19180 CI =  VAL (A$)
19190  TEXT : HOME : VTAB 3
19200  PRINT  TAB( 14);VAR$(VI)
19210  PRINT  TAB( 14);"-------"
19220  PRINT
19230  PRINT "CONSTRAINT   CURRENT    CHANGE TO"
19240  PRINT "----------   -------    ------ --"
19250  PRINT
19260  PRINT CI; TAB( 5);CST$(CI); TAB( 14);T$(CI,VI); TAB( 25);: INPUT
 "";A$: IF A$ = "" THEN 19290
19270 T$(CI,VI) = A$
19280  POKE 34,23: GOSUB 14830: IF A$ = "Y" THEN 19190
```

```
19290  GOSUB 14800: IF A$ = "Y" THEN 19090
19300  GOTO 17260
19310
19320  REM   - CHANGE RHS VALUES, B(I)
19330  HOME : VTAB 3
19340  PRINT "CHANGE RHS VALUES"
19350  PRINT "------ --- ------"
19360  PRINT : PRINT
19370  PRINT  TAB( 14);"RHS"
19380  PRINT  TAB( 14);"---"
19390  PRINT
19400  PRINT "CONSTRAINT   CURRENT    CHANGE TO"
19410  PRINT "----------   -------    ------ --"
19420  PRINT
19430  FOR I = 1 TO NC
19440  PRINT I; TAB( 5);CST$( I ); TAB( 14);T$( I,NV + 2 ); TAB( 25);: INPUT
  "";A$: IF A$ = "" THEN 19460
19450 T$( I,NV + 2 ) = A$
19460  NEXT I
19470  GOSUB 14830: IF A$ = "Y" THEN 19330
19480  GOTO 17260
19490
19500  REM  - CHANGE OBJECTIVE FUNCTION
19510  HOME : VTAB 2
19520  PRINT "CHANGE OBJECTIVE FUNCTION"
19530  PRINT "------ --------- --------"
19540  PRINT : PRINT
19550  PRINT "VARIABLE     CURRENT    CHANGE TO"
19560  PRINT "--------     -------    ------ --"
19570  PRINT
19580  POKE 34,8
19590  PRINT "OBJECTIVE"; TAB( 14);M$;"IMIZE"; TAB( 25);: INPUT "";A$: I
 F A$ = "" THEN 19610
19600 M$ =  LEFT$ (A$,3)
19610  PRINT : PRINT  TAB( 14); LEFT$ (OBJ$,10); TAB( 25);: INPUT "";A$:
  IF A$ = "" THEN 19630
19620 OBJ$ =  LEFT$ (A$,15)
19630  PRINT
19640  FOR J = 1 TO NV
19650  PRINT J; TAB( 5);VAR$( J ); TAB( 14); VAL (T$(0,J)) * Z1; TAB( 25);
 : INPUT "";A$: IF A$ = "" THEN 19670
19660 T$( 0,J ) =  STR$ ( VAL (A$) * Z1 )
19670  NEXT J
19680  GOSUB 14770
19690  GOTO 17260
19700
19710  REM  ** DISMANTLE STRING ARRAY INTO LP FORMAT **
19720  GOSUB 20160
19730  REM - SORT CONSTRAINTS
19740  FOR I = 1 TO NC - 1
19750    FOR J = I + 1 TO NC
19760      IF T$( J,NV + 1 ) >  = T$( I,NV + 1) THEN 19830
```

```
19770     S$ = CST$(I):CST$(I) = CST$(J):CST$(J) = S$
19780      FOR Q = 1 TO NV + 2
19790       SAV$ = T$(I,Q)
19800       T$(I,Q) = T$(J,Q)
19810       T$(J,Q) = SAV$
19820      NEXT Q
19830    NEXT J
19840  NEXT I
19850
19860  REM   *** BREAK UP STRING ARRAY ***
19870
19880  REM  -CONSTRAINT COEFF'S AND INEQUALITY SIGNS
19890  FOR I = 1 TO NC
19900   CT$(I) = T$(I,NV + 1)
19910    FOR J = 1 TO NV
19920     A(I,J) =  VAL (T$(I,J))
19930    NEXT J
19940  NEXT I
19950  FOR J = 1 TO NV:C(J) =  VAL (T$(0,J)): NEXT J
19960
19970  REM  - RIGHT HAND SIDE, INCL ADJUSTMENT FOR NEGATIVE RHS
19980  FOR I = 1 TO NC
19990   B(I) =  VAL (T$(I,NV + 2))
20000    IF B(I) >  = 0 THEN 20050
20010      FOR J = 1 TO NV:A(I,J) =  - A(I,J): NEXT
20020      IF CT$(I) = "<" THEN CT$(I) = ">"
20030      IF CT$(I) = ">" THEN CT$(I) = "<"
20040     B(I) =  - B(I)
20050  NEXT I
20060
20070  REM - COUNT LESS THAN'S, EQUALITIES, GREATER THAN'S
20080 L = 0:E = 0:G = 0
20090  FOR I = 1 TO NC
20100    IF CT$(I) = "<" THEN L = L + 1
20110    IF CT$(I) = "=" THEN E = E + 1
20120    IF CT$(I) = ">" THEN G = G + 1
20130  NEXT I
20140  RETURN
20150
20160  REM  ** CLEAR COMPLETE ARRAY **
20170  FOR I = 0 TO NC
20180   B(I) = 0:D(I) = 0
20190    FOR J = 1 TO NV + NC + G
20200     A(I,J) = 0:C(J) = 0
20210    NEXT J
20220  NEXT I
20230  RETURN
20240
20250  REM   ** SAVE DISK FILE **
20260  GOSUB 19710
20270  HOME : VTAB 3
20280  PRINT  TAB( 11);"SAVE MODEL TO DISK"
```

```
20290  PRINT  TAB( 11);"---- ----- -- ----"
20300  PRINT : PRINT : PRINT
20310  HTAB 8: INPUT "SAVE UNDER WHAT NAME?  ";SFILE$: IF SFILE$ = "" TH
 EN  RETURN
20320  PRINT D$;"OPEN ";SFILE$;D$;"DELETE ";SFILE$;D$;"OPEN ";SFILE$
20330  PRINT D$;"WRITE ";SFILE$
20340  PRINT DE: PRINT NV: PRINT NC: PRINT M$: PRINT OBJ$: PRINT Z1: PRI
 NT L: PRINT E: PRINT G
20350  FOR I = 0 TO NC
20360    PRINT CST$(I)
20370    FOR J = 1 TO NV + 2
20380      PRINT VAR$( J): PRINT T$(I,J)
20390    NEXT J
20400  NEXT I
20410  PRINT D$;"CLOSE ";SFILE$
20420  RETURN
20430
20440  REM  ** READ DISK FILE **
20450  GOSUB 20160
20460  HOME : VTAB 5
20470  INPUT "DO YOU WANT TO SEE THE DISK CATALOG?  ";A$
20480  IF  LEFT$ (A$,1) = "Y" THEN  HOME : PRINT D$;"CATALOG": VTAB 23:
 GOSUB 14860: GOTO 20530
20490  HOME : VTAB 3
20500  PRINT  TAB( 8);"READ DATA FILE FROM DISK"
20510  PRINT  TAB( 8);"---- ---- ---- ---- ----"
20520  PRINT : PRINT : PRINT
20530  HTAB 8: INPUT "WHAT DATA FILE NAME?  ";RFILE$: IF RFILE$ = "" THE
 N  RETURN
20540  PRINT D$;"OPEN ";RFILE$
20550  PRINT D$;"READ ";RFILE$
20560  INPUT DE,NV,NC,M$,OBJ$,Z1,L,E,G
20570  FOR I = 0 TO NC
20580    INPUT CST$(I)
20590    FOR J = 1 TO NV + 2
20600      INPUT VAR$( J),T$(I,J)
20610    NEXT J
20620  NEXT I
20630  PRINT D$;"CLOSE ";RFILE$
20640  RETURN
20650
20660  REM   ** INTRODUCTORY REMARKS **
20670  HOME : VTAB 7
20680  PRINT "LINEAR PROGRAMMING IS USED TO DETERMINE THE BEST ALLOCATIO
 N FOR SCARCE RESOURCESIN ORDER TO"
20690  PRINT : PRINT
20700  PRINT  TAB( 10);"MAXIMIZE OR MINIMIZE"
20710  PRINT : PRINT
20720  PRINT "A LINEAR FUNCTION DEFINING AN OBJECTIVE SUCH AS PROFIT, CO
 ST, TONNAGE, ETC."
20730  GOSUB 14890
20740  RETURN
```

Section 7, DECISION

```
10000 BELL$ = "": REM - BELL
10010  PRINT BELL$
10020  TEXT : HOME : VTAB 5
10030  PRINT "* * * * * * * * * * * * * * * * * * * *"
10040  PRINT
10050  PRINT "      V A L U E   M A T R I X"
10060  PRINT
10070  PRINT "          D E C I S I O N"
10080  PRINT
10090  PRINT "          A N A L Y S I S"
10100  PRINT : PRINT
10110  PRINT "                BY"
10120  PRINT
10130  PRINT "          ROBERT D. CONTE"
10140  PRINT
10150  PRINT "* * * * * * * * * * * * * * * * * * * *"
10160  VTAB 24: INPUT "  DO YOU WANT INTRODUCTORY REMARKS?  ";A$: IF  LE
 FT$ (A$,1) = "Y" THEN  GOSUB 17830
10170
10180  CLEAR
10190  REM - DIMENSIONED FOR 15 OPTIONS, 20 CRITERIA, AND 5 STATES
10200  DIM OP$(15),CR$(20),SN$(5),W(20),NW(20),P(5),NP(5)
10210  DIM V(15,5,20),CV(15,5),EV(15),CP(5),RP(20)
10220  DEF  FN R(X) =  INT (X * 1000 + .5) / 1000: REM   - 3 PLACE ROUND
  OFF FUNCTION
10230  DEF  FN S(X) =  INT (X * 100 + .5) / 100: REM    - 2 PLACE ROUND
 OFF  FUNCTION
10240 D$ =  CHR$ (13) +  CHR$ (4)
10250 OP$(0) = "** DECISION OPTIONS **"
10260 CR$(0) = "** JUDGEMENT CRITERIA **"
10270 SN$(0) = "** STATES OF NATURE **"
10280
10290  HOME : POKE 33,36: POKE 32,4
10300  VTAB 3
10310  PRINT "    TO ENTER A MODEL, YOU CAN:"
10320  PRINT "    -- ----- - -----  --- ---"
10330  PRINT : PRINT : PRINT
10340  PRINT "1.  READ EXISTING MODEL FROM DISK"
10350  PRINT : PRINT
10360  PRINT "2.  CREATE MODEL INTERACTIVELY"
10370  TEXT
10380  VTAB 15: HTAB 9: INPUT "WHICH?  ";A$:DE =  VAL (A$): IF A$ = "" O
 R DE = 1 THEN  GOSUB 15530: GOTO 10420
10390  IF DE = 2 THEN  GOSUB 11070: GOTO 10420
10400  CALL  - 998: GOTO 10380
10410
10420  REM - MAIN MENU
```

```
10430  TEXT : HOME : POKE 33,33: POKE 32,7
10440  VTAB 3
10450 X =  FRE (0): REM  - CLEANS UP UNUSED STRING VALUES
10460  PRINT "    DATA MANAGEMENT"
10470  PRINT "    ---- ----------"
10480  PRINT : PRINT
10490  PRINT "1.  DISPLAY MODEL"
10500  PRINT
10510  PRINT "2.  EDIT MODEL"
10520  PRINT
10530  PRINT "3.  SAVE MODEL TO DISK"
10540  PRINT
10550  PRINT "4.  ENTER ANOTHER MODEL"
10560  PRINT
10570  PRINT "5.  QUIT THE PROGRAM"
10580  PRINT : PRINT
10590  PRINT "6.  EVALUATE DECISION OPTIONS"
10600  PRINT : PRINT : PRINT
10610  INPUT "    WHICH OPTION?  ";A$:DO =  VAL (A$): TEXT : IF A$ = ""
 THEN 10660
10620  IF DO < 1 OR DO > 6 THEN  CALL  - 998: GOTO 10610
10630  ON DO GOSUB 17350,15990,15780: IF DO < 4 THEN 10430
10640  ON DO - 3 GOTO 10180,10700,10660
10650
10660  REM  - FIND OPTIMUM OPTION FOR CURRENT MODEL
10670  GOSUB 13600
10680  INPUT "WANT TO RUN SENSITIVITY ANALYSIS?  ";A$: IF  LEFT$ (A$,1)
 = "Y" THEN  GOSUB 14280
10690  GOTO 10430
10700  PRINT D$;"BLOAD CHAIN, A520"
10710  CALL 520"CAAMM MASTER"
10720  END
10730
10740  REM    *********************************************************

10750
10760  REM  ** UTILITY SUBROUTINES **
10770  VTAB 23: GOSUB 10830: PRINT "    HIT 'RETURN  TO PROCEED...  ";:
  GET A$: PRINT
10780  RETURN
10790
10800  TEXT : POKE 34,23: VTAB 23: GOSUB 10830: INPUT "         NEED TO M
 AKE CHANGES?  ";A$:A$ =  LEFT$ (A$,1): PRINT
10810  RETURN
10820
10830  PRINT "- - - - - - - - - - - - - - - - - - - -"
10840  RETURN
10850
10860  TEXT : POKE 34,23: VTAB 23: GOSUB 10830: INPUT "      NEED TO READ
 JUST VALUES?  ";A$:A$ =  LEFT$ (A$,1): PRINT
10870  RETURN
10880
```

```
10890  VTAB 23: GOSUB 10830: PRINT "HIT 'RETURN' TO GO ON, OR 'Q' TO QUI
 T ";
10900  GET A$: IF  ASC (A$ ) = 81 THEN 10700
10910  PRINT
10920  RETURN
10930
10940  VTAB 23: GOSUB 10830: PRINT "HIT 'RETURN' TO GO ON, OR 'S' TO STO
 P ";
10950  GET A$: IF  ASC (A$ ) = 83 THEN  POP : POP : GOTO 17350
10960  PRINT
10970  RETURN
10980
10990  PRINT : VTAB 23: GOSUB 10830: INPUT "       EDIT ANOTHER?  ";A$:A
 $ =  LEFT$ (A$,1): PRINT
11000  RETURN
11010
11020  REM  ** TURN ON PRINTER **
11030  PRINT D$;"PR#4": PRINT : POKE 1148,32: POKE 1788,80
11040  PRINT : GOSUB 10830: PRINT : PRINT
11050  RETURN
11060
11070  REM   ** DECISION MATRIX DATA ENTRY **
11080  TEXT : HOME : VTAB 3
11090  PRINT "DECISION MATRIX DATA ENTRY"
11100  PRINT "-------- ------ ---- -----"
11110  PRINT : PRINT : PRINT
11120  PRINT "THE DECISION MATRIX ALLOWS 3 ELEMENTS:"
11130  PRINT : PRINT
11140  PRINT "  DECISION OPTIONS / ALTERNATIVES"
11150  PRINT
11160  PRINT "  JUDGEMENT CRITERIA / ATTRIBUTES"
11170  PRINT
11180  PRINT "  UNCERTAIN STATES OF NATURE  (OPTIONAL)"
11190  VTAB 23: GOSUB 10830: HTAB 14: INPUT "NEED HELP?  ";A$: IF  LEFT$
  (A$,1) = "Y" THEN  GOSUB 18120
11200  REM  - DECISION OPTIONS INFO
11210  HOME
11220  PRINT "LIST DECISION OPTIONS BEING CONSIDERED   (MAX 15),  USING
  1-9 CHARACTER NAMES."
11230  PRINT
11240  PRINT "TO STOP INPUT,  HIT 'RETURN' W/O ENTRY."
11250  GOSUB 10830
11260 J = 1
11270  VTAB 7
11280  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
11290  PRINT J; TAB( 5);: INPUT "";A$: IF A$ = "" THEN  CALL  - 998: CAL
 L  - 958:NOP = J - 1: GOTO 11330
11300 OP$(J) =  LEFT$ (A$,9)
11310 J = J + 1: IF J > 15 THEN NOP = 15: GOTO 11390
11320  GOTO 11280
11330  IF NOP > 1 THEN 11390
11340  HOME : VTAB 7
```

```
11350  PRINT BELL$;"*** YOU CAN'T HAVE MUCH OF A DECISION  UNLESS YOU H
 AVE AT LEAST TWO OPTIONS !!!"
11360  PRINT "THINK ABOUT IT AND HIT 'RETURN' WHEN YOUARE READY TO START
   AGAIN...";
11370  GET A$: GOTO 11200
11380  REM - CHANGES?
11390  GOSUB 10800: IF A$ >  < "Y" THEN 11450
11400 LIM = NOP:H = 0
11410  GOSUB 13370: IF CE = 0 THEN 11450
11420 OP$(CE) =  LEFT$ (A$,10)
11430  GOTO 11410
11440
11450  REM - ENTER JUDGEMENT CRITERION INFO
11460  TEXT : HOME
11470  PRINT "LIST JUDGEMENT CRITERIA BEING CONSIDERED  (MAX 20),  USING
   1-8 CHARACTER NAMES."
11480  PRINT
11490  PRINT "TO STOP INPUT,  HIT 'RETURN' W/O ENTRY."
11500  GOSUB 10830
11510 J = 1
11520  VTAB 7
11530  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
11540  PRINT J; TAB( 5);: INPUT "";A$: IF A$ = "" THEN  CALL  - 998: CAL
 L  - 958:NC = J - 1: GOTO 11580
11550 CR$(J) =  LEFT$ (A$,8)
11560 J = J + 1: IF J > 20 THEN NC = 20: GOTO 11630
11570  GOTO 11530
11580  IF NC > 1 THEN 11630
11590  HOME : VTAB 7
11600  PRINT BELL$;"*** YOU CAN'T HAVE MUCH OF A DECISION  UNLESS YOU H
 AVE AT LEAST TWO JUDGEMENT  CRITERIA !!!"
11610  PRINT "THINK ABOUT IT AND HIT 'RETURN' WHEN YOUARE READY TO START
   AGAIN...";
11620  GET A$: GOTO 11450
11630  REM - CHANGES?
11640  GOSUB 10800: IF A$ >  < "Y" THEN 11700
11650 LIM = NC:H = 0
11660  GOSUB 13370: IF CE = 0 THEN 11700
11670 CR$(CE) =  LEFT$ (A$,8)
11680  GOTO 11660
11690
11700  REM - CRITERION WEIGHTINGS
11710  TEXT : HOME
11720  PRINT "ASSESS RELATIVE IMPORTANCE OF JUDGEMENT   CRITERIA BY ENTE
 RING WEIGHT FACTORS."
11730  PRINT
11740  PRINT "WEIGHTS WILL BE NORMALIZED AUTOMATICALLY";
11750  GOSUB 10830
11760  VTAB 7
11770  FOR J = 1 TO NC
11780  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
11790    PRINT J; TAB( 5);CR$( J);: IF RFLAG = 1 THEN  PRINT  TAB( 15);W(
 J): GOTO 11810
```

```
11800  PRINT
11810  NEXT J
11820  POKE 33,19: POKE 32,1
11830  VTAB 7
11840  FOR J = 1 TO NC
11850    IF J > 15 THEN  POKE 33,19: POKE 32,21: VTAB J - 9
11860    HTAB 14: INPUT "";A$:W(J) =  VAL (A$): IF W(J) <  = 0 THEN  CAL
 L  - 998: GOTO 11860
11870  NEXT J
11880  REM - CHANGES?
11890  GOSUB 10800: IF A$ >  < "Y" THEN 11940
11900 LIM = NC:H = 10
11910  GOSUB 13370: IF CE = 0 THEN 11940
11920 W(CE) =  VAL (A$)
11930  GOTO 11910
11940  REM  - NORMALIZE AND LIST
11950 Q = 0
11960  FOR J = 1 TO NC:Q = Q + W(J): NEXT J
11970  FOR J = 1 TO NC:NW(J) = W(J) / Q: NEXT J
11980  TEXT : HOME : VTAB 3: PRINT  TAB( 4);"** NORMALIZED CRITERION WEI
 GHTS **": PRINT : GOSUB 10830
11990  VTAB 7
12000  FOR J = 1 TO NC
12010    IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
12020    PRINT J; TAB( 5);CR$(J); TAB( 14); FN R(NW(J))
12030  NEXT J
12040  IF MD = 2 THEN  RETURN
12050  REM - READJUST WEIGHTS?
12060  GOSUB 10860: IF A$ = "Y" THEN RFLAG = 1: GOTO 11700
12070 RFLAG = 0
12080  IF DC = 4 THEN  RETURN
12090
12100  REM  -ENTER STATES OF NATURE INFO
12110  TEXT : HOME : VTAB 7
12120  PRINT "IS YOUR DECISION AFFECTED BY UNCERTAIN"
12130  PRINT
12140  HTAB (11): INPUT "STATES OF NATURE?  ";A$:A$ =  LEFT$ (A$,1)
12150  IF A$ >  < "Y" THEN NS = 1:P(1) = 1:NP(1) = 1:SN$(1) = "CERTAINTY
 ": GOTO 12820
12160
12170  HOME
12180  PRINT "LIST STATES OF NATURE BEING CONSIDERED    (MAX 5),  USING
 1-8 CHARACTER NAMES."
12190  PRINT
12200  PRINT "TO STOP INPUT,  HIT 'RETURN' W/O ENTRY."
12210  GOSUB 10830
12220 J = 1
12230  VTAB 7
12240  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
12250  PRINT J; TAB( 5);: INPUT "";A$: IF A$ = "" THEN  CALL  - 998: CAL
 L  - 958:NS = J - 1: GOTO 12290
12260 SN$(J) =  LEFT$ (A$,8)
12270 J = J + 1: IF J > 5 THEN NS = 5: GOTO 12360
```

```
12280  GOTO 12240
12290  IF NS > 1 THEN 12350
12300  HOME : VTAB 7
12310  PRINT BELL$;"*** WHERE IS THE UNCERTAINTY IF YOU ONLYHAVE ONE STA
 TE OF NATURE ? !!!"
12320  PRINT
12330  PRINT "THINK ABOUT IT AND HIT 'RETURN' WHEN YOUARE READY TO START
  AGAIN...";
12340  GET A$: GOTO 12110
12350  REM - CHANGES?
12360  GOSUB 10800: IF A$ >  < "Y"  THEN 12420
12370 LIM = NS:H = 0
12380  GOSUB 13370: IF CE = 0 THEN 12420
12390 SN$(CE) =  LEFT$ (A$,8)
12400  GOTO 12380
12410
12420  REM - STATE PROBABILITIES
12430  TEXT : HOME
12440  PRINT "ESTIMATE THE PROBABILITY OR LIKELIHOOD    OF EACH OF THE U
 NCERTAIN OUTCOMES."
12450  PRINT
12460  PRINT "NUMBERS WILL BE NORMALIZED AUTOMATICALLY";
12470  GOSUB 10830
12480  VTAB 7
12490  FOR J = 1 TO NS
12500  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
12510    PRINT J; TAB( 5);SN$( J);: IF RFLAG = 1 THEN  PRINT  TAB( 15);P(
 J): GOTO 12530
12520    PRINT
12530  NEXT J
12540  POKE 33,19: POKE 32,1
12550  VTAB 7
12560  FOR J = 1 TO NS
12570  IF J > 15 THEN  POKE 33,19: POKE 32,21: VTAB J - 9
12580    HTAB 14: INPUT "";A$:P(J) =  VAL (A$): IF P(J) <  = 0 THEN  CAL
 L - 998: GOTO 12580
12590  NEXT J
12600  REM - CHANGES?
12610  GOSUB 10800: IF A$ >  < "Y"  THEN 12660
12620 LIM = NS:H = 10
12630  GOSUB 13370: IF CE = 0 THEN 12660
12640 P(CE) =  VAL (A$)
12650  GOTO 12630
12660  REM  - NORMALIZE AND LIST
12670 Q = 0
12680  FOR J = 1 TO NS:Q = Q + P(J): NEXT J
12690  FOR J = 1 TO NS:NP(J) = P(J) / Q: NEXT J
12700  TEXT : HOME : VTAB 3: PRINT  TAB( 3);"** NORMALIZED STATE PROBABI
 LITIES **": PRINT : GOSUB 10830
12710  VTAB 7
12720  FOR J = 1 TO NS
12730  IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
```

```
12740    PRINT J; TAB( 5);SN$( J); TAB( 14 ); FN R(NP(J))
12750  NEXT J
12760  IF MD = [illegible] THEN  RETURN
12770  REM - [illegible] 'UST PROBABILITIES?
12780  GOSUB [illegible] : IF A$ = "Y" THEN RFLAG = 1: GOTO 12420
12790 RFLAG = [illegible]
12800  IF DC = 5 THEN  RETURN
12810
12820  REM   ** VALUE MATRIX ENTRY **
12830  TEXT : HOME : VTAB 2: HTAB 5
12840  PRINT "** VALUE MATRIX DATA ENTRY **"
12850  POKE 34,3
12860  VTAB 8: HTAB 10
12870  INPUT "NEED INSTRUCTIONS?  ";A$: IF  LEFT$ (A$,1) = "Y" THEN  GOS
 UB 18280
12880 AO = 0: IF NOP / 15 >  INT (NOP / 15) THEN AO = 1
12890 AS = 0: IF NS / 3 >  INT (NS / 3) THEN AS = 1
12900  REM - FIRST CRITERION
12910 CR = 1
12920  TEXT : HOME
12930  PRINT "CRITERION ";CR;": ";CR$(CR); TAB( 25);"WEIGHTING: "; FN R(
 NW(CR))
12940  PRINT
12950  REM   - FIRST BLOCK OF 3 STATES
12960 BS = 1
12970  VTAB 3: PRINT "  STATES -->"; TAB( 14);
12980  FOR S1 = BS * 3 - 2 TO BS * 3
12990    PRINT SN$(S1); TAB( 14 + 9 * (S1 - (BS - 1) * 3))
13000    IF S1 = NS THEN 13020
13010  NEXT S1
13020  PRINT : VTAB 4: PRINT "OPTIONS"; TAB( 14);
13030  FOR S2 = BS * 3 - 2 TO BS * 3
13040    PRINT S2; TAB( 14 + 9 * (S2 - (BS - 1) * 3))
13050    IF S2 = NS THEN 13070
13060  NEXT S2
13070  PRINT : VTAB 5: GOSUB 10830: PRINT
13080  REM  - FIRST BLOCK OF 15 OPTIONS
13090 BO = 1
13100  FOR OP = BO * 15 - 14 TO BO * 15
13110    PRINT OP; TAB( 4);OP$( OP)
13120    IF OP = NOP THEN  PRINT : GOTO 13140
13130  NEXT OP
13140 ST = BS * 3 - 2
13150    POKE 33,7: POKE 32,6 + (9 * (ST - (BS - 1) * 3))
13160    VTAB 7
13170    FOR OP = BO * 15 - 14 TO BO * 15
13180      IF RFLAG = 1 THEN  PRINT V(OP,ST,CR): GOTO 13200
13190      INPUT "";A$:V(OP,ST,CR) =  VAL (A$)
13200      IF OP = NOP THEN  PRINT : GOTO 13220
13210    NEXT OP
13220  IF ST = NS THEN  PRINT : GOTO 13240
13230 ST = ST + 1: IF ST <  = BS * 3 THEN 13150
```

```
13240  IF MD = 4 THEN  TEXT : POKE 34,23: GOSUB 10940: GOTO 13320
13250  IF DC = 6 THEN  TEXT : POKE 34,23: VTAB 23: GOSUB 10830: GOTO 132
 80
13260  REM - CHANGES?
13270  GOSUB 10800: IF A$ >  < "Y" THEN 13320
13280  GOSUB 13490: IF OP = 0 OR S = 0 THEN 13320
13290 V(OP,S,CR) =  VAL (A$)
13300  GOTO 13280
13310
13320 BO = BO + 1: IF BO <  =  INT (NOP / 15) + AO THEN  POKE 34,6: HOME
 : GOTO 13100
13330 BS = BS + 1: IF BS <  =  INT (NS / 3) + AS THEN  POKE 34,2: HOME :
 GOTO 12970
13340 CR = CR + 1: IF CR <  = NC THEN 12920
13350  RETURN
13360
13370  REM    ** PARAMETER ENTRY CORRECTION ROUTINE **
13380  VTAB 24: INPUT "INDEX OF ITEM TO CHANGE (0 TO STOP)? ";A$:CE =  V
 AL ( A$): IF A$ = "" OR CE = 0 THEN  RETURN
13390  IF CE > LIM OR CE < 0 THEN 13380
13400  IF CE > 15 THEN 13440
13410  REM - SET WINDOW TO PROTECT PARTS OF SCREEN
13420  POKE 33,19: POKE 32,1: VTAB CE + 6: HTAB 4 + H
13430  GOTO 13450
13440  POKE 33,20: POKE 32,20: VTAB CE - 9: HTAB 5 + H
13450  INPUT "";A$
13460  POKE 32,0: POKE 33,40
13470  RETURN
13480
13490  REM    ** MATRIX VALUE CORRECTION ROUTINE **
13500  VTAB 24: HTAB 10: INPUT "WHICH OPTION?  ";A$:OP =  VAL (A$): IF A
 $ = "" OR OP = 0 THEN  RETURN
13510  IF OP > NOP OR OP < 0 THEN 13500
13520  HTAB 10: INPUT "WHICH STATE?  ";A$:S =  VAL (A$): IF A$ = "" OR S
 = 0 THEN  RETURN
13530  IF S > BS * 3 OR S < BS * 3 - 2 THEN 13520
13540  POKE 33,7: POKE 32,6 + (9 * (S - (BS - 1) * 3))
13550  VTAB OP + 6 - (BO - 1) * 15
13560  INPUT "";A$
13570  POKE 32,0: POKE 33,40
13580  RETURN
13590
13600  REM   ** ADDITIVE WEIGHTING ALGORITHM TO DETERMINE OPTIMUM OPTION
 **
13610  TEXT : HOME : VTAB 4
13620  PRINT "TO DETERMINE THE OPTIMUM DECISION OPTION";
13630  PRINT "-- --------- --- ------- -------- ------"
13640  PRINT
13650  PRINT  TAB( 17);"YOU CAN:"
13660  PRINT  TAB( 17);"--- ---"
13670  PRINT : PRINT : PRINT
13680  PRINT  TAB( 4);"1.  MAXIMIZE THE DECISION VALUE"
```

```
13690  PRINT : PRINT
13700  PRINT  TAB( 4);"2.  MINIMIZE THE DECISION VALUE"
13710  VTAB 20: HTAB 8
13720  INPUT "WHICH?  ";A$:M =  VAL (A$): IF A$ = "" THEN M = 1
13730  IF M < 1 OR M > 2 THEN  CALL  - 998: GOTO 13720
13740  PRINT : PRINT : INPUT "DO YOU WANT OUTPUT TO THE PRINTER?  ";P$:P
 $ =  LEFT$ (P$,1)
13750  IF M = 1 THEN M$ = "MAX": GOTO 13780
13760  IF M = 2 THEN M$ = "MIN"
13770
13780  REM - WEIGHT AND SUM MATRICES ACROSS ALL CRITERIA
13790  FOR I = 1 TO NOP: FOR J = 1 TO NS:CV(I,J) = 0: NEXT J: NEXT I
13800  FOR K = 1 TO NC
13810    FOR J = 1 TO NS
13820      FOR I = 1 TO NOP
13830       CV(I,J) = CV(I,J) + V(I,J,K) * NW(K)
13840      NEXT I
13850    NEXT J
13860  NEXT K
13870
13880  REM  - CALCULATE EXPECTED VALUE BY INCORPORATING STATE PROBABILIT
 IES
13890  FOR I = 1 TO NOP:EV(I) = 0: NEXT I
13900  FOR J = 1 TO NS
13910    FOR I = 1 TO NOP
13920     EV(I) = EV(I) + CV(I,J) * NP(J)
13930    NEXT I
13940  NEXT J
13950
13960  REM - FIND MAX OR MIN
13970 T = EV(1):MO = 1
13980  FOR J = 2 TO NOP
13990    IF M = 2 AND EV(J) >  = T THEN 14030
14000    IF M = 2 THEN 14020
14010    IF EV(J) <  = T THEN 14030
14020   T = EV(J):MO = J
14030  NEXT J
14040
14050  IF P$ = "Y" THEN  GOSUB 11020
14060  HOME : VTAB 5
14070  PRINT BELL$ + BELL$;"***  WHEN DECISION VALUE IS ";M$;"IMIZED..."

14080  PRINT : PRINT   PRINT
14090  PRINT "THE OPTIMUM OPTION IS  ** ";OP$(MO);" **"
14100  PRINT : PRINT
14110  PRINT "   WITH AN EXPECTED VALUE OF  "; FN S(EV(MO))
14120  GOSUB 10770
14130
14140  REM  ** EXPECTED VALUE TABLE **
14150  HOME
14160  PRINT  TAB( 10);"EXPECTED VALUE TABLE"
14170  PRINT  TAB( 10);"-------- ----- -----"
```

```
14180  PRINT
14190  PRINT "OPTION          VALUE"
14200  PRINT "------          -----"
14210  FOR J = 1 TO NOP
14220    IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 9
14230    PRINT J; TAB( 4);OP$( J); TAB( 15); FN S(EV(J))
14240  NEXT J
14250  POKE 32,0: POKE 33,40: VTAB 23: GOSUB 10830
14260  RETURN
14270
14280  REM   ** SENSITIVITY ANALYSIS **
14290  HOME : VTAB 3
14300  PRINT "SENSITIVITY ANALYSIS"
14310  PRINT "----------- --------"
14320  POKE 34,4
14330  IF CS > 0 THEN 14360
14340  PRINT : PRINT : INPUT "DO YOU NEED INSTRUCTIONS?  ";A$
14350  IF  LEFT$ ( A$,1 ) = "Y" THEN  GOSUB 18400
14360  HOME : VTAB 7
14370  PRINT "YOUR CHOICES ARE:"
14380  PRINT : PRINT
14390  PRINT "   1.  VARY CRITERION WEIGHTS"
14400  PRINT
14410  PRINT "   2.  VARY STATE PROBABILITIES"
14420  PRINT : PRINT
14430  PRINT "   3.  RETURN TO LAST MENU"
14440  PRINT : PRINT : PRINT
14450  INPUT "         WHICH?  ";A$:CS =  VAL (A$): IF A$ = "" OR CS = 3 T
 HEN 10430
14460  PRINT : PRINT : INPUT "WHAT RANGE DEVIATION (1-100%)?  ";A$:RD =
   VAL (A$): IF A$ = "" THEN RD = 20
14470  IF RD < 1 OR RD > 100 THEN  CALL  - 998: GOTO 14460
14480  PRINT : PRINT "WHICH DECISION OPTION (1-";NOP;")?     ";: INPUT "
 ";A$:OS =  VAL (A$)
14490  IF OS = 0 THEN OS = NO
14500  IF OS < 1 OR OS > NOP THEN  CALL  - 998: GOTO 14480
14510 RD = RD / 100:INCR = RD / 5
14520  TEXT
14530  ON CS GOTO 14610,15100
14540
14550  REM  - NORMALIZE AS EACH PARAMETER IS CHANGED
14560 Q = 0
14570  FOR T = 1 TO N:Q = Q + RP(T): NEXT T
14580  FOR T = 1 TO N:RP(T) = RP(T) / Q: NEXT T
14590  RETURN
14600
14610  REM - VARY CRITERION WEIGHTS
14620  FOR T = 1 TO NC:RP(T) = NW(T): NEXT T
14630 AC = 0: IF NC / 4 >  INT (NC / 4) THEN AC = 1
14640  IF P$ = "Y" THEN  GOSUB 11020
14650 K = 1
14660 FAC = 1 - RD
```

```
14670  HOME
14680  PRINT "SENSITIVITY DUE TO CRITERION WEIGHTS"
14690  PRINT "----------- --- -- --------- -------"
14700  PRINT : PRINT : HTAB 8
14710  FOR L = K * 4 - 3 TO K * 4
14720    PRINT "("; FN R(NW(L));")"; TAB( 8 * (L - (K - 1) * 4) + 8)
14730    IF L = NC THEN 14750
14740  NEXT L
14750  PRINT
14760  PRINT "DEV"
14770  VTAB 7: PRINT "FAC"; TAB( 8);
14780  FOR L = K * 4 - 3 TO K * 4
14790    PRINT  LEFT$ (CR$(L),6); TAB( 8 * (L - (K - 1) * 4) + 8)
14800    IF L = NC THEN 14820
14810  NEXT L
14820  PRINT
14830  PRINT "---"
14840  FOR L = 1 TO NC
14850   CP(L) = 0
14860    FOR J = 1 TO NS
14870     CP(L) = CP(L) + V(OS,J,L) * NP(J)
14880    NEXT J
14890  NEXT L
14900    PRINT FAC; TAB( 8);
14910  FOR L = K * 4 - 3 TO K * 4
14920    FOR T = 1 TO NC:RP(T) = NW(T): NEXT T
14930 RP(L) = NW(L) * FAC: IF RP(L) > 1 THEN RP(L) = 1
14940 N = NC: GOSUB 14550
14950 SA = 0
14960    FOR J = 1 TO NC
14970     SA = SA + CP(J) * RP(J)
14980    NEXT J
14990      PRINT  FN S(SA); TAB( 8 * (L - (K - 1) * 4) + 8)
15000   IF L = NC THEN 15020
15010    NEXT L
15020  PRINT
15030   IF FAC + INCR = 1 OR FAC = 1 THEN  PRINT
15040 FAC = FAC + INCR: IF FAC <  = 1 + RD THEN 14900
15050 K = K + 1: IF K <  =  INT (NC / 4) + AC THEN  GOSUB 10770: GOTO 14
 660
15060  PRINT D$;"PR#0"
15070  GOSUB 10770
15080  GOTO 14290
15090
15100  REM - VARY STATE PROBABILITIES
15110  FOR T = 1 TO NS:RP(T) = NP(T): NEXT T
15120 AS = 0: IF NS / 4 >  INT (NS / 4) THEN AS = 1
15130  IF P$ = "Y" THEN  GOSUB 11020
15140 K = 1
15150 FAC = 1 - RD
15160  HOME
15170  PRINT "SENSITIVITY DUE TO STATE PROBABILITIES"
```

```
15180  PRINT "----------- --- -- ----- ------------"
15190  PRINT : PRINT : HTAB 8
15200  FOR L = K * 4 - 3 TO K * 4
15210    PRINT "("; FN R(NP(L));")"; TAB( 8 * (L - (K - 1) * 4) + 8)
15220    IF L = NS THEN 15240
15230  NEXT L
15240  PRINT
15250  PRINT "DEV"
15260  VTAB 7: PRINT "FAC"; TAB( 8);
15270  FOR L = K * 4 - 3 TO K * 4
15280    PRINT  LEFT$ (SN$(L),6); TAB( 8 * (L - (K - 1) * 4) + 8)
15290    IF L = NS THEN 15310
15300  NEXT L
15310  PRINT
15320  PRINT "---"
15330  PRINT FAC; TAB( 8);
15340  FOR L = K * 4 - 3 TO K * 4
15350    FOR T = 1 TO NS:RP(T) = NP(T): NEXT T
15360   RP(L) = NP(L) * FAC: IF RP(L) > 1 THEN RP(L) = 1
15370   N = NS: GOSUB 14550
15380   SA = 0
15390    FOR J = 1 TO NS
15400     SA = SA + CV(OS,J) * RP(J)
15410    NEXT J
15420    PRINT  FN S(SA); TAB( 8 * (L - (K - 1) * 4) + 8)
15430    IF L = NS THEN 15450
15440  NEXT L
15450  PRINT
15460  IF FAC + INCR = 1 OR FAC = 1 THEN  PRINT
15470 FAC = FAC + INCR: IF FAC < = 1 + RD THEN 15330
15480 K = K + 1: IF K < =  INT (NS / 4) + AS THEN  GOSUB 10770: GOTO 15
 150
15490  PRINT D$;"PR#0"
15500  GOSUB 10770
15510  GOTO 14290
15520
15530  REM  ** READ DATA FROM DISK FILE **
15540  HOME : VTAB 5
15550  INPUT "DO YOU WANT TO SEE THE DISK CATALOG? ";A$
15560  IF  LEFT$ (A$,1) = "Y" THEN  HOME : PRINT D$;"CATALOG": VTAB 23:
 GOSUB 10830: GOTO 15610
15570  HOME : VTAB 3
15580  PRINT "READ DATA BASE FROM DISK"
15590  PRINT "---- ---- ---- ---- ----"
15600  PRINT : PRINT : PRINT
15610  INPUT "WHAT DATA FILE NAME?  ";RFILE$: IF RFILE$ = "" THEN  RETUR
 N
15620  PRINT D$;"OPEN ";RFILE$
15630  PRINT D$;"READ ";RFILE$
15640  INPUT NOP,NS,NC
15650  FOR K = 1 TO NC
15660    INPUT CR$(K),W(K)
```

```
15670    FOR J = 1 TO NS
15680      INPUT SN$(J),P(J)
15690      FOR I = 1 TO NOP
15700        INPUT OP$(I),V(I,J,K)
15710      NEXT I
15720    NEXT J
15730  NEXT K
15740  PRINT D$;"CLOSE ";RFILE$
15750  GOSUB 16500
15760  RETURN
15770
15780  REM  ** SAVE DATA TO DISK FILE **
15790  HOME : VTAB 3
15800  PRINT "SAVE DATA BASE TO DISK"
15810  PRINT "---- ---- ---- -- ----"
15820  PRINT : PRINT : PRINT
15830  INPUT "SAVE UNDER WHAT FILE NAME?  ";SFILE$: IF SFILE$ = "" THEN
 RETURN
15840  PRINT D$;"OPEN ";SFILE$;D$;"DELETE ";SFILE$;D$;"OPEN ";SFILE$
15850  PRINT D$;"WRITE ";SFILE$
15860  PRINT NOP: PRINT NS: PRINT NC
15870  FOR K = 1 TO NC
15880    PRINT CR$(K): PRINT W(K)
15890    FOR J = 1 TO NS
15900      PRINT SN$(J): PRINT P(J)
15910      FOR I = 1 TO NOP
15920        PRINT OP$(I): PRINT V(I,J,K)
15930      NEXT I
15940    NEXT J
15950  NEXT K
15960  PRINT D$;"CLOSE ";SFILE$
15970  RETURN
15980
15990  REM   ** DECISION MODEL EDITING **
16000  TEXT : HOME : VTAB 2
16010  PRINT "     DECISION MODEL EDIT FUNCTIONS:"
16020  PRINT "     -------- ----- ---- ----------"
16030  PRINT : PRINT
16040  POKE 34,5
16050  IF DC > 0 THEN 16070
16060  INPUT "        DO YOU NEED INSTRUCTIONS?  ";A$: IF  LEFT$ (A$,1) =
 "Y" THEN  GOSUB 18500
16070  HOME : PRINT "1.  DELETE ANY PARAMETER"
16080  PRINT
16090  PRINT "2.  ADD ANY PARAMETER"
16100  PRINT
16110  PRINT "3.  CHANGE AN OPTION"
16120  PRINT
16130  PRINT "4.  CHANGE CRITERION AND/OR WEIGHTING"
16140  PRINT
16150  PRINT "5.  CHANGE STATE AND/OR PROBABILITY"
16160  PRINT
```

```
16170  PRINT "6.  CHANGE VALUE MATRIX"
16180  PRINT : PRINT
16190  PRINT "7.  RETURN TO LAST MENU"
16200  PRINT : PRINT : PRINT
16210  INPUT "    WHICH?  ";A$:DC =  VAL (A$)
16220  IF A$ = "" OR DC = 7 THEN DC = 0: GOSUB 16500: RETURN
16230  TEXT : ON DC GOTO 16720,16920,17090,17190,17250,17310
16240  IF DC < 0 OR DC > 7 THEN  CALL  - 998: GOTO 16210
16250
16260  REM - PARAMETER TYPES
16270  TEXT : HOME : POKE 33,32: POKE 32,8
16280  VTAB 5
16290  PRINT "DECISION PARAMETERS"
16300  PRINT "-------- ----------"
16310  PRINT : PRINT
16320  PRINT : PRINT "1.  DECISION OPTION"
16330  PRINT : PRINT "2.  JUDGEMENT CRITERION"
16340  PRINT : PRINT "3.  STATE OF NATURE"
16350  PRINT : PRINT : PRINT
16360  TEXT : VTAB 19
16370  RETURN
16380
16390  REM  - COMPRESS DATA BASE
16400 Q = 0
16410  FOR J = 1 TO N
16420    IF J = PD THEN 16470
16430   Q = Q + 1
16440    IF TP = 1 THEN OP$(Q) = OP$(J)
16450    IF TP = 2 THEN CR$(Q) = CR$(J)
16460    IF TP = 3 THEN SN$(Q) = SN$(J)
16470  NEXT J
16480  RETURN
16490
16500  REM   - NORMALIZE CURRENT WEIGHTS AND PROBABILITIES
16510 Q = 0
16520  FOR J = 1 TO NC:Q = Q + W(J): NEXT J
16530  FOR J = 1 TO NC:NW(J) = W(J) / Q: NEXT J
16540 R = 0
16550  FOR K = 1 TO NS:R = R + P(K): NEXT K
16560  FOR K = 1 TO NS:NP(K) = P(K) / R: NEXT K
16570  RETURN
16580
16590  REM - LIST OPTIONS, CRITERIA, OR STATES
16600  VTAB 7: CALL  - 958
16610  FOR J = 1 TO N
16620    IF J > 15 THEN  POKE 33,20: POKE 32,20: VTAB J - 11
16630  PRINT J; TAB( 5);
16640  IF TP = 1 THEN  PRINT OP$(J): GOTO 16670
16650  IF TP = 2 THEN  PRINT CR$(J): GOTO 16670
16660  IF TP = 3 THEN  PRINT SN$(J)
16670  NEXT J
16680  TEXT : POKE 34,23
```

```
16690  VTAB 23: GOSUB 10830: VTAB 24
16700  RETURN
16710
16720  REM  - DELETE A PARAMETER
16730  GOSUB 16260
16740  INPUT "WHAT TYPE PARAMETER TO DELETE?  ";TP
16750  HOME : VTAB 2
16760  IF  TP = 1 THEN  N = NOP: PRINT OP$(0): GOTO 16800
16770  IF  TP = 2 THEN  N = NC:  PRINT CR$(0): GOTO 16800
16780  IF  TP = 3 THEN  N = NS:  PRINT SN$(0): GOTO 16800
16790  CALL  - 998: GOTO 16740
16800  GOSUB 16590
16810  POKE 34;23: PRINT "WHICH TO DELETE (1-";N;")?  ";: INPUT "";A$: I
 F A$ = "" THEN 16000
16820 PD =  VAL (A$)
16830  GOSUB 16390
16840  HOME : VTAB 10
16850  IF  TP = 1 THEN  NOP  = NOP - 1:N = NOP: GOTO 16880
16860  IF  TP = 2 THEN  NC = NC  - 1:N = NC: GOTO 16880
16870  IF  TP = 3 THEN  NS = NS  - 1:N = NS
16880  PRINT BELL$: GOSUB 16590
16890  VTAB 23: GOSUB 10830: INPUT "    DELETE ANOTHER?  ";A$: IF  LEFT$
  (A$,1) = "Y" THEN 16730
16900  GOTO 16000
16910
16920  REM - ADD A PARAMETER
16930  GOSUB 16260
16940  INPUT "WHAT TYPE PARAMETER TO ADD?  ";TP
16950  HOME : VTAB 2
16960  IF  TP = 1 THEN  N = NOP: PRINT OP$(0): GOTO 17000
16970  IF  TP = 2 THEN  N = NC:  PRINT CR$(0): GOTO 17000
16980  IF  TP = 3 THEN  N = NS:  PRINT SN$(0): GOTO 17000
16990  CALL  - 998: GOTO 16940
17000  GOSUB 16590
17010  POKE 34,23: INPUT "    NAME TO ADD?  ";A$: IF A$ = "" THEN 16000
17020  IF  TP = 1 THEN  NOP  = NOP + 1:N = NOP:OP$(NOP) =  LEFT$ (A$,10): G
 OTO  17050
17030  IF  TP = 2 THEN  NC = NC  + 1:N = NC:CR$(NC) =  LEFT$ (A$,8): GOTO 1
 7050
17040  IF  TP = 3 THEN  NS = NS  + 1:N = NS:SN$(NS) =  LEFT$ (A$,8)
17050  PRINT BELL$: GOSUB 16590: ON TP  GOTO 17090,17190,17250
17060  VTAB 23: GOSUB 10830: INPUT "    ADD ANOTHER?  ";A$: IF  LEFT$ (A
 $,1 ) = "Y" THEN 16930
17070  GOTO 16000
17080
17090  REM - CHANGE AN OPTION
17100  HOME : VTAB 2: PRINT OP$(0)
17110 N = NOP:TP = 1: GOSUB 16590
17120  POKE 34,23: VTAB 23: GOSUB 10830
17130 LIM = NOP:H = 0
17140  GOSUB 13370: IF CE = 0 THEN 17170
17150 OP$(CE) =  LEFT$ (A$,10)
```

```
17160  GOTO 17140
17170  GOTO 16000
17180
17190  REM  - CHANGE CRITERION AND/OR WEIGHTS
17200  HOME : VTAB 2: PRINT CR$(0)
17210 N = NC:TP = 2: GOSUB 16590
17220 RFLAG = 1: GOSUB 11630
17230  GOTO 16000
17240
17250  REM  - CHANGE STATE AND/OR PROBABILITIES
17260  HOME : VTAB 2: PRINT SN$(0)
17270 N = NS:TP = 3: GOSUB 16590
17280 RFLAG = 1: GOSUB 12350
17290  GOTO 16000
17300
17310  REM - CHANGE VALUE MATRIX
17320 RFLAG = 1: GOSUB 12880
17330  GOTO 16000
17340
17350  REM   ** DISPLAY MODEL **
17360  PRINT D$;"PR#0"
17370  TEXT : HOME : VTAB 3
17380  PRINT "    DISPLAY MODEL OPTIONS:"
17390  PRINT "    ------- ----- --------"
17400  PRINT : PRINT
17410  PRINT "1.  DECISION OPTIONS"
17420  PRINT
17430  PRINT "2.  JUDGEMENT CRITERIA AND WEIGHTS"
17440  PRINT
17450  PRINT "3.  STATES OF NATURE AND PROBABILITIES"
17460  PRINT
17470  PRINT "4.  VALUE MATRICES"
17480  PRINT : PRINT
17490  PRINT "5.  RETURN TO LAST MENU"
17500  PRINT : PRINT : PRINT : INPUT "    WHICH OPTION?  ";A$:MD =  VAL
 (A$)
17510  IF A$ = "" OR MD = 5 THEN MD = 0: RETURN
17520  IF MD < 1 OR MD > 5 THEN  CALL  - 998: GOTO 17500
17530  CALL  - 998: INPUT "    DO YOU WANT OUTPUT TO PRINTER? ";P$: IF
 LEFT$ (P$,1) = "Y" THEN  GOSUB 5560
17540  HOME
17550  ON MD GOTO 17570,17650,17720,17790
17560
17570  REM - DISPLAY OPTIONS
17580 N = NOP:TP = 1
17590  VTAB 3: HTAB 6
17600  PRINT OP$(0)
17610  GOSUB 16590
17620  GOSUB 10770
17630  GOTO 17350
17640
17650  REM - DISPLAY CRITERIA
```

```
17660  VTAB 3: HTAB 7
17670  PRINT CR$(0)
17680  GOSUB 11990
17690  GOSUB 10770
17700  GOTO 17350
17710
17720  REM - DISPLAY STATES
17730  VTAB 3: HTAB 7
17740  PRINT SN$(0)
17750  GOSUB 12710
17760  GOSUB 10770
17770  GOTO 17350
17780
17790  REM  - DISPLAY VALUE MATRICES
17800 RFLAG = 1: GOSUB 12880
17810  GOTO 17350
17820
17830  REM   ** INTRODUCTORY REMARKS **"
17840  HOME : VTAB 2
17850  PRINT  TAB( 8);"MATRIX DECISION ANALYSIS"
17860  PRINT  TAB( 8);"------ -------- --------"
17870  PRINT
17880  PRINT "DECISION ANALYSIS PROVIDES A STRUCTURED APPROACH TO COMPAR
 ING THE RELATIVE MERITOF VARIOUS ALTERNATIVES OR OPTIONS."
17890  PRINT
17900  PRINT "REGARDLESS WHETHER AN ACTUAL DECISION ISMADE, THE ANALYSIS
  CAN ILLUMINATE PARTS OF THE PROBLEM PREVIOUSLY UNCONSIDERED  AND CAN P
 ROVIDE THE BASIS FOR ELABORATE SUBSEQUENT STUDIES."
17910  PRINT
17920  PRINT "AMONG THE VARIOUS FORMS OF QUANTITATIVE DECISION ANALYSIS
 ARE:"
17930  PRINT
17940  PRINT "   DECISION TREES"
17950  PRINT "   COST/BENEFIT ANALYSIS"
17960  PRINT "   UTILITY THEORY"
17970  PRINT "   PROCESS DIAGRAMS"
17980  GOSUB 10890
17990  HOME : VTAB 2
18000  PRINT "THE MATRIX DECISION ALGORITHM USES THE  ADDITIVE WEIGHTING
  OF PARAMETERS AND IS USEFUL FOR QUICK INITIAL ANALYSIS OF    HIGHLY SU
 BJECTIVE, MULTI-ATTRIBUTE PROB-LEMS."
18010  PRINT
18020  PRINT "THIS PROGRAM IS BASED ON THE TECHNIQUE  DESCRIBED IN DECIS
 IONS AND DESIGNS, INC.TECHNICAL REPORT 76-12, 'RAPID SCREENINGOF DECISI
 ON OPTIONS' BY JUDITH SELVIDGE."
18030  PRINT : PRINT
18040  PRINT "THE TECHNIQUE ALLOWS:"
18050  PRINT
18060  PRINT "  ONE DECISION (MULTIPLE OPTIONS)"
18070  PRINT "  ONE UNCERTAIN EVENT (MULTIPLE STATES)"
18080  PRINT "  MULTIPLE JUDGEMENT CRITERIA"
18090  GOSUB 10890
18100  RETURN
```

```
18110
18120  REM  ** DECISION MATRIX DEFINITIONS **
18130  HOME
18140  PRINT "OPTIONS:"
18150  PRINT "-------"
18160  PRINT "   THE OPTIONS OR ALTERNATIVES ARE THE  DIFFERENT POSSIBLE
  COURSES OF ACTION THEDECISION-MAKER IS CONSIDERING. THEY MUSTBE INDEPE
 NDENT AND NON-REDUNDANT."
18170  PRINT
18180  PRINT "CRITERIA:"
18190  PRINT "--------"
18200  PRINT "   JUDGEMENT CRITERIA (OR ATTRIBUTES OR DECISION CRITERIA)
  ARE USED TO COMPARE  THE RELATIVE VALUE OF DIFFERENT OPTIONS.THEY MUST
  INCLUDE ALL RELEVANT CONCERNS AND SHOULD BE NON-REDUNDANT."
18210  PRINT
18220  PRINT "STATES:"
18230  PRINT "------"
18240  PRINT "   STATES OF NATURE REFER TO THE SET OF UNCERTAIN OUTCOMES
  THAT CAN RESULT FROM AN EVENT OVER WHICH WE HAVE LITTLE OR   NO CONTRO
 L."
18250  GOSUB 10770
18260  RETURN
18270
18280  REM  ** VALUE MATRIX INSTRUCTIONS **
18290  HOME
18300  PRINT : PRINT
18310  PRINT "   TO USE THE VALUE MATRIX, YOU MUST    QUANTIFY YOUR SUBJ
 ECTIVE ESTIMATE OF THEVALUE OF EACH OPTION,  RELATIVE TO EACH JUDGEMENT
  CRITERION";: IF NS > 0 THEN  PRINT " AND STATE OF NATURE.": GOTO 18330

18320  PRINT ".": PRINT
18330  PRINT
18340  PRINT "   VALUES MUST BE CORRELATED AMONG THE  OPTIONS, CRITERIA,

  AND STATES OF NATURE FOR RESULTS TO BE MEANINGFUL,  SO TAKE  ADVANTAGE
  OF THE OPTIONS FOR CHANGING   INPUT DATA."
18350  PRINT : PRINT
18360  PRINT "   VALUES MAY BE POSITIVE OR NEGATIVE,  THE RECOMMENDED SC
 ALES BEING  -100 TO 0 OR  0 TO 100.  LATER YOU WILL HAVE THE  OPTION TO
  MAXIMIZE OR MINIMIZE."
18370  GOSUB 10770
18380  RETURN
18390
18400  REM  ** SENSITIVITY ANALYSIS INSTRUCTIONS **
18410  HOME : VTAB 6
18420  PRINT "SENSITIVITY ANALYSIS SHOWS THE VARIATIONIN THE EXPECTED VA
 LUE OF ANY OPTION WHENEITHER STATE PROBABILITIES OR CRITERION WEIGHTS A
 RE VARIED WITHIN A SPECIFIED   RANGE."
18430  PRINT
18440  PRINT "IF YOU SPECIFY A DEVIATION RANGE OF 20%,THE DEVIATION FACT
 OR WOULD RUN FROM .80 TO 1.20 (A MULTIPLE OF ORIGINAL WEIGHTS OR PROBAB
 ILITIES)."
18450  PRINT
```

```
18460  PRINT "THE ACTUAL RANGE IN WEIGHTING OR PROBA- BILITY, HOWEVER, W
 ILL DIFFER FOR EACH   PARAMETER BECAUSE THE ORIGINAL VALUES   ARE DIFFE
 RENT AND WILL BE RENORMALIZED."
18470  GOSUB 10770
18480  RETURN
18490
18500  REM  ** EDIT FUNCTION INSTRUCTIONS **
18510  HOME
18520  PRINT "THE EDIT MODE ALLOWS YOU TO RESHAPE YOURDECISION MODEL TO
 MEET CHANGING NEEDS.  YOU CAN ADD OR DELETE PARAMETERS, AND   YOU CAN C
 HANGE CRITERION WEIGHTS, STATE PROBABILITIES, AND MATRIX VALUES."
18530  PRINT
18540  PRINT "NUMBERS WILL BE RENORMALIZED AFTER YOU  FINISH EDITING."
18550  PRINT : PRINT
18560  PRINT  TAB( 14);"** NOTE **"
18570  PRINT
18580  PRINT "DON'T FORGET TO ADD OR CHANGE WEIGHTINGSAND/OR PROBABILITI
 ES AND MATRIX VALUES  IF YOU ADD/DELETE PARAMETERS.  OTHERWISEYOUR RESU
 LTS MAY NOT BE MEANINGFUL !!!"
18590  GOSUB 10770
18600  RETURN
```

VITA

Robert D. Conte was born in Clemson, South Carolina on 26 August 1948 and graduated from high school in North Olmsted, Ohio in 1966. He attended the U.S. Military Academy and graduated in 1970 with a Bachelor of Science degree, concentrating in international relations and area studies. He was commissioned in the U.S. Army Field Artillery and following Airborne and Ranger training was assigned to the 2d Battalion, 30th Field Artillery in Vicenza, Italy, serving as platoon leader, assistant battalion operations officer, and battery executive officer.

Following the assignment to Italy, he transferred to Military Intelligence and in 1974 was assigned to the 3d Armored Cavalry Regiment at Fort Bliss, Texas, serving as interrogation section leader, intelligence analysis section leader, and squadron intelligence officer. In 1977 he attended the Defense Intelligence Agency Post-graduate Intelligence Course in Washington, D.C., and in June 1978 he entered the Air Force Institute of Technology.

He is married to the former Maureen E. Culkin of Ridgewood, New Jersey and has one son, David.

Permanent address: 9910 Edgewater Dr.
Cleveland, Ohio 44102

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GOR/MA/79d-3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER ASSISTED ANALYSIS FOR MILITARY MANAGERS		5. TYPE OF REPORT & PERIOD COVERED MS THESIS
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) ROBERT D. CONTE CPT, U.S. ARMY		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Department of Mathematics AFIT-EN		12. REPORT DATE December 1979
		13. NUMBER OF PAGES 218
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)

18. SUPPLEMENTARY NOTES

Approved for public release IAW AFR 190-17.

JOSEPH P. HIPPS, MAJ, USAF
Director of Public Affairs

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Programs
Computer Support
BASIC
Desk-Top Computers
Microcomputers
Quantitative Analysis
Decision Aids
Regression Analysis
Linear Programming

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

Modern decision making frequently demands timely and accurate, quantitatively-based analysis. Computer support is necessary, but mainframe systems are not sufficiently responsive, flexible, or accessible. Desk-top "micro-" computers have the potential to provide effective, time-sensitive analysis support because of their low cost, portability, flexibility, and independent memory. But there is little analysis software that truly exploits the advantages of desk-top computers. The objective of this study was to demonstrate the power and utility of desk-top computers by producing a useful desk-top computer quantitative analysis

over

Block 20 Cont.

software package. Analysis techniques implemented were Bivariate and Multivariate Regression Analysis, Linear Programming, and a Value Matrix Decision Aid All the programs are fully interactive and user-oriented. They provide for "paper and pencil" style data entry or model formulation, built-in screen-oriented editing, user-defined models, disk storage/retrieval, and printer options. The entire package requires 32K bytes Random Access Memory per program and a total of 92K bytes storage.