



Energy, Mines and
Resources Canada

Énergie, Mines et
Ressources Canada

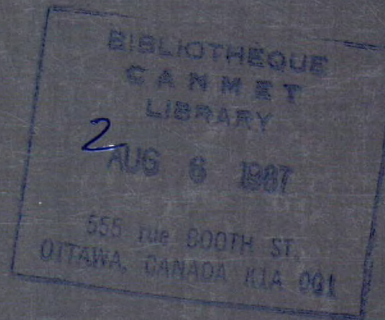
CANMET

Canada Centre
for Mineral
and Energy
Technology

Centre canadien
de la technologie
des minéraux
et de l'énergie

SPOC

Simulated Processing of Ore and Coal



Chapter 7.1 STAMP Computer Program

CANMET

Canada Centre
for Mineral
and Energy
Technology

Centre canadien
de la technologie
des minéraux
et de l'énergie

The **SPOC** Manual

Chapter 7.1 STAMP Computer Program

STAMP — Program for Statistical Analysis and Modelling

F. Seguin, J. Leung, F. Flament, and D. Laguitton

Editor: D. Laguitton

© Minister of Supply and Services Canada 1985

Available in Canada through

Authorized Bookstore Agents
and other bookstores

or by mail from

Canadian Government Publishing Centre
Supply and Services Canada
Ottawa, Canada K1A 0S9

Catalogue No. M38-16/7-1-1985E Canada: \$7.00
ISBN 0-660-11869-6 Other Countries: \$8.40

Price subject to change without notice
Disponible en français

THE SPOC MANUAL

The **SPOC*** manual consists of eighteen chapters, published separately. Their numbers and short titles are as follows:

- | | |
|------------------------------------|--------------------------------------|
| 1. Summary | 5. Unit Models: Part A |
| 2. Sampling Methodology | 5.1 Unit Models: Part B |
| 2.1 SAMBA Computer Program | 5.2 Unit Models: Part C |
| 2.2 Grinding Circuit Sampling | 6. Flowsheet Simulators |
| 3. Material Balance | 7. Model Calibration |
| 3.1 BILMAT Computer Program | 7.1 STAMP Computer Program |
| 3.2 MATBAL Computer Program | 7.2 FINDBS Computer Program |
| 4. Modelling and Simulation | 7.3 RTD and MIXERS Computer Programs |
| 4.1 Industrial Ball Mill Modelling | 8. Miscellaneous Computer Programs |

These chapters are available from: CANMET, Energy, Mines and Resources Canada
Technology Information Division
555 Booth Street
Ottawa, Ontario

*Simulated Processing of Ore and Coal

FOREWORD

High energy costs and depleting ore reserves combine to make process evaluation and optimization a challenging goal in the 80's. The spectacular growth of computer technology in the same period has resulted in widely available computing power that can be distributed to the most remote mineral processing operations. The SPOC project, initiated at CANMET in 1980, has undertaken to provide Canadian industry with a coherent methodology for process evaluation and optimization assisted by computers. The SPOC Manual constitutes the written base of this methodology and covers most aspects of steady-state process evaluation and simulation. It is expected to facilitate industrial initiatives in data collection and model upgrading.

Creating a manual covering multidisciplinary topics and involving contributions from groups in universities, industry and government is a complex endeavour. The reader will undoubtedly notice some heterogeneities resulting from the necessary compromise between ideals and realistic objectives or, more simply, from oversight. Critiques to improve future editions are welcomed.

D. Laguitton
SPOC Project Leader
Canada Centre for Mineral and Energy Technology

AVANT-PROPOS

La croissance des coûts de l'énergie et l'appauvrissement des gisements ont fait de l'évaluation et de l'optimisation des procédés un défi des années 80 au moment même où s'effectuait la dissémination de l'informatique jusqu'aux concentrateurs les plus isolés. Le projet SPOC, a été lancé en 1980 au CANMET, en vue de développer pour l'industrie canadienne, une méthodologie d'application de l'informatique à l'évaluation et à l'optimisation des procédés minéralurgiques. Le Manuel SPOC constitue la documentation écrite de cette méthodologie et en couvre les différents éléments. Les retombées devraient en être une vague nouvelle d'échantillonnages et d'amélioration de modèles.

La rédaction d'un ouvrage couvrant différentes disciplines et rassemblant des contributions de groupes aussi divers que les universités, l'industrie et le gouvernement est une tâche complexe. Le lecteur notera sans aucun doute des ambiguïtés ou contradictions qui ont pu résulter de la diversité des sources, de la traduction ou tout simplement d'erreurs. La critique constructive est encouragée afin de parvenir au format et au contenu de la meilleure qualité possible.

D. Laguitton
Chef du projet SPOC,
Centre canadien de la technologie des minéraux et de l'énergie

ABSTRACT

The STAMP program is a conversational package dedicated to experimental data analysis. STAMP features data entry and data file maintenance, various regression methods (including simple linear regression, multiple linear regression, stepwise linear regression, polynomial regression) and three non-linear optimization algorithms (Powell, Simplex and Rosenbrock). Options selection is menu driven.

This manual is a user's guide to the program and includes a detailed sample run.

The version of the program described here (STAMP84) was written in FORTRAN for a Cyber 730 computer. It easily runs in a 70 K octal memory (56 K decimal) but uses file structure and FORTRAN routines specific to Cyber. A version is also available for an IBM environment.

RÉSUMÉ

Le programme STAMP est un logiciel conversationnel pour le traitement de données expérimentales. Il inclut un système d'édition et de maintenance d'un fichier de données, plusieurs types de régressions (régressions linéaires simples, régression linéaire multiple, régression linéaire par pas, régression polynomiale) et trois programmes d'optimisation non linéaire (méthodes de Powell, du Simplex et de Rosenbrock). Un système hiérarchisé de menus permet un accès aisé à chacune des options.

Ce manuel est un guide de l'utilisateur. À ce titre un exemple détaillé d'utilisation du programme est donné en appendice.

Le langage de programmation utilisé est le FORTRAN. La version décrite dans le présent manuel (STAMP84) a été mise au point sur un Cyber 730 et fonctionne aisément dans une mémoire interactive de 70 K octal (56 K décimal). Elle fait appel à divers outils de programmation spécifiques à Cyber. Toutefois une autre version est aussi disponible pour un environnement IBM.

ACKNOWLEDGEMENTS

The SPOC project has benefited from such a wide range of contributions throughout the industry, the university, and the government sectors that a nominal acknowledgement would be bound to make unfair omissions. The main groups that contributed are: the various contractors who completed project elements; the Industrial Steering Committee members who met seven times to provide advice to the project leader; the various users of project documents and software who provided feedback on their experience; the CANMET Mineral Sciences Laboratories staff members who handled the considerable in-house task of software development, maintenance, and documentation; the EMR Computer Science Centre staff who were instrumental in some software development; and the CANMET Publications Section. Inasmuch as in a snow storm, every flake is responsible, their contributions are acknowledged.

CONTENTS

THE SPOC MANUAL	i
FOREWORD	iii
AVANT-PROPOS	iii
ABSTRACT	v
RÉSUMÉ	v
ACKNOWLEDGEMENTS	v
 1. PROGRAM IDENTIFICATION	 1
2. ENGINEERING DOCUMENTATION	2
2.1 Narrative Description	2
2.1.1 Introduction	2
2.1.2 History	2
2.1.3 The STAMP Program	2
2.1.3.1 Editing of data	2
2.1.3.2 Plotting of variables	2
2.1.3.3 Descriptive statistics	2
2.1.3.4 Correlation	3
2.1.3.5 Regressions	3
2.1.3.6 Non-linear optimization	3
2.2 Program Capabilities	3
2.3 Data Input	3
2.3.1 Entering Variables and Observations	3
2.3.1.1 On-line entry	3
2.3.1.2 Batch entry	4
2.3.1.3 Data file maintenance	4
2.3.2 Responding to Queries	4
2.4 Data Output	4
2.5 Program Structure	4
2.5.1 Main Line	4
2.5.2 Regression Module	4
2.5.2.1 INTMLR: Interactive multiple linear regression	5
2.5.2.2 POLYRG	5
2.5.2.3 SPOCLR: Simple weighted regression	6
2.5.2.4 SR: Stepwise regression	6
2.5.3 EDIT	6
2.5.3.1 FRMEQU	7
2.5.3.2 MODIFY	8
2.5.4 PLOTS	8
2.5.4.1 EXAVAR	8
2.5.5 Non-linear optimization module	8
2.5.5.1 EDFUNC: Editing of a non-linear function	8
2.5.5.2 POWELL: BOTM algorithm option	9
2.5.5.3 SIMPLX: Simplex algorithm option	9
2.5.5.4 ROSENB: ROSENB algorithm option	9
2.6 ADSTAMP: Batch Entry	9
2.7 CLEANF: Cleaning Data Files	9
2.8 Installation-Dependent Routines	10
3. SYSTEM DOCUMENTATION	11
3.1 Computer Equipment	11
3.2 Source Program	11
3.3 Storage Requirements	11
3.4 Maintenance and Update	11
3.4.1 Introduction	11
3.4.2 Changes to the System Routines	11
3.4.2.1 Analysis of module(s)	11
3.4.2.2 Source library	11
3.4.2.3 Relocatable library	14
3.4.2.4 Redefining the tree structure	14

3.4.3 STAMP Files	14
3.4.3.1 INDEX file	14
3.4.3.2 DATA file	14
3.5 Control Cards	14
4. OPERATING DOCUMENTATION	15
4.1 Operating Instructions	15
4.1.1 EDIT	15
4.1.1.1 MODIFY A VARIABLE (EDIT)	15
4.1.1.2 CREATE A VARIABLE (EDIT)	18
4.1.1.3 DELETE A VARIABLE (EDIT)	20
4.1.1.4 EXAMINE VARIABLES (EDIT)	20
4.1.2 PLOTS	20
4.1.2.1 X(I) vs I (PLOTS)	20
4.1.2.2 Y(I) vs X(I) (PLOTS)	20
4.1.2.3 EXAMINE VARIABLES (PLOTS)	20
4.1.3 STATS	20
4.1.4 CORRELATION	21
4.1.5 REGRESSIONS	21
4.1.5.1 MULTIPLE LINEAR REGRESSION	21
4.1.5.2 SIMPLE WEIGHTED REGRESSION	21
4.1.5.3 STEPWISE REGRESSION	21
4.1.5.4 POLYNOMIAL REGRESSION	21
4.1.6 NON-LINEAR OPTIMIZATION	21
4.1.6.1 EDITING FUNCTION: Non-linear optimization	22
4.1.6.2 POWELL METHOD: Non-linear optimization	23
4.1.6.3 SIMPLEX METHOD: Non-linear optimization	23
4.1.6.4 ROSENBROCK METHOD: Non-linear optimization	24
4.2 Error Messages	24
4.2.1 STAMP Error Messages	24
4.2.2 Operating System Error Messages	24
4.2.3 Fatal Errors	24
4.3 Run Time	24
5. REFERENCES	25
APPENDIX A – SAMPLE RUN OF STAMP	27
APPENDIX B – EDITING FUNCTIONS FOR NON-LINEAR OPTIMIZATION	45
APPENDIX C – SAMPLE OF A CONTROL CARD DECK	53
TABLES	
<u>No.</u>	
1. Correspondence between routines associated with FRMEQU and EDFUNC	8
2. System-dependent subroutines and their occurrence	10
FIGURES	
1. STAMP tree	12
2. Summary of STAMP options	16

1. PROGRAM IDENTIFICATION

<u>Program Title:</u>	STATistical Modelling Package (STAMP).	<u>Source Language:</u>	CDC FORTRAN extended 4.8. (Most routines dealing with statistical methods comply with American National Standards Institute FORTRAN language as described in X3.9 – 1966.) Routines related to file activity are installation dependent. These routines are identified in Section 2.8.
<u>Program Code Name:</u>	STAMP.		
<u>Authors:</u>	F. Séguin, J. Leung, F. Flament, and D. Laguitton.	<u>Availability:</u>	This is the STAMP84 version. All enquiries should be directed to: CANMET, Energy, Mines and Resources, Technology Information Division, 555 Booth Street, Ottawa, Ontario, K1A 0G1.
<u>Organization:</u>	Energy, Mines and Resources Canada Centre for Mineral and Energy Technology, Computer Science Centre.		
<u>Date:</u>	June 1983.		
<u>Updates:</u>	Updated June 1984. This program may be updated as new statistical methods become available.		

SSSSSS	TTTTTTT	AAAAA	MMMMMMMMMM	PPPPPP
SSSSSSS	TTTTTTT	AAAAAAA	MMMMMMMMMM	PPPPPPP
SSSSSSS	TTTTTTT	AAAAAAA	MMMMMMMMMM	PPPPPPP
SSS	TTT	AAA AAA	MMM MMM MMM	PPP PPP
SSSSSSS	TTT	AAA AAA	MMM MMM MMM	PPPPPPP
SSSSSSS	TTT	AAA AAA	MMM MMM MMM	PPPPPPP
SSSSSSS	TTT	AAAAAAA	MMM MMM MMM	PPPPPP
SSS	TTT	AAAAAAA	MMM MMM MMM	PPP
SSSSSSS	TTT	AAAAAAA	MMM MMM MMM	PPP
SSSSSSS	TTT	AAA AAA	MMM MMM MMM	PPP
SSSSSSS	TTT	AAA AAA	MMM MMM MMM	PPP

2. ENGINEERING DOCUMENTATION

2.1 NARRATIVE DESCRIPTION

2.1.1 Introduction

The concepts of modelling and simulation by computers have been used for about two decades. During this time-span, several statistical packages have been developed, most of which cover the spectrum of statistical methods.

In most of these packages, the user is required to have a good knowledge of the data in terms of where it is stored, access methods and the characteristics of the operating system.

The STAMP program attempts to remove from its users most of the data management activities associated with statistical methods by providing a useful array of linear and non-linear modelling techniques and a comprehensive data management facility.

2.1.2 History

The STAMP program was inspired by the SACADOS system developed at the Canadian Forestry Service, Department of the Environment (1). This system consists of several statistical procedures and offers its users a comprehensive data management facility. It is written in APL and makes good use of APL's powerful operators.

Another feature of APL used in the SACADOS system is the dynamic loading of required modules and data elements. This permits an efficient use of the limited workspace.

The STAMP program offers a similar capability by using module segmentation and direct access to the data. The following is a detailed presentation of the STAMP program.

2.1.3 The STAMP Program

The purpose of the STAMP program is to offer comprehensive linear and non-linear modelling capabilities. This is accomplished by providing the users with the following principal modules:

- editing of data
- plots/scattergrams
- descriptive statistics
- simple correlation
- regressions
- non-linear optimization.

In each module, the user is guided with comprehensive menus. All menus contain two common choices that

allow the user **to view** the options of the current module (by entering 99) or **to exit** the current module (by entering 0).

The following is a description of these modules.

2.1.3.1 Editing of data

The **EDIT** module allows the user to modify the contents of the data files. The user can modify an existing variable, add a new variable, delete an existing variable or examine the file content.

A — Modifying a variable

When this option is selected, the user can add an observation to an existing variable, delete an observation, reenter a complete set of observations, reenter a complete set via a function, or modify an observation.

B — Creating a variable

There are two ways to add a new variable in the data files. The first method is to enter observations one at a time and the second method is to enter them with a function of existing variables.

C — Deleting a variable

This option allows the user to permanently remove a variable from the data files.

D — Examining variables

This option allows the user to list the variables in the data files. It lists the variable name, description and the number of observations. It also permits the user to examine the values of the variables.

2.1.3.2 Plotting of variables

This module produces two types of graphics. The first type allows the user to plot the observations of a variable against its position in the array ($X(I)$ vs I) and the second type allows the user to plot variable pairs (scattergram). The module also provides the option of examining the data file content.

2.1.3.3 Descriptive statistics

This module allows the user to obtain descriptive statistics on a given variable. Observations of the selected variable are also listed. The descriptive statistics are:

- minimum value
- maximum value
- average
- standard deviation.

2.1.3.4 Correlation

The correlation option allows the user to create a correlation matrix of up to ten variables. The lower triangle of the matrix is displayed.

2.1.3.5 Regressions

This module allows the user to build and verify the adequacy of a model by using regression techniques. Four regression techniques are offered in STAMP and make up the options of this module. The user may also examine the variables in the data files. The four techniques are:

- multiple linear regression
- simple weighted linear regression
- stepwise linear regression
- polynomial regression.

In each of these techniques, the user may save the residuals and estimates in the files as new variables.

Details regarding the use of these modules and related options are given in Sections 2.3 (Data Input) and 4.1 (Operating Instructions).

2.1.3.6 Non-linear optimization

The purpose of this module is to solve the problem of the minimization of a non-linear multivariable function. That function must be edited by the user before running the non-linear optimization routines. The editing work has been reduced to the minimum: write the formula using conventional FORTRAN symbols and implicit functions. It must fit in a line of 80 characters. Three kinds of variables may be used: indexed variables, real constants and search variables. When edited, the non-linear function is computed using the values of the variables and constants provided by the user. Before calling the non-linear optimization routine, it is possible to guess repeatedly the values of the search variables and examine the corresponding function value. Therefore, the optimization process may start from a point close to the solution whenever possible.

Three optimization packages are proposed:

- Powell method
- Simplex method
- Rosenbrock method.

Details regarding the NONLIN module are given in Sections 2.5.5 and 4.1.6.

2.2 PROGRAM CAPABILITIES

The STAMP program limits the user to a maximum of 42 observations. For correlation matrices and regressions, the user is allowed a maximum of ten variables. When editing the function in the non-linear optimization module, a maximum of five indexed variables, nine real constants and nine search variables may be used. The non-linear function may use FORTRAN implicit functions and must follow FORTRAN rules regarding levels of parentheses and priority of execution of operations. The function must fit in a an 80-character line, e.g.,

$$(K1*Z1 + Z2*V1 - V2)**2$$

where K1 is a real constant, Z_i are search variables and V_i are experimental variables from the user's own files. This particular expression defines a least-square criterion of the form $(\hat{y} - y)^2$ where \hat{y} is a linear function of variable V_i . The intercept coefficient Z1 (here multiplied by a user-supplied constant K1) and the slope Z2 are the search variables, Z_i .

2.3 DATA INPUT

This section describes two types of data entry in STAMP. The first type is associated with entering variables and observations in the user's data files and the second type, with responding to queries of STAMP.

2.3.1 Entering Variables and Observations

Variables and observations can be entered in STAMP on-line or in batch mode.

2.3.1.1 On-line entry

On-line entry is achieved in the EDIT module of STAMP (See 2.1.3.1 B). The system requests a variable name (up to four alphanumeric characters; extra characters are ignored). This code must be unique and must not be STOP.

Following this, the system requests a variable description (up to 16 alphanumeric characters).

The number of observations (numeric) is then requested (42 is the maximum allowed).

Then the observations are entered (numeric). They may be entered in the same line separated by blanks or commas, or on different lines separated by a carriage return.

2.3.1.2 Batch entry

The on-line method of entering data may be tedious and costly, particularly if there are several variables and observations.

To speed up data entry, the creation of STAMP compatible data files can be made outside the STAMP package. This operation is controlled by an independent program named ADSTAMP (see Section 2.6) and allows the user to create new data files or to extend old data files. ADSTAMP uses a conventional sequential file where new data is recorded according to the following format:

BLOCK A: The status (format A3) of STAMP data files: NEW (to create new data files) or OLD (to append existing data files).

BLOCK B: FORTRAN format to be used to read BLOCK E. The format should be written between brackets and use one record of 80 characters maximum.

BLOCK C: Each record in this block contains the name (format A4) and description (format 4A4) of a variable to be added. The number of records in BLOCK C should be equal to the number of columns in BLOCK E. A maximum of 20 new variables is allowed.

BLOCK D: STOP (mandatory, format A4).

BLOCK E: The table of observations. The number of columns must be equal to the number of records in BLOCK C. A maximum of 42 records will be read in the format specified in BLOCK B.

2.3.1.3 Data file maintenance

Every time a variable is modified by the user, the corresponding records in STAMP data files are blanked and new records are appended. It is therefore necessary to clean up the STAMP data files from time to time in order to remove unused records. This operation is performed by a program called CLEANF. This program (see Section 2.7) is independent of the STAMP package. It is the responsibility of the user to clean up the data files.

2.3.2 Responding to Queries

For all menus presented in STAMP, the user responds with a numeric entry. If an alphanumeric entry is made, the system requests a numeric entry. If the entry is outside the range of the menu, the menu is displayed.

For Yes/No responses, the user enters Y for yes and N for no.

When a listing is presented by STAMP, the user enters a "C" when the prompt "ENTER C to CONTINUE" appears.

When a user enters variable names for a correlation matrix or a regression, a check is made on the number of observations. The number of observations must be the same for all variables involved.

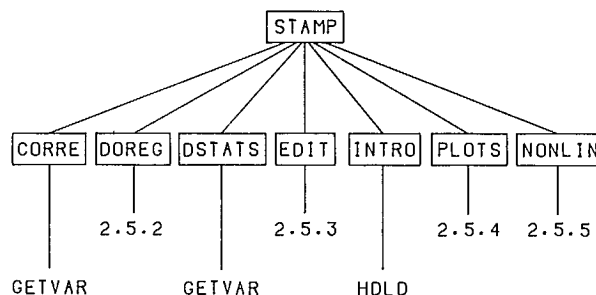
2.4 DATA OUTPUT

A sample run showing the various modules and options is given in Appendix A.

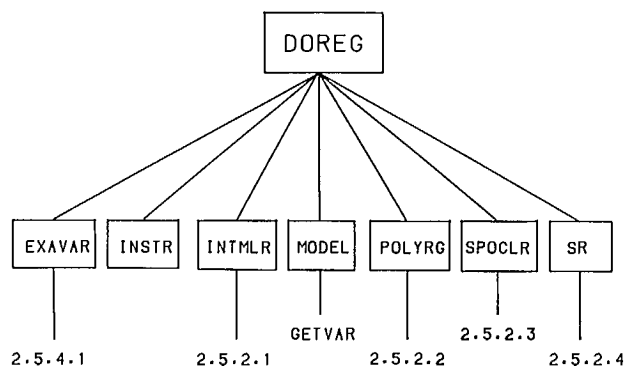
2.5 PROGRAM STRUCTURE

In this section, all the routines in STAMP are presented in hierarchical form along with a brief description of their purpose.

2.5.1 Main Line

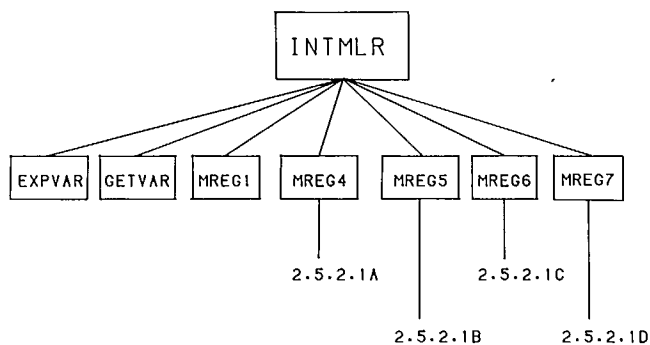


2.5.2 Regression Module

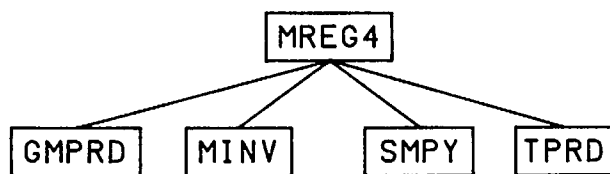


Subroutine DOREG is one of the principal modules. Its purpose is to present the options on regressions and access the programs selected by the user.

2.5.2.1 INTMLR: Interactive multiple linear regression



A — MREG4

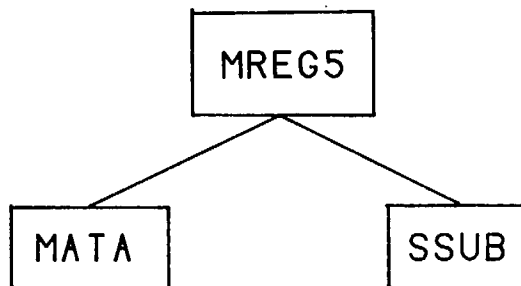


Subroutine MREG4 is called by the subroutine INTMLR. Its purpose is to calculate regression coefficients, covariance matrix, T-values, Y-estimates and residual mean square for multivariable linear regression.

The subroutines called by MREG4 are from the IBM Scientific Subroutine Package:

- MINV:** inverts a matrix
- TPRD:** transposes a matrix and post-multiplies it by another one
- GMPRD:** multiplies two matrices
- SMPY:** multiplies a matrix by a scalar.

B — MREG5

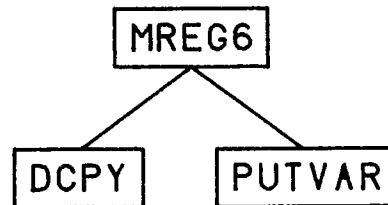


Subroutine MREG5 is called by subroutine INTMLR. Its purpose is to find the standard deviation for each variable, correlation of X versus Y, and the multiple correlation coefficient.

It calls two subroutines which are part of the IBM Scientific Subroutine Package:

- MATA:** premultiplies a matrix by its transpose
- SSUB:** subtracts a scalar from a matrix.

C — MREG6

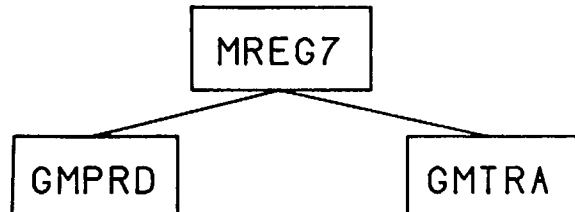


Subroutine MREG6 is called by the subroutine INTMLR. Its purpose is to display the regression calculation results.

Subroutine DCPY is from the IBM Scientific Subroutine Package:

- DCPY:** copies diagonal elements of a matrix into a vector
- PUTVAR:** puts variables in the data file.

D — MREG7

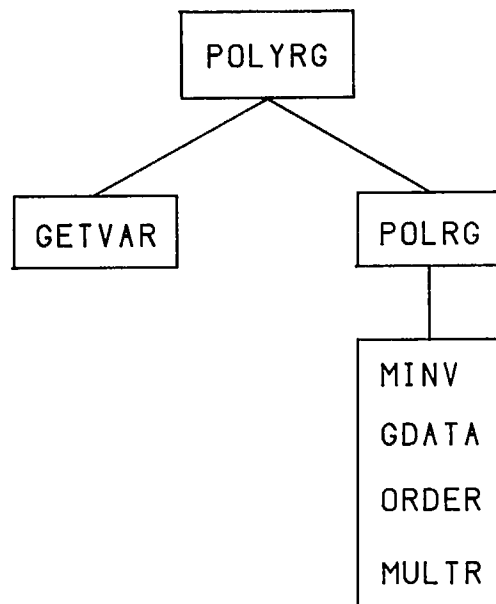


Subroutine MREG7 is called by the subroutine INTMLR. Its purpose is to find the 95% confidence intervals for the Y estimate and the parameters as well as the table of variance.

It calls routines from the IBM Scientific Subroutine Package. They are GMTRA and GMPRD:

- GMPRD:** multiplies two matrices
- GMTRA:** transposes a matrix.

2.5.2.2 POLYRG

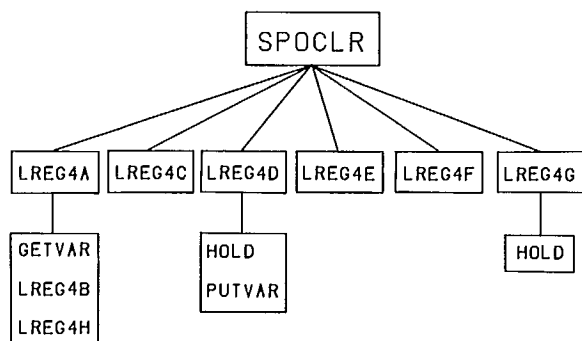


Subroutine POLYRG sets up a model based on the algorithm of B. Ostle (2). The user needs to identify the dependent variable Y, the independent variable X, and the power assigned to X. The resulting model is:

$$Y = B_0 + B_1X_1 + B_2X_2^2 + \dots + B_nX_n^n$$

The largest power that may be entered is ten. Powers of an independent variable are generated to calculate polynomials of successively increasing degrees. If there is no reduction in the residual sum of squares between two successive degrees of polynomials, the program terminates the problem before completing the analysis for the highest degree polynomial specified.

2.5.2.3 SPOCLR: Simple weighted regression



The subroutine SPOCLR performs the analysis of linear data. The user selects one of the following linear regressions:

- X on Y
- Y on X
- weighted X and Y
- orthogonal.

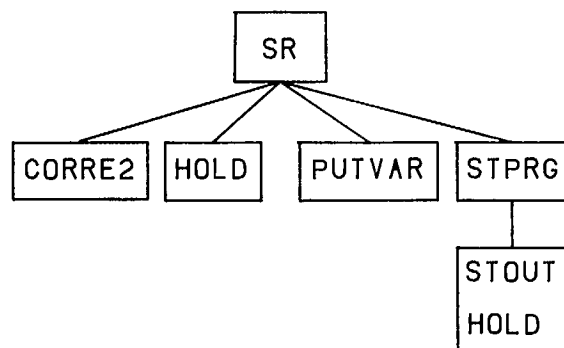
The following is a brief description of subroutines:

- LREG4A:** is called by the subroutine SPOCLR. Its purpose is to capture dependent and independent variables and to identify which type of linear regression is to be performed. It calls GETVAR, LREG4H.
- LREG4B:** computes the required summations for regression.
- LREG4C:** computes summary statistics.
- LREG4D:** displays regression results.
- LREG4E:** calculates table of residuals.
- LREG4F:** calculates the analysis of variance table.
- LREG4G:** calculates the confidence intervals for regression coefficients and estimated points.

LREG4H: calculates the mean experimental variance for repeat measurements.

GETVAR: gets a variable from the file.

2.5.2.4 SR: Stepwise regression

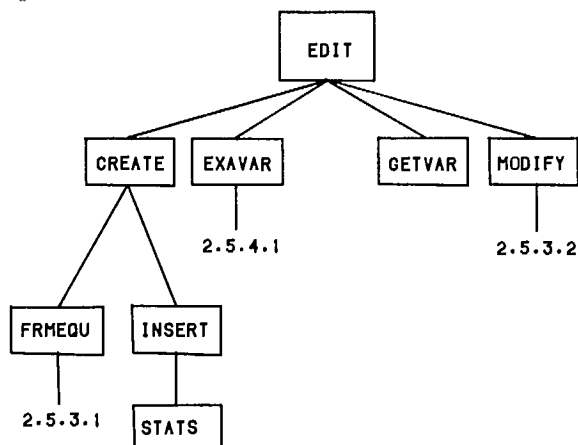


Subroutine SR is the stepwise regression used in STAMP. It is based on the algorithm of Bennett and Franklin (3).

To initiate the stepwise procedure, the user is required to enter control information as follows:

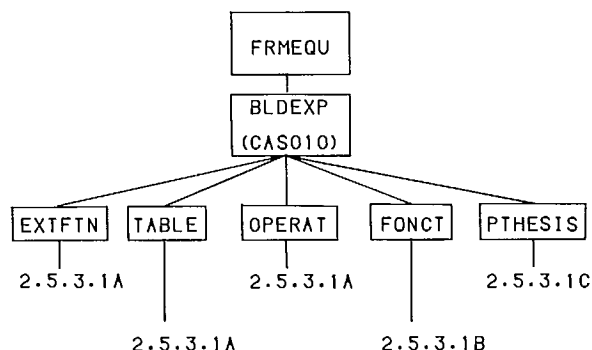
- A constant value of proportion of sum of squares that will be used to limit the number of variables entering in the regression. (Unless the proportion of the sum of squares attributable to an entering variable is greater than this value, the variable will not be entered.)
- A selection code (0 to 3) assigned to each variable where:
 - 0 = independent variable and available for selection
 - 1 = independent variable to be forced in the regression
 - 2 = independent variable that may be deleted
 - 3 = dependent variable.

2.5.3 EDIT



Subroutine EDIT allows the user to manipulate the data by modifying existing variables, adding a new variable, or deleting an existing variable. The user may also list the content of the data files.

2.5.3.1 FRMEQU



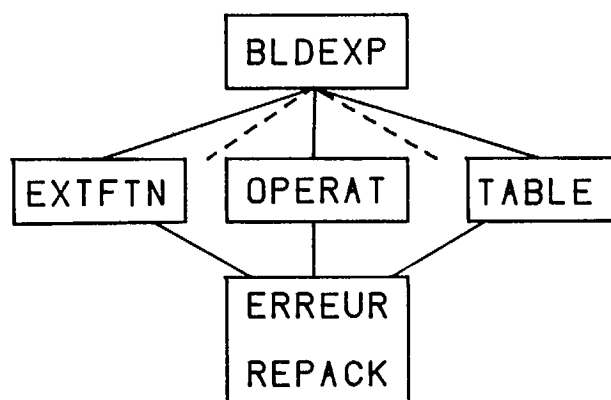
Subroutine FRMEQU allows a user to form a mathematical expression using up to five different variables, i.e.,

$$Y = f(X_1, X_2, \dots, X_n), n \leq 5$$

It is a version of the CASO10 routine available in EMRLIB*. It was modified and segmented to reduce its working space and was renamed BLDEXP.

The routines associated with FRMEQU and BLDEXP are as follows:

A — EXTFTN, OPERAT, TABLE

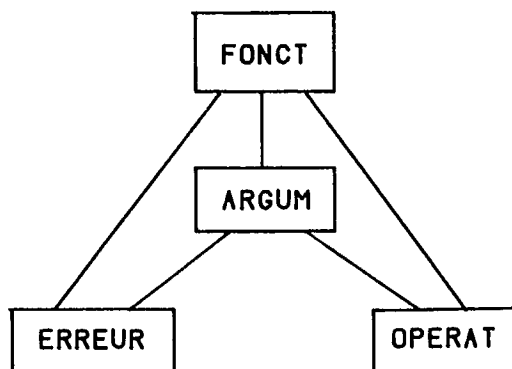


These three subroutines call the same subroutines ERREUR and REPACK.

Following is a brief description of each.

- EXTFTN:** searches for external FORTRAN functions.
- OPERAT:** performs the basic FORTRAN operations.
- TABLE:** sets up a table of codes to identify all legal characters as well as function names.
- ERREUR:** provides a comprehensive error message.
- REPACK:** redefines the input strings.

B — FONCT

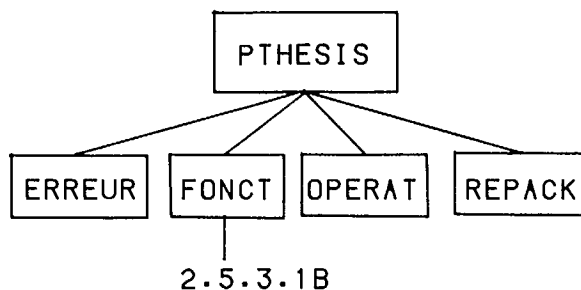


Subroutine FONCT performs the computation associated with the following functions:

EXP	SIN	COS	TAN	SQRT
DIM	ALOG	TANH	COSH	SINH
ATAN	ACOS	ASIN	SIGN	
AINT	AMOD	AMAX1	AMIN1	
ATAN2	FLOAT	ALOG10	ABS	

Subroutine ARGUM is called when functions with two arguments are encountered to calculate values of each argument separately.

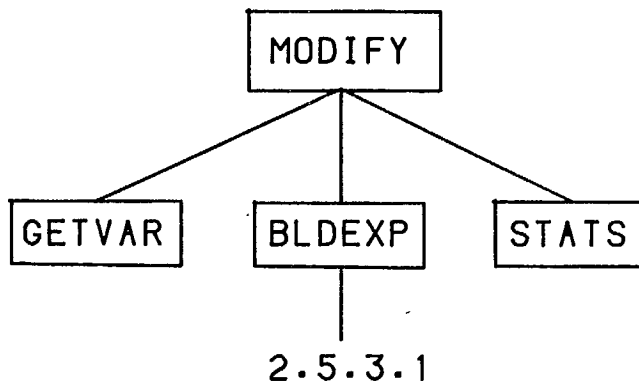
C — PTHESIS



Subroutine PTHESIS scans for matching parentheses and determines whether the contents of a set of parentheses is an argument of a function or a group of variables separated by operators.

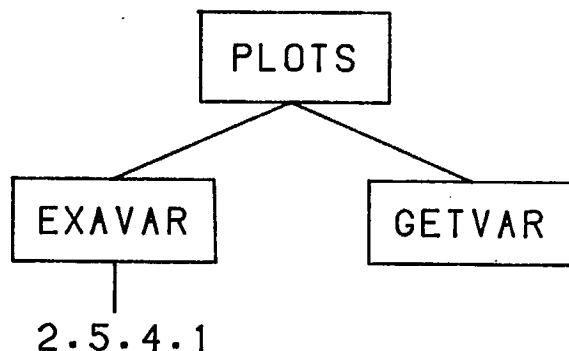
*Computer Science Centre, Energy, Mines and Resources Canada, Ottawa

2.5.3.2 MODIFY



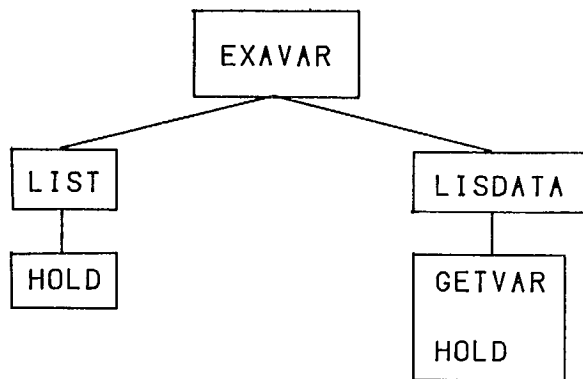
Subroutine MODIFY allows a user to change the observations of a given variable.

2.5.4 PLOTS



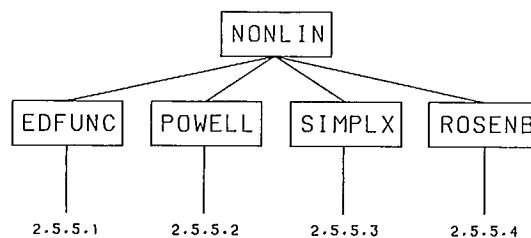
Subroutine PLOTS offers two types of line plots described previously. When in this module, the user may examine the contents of the data files.

2.5.4.1 EXAVAR



Subroutine EXAVAR allows the user to display information about the variables.

2.5.5 Non-Linear Optimization Module



When the NONLIN module is entered, the user is first prompted to edit the function to be minimized (subroutine EDFUNC). Then the menu of NONLIN is presented and the user can choose one of the proposed non-linear optimization algorithms.

2.5.5.1 EDFUNC: Editing of a non-linear function

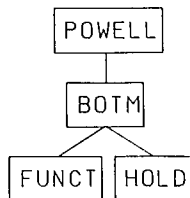
EDFUNC is the main subroutine involved in the editing of a non-linear function. As FRMEQU (Section 2.5.3.1) it is based on the CASO10 routine. The main difference between FRMEQU and EDFUNC is that the latter can handle indexed variables. Otherwise, all the subroutines used by EDFUNC have the same purpose as those used by FRMEQU. Table 1 gives the correspondence between the names of routines. Section 2.5.3.1 gives more details about the purpose of each routine.

Table 1 – Correspondence between routines associated with FRMEQU and EDFUNC

FRMEQU	EDFUNC
BLDEXP	BUILD
PTHESES	BRACKET
REPACK	COMPRS
TABLE	TABLE2
OPERAT	BASICO
FONCT	FONCT2
EXTFTN	EXTFNC
ERREUR	ERROR
ARGUM	ARGUM2

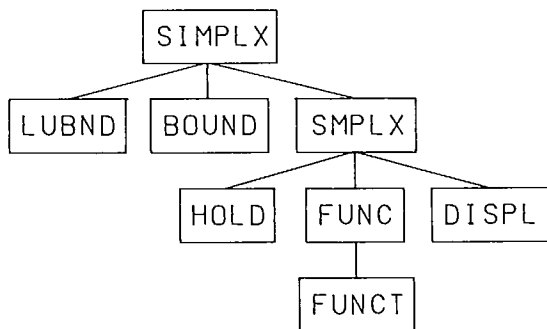
When the function is edited, EDFUNC calls subroutine FUNCT to evaluate it and then displays the value.

2.5.5.2 POWELL: BOTM algorithm option



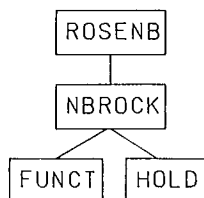
The purpose of subroutine POWELL is to read the values of various search conditions and to call subroutine BOTM to execute the BOTM algorithm. Subroutine FUNCT is used to compute the value of the non-linear function to be minimized. More details about the algorithm proposed by Powell (11) are given in Section 4.1.6.2.

2.5.5.3 SIMPLX: Simplex algorithm option



The purpose of subroutine SIMPLX is to read the values of various search conditions, read eventual bounds for search variables, initiate the search process and call the subroutine SMPLX which executes the simplex algorithm. DISPL is used to display intermediate results and FUNC to call subroutine FUNCT which computes the values of the function to be minimized (see Section 4.1.6.3).

2.5.5.4 ROSENB: ROSENB algorithm option



The purpose of subroutine ROSENB is to read various search conditions and to call subroutine NBROCK which performs the minimization using the algorithm proposed by Rosenbrock (15). FUNCT computes the values of the function to be minimized (see Section 4.1.6.4).

2.6 ADSTAMP: BATCH ENTRY

To create new STAMP data files or to enter many variables at the same time into old data files, the following program is proposed: ADSTAMP. It is a stand-alone program. It requires that an ordinary sequential file containing the data has been created and named TAPE10 (see Section 2.3.1.2). As a result, ADSTAMP creates or extends TAPE25 and TAPE26, the INDEX and DATA files of STAMP.

In a first step, ADSTAMP reads TAPE10 and checks for possible errors. In a second step, it computes general statistics about the new variables that will be written as well as the observations in the DATA file.

Finally, new records are entered into TAPE25 and TAPE26 if new variable names are not already used in the old files.

If necessary, ADSTAMP prints its own error messages. They are related to the status of the STAMP data files, the format to be used to read TAPE10, the variable names, the numbers of variables and observations, and the STOP parameter.

The procedure in Appendix C gives an example of a control card deck used to run ADSTAMP.

2.7 CLEANF: CLEANING DATA FILES

The program named CLEANF is independent from the STAMP package. Its purpose is to remove empty records from the INDEX and DATA files. Every time a variable or an observation is modified by the user, empty records mark off old records while new records are appended to the files. Therefore, CLEANF should be activated from time to time to keep STAMP data files in a compressed format.

CLEANF uses four data files:

- TAPE25: old INDEX file
- TAPE26: old DATA file
- TAPE35: new INDEX file
- TAPE36: new DATA file.

These files must be specified before CLEANF is executed. Their structure is installation dependent as well as the routines used to reorganize them.

The procedure in Appendix C gives an example of the control cards used to run CLEANF on a Cyber using NOS/BE operating system.

2.8 INSTALLATION-DEPENDENT ROUTINES

For an efficient retrieval of data from files, STAMP uses indexed sequential (IS) and actual key (AK) data files. These files and the subroutines used to manipulate them are system dependent. Table 2 gives a cross reference of the system-dependent subroutines and the main programs STAMP, ADSTAMP and CLEANF.

Table 2 – System-dependent subroutines and their occurrence

	STAMP	ADSTAMP	CLEANF
FILEAK	X	X	X
FILEIS	X	X	X
OPENM	X	X	X
CLOSEM	X	X	X
IFETCH	X	X	
REPLC	X	X	
PUT	X	X	
GET	X		X
GETN	X		X
DLTE	X		X
REWND	X		
CONNEC	X		
RECOVR	X		

The following gives a brief description of the purpose of each subroutine.

- FILExx:** FILEIS and FILEAK define INDEX and DATA files structure.
- OPENM:** opens and prepares a file for further processing.
- CLOSEM:** properly closes a file after processing is completed.
- IFETCH:** is an integer function that returns the value of one of the parameters located in the File Information Table of a file.
- REPLC:** replaces a record in a file.
- PUT:** writes a record in a file.
- GET:** reads a record from a file.
- GETN:** reads the next record (in sequential order) from a non-sequential file (IS or AK file).
- DLTE:** deletes a record in a file.
- REWND:** positions a file to the Beginning of Information.
- CONNEC:** connects INPUT and OUTPUT files for interactive users.
- RECOVR:** enables a last process (i.e., files closing) to be executed when a fatal error occurs during execution of STAMP.

3. SYSTEM DOCUMENTATION

3.1 COMPUTER EQUIPMENT

STAMP was developed on a CYBER 730/30 operating under NOS/BE, Energy, Mines and Resources. It requires a CRT terminal and uses two logical disc files.

These files are referred to as INDEX and DATA files respectively. The INDEX and DATA files are index sequential and random access files under CDC Record Manager.

All routines dealing with Record Manager are installation dependent.

3.2 SOURCE PROGRAM

Enquiries about the STAMP program should be directed to CANMET (see Section 1).

3.3 STORAGE REQUIREMENTS

The STAMP program requires a maximum of 70 K octal words (56 K decimal) of memory. It can run under 60 K words when the function entry subroutine (Section 2.5.3.1) is not involved.

3.4 MAINTENANCE AND UPDATE

3.4.1 Introduction

The STAMP package consists of several routines. When changes to the package are required, the following libraries/files are impacted:

- source library
- relocatable library
- SEGLOAD file
- STAMP data files.

Changes to the package occur when new routines are to be added, existing routines to be replaced or deleted or when the data files are to be reformatted.

A discussion on these changes follows.

3.4.2 Changes to the System Routines

The following steps must be adhered to:

- analyzing module(s)
- inserting revised or new module(s) in the source library
- inserting revised or new relocatable(s) in the relocatable library
- redefining the tree structure (if necessary) and creating a new segload file.

3.4.2.1 Analysis of module(s)

During this analysis, the following items must be considered:

- argument list
- common area WORK
- subroutine(s) called
- I/O activity.

In general, the argument list of the revised or new subroutine will be moved to the common area WORK which will be redefined according to the requirements of the module.

NOTE: The common area WORK is global in the STAMP package. Each module in STAMP uses it to suit its own purpose. The maximum size of this area is 966 words.

The user must ensure that all subroutines called by the new module are available, either in source code or in existing libraries. These libraries must be available during the "SEGLOAD" phase.

I/O activities in STAMP are carried out in five files. When changes to the package are required, the impact on these files should be considered. Section 3.4.3 describes these files.

3.4.2.2 Source library

STAMP source programs are maintained in a library called UPDATE. This is a CDC software package and the user should refer to the UPDATE manuals (user guide and reference manual).

3.4.2.3 Relocatable library

The STAMP relocatable library is maintained with the EDITLIB facility. Again, this is a CDC software package.

When changes are performed on the source programs, the output of the UPDATE run is assumed to be on the COMPILE file.

To insert these changes in the relocatable library, the user must compile the new modules with the SYSEDT option on, i.e.:

FTN, I=COMPILE, SYSEDT.

The relocatables will be on the file LGO.

If the user is adding new modules, then the EDITLIB directives are:

- ADD (*,LGO,AL=1)
- FINISH.
- ENDRUN.

To replace existing modules, the directives are:

- REPLACE (*,LGO,AL=1)
- FINISH
- ENDRUN.

NOTE: Refer to the CDC's EDTLIB manuals for more details.

3.4.2.4 Redefining the tree structure

Since the STAMP package is segmented, any changes require a SEGLOAD run. The user must be familiar with the SEGLOAD section of the CDC's LOADER manual. Figure 1 gives the current tree structure of STAMP. The file required for a SEGLOAD run is given in Appendix C.

3.4.3 STAMP Files

The STAMP system uses four files. They are:

- INPUT file
- OUTPUT file
- INDEX file
- DATA file.

The INPUT and OUTPUT files serve to provide direction to the STAMP program and relay information to the user.

The structure of both INDEX and DATA files is Cyber dependent.

3.4.3.1 INDEX file

The INDEX file is an indexed sequential file with fixed length records of eight words. It contains the names assigned by the users to the variables, their descriptors and their address in the DATA file.

3.4.3.2 DATA file

The DATA file contains all the values plus descriptive statistics associated with a variable.

The descriptive statistics occupy the first five words of the record and the remaining 59 words are the observations themselves. Note that STAMP uses 42 words to store the observations.

This is an actual key file. Whenever a new variable is inserted, the system provides the storing address of the data.

3.5 CONTROL CARDS

A listing of a procedure to run STAMP, ADSTAMP and CLEANF is presented in Appendix C. It uses Cyber Control Language (refer to Cyber's CCL user's guide).

The procedure first attaches the data files INDEX and DATA and takes security copies of them. If these files do not exist, test files are used instead.

Depending on the value of parameter UMODE, data are then edited in BATCH mode using ADSTAMP program, or data files are cleaned up using CLEANF program. Then, in all cases, STAMP is executed. If everything worked well, security data files are purged and old files extended to keep the last modifications to the files permanent. If STAMP program aborted during execution, an error message prints the name of security files. STAMP has its own way to avoid file destruction due to most problems occurring during execution, but it cannot prevent special problems such as an electrical breakdown. This procedure proved to be useful in several cases.

4. OPERATING DOCUMENTATION

The purpose of this section is to assist the user in utilizing the STAMP package.

The STAMP package offers the user the tools to edit the data files, several regression methods, and optimization algorithms.

Four types of regressions are proposed:

1. multiple linear regression
2. simple weighted linear regression
3. stepwise linear regression
4. polynomial regression.

Three optimization algorithms are proposed:

1. BOTM algorithm
2. Simplex algorithm
3. ROSENB algorithm.

The following sections describe how to operate the various options. Figure 2 shows the network of all the options.

4.1 OPERATING INSTRUCTIONS

The first menu which shows up is the STAMP menu. It gives access to six different modules. The first module allows the user to modify the content of the data file and the second to produce simplified graphics. The third one presents the descriptive statistics of a given variable while the fourth one allows the user to build the multiple correlation matrix of several variables. The fifth module contains the regression programs and the sixth one the non-linear optimization algorithms. Two options, standard in all the menus, are also presented. They permit the user to view the modules or options of the current module or to exit it and go one step back in the hierarchy of the menus. The following presents the STAMP menu.

- | | |
|------------------------------------|---|
| 1 = EDIT | to modify data file content. |
| 2 = PLOTS | to produce a plot of one or two variables. |
| 3 = STATS | descriptive statistics on a given variable. |
| 4 = CORRELATION | multiple correlation matrix of several variables. |
| 5 = REGRESSIONS | to create a model using various regression techniques. |
| 6 = NON-LINEAR OPTIMIZATION | to create a non-linear model using various optimization algorithms. |
| 99 = HELP | to display menu. |
| 0 = STOP | to exit STAMP package. |

4.1.1 EDIT

The EDIT module allows a user to modify the contents of the data files. This can be accomplished by:

- modifying the observations of a variable
- inserting a new variable
- purging an existing variable
- listing values of variables.

In each of the following options, the user must supply a variable name. When this variable name is not found in the INDEX file, the program returns to the menu of the entered option.

Options

- | | |
|------------------------------|---|
| 1 = MODIFY A VARIABLE | to modify the observations of a variable. |
| 2 = CREATE A VARIABLE | to insert a new variable. |
| 3 = DELETE A VARIABLE | to purge an existing variable. |
| 4 = EXAMINE VARIABLES | to view the content of data files. |
| 99 = HELP | to display menu. |
| 0 = EXIT | to return to the STAMP menu. |

4.1.1.1 MODIFY A VARIABLE (EDIT)

There are five ways in which the observation of a given variable may be modified. The first one consists of adding an observation in any position. The second one allows deletion of an observation in any position. The third one permits the user to reenter all the observations of the variable. The fourth one allows the user to reenter all the observations with a function and the last one, by changing a specific observation.

Options

- | | |
|--|---|
| 1 = ADD AN OBSERVATION | to add an observation in any position. |
| 2 = DELETE AN OBSERVATION | to purge an observation in any position. |
| 3 = REENTER COMPLETE OBSERVATIONS | to reenter all the observations. |
| 4 = REENTER WITH FUNCTION | to transform a variable using a function. |

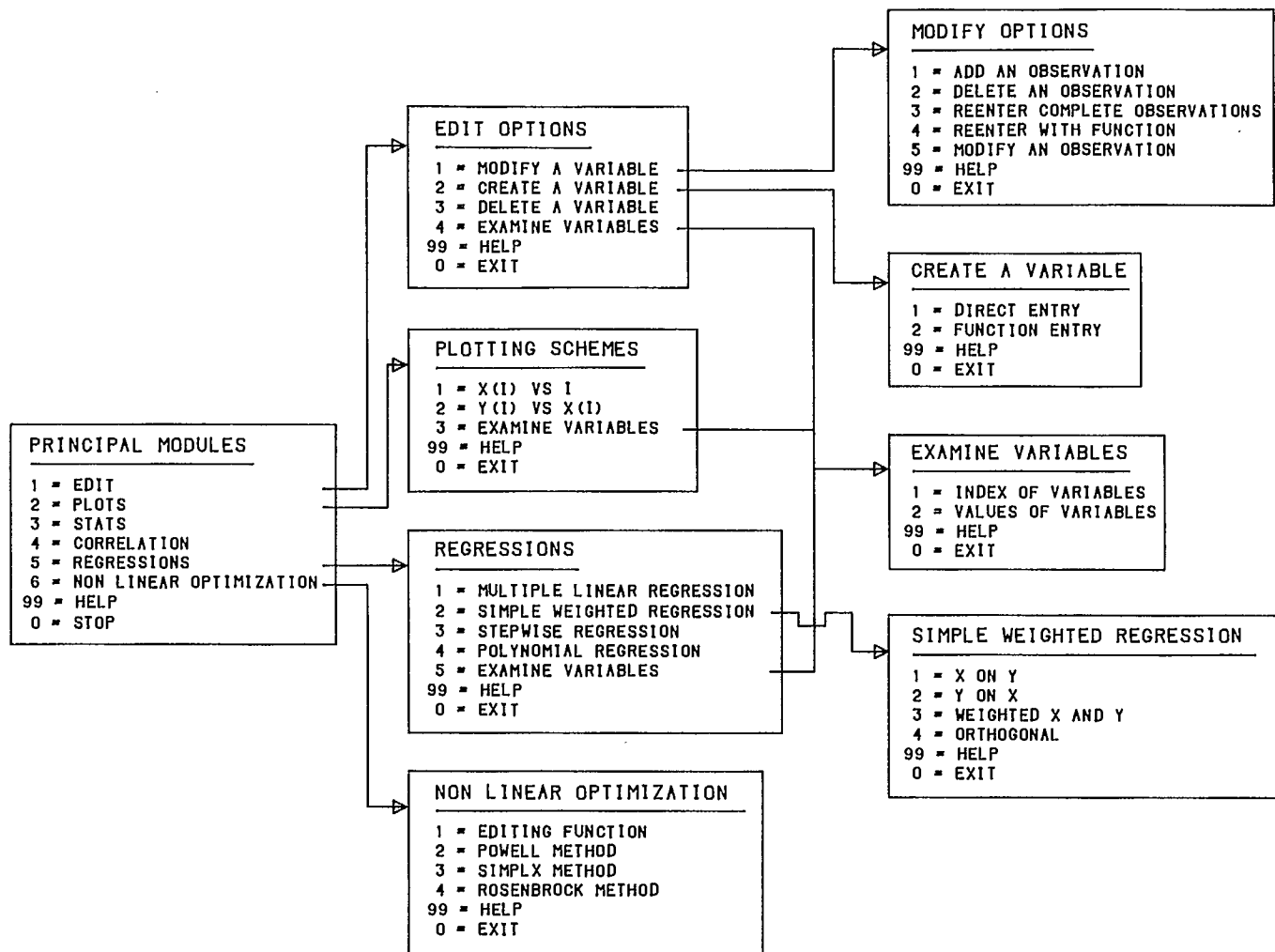


Fig. 2 — Summary of STAMP options

5 = **MODIFY AN
OBSERVATION**

to change a specific
observation.

99 = **HELP**

to display menu.

0 = **EXIT**

to return to the EDIT
menu.

selected, the observations for this variable are displayed
along with a position number.

1 = xxxx 2 = xxxx 3 = xxxx N = xxxx

1 — **ADD AN OBSERVATION** [MODIFY (EDIT)]

When this module is entered and a variable name
selected, the observations for this variable are displayed
along with a position number.

1 = xxxx 2 = xxxx 3 = xxxx N = xxxx

The user enters the position number where the observa-
tion is to be added and all subsequent observations in
the array are shifted by one position. The user is then
prompted to enter the new observation. The resulting
variable may be reviewed if so desired.

This process is stopped when a "0" is entered as a
position number. This results in the display of the MOD-
IFY menu.

2 — **DELETE AN OBSERVATION** [MODIFY(EDIT)]

When this module is entered and a variable name

The user enters the position number of the observation
to be deleted. The resulting variable may be reviewed if
so desired.

This process is stopped when a "0" is entered as a
position number. This results in the display of the
MODIFY menu.

3 — **REENTER COMPLETE OBSERVATIONS**

[MODIFY(EDIT)]

When this module is entered and a variable name
selected, the user is asked to enter the number of
observations to be entered. Then the observation is read
in. Note that all the old observations are purged. The
program returns to the MODIFY option of the EDIT
module.

4 — **REENTER WITH FUNCTION** [MODIFY(EDIT)]

In this option, the user transforms each observation with
a function. The function statement recognizes the vari-
able name by "V1". (See example below.)

Example

SYSTEM:	MODIFY	(HELP = 99):4
SYSTEM:	ENTER NAME () :FLOW
SYSTEM:	V1 = FLOW	
SYSTEM:	ENTER THE FUNCTION	
USER:	(SIN(V1) + SQRT (V1))/3.14	
SYSTEM:	System lists the new observations of variable FLOW	
SYSTEM:	MODIFY	(HELP = 99):
	The program returns to the MODIFY menu.	

5 — **MODIFY AN OBSERVATION** [MODIFY(EDIT)]

In this option, the observations of the selected variable are displayed along with a position number. The user then enters the desired position number and the new observation. This process is continued until a "0" is entered as a position number. When "0" is entered, the program returns to the MODIFY menu.

4.1.1.2 **CREATE A VARIABLE (EDIT)**

A user may add a variable in two ways. The first method allows the user to enter the observations directly. The second method allows the user to create a variable from existing variables.

Options

- 1 = **DIRECT ENTRY** to enter the variable.
- 2 = **FUNCTION ENTRY** to enter the variable with a function.
- 99 = **HELP** to display menu.
- 0 = **EXIT** to return to the EDIT menu.

1 — **DIRECT ENTRY** [CREATE A VARIABLE(EDIT)]

When this option is selected, the user is asked to enter the following information:

- | | |
|--|--|
| — Name of the variable | This name must be unique and made of up to four alpha-numeric characters, the first character being non-numeric. |
| — Description of the variable | 16 alphanumeric characters, with no restrictions. |
| — Number of observations | |
| — Observations associated with this variable | |

When this information is entered, the program returns to the option CREATE A VARIABLE of the EDIT menu.

2 — FUNCTION ENTRY [CREATE A VARIABLE(EDIT)]

This option permits the user to construct a new variable from existing variables (up to five variables may be used). This is done by entering a function whose arguments are existing variables.

The variable names serving as arguments are then entered. If fewer than five variables are required, the user enters "STOP" after the last variable. If the number of the observations of the variable last entered is different from the previous one, the user is informed and must replace it with a new variable of the same number of observations.

The user may enter up to nine constants. If fewer than nine are required, the user enters a "0" after the last one.

An identification code (V1 to V5 for variables and P1 to P9 for constants) is associated to each selected variable and

each entered constant. These are listed before the user can enter the function.

If no errors are detected, the new observations are displayed on the screen. The user is then asked to enter the variable name and a description for the newly created variable to be inserted into the data files.

If errors are detected during compilation of the function, a message describing the error is printed and the user is asked if the same variables are to be kept for a second attempt. If so, the corrected function can be entered.

When the process is completed, the program returns to the option CREATE A VARIABLE of the EDIT module. (See example below.)

Example

SYSTEM:	STAMP	(HELP = 99):1	(enter EDIT module)
SYSTEM:	EDIT	(HELP = 99):2	(enter CREATE A VARIABLE option)
SYSTEM:	CREATE A VARIABLE	(HELP = 99):2	function entry
SYSTEM:	ENTER "STOP" TO FINISH		
SYSTEM:	ENTER NAME (IND):ACC		
SYSTEM:	ENTER NAME (IND):PRIC		
SYSTEM:	ENTER NAME (IND):STOP		(2 variables required)
SYSTEM:	ENTER A PARAMETER: 3.141592		
SYSTEM:	ENTER A PARAMETER: 2.8182		
SYSTEM:	ENTER A PARAMETER: 0		(2 constants required)
SYSTEM:	V1 = ACC		
SYSTEM:	V2 = PRIC		
SYSTEM:	P1 = 3.141592		
SYSTEM:	P2 = 2.8182		
SYSTEM:	ENTER THE FUNCTION		
USER:	SIN(V1 + P2) - (ALOG10(P1*V2)/(10.0*P1))		

4.1.1.3 DELETE A VARIABLE (EDIT)

This option allows the user to purge a variable from the files. If the variable name is not found, the program returns to the menu of the EDIT module. Otherwise, the variable is deleted and the program returns to the menu of the EDIT module.

4.1.1.4 EXAMINE VARIABLES (EDIT)

This option allows the user to view the files content. The variable index or the observations associated with selected variable names may be listed. Observations of up to five variables can be listed at the same time.

Options

- | | |
|--------------------------------|---|
| 1 = INDEX OF VARIABLES | to list variable names, description and number of observations. |
| 2 = VALUES OF VARIABLES | to display observations of selected variables. |
| 99 = HELP | to display menu. |
| 0 = EXIT | to return to calling modules (EDIT, PLOTS, REGRESSIONS). |

1 — **INDEX OF VARIABLES [EXAMINE VARIABLES (EDIT-PLOTS-REGRESSIONS)]**

This option allows the user to list in tabular form the name, description and number of observations of the variables in the files. This list is produced in alphabetical order.

A range of names must be selected. The user is asked the initials of the first and last names of the range to be listed. Only variable names within this range are displayed.

The user may enter "0" to halt the listing, or "C" to continue the listing if it contains more than 17 names.

When the list is finished, the program returns to the menu of EXAMINE VARIABLES.

2 — **VALUES OF VARIABLES [EXAMINE VARIABLES (EDIT-PLOTS-REGRESSIONS)]**

This option allows the user to view the observations of up to five variables in tabular form. The user is asked for the number of variables to be displayed and then the variable names.

If the numbers of observations of the variables entered are different, the user is informed and must enter new variables of the same number of observations.

To exit this module, the user enters a "0" when asked for the number of variables to be displayed. The program returns to the menu of EXAMINE VARIABLES.

4.1.2 PLOTS

This module offers the user two types of plots. The first type plots the observations of a variable versus the index of the observations in the array and the second type plots a variable versus another one. An option allows the user to view the content of the files.

Options

- | | |
|------------------------------|---|
| 1 = X(I) vs I | to plot variable X versus its index in array. |
| 2 = Y(I) vs X(I) | to plot observations of two variables. |
| 3 = EXAMINE VARIABLES | to view file content. |
| 99 = HELP | to display menu. |
| 0 = EXIT | to return to STAMP menu. |

4.1.2.1 X(I) vs I (PLOTS)

The user enters the variable name to be plotted. An invalid variable name returns the user to the PLOTS menu. For a valid variable name, a plot is produced followed by a return to the menu of PLOTS.

4.1.2.2 Y(I) vs X(I) (PLOTS)

The user needs to enter two variable names. The horizontal variable is labelled (HOR) and the vertical variable (VER). The graph is then produced followed by a return to the PLOTS menu. In the event of an invalid variable name, the user is returned to the menu of PLOTS.

4.1.2.3 EXAMINE VARIABLES (PLOTS)

This option allows the user to view the files content (see Section 4.1.1.4).

4.1.3 STATS

This module allows the user to list the descriptive statistics of the variables. They are:

- minimum value
- maximum value
- average
- standard deviation.

The observations associated with the variables are also displayed.

The user is prompted for a variable name. If "STOP" is entered instead, a return to the menu of STAMP is performed.

4.1.4 CORRELATION

This module allows the user to produce a correlation matrix. The user enters successively up to ten variable names. A "STOP" terminates the entry of variable names. When "STOP" is entered, the correlation matrix is displayed and the user is returned to the menu of STAMP.

4.1.5 REGRESSIONS

This module permits the user to enter four regression techniques. They are:

- multiple linear regression
- simple weighted linear regression
- stepwise linear regression
- polynomial regression.

Options

- 1 = **MULTIPLE LINEAR REGRESSION**
- 2 = **SIMPLE WEIGHTED REGRESSION**
- 3 = **STEPWISE REGRESSION**
- 4 = **POLYNOMIAL REGRESSION**
- 5 = **EXAMINE VARIABLE**
- 99 = **HELP** to display menu.
- 0 = **EXIT** to return to STAMP menu.

4.1.5.1 MULTIPLE LINEAR REGRESSION

When this module is entered, the user must first enter the variable name of the dependent variable and then, successively, the variable names of the independent variables. When "STOP" is entered as one of the independent variables, the model is complete and the algorithm for multiple linear regression is initiated.

If the variance of the dependent variable is required for modelling purposes, the user must have prepared it previously using the EDIT options (see Section 4.1.1.2).

The results of the model are then displayed. The user has the opportunity to save the Y-estimates and residuals for plotting purposes or later use. This method is discussed in detail in a separate report (4).

4.1.5.2 SIMPLE WEIGHTED REGRESSION

The user must enter the names of the dependent and independent variables and the standard deviations of the observations to be used in the weighted regression. This option offers several sub-options such as regressions of:

- Y on X
- X on Y
- on X and Y
- orthogonal.

These methods are documented in a separate report (5).

4.1.5.3 STEPWISE REGRESSION

The stepwise regression program has been taken from the IBM Scientific Subroutine Package (6) where it is documented as subroutine STPRG.

4.1.5.4 POLYNOMIAL REGRESSION

The polynomial regression program has also been taken from the IBM Scientific Subroutine Package (6) where it is documented as main program POLRG.

4.1.6 NON-LINEAR OPTIMIZATION

This STAMP module gives access to the non-linear optimization program. Six options are available. The first one permits the user to define the function to be minimized (later called objective function). Since this step has to be performed prior to access to any of the optimization programs, the first option is automatically called as soon as the user enters the NONLIN module. The second, third and fourth options are used to run one of the three proposed optimization programs using BOTM (7), SIMPLX (8) and ROSENB algorithms (7) respectively. The last two options are the standard HELP and EXIT facilities common to all modules.

Options

- 1 = **EDITING FUNCTION** to edit the objective function.
- 2 = **POWELL METHOD** to access BOTM algorithm.
- 3 = **SIMPLX METHOD** to access SIMPLX algorithm.
- 4 = **ROSENBROCK METHOD** to access ROSENB algorithm.
- 99 = **HELP** to display menu.
- 0 = **EXIT** to return to STAMP menu.

4.1.6.1 EDITING FUNCTION: Non-linear optimization

The traditional regression methods (see Section 4.1.5) are used to determine the values of the coefficients of a linear model or of a polynomial model. When the model to be fitted to experimental data is highly non linear, one needs more powerful tools to calculate the best values of the coefficients. Several methods have been proposed and compared (9) in the literature. Their main drawback is that they always require the user to program a subroutine which computes the value of the objective function. Therefore, the user needs some programming skills. The editing option of the NONLIN module presents a very simple way to edit the objective function formula. All the programming is then done automatically by the STAMP package.

Two general classes of functions have been identified. These are: 1) functions which use only scalar variables such as:

$$F = a_1 \exp X_1 + a_2 X_2^{1.414} - a_3 X_3 \log X_4;$$

and 2) functions which use indexed variables (i.e., arrays) such as:

$$F = \sum_i a_1 \exp X_{1(i)} + a_2 X_{2(i)}^{1.414} - a_3 X_{3(i)} \log X_{4(i)}.$$

For simplicity, the second class of functions is edited the same way as the first class except that the user must identify the indexed variables as such to allow the program to recognize them. However, during execution, the program computes F_i , the function value for all observations i of indexed variables, and then computes $F = \sum_i F_i$ and uses that value for the objective function.

The following describes the way to edit a function in STAMP. The user will note that the indexed variables must contain the same number of observations to be accepted in the objective function.

When the NONLIN module is accessed or when the option EDITING FUNCTION is chosen, the system responds with:

ENTER FUNCTION USING
V1 to V5, K1 to K9, Z1 to Z9.
eg.: $(K1 \cdot Z1 + Z2 \cdot V2 - V1)^{**2}$

Vi are indexed variables from your file.

Ki are real constant values.

Zi are search variables.

You will be prompted to identify Vi, Ki, and Zi later on.

When writing the function, the user must follow all FORTRAN rules, use FORTRAN symbols (+, -, *, /, **) and FORTRAN implicit function names (see Section 2.5.3.1 B). The formula must fit in a line of up to 80 characters. For example, this is the way to write the formula given at the beginning of this section:

$$Z1 * \exp(V1) + Z2 \cdot V2^{**K1} - Z3 \cdot V3 * \text{ALOG}(V4).$$

Other examples of how to edit a function are given in Appendix B.

When the function is entered, the user is prompted to define Vi, Ki and Zi:

ENTER NAMES OF INDEXED VARIABLES V

ENTER NAME (i): where i varies from 1 to the number of indexed variables used in the formula, if any.

A maximum of five variables can be used. They must have the same number of observations and they must have been entered into the data files before the NONLIN module is accessed (see Section 4.1.1). As explained previously, all objective functions do not necessarily require the use of indexed variables; therefore, the prompt is displayed only when required.

When all indexed variables are defined, the following prompt is displayed:

ENTER REAL CONSTANT K VALUES

Ki = where i varies from 1 to the number of K constants used in the formula, if any.

Here the user is asked to define the real constants. It is not mandatory to use the Ki in the formula, it is only practical to save space in the function definition record (80 characters maximum): for instance, the following function is also acceptable:

$$(1.414 \cdot Z1 + Z2 \cdot V2 - V1)^{**2}$$

Then the user is asked to give estimated values to the search variables Zi:

ENTER ESTIMATES FOR SEARCH
VARIABLES Z

Zi = where i varies from 1 to the number of search variables used in the formula.

These variables are mandatory: they are, for instance, the coefficients of a model the user wishes to fit to experimental data. Up to nine Z variables may be used in the formula.

When this step is completed, STAMP compiles the function and evaluates it. The value is displayed and the user is asked if it is desirable to change the estimated values of the search variables. This process is repeated until the user is satisfied. STAMP keeps in memory the best estimated values; therefore, the optimization program starts searching for the optimum values from a location close to the solution. If an error is detected during compilation, a message is printed out and the user has to redefine the entire function. When the user is satisfied, the menu of the NONLIN module is displayed. At this point, any of the three optimization algorithms may be accessed.

4.1.6.2 POWELL METHOD: Non-linear optimization

The program used here is described by Kuester-Mize (7). It has been slightly modified (10) and adapted to the STAMP package. It uses the Powell method (11) to find the minimum value of an unconstrained non-linear function.

The method requires the user to set the values of various search conditions; to make it easier, default values have been entered but can be interactively modified by the user if necessary:

- The maximum number of iterations allowed is 40. This should be enough in all cases.
- The printing option is 2. Partial results (objective function and search variables values) are printed at the end of each iteration. When the printing option equals 1, partial results are printed when a minimum is found for each search variable within each iteration. When the printing option equals 3, only the final results are printed.
- The accuracy (i.e., tolerance) required on each search variable value at the end of the minimization process is 5% of the estimated values given by the user. For a more accurate result, values as low as 0.1% can be used but, in this case, the estimated values of the search variables should be close to the solution.
- The ESCALE factor is 1. That factor is used to define the step-length modifier applied on each variable during computation. To keep a reasonable computation time, the user who decreases the percentage value of the accuracy should increase the ESCALE factor.

4.1.6.3 SIMPLEX METHOD: Non-linear optimization

The program used here is based on the algorithm proposed by Nelder and Mead (8,12). It has been modified by Kelly and Pilgrim (13) and adapted to the STAMP package.

In SIMPLX, search variables can be constrained by upper or lower or both bounds (14). This option is not available in the Powell and Rosenbrock methods.

Default values have been entered to set the various search conditions but can be modified by the user:

- the maximum number of iterations is 1000.
- the number of significant digits of the values of the objective function is 5.
- the variance convergence is 0.2.
- the step-length modifier is 1.0.

The number of significant digits and the variance convergence value are both used to determine when the search for the optimum value of the objective function should stop. Whenever a new objective function value is obtained, the following calculation is performed using the last five objective function values truncated to the number of significant digits:

$$TT = \frac{1}{4} \sum_{i=1}^5 \left(f_i - \frac{1}{5} \sum_{j=1}^5 f_j \right)^2$$

If TT is greater than the variance convergence value, the program goes on. If TT is lower than the variance convergence value, SIMPLEX is restarted. When TT is again lower than the variance convergence value, the program tests if the minimum found is a local minimum. If it happens to be a local minimum, SIMPLEX is restarted a second time; if not, computation stops and final results are printed. A maximum of two restarts is allowed.

Partial results are printed every ten iterations and before each restart occurs. At the end of the minimization process, final results are printed out.

The user who desires to constrain the search variable values is first requested to enter a boundary code for the search variables. The possible boundary codes are:

- 0 = no bounds
- 1 = upper bound only
- 2 = lower bound only
- 3 = both lower and upper bounds.

Then the program prompts the user to enter the values of the bounds according to the entered codes.

4.1.6.4 ROSENBROCK METHOD: Non-linear optimization

The program uses the algorithm proposed by Rosenbrock (15) as described by Kuester and Mize (7). It has been adapted for the STAMP package.

Default values have been entered to set the various search conditions:

- The maximum number of evaluations of objective function is 1000.
- The maximum number of axes rotation is 20.
- The number of successive failures in all directions before termination is 5.
- The scaling code is 1, i.e., initial step-length multipliers are used after each axis rotation. When the code equals 2, final step-length multipliers are used to start a new stage.
- The accuracy required on the minimum value of the objective function is 0.01. This value is compared to the difference between the last and the previous minima found. This should be remembered and modified when dealing with a function starting already with a small value, otherwise the stopping criterion can be met at the very first iteration and the program stops searching.
- The scaling-up multiplier is 1.1. In all cases it must be greater than 1.
- The scaling-down multiplier is 0.9. In all cases it must be less than 1.
- The initial step-length modifiers are 5% of the estimated values of search variables.

During the minimization process, partial results are printed whenever a stage is completed. Final results are printed at the end of the process.

4.2 ERROR MESSAGES

Error messages are grouped under three sections. They are:

1. STAMP error messages
2. operating system error messages
3. fatal errors.

4.2.1 STAMP Error Messages

These messages are non-fatal and are related to the I/O processes of STAMP.

When two or more variables are required as in PLOTS, CORRELATION or REGRESSION, they must all have the same number of observations.

Other types of errors are assigning variable names which are already in use or attempting to delete or modify a variable using a non-existing name.

When such errors occur, the user is returned to the option of the menu under which the error occurred.

4.2.2 Operating System Error Messages

These are I/O related errors. They are non fatal. They occur when the system expects numeric values rather than alphanumeric values.

4.2.3 Fatal Errors

These occur mainly when using the FUNCTION capability of the package, i.e.,

- $\text{ALOG10}(X)$ when $X \leq 0.0$
- $X/P1$ when $P1 = 0$
- $\text{TAN}(X)$ when $X = \pi/2$

When such errors occur, the STAMP program is aborted and data files closed.

4.3 RUN TIME

Run time varies with problem size and length of interaction. A sample run where all options were used required a total of 8 CP sec. The connect time was 50 minutes.

5. REFERENCES

1. Boudoux, M., and Bonenfant, F. "SACADOS, System automatique conversationnel d'analyse de données statistiques" Rapp. Inf. LAU-X-39. 22 p.; Centre de Recherche Forestière des Laurentides; Québec; November 1979.
2. Ostle, B. Statistics In Research; Ch 6; Iowa State College Press; 1954.
3. Bennett, C.A., and Franklin, N.L. Statistical Analysis in Chemistry and the Chemical Industry; Appendix 6A; John Wiley and Sons; 1954.
4. Leung, J. "INTMLR, Interactive multiple linear regression"; Division Report MRP/MSL 82-48(IR); CANMET, Energy, Mines and Resources Canada; 1982.
5. Leung, J., Fortin, S., and Laguitton, D. "SPOCLR — A conversational FORTRAN program for simple regression analysis"; Division Report MRP/MSL 82-95(IR); CANMET, Energy, Mines and Resources Canada; 1982.
6. IBM. "System 360 Scientific Subroutine Package, Version III. Programmer's Manual"; Publication No. GH20-0205-4; Program No. 360A-CM-03X; 1970.
7. Kuester, J.L., and Mize, J.H. Optimization Techniques with FORTRAN; McGraw Hill Company; 1973.
8. Nelder, J.A., and Mead, R. "A simplex method for function minimization"; Computer J 7:308-313; 1965.
9. Himmelblau, D.M. "A uniform evaluation of unconstrained optimization techniques"; Numerical Methods for Non-Linear Optimization; Edited by F.A. Lootsma; Academic Press; New York; 1972.
10. Leung, J., and Laguitton, D. "INTDS, Interactive driver for an optimization program with direct search method." Division Report MRP/MSL 80-115(IR). CANMET, Energy, Mines and Resources Canada; 1980.
11. Powell, M.J.D. "An efficient method for finding the minimum of a function of several variables without calculating derivatives." Computer J 7:155-162; 1964.
12. Olsson, D.M., and Nelson, L.S. "The Nedler-Mead simplex procedure for function minimization." Technometrics, 17(1), 45-51; 1975.
13. Kelly, F., and Pilgrim, R.F. Personal communication. CANMET, Energy, Mines and Resources Canada; 1984.
14. Atwood, J.R., and Foster, W.W. "Transformation of bounded variables in simplex optimization techniques." Ind. Eng. Chem. Process. Des. Develop. 12(4), 485-486; 1973.
15. Rosenbrock, H.M. "An automatic method for finding the greatest or least value of a function." Computer J 3:175-185; 1960.

APPENDIX A
SAMPLE RUN OF STAMP

WELCOME TO THE STAMP PACKAGE
 HERE IS A LIST OF MODULES AND THEIR OPTIONS

MODULES	OPTIONS	SUB-OPTIONS
EDIT	MODIFY A VARIABLE	ADD AN OBSERVATION DELETE AN OBSERVATION REENTER COMPLETE OBSERVATIONS REENTER WITH FUNCTION MODIFY AN OBSERVATION DIRECT ENTRY FUNCTION ENTRY
	CREATE A VARIABLE	INDEX OF VARIABLES VALUES OF VARIABLES
	DELETE A VARIABLE	
	EXAMINE VARIABLES	
.....		
PLOTS	X(I) VS I Y(I) VS X(I) EXAMINE VARIABLES	
.....		
C TO CONTINUE : C		
.....		
STATS		
.....		
CORRELATION		
.....		
LINEAR	MULTIPLE LINEAR REGRESSION	
REGRESSIONS	SIMPLE LINEAR REGRESSION	X ON Y Y ON X WEIGHTED X AND Y ORTHOGONAL
	STEPWISE REGRESSION POLYNOMIAL REGRESSION EXAMINE VARIABLES	
.....		
NON LINEAR	EDITING FUNCTION	
OPTIMIZATION	POWELL METHOD SIMPLEX METHOD ROSENBROCK METHOD	

TO CONSULT A MENU, ENTER 99
 AND TO EXIT OF A MODULE, ENTER 0.

C TO CONTINUE : C
 PRINCIPAL MODULES

1 = EDIT
 2 = PLOTS
 3 = STATS
 4 = CORRELATION
 5 = LINEAR REGRESSION
 6 = NON LINEAR OPTIMIZATION
 99 = HELP
 0 = STOP

STAMP (HELP=99) : 1
 EDIT (HELP=99) : 99

EDIT OPTIONS

1 = MODIFY A VARIABLE
 2 = CREATE A VARIABLE
 3 = DELETE A VARIABLE
 4 = EXAMINE VARIABLES
 99 = HELP
 0 = EXIT

EDIT (HELP=99) : 4

```

EXAMINE VAR          (HELP=99) : 99
EXAMINE VARIABLES
-----
1  = INDEX OF VARIABLES
2  = VALUES OF VARIABLES
99 = HELP
0  = EXIT
EXAMINE VAR          (HELP=99) : 1
FROM LETTER : A
TO LETTER   : Z
THERE ARE    19 VARIABLES IN THE FILE
VARIABLES    DESCRIPTIONS          NB. OBSERV.
-----
DEM1          DEMO DATA              10.
DEM2          DEMO DATA              4.
LRXX          IND VA LINEAR R         5.
LRY Y        DEP VA LINEAR R         5.
MLR1          IND VA MULTI R         22.
MLR2          IND VA 2 MULTI R        22.
MLR3          DEP VA MULTI R         22.
MLRR          SQUARE OF MLR3         22.
NLX          SIEVE DIMENSIONS        10.
NLY          EFFICIENCY CURVE        10.
NLY*         PRED. EFFI. CURV        10.
POXX          IND VA POLY R          15.
POYY          DEP VA POLY R          15.
STP1          IND VA STEPWISE        30.
STP2          IND VA STEPWISE        30.
STP3          IND VA STEPWISE        30.
STP4          IND VA STEPWISE        30.
ENTER 0 TO EXIT
C TO CONTINUE : C
THERE ARE    19 VARIABLES IN THE FILE
VARIABLES    DESCRIPTIONS          NB. OBSERV.
-----
STP5          IND VA STEPWISE        30.
STP6          DEP VA STEPWISE        30.
EXAMINE VAR          (HELP=99) : 2
HOW MANY VARIABLES WOULD YOU LIKE TO LIST (MAX = 5, STOP = 0)?2
ENTER NAME (      ) : LRXX
ENTER NAME (      ) : LRY Y
SEQUENCE LRXX      LRY Y
NUMBER  ----      ----
1  .200E+01  .910E+00
2  .600E+01  .503E+01
3  .600E+01  .481E+01
4  .600E+01  .504E+01
5  .100E+02  .843E+01
EXAMINE VAR          (HELP=99) : 0
EDIT                (HELP=99) : 1
MODIFY              (HELP=99) : 99
MODIFY OPTIONS
-----
1  = ADD AN OBSERVATION
2  = DELETE AN OBSERVATION
3  = REENTER COMPLETE OBSERVATIONS
4  = REENTER WITH FUNCTION
5  = MODIFY AN OBSERVATION
99 = HELP
0  = EXIT
MODIFY              (HELP=99) : 1

```

```

ENTER NAME ( ) : DEM2
  1 = .100E+01,  2 = .141E+01,  3 = .173E+01,  4 = .200E+01,
INSERT POSITION (EXIT = 0) : 2
ENTER THE NEW OBSERVATION : 999
DISPLAY Y/N? : Y
  1 = .100E+01,  2 = .999E+03,  3 = .141E+01,  4 = .173E+01,  5 = .200E+01,
INSERT POSITION (EXIT = 0) : 0
MODIFY (HELP=99) : 2
ENTER NAME ( ) : DEM2
  1 = .100E+01,  2 = .999E+03,  3 = .141E+01,  4 = .173E+01,  5 = .200E+01,
POSITION TO BE DELETED (EXIT = 0) : 2
DISPLAY Y/N? : Y
  1 = .100E+01,  2 = .141E+01,  3 = .173E+01,  4 = .200E+01,
POSITION TO BE DELETED (EXIT = 0) : 0
MODIFY (HELP=99) : 3
ENTER NAME ( ) : XXXX
NAME XXXX NOT FOUND
MODIFY (HELP=99) : 3
ENTER NAME ( ) : DEM2
ENTER NUMBER OF OBSERVATIONS : 4
ENTER MODIFICATION
1 2 5 9
MODIFY (HELP=99) : 4
ENTER NAME ( ) : DEM2
      V1 = DEM2
ENTER THE FUNCTION
3.14+V1**2-SQRT(2+V1)
MODIFY (HELP=99) : 5
ENTER NAME ( ) : DEM2
  1 = .241E+01,  2 = .514E+01,  3 = .255E+02,  4 = .808E+02,
ENTER POSITION (EXIT = 0) : 2
ENTER OBSERVATION  2 : 9.9
DISPLAY Y/N? : Y
  1 = .241E+01,  2 = .990E+01,  3 = .255E+02,  4 = .808E+02,
ENTER POSITION (EXIT = 0) : 0
MODIFY (HELP=99) : 0
EDIT (HELP=99) : 2
CREATE A VAR (HELP=99) : 99
CREATE A VARIABLE
-----
  1 = DIRECT ENTRY
  2 = FUNCTION ENTRY
  99 = HELP
  0 = EXIT
CREATE A VAR (HELP=99) : 1
ENTER A NAME FOR YOUR VARIABLE : BB
ENTER A DESCRIPTOR (16 CHAR.) FOR THIS NAME : BB
DESCRIPTOR : TEST DATA
ENTER THE NUMBER OF OBSERVATIONS (MAX = 42) : 4
ENTER THE OBSERVATIONS
10 20 30 40
CREATE A VAR (HELP=99) : 2
ENTER STOP TO EXIT
ENTER NAME ( IND ) : BB
ENTER NAME ( IND ) : DEM2
ENTER NAME ( IND ) : STOP

```


[illegible]

REGRESSIONS (HELP=99) : 99

REGRESSIONS

1 = MULTIPLE LINEAR REGRESSION
2 = SIMPLE WEIGHTED REGRESSION
3 = STEPWISE REGRESSION
4 = POLYNOMIAL REGRESSION
5 = EXAMINE VARIABLES
99 = HELP
0 = EXIT

REGRESSIONS (HELP=99) : 1

INSTRUCTION Y/N?Y

ENTERING DATA IDENTIFY THE DEPENDANT VARIABLE
IDENTIFY THE INDEPENDANT VARIABLES
ENTER STOP TO FINISH

RESULTS 1 SCREEN : STATISTICS ON PARAMETERS
2 SCREEN : ANOVA TABLE
3 SCREEN : INTERVALS ESTIMATED Y AND PARAMETERS

ENTER "STOP" FOR FINISH

ENTER NAME (DEP) : MLR3

ENTER NAME (IND) : MLR1

ENTER NAME (IND) : MLR2

ENTER NAME (IND) : STOP

CURVE FITTING FOR MULTI-VARIABLE LINEAR REGRESSION

$Y = B_0 + B_1(X_1 - \bar{X}_1) + B_2(X_2 - \bar{X}_2) + \dots$

WHERE \bar{X}_1 , \bar{X}_2 ARE THE (WEIGHTED) MEAN VALUES

OF X_1 , X_2 RESPECTIVELY

ANY REPEAT Y FOR SAME X VALUES ?

N

DID YOU PREPARE THE ERROR VARIANCE OF Y ?

Y

ENTER NAME (ERV) : MLRR

IN CASE OF ERROR VARIANCE EQUALS TO $K \cdot (\text{ANY FUNCTION})$,

DID YOU ENTER THE VALUE OF PROPORTIONALITY FACTOR K ?

N

VARIABLE NO.	WEIGHTED MEAN	STANDARD DEVIATION	CORRELATION X VS Y	REGRESSION COEFFICIENT	COMPUTED T VALUE
2	.03542	.73585	1.66962	13.99964	94.09978
3	.00368	1.01507	2.03110	-.96653	6.63069
DEP.	7.64455	5.93245			
T (19), 95 %: 2.09 (TWO TAILS)					
INTERCEPT			.47620		
INTERCEPT $B_0 - B_1 \cdot \bar{X}_1 - B_2 \cdot \bar{X}_2 - \dots$.45758		
MULTI-CORRELATION COEFFICIENT			.99645		
STD. ERROR OF ESTIMATE			.26060		
ESTIMATED PROPORTIONALITY FACTOR FOR ERROR VARIANCE				.00040671	

CASE NO.	ERROR VARIANCE (ESTIMATE)
1	.06354863
2	.03748251
3	.04743880
4	.07971540
5	.09396656
6	.06354863
7	.01593804
8	.06354863
9	.03662845
10	.17259197
11	.03647423
12	.03655130
13	.03686040
14	.03540376
15	.00005568
16	.00005568
17	.00006507
18	.00006507
19	.00012754
20	.00012754
21	.00189755
22	.00189755

COVARIANCE MATRIX OF REGRESSION COEFFICIENTS :

.1195E-04	0.	0.
0.	.2213E-01	-.1912E-01
0.	-.1912E-01	.2125E-01

DO YOU WANT THE TABLE OF RESIDUALS ?Y

CASE NO.	Y VALUE	Y ESTIMATE	RESIDUAL
1	12.50000	13.01702	-.51702
2	9.60000	9.68369	-.08369
3	10.80000	10.94008	-.14008
4	14.00000	14.34903	-.34903
5	15.20000	14.32003	.87997
6	12.50000	12.55940	-.05940
7	6.26000	6.19474	.06526
8	12.45000	12.50089	-.05089
9	9.49000	9.48913	.00087
10	20.60000	20.35679	.24321
11	9.47000	9.46936	.00064
12	9.48000	9.59635	-.11635
13	9.52000	9.42256	.09744
14	9.33000	9.39492	-.06492
15	.37000	.36846	.00154
16	.37000	.37389	-.00389
17	.40000	.40022	-.00022
18	.40000	.40022	-.00022
19	.56000	.56673	-.00673
20	.56000	.55164	.00836
21	2.16000	2.11869	.04131
22	2.16000	2.11976	.04024

DO YOU WISH TO SAVE Y-ESTIMATES AND RESIDUALS?N

DO YOU WANT THE ANALYSIS OF VARIANCE FOR THE MODEL ?Y

```

      ANALYSIS OF VARIANCE TABLE
SOURCE      SUM OF      DEGREES OF      MEAN      F-RATIO
            SQUARES    FREEDOM      SQUARES
REG         733.8321         2         366.9161    5402.7762
RES          1.2903         19          .0679
TTL         735.1224         21         35.0058
R**2 (REG/TTL)=          .99824

```

```

-----
WANT THE 95% CONFIDENCE INTERVALS FOR Y ESTIMATE
AND THE PARAMETERS ?Y
-----

```

95% CONFIDENCE INTERVALS FOR Y AND Y*

```

      Y = B0 + B1*(X1-X1BAR) + B2*(X2-X2BAR) + ...
      Y* = B0 + B1*(X1-X1BAR) + B2*(X2-X2BAR) + ... + E
      (E IS THE ERROR ON Y)

```

Y	Y		Y*	
ESTIMATE	LOWER LIMIT	UPPER LIMIT	LOWER LIMIT	UPPER LIMIT
13.01702	12.86993	13.16412	12.46934	13.56471
9.68369	9.57558	9.79180	9.26434	10.10303
10.94008	10.82291	11.05725	10.46944	11.41071
14.34903	14.15190	14.54615	13.72614	14.97191
14.32003	14.11739	14.52268	13.64727	14.99280
12.55940	12.41609	12.70271	12.01272	13.10608
6.19474	6.10695	6.28254	5.91633	6.47315
12.50089	12.35416	12.64763	11.95330	13.04849
9.48913	9.38512	9.59314	9.07532	9.90294
20.35679	19.64798	21.06560	19.23505	21.47854
9.46936	9.36390	9.57482	9.05600	9.88272
9.59635	9.48583	9.70688	9.18126	10.01144
9.42256	9.31568	9.52944	9.00679	9.83832
9.39492	9.29104	9.49880	8.98767	9.80216
.36846	.36104	.37588	.35117	.38575
.37389	.36648	.38129	.35660	.39117
.40022	.39290	.40753	.38182	.41861
.40022	.39290	.40753	.38182	.41861
.56673	.55914	.57433	.54191	.59156
.55164	.54412	.55917	.52684	.57645
2.11869	2.08576	2.15162	2.02177	2.21562
2.11976	2.08655	2.15296	2.02273	2.21678

95% CONFIDENCE INTERVALS FOR PARAMETERS

	LOWER LIMIT	UPPER LIMIT
B 0	.46897	.48344
B 1	13.68829	14.31099
B 2	-1.27159	-.66148

```

-----
REGRESSIONS      (HELP=99) : 2

```

INSTRUCTION Y/N?Y

ENTERING DATA IDENTIFY THE TYPE OF LINEAR REGRESSION

IDENTIFY THE DEPENDANT VARIABLE

IDENTIFY THE INDEPENDANT VARIABLE

IDENTIFY THE REPEAT DATA

IDENTIFY THE VARIANCE OF X AND Y

IDENTIFY THE LEVEL OF SIGNIFICANCE

RESULTS 1 SCREEN : STATISTICS ON PARAMETERS

2 SCREEN : ANOVA TABLE

3 SCREEN : LACK OF FIT TEST

4 SCREEN : COVARIANCE MATRIX AND INTERVALS SLOPE AND INTERCEPT

5 SCREEN : INTERVALS ESTIMATED X AND Y

SIM. LIN. REG. (HELP=99) : 99

SIMPLE LINEAR REGRESSION

 1 = X ON Y
 2 = Y ON X
 3 = WEIGHTED X AND Y
 4 = ORTHOGONAL
 99 = HELP
 0 = EXIT

SIM. LIN. REG. (HELP=99) : 4

ENTER NAME (DEP) : LRYX

ENTER NAME (IND) : LRXX

DO YOU KNOW THE STANDARD DEVIATION ON EACH X, Y/N ? : N

DO YOU KNOW THE STANDARD DEVIATION ON EACH Y, Y/N ? : N

AT WHICH LEVEL OF SIGNIFICANCE DO YOU WANT TO WORK?

1: 68%

2: 86.6%

3: 95%

4: 99%

5: 99.7%

(ENTER 1,2,3,4 OR 5)

3

SPOCLR RESULTS

VARIABLE	MEAN	STANDARD DEVIATION	COVARIANCE X VS Y
X	.6000E+01	.2828E+01	.7520E+01
Y	.4844E+01	.2665E+01	
T (NOBV-2) 95.0%:	3	TDIS (TWO-TAILS)	.3080E+01
EQUATION: Y = A + B * X			
SLOPE (B)		.9421E+00	
INTERCEPT (A)		-.8086E+00	
CORRELATION COEFFICIENT (R)		.9976E+00	
STD. ERROR OF ESTIMATE		.1543E+00	
WEIGHTING FACTOR (RESIDUALS SLOPE)		-.1061E+01	

C TO CONTINUE : C

ANALYSIS OF VARIANCE TABLE

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARES	F-RATIO
REG	.149955E+02	1	.149955E+02	.629644E+03
RES	.714476E-01	3	.238159E-01	
TTL	.150670E+02	4	.376674E+01	
R**2=	.9953E+00			

F (1, 3) 95.0%: .9486E+01

EXAMINE RESIDUALS Y-EST Y/N?Y

CASE	OBSERVED		ESTIMATED		RESIDUAL
	X	Y	X	Y	
1	.20000E+01	.91000E+00	.19174E+01	.99773E+00	-.12052E+00
2	.60000E+01	.50300E+01	.60928E+01	.49315E+01	.13538E+00
3	.60000E+01	.48100E+01	.59830E+01	.48280E+01	-.24747E-01
4	.60000E+01	.50400E+01	.60978E+01	.49362E+01	.14266E+00
5	.10000E+02	.84300E+01	.99090E+01	.85266E+01	-.13277E+00

SAVE RESIDUALS Y/N?N

COVARIANCE MATRIX OF

REGRESSION COEFFICIENTS

.280973E-01	-.30348E-02
-.303481E-02	.39470E-03

95.0% CONFIDENCE INTERVALS FOR

SLOPE AND INTERCEPT

	LOWER LIMIT	UPPER LIMIT
SLOPE	.83290E+00	.10647E+01
INTERCEPT	-.15179E+01	-.99327E-01

C TO CONTINUE : C

```

      95.0% CONFIDENCE INTERVALS FOR ESTIMATED X AND Y
CASE      LOWER LIMIT      UPPER LIMIT
           X      Y      X      Y
  1      .14897E+01  .54381E+00  .23450E+01  .14516E+01
  2      .57358E+01  .45524E+01  .64499E+01  .53105E+01
  3      .56260E+01  .44490E+01  .63401E+01  .52070E+01
  4      .57407E+01  .45571E+01  .64549E+01  .53152E+01
  5      .94868E+01  .80785E+01  .10331E+02  .89748E+01
C TO CONTINUE : C
SIM. LIN. REG.      (HELP=99) : 0
REGRESSIONS      (HELP=99) : 3
INSTRUCTION Y/N?Y
ENTERING DATA      IDENTIFY THE VARIABLES
                     ENTER STOP TO FINISH
                     IDENTIFY THE CONSTANT TO LIMIT VARIABLES
                     (ENTER 0 IF ALL VARIABLES TO BE USED)
                     IDENTIFY THE TYPE OF EACH VARIABLES
RESULTS      1 SCREEN : VARIABLES STATISTICS
              2 SCREEN : ANOVA TABLE OF EACH STEP
              3 SCREEN : STATISTICS ON PARAMETERS OF EACH STEP
ENTER "STOP" FOR FINISH
ENTER NAME ( DEP ) : STP6
ENTER NAME ( IND ) : STP1
ENTER NAME ( IND ) : STP2
ENTER NAME ( IND ) : STP3
ENTER NAME ( IND ) : STP4
ENTER NAME ( IND ) : STP5
ENTER NAME ( IND ) : STOP
ENTER THE CONSTANT TO LIMIT VARIABLES : 0
                     STEP-WISE MULTIPLE REGRESSION
CONSTANT TO LIMIT VARIABLES 0.00000
VARIABLE      MEAN      STANDARD
NO.      DEVIATION
STP6      .2266667E+01  .1412587E+01
STP1      .4313333E+02  .6521758E+01
STP2      .3161667E+03  .1144299E+03
STP3      .2418000E+03  .3643076E+02
STP4      .1056667E+03  .1785639E+02
STP5      .3413333E+02  .1597570E+02
CHOOSE A TYPE FOR EACH VARIABLE
  0 = INDEPENDENT AVAILABLE
  1 = INDEPENDENT FORCED
  2 = VARIABLE TO BE DELETED
  3 = DEPENDENT VARIABLE
3 0 0 0 0 0
DEPENDENT VARIABLE.....STP6
NUMBER OF VARIABLES FORCED.... 0
NUMBER OF VARIABLES DELETED... 0
C TO CONTINUE : C
STEP 1
VARIABLE ENTERED.....STP2
SUM OF SQUARES REDUCED IN THIS STEP.... .102998E+02
PROPORTION REDUCED IN THIS STEP..... .177992E+00
CUMULATIVE SUM OF SQUARES REDUCED..... .102998E+02
CUMULATIVE PROPORTION REDUCED..... .177992E+00 OF .578667E+02
FOR 1 VARIABLES ENTERED
  MULTIPLE CORRELATION COEFFICIENT... .421891E+00
  (ADJUSTED FOR D.F.)..... .421891E+00
  F-VALUE FOR ANALYSIS OF VARIANCE... .606293E+01
  STANDARD ERROR OF ESTIMATE..... .130339E+01
  (ADJUSTED FOR D.F.)..... .130339E+01
C TO CONTINUE : C

```

VARIABLE NUMBER	REGRESSION COEFFICIENT	STD. ERROR OF REG. COEFF.	COMPUTED T-VALUE
3	.520806E-02	.211512E-02	.246230E+01
INTERCEPT	.620053E+00		

C TO CONTINUE : C

STEP 2

VARIABLE ENTERED.....STP5

SUM OF SQUARES REDUCED IN THIS STEP....	.133241E+02		
PROPORTION REDUCED IN THIS STEP.....	.230255E+00		
CUMULATIVE SUM OF SQUARES REDUCED.....	.236239E+02		
CUMULATIVE PROPORTION REDUCED.....	.408247E+00	OF	.578667E+02

FOR 2 VARIABLES ENTERED

MULTIPLE CORRELATION COEFFICIENT...	.638942E+00		
(ADJUSTED FOR D.F.).....	.622184E+00		
F-VALUE FOR ANALYSIS OF VARIANCE...	.931357E+01		
STANDARD ERROR OF ESTIMATE.....	.112617E+01		
(ADJUSTED FOR D.F.).....	.114610E+01		

C TO CONTINUE : C

VARIABLE NUMBER	REGRESSION COEFFICIENT	STD. ERROR OF REG. COEFF.	COMPUTED T-VALUE
3	.631573E-02	.185921E-02	.339701E+01
6	.431642E-01	.133170E-01	.324128E+01
INTERCEPT	-.120349E+01		

C TO CONTINUE : C

STEP 3

VARIABLE ENTERED.....STP3

SUM OF SQUARES REDUCED IN THIS STEP....	.757218E+01		
PROPORTION REDUCED IN THIS STEP.....	.130856E+00		
CUMULATIVE SUM OF SQUARES REDUCED.....	.311961E+02		
CUMULATIVE PROPORTION REDUCED.....	.539103E+00	OF	.578667E+02

FOR 3 VARIABLES ENTERED

MULTIPLE CORRELATION COEFFICIENT...	.734236E+00		
(ADJUSTED FOR D.F.).....	.710607E+00		
F-VALUE FOR ANALYSIS OF VARIANCE...	.101372E+02		
STANDARD ERROR OF ESTIMATE.....	.101281E+01		
(ADJUSTED FOR D.F.).....	.104966E+01		

C TO CONTINUE : C

VARIABLE NUMBER	REGRESSION COEFFICIENT	STD. ERROR OF REG. COEFF.	COMPUTED T-VALUE
3	.743547E-02	.172211E-02	.431765E+01
6	.536255E-01	.125803E-01	.426264E+01
4	.149740E-01	.551135E-02	.271694E+01
INTERCEPT	-.553532E+01		

C TO CONTINUE : C

STEP 4

VARIABLE ENTERED.....STP1

SUM OF SQUARES REDUCED IN THIS STEP....	.127302E+00		
PROPORTION REDUCED IN THIS STEP.....	.219991E-02		
CUMULATIVE SUM OF SQUARES REDUCED.....	.313234E+02		
CUMULATIVE PROPORTION REDUCED.....	.541302E+00	OF	.578667E+02

FOR 4 VARIABLES ENTERED

MULTIPLE CORRELATION COEFFICIENT...	.735733E+00		
(ADJUSTED FOR D.F.).....	.698839E+00		
F-VALUE FOR ANALYSIS OF VARIANCE...	.737554E+01		
STANDARD ERROR OF ESTIMATE.....	.103040E+01		
(ADJUSTED FOR D.F.).....	.108823E+01		

C TO CONTINUE : C

VARIABLE NUMBER	REGRESSION COEFFICIENT	STD. ERROR OF REG. COEFF.	COMPUTED T-VALUE
3	.740639E-02	.175403E-02	.422249E+01
6	.507639E-01	.152350E-01	.333205E+01
4	.149279E-01	.560864E-02	.266159E+01
2	.122620E-01	.354122E-01	.346266E+00
INTERCEPT	-.594620E+01		

C TO CONTINUE : C
 STEP 5
 VARIABLE ENTERED.....STP4
 SUM OF SQUARES REDUCED IN THIS STEP.... .186068E-02
 PROPORTION REDUCED IN THIS STEP..... .321546E-04
 CUMULATIVE SUM OF SQUARES REDUCED..... .313252E+02
 CUMULATIVE PROPORTION REDUCED..... .541335E+00 OF .578667E+02
 FOR 5 VARIABLES ENTERED
 MULTIPLE CORRELATION COEFFICIENT... .735754E+00
 (ADJUSTED FOR D.F.)..... .684067E+00
 F-VALUE FOR ANALYSIS OF VARIANCE... .566515E+01
 STANDARD ERROR OF ESTIMATE..... .105161E+01
 (ADJUSTED FOR D.F.)..... .113262E+01

C TO CONTINUE : C

VARIABLE NUMBER	REGRESSION COEFFICIENT	STD. ERROR OF REG. COEFF.	COMPUTED T-VALUE
3	.738532E-02	.186241E-02	.396546E+01
6	.491895E-01	.414116E-01	.118782E+01
4	.150406E-01	.634940E-02	.236883E+01
2	.124215E-01	.363496E-01	.341723E+00
5	.150902E-02	.367888E-01	.410185E-01

 INTERCEPT -.607939E+01

C TO CONTINUE : C
 EXAMINE RESIDUALS Y-EST Y/N?Y

NB. OBSERV.	Y ESTIMATED	RESIDUALS
1	.4809E+00	.5191E+00
2	.1777E+01	.2233E+00
3	.2146E+01	-.1459E+00
4	.8288E+00	-.8288E+00
5	.1905E+01	.9478E-01
6	.1521E+01	.4788E+00
7	.3464E+01	-.4645E+00
8	.2259E+01	-.2589E+00
9	.3803E+01	-.8026E+00
10	.1020E+01	-.1020E+01
11	.2497E+01	.1503E+01
12	.2001E+01	-.1001E+01
13	.2007E+01	-.1007E+01
14	.1153E+01	-.1531E+00
15	.2904E+01	.9555E-01

C TO CONTINUE : C

NB. OBSERV.	Y ESTIMATED	RESIDUALS
16	.1835E+01	.1647E+00
17	.2560E+01	.4400E+00
18	.3452E+01	.5477E+00
19	.3627E+01	.3734E+00
20	.2681E+01	.3193E+00
21	.3649E+01	.3511E+00
22	.1865E+01	.2135E+01
23	.2099E+01	-.1099E+01
24	.1972E+01	-.1972E+01
25	.1413E+01	.2587E+01
26	.1880E+01	-.8803E+00
27	.2276E+01	.7235E+00
28	.4511E+01	-.5108E+00
29	.3957E+01	.4254E-01
30	.4546E+00	-.4546E+00

SAVE RESIDUALS Y/N?N
 REGRESSIONS (HELP=99) : 4
 INSTRUCTION Y/N?Y

ENTERING DATA IDENTIFY THE DEPENDANT VARIABLE
 IDENTIFY THE INDEPENDANT VARIABLE
 IDENTIFY THE DEGREE OF POLINOMIAL
 RESULTS 1 SCREEN : STATISTICS ON PARAMETERS
 2 SCREEN : ANOVA TABLE

ENTER NAME (DEP) : POYY
 ENTER NAME (IND) : POXX
 ENTER DEGREE OF POLYNOMIAL : 5
 POLYNOMIAL REGRESSION OF DEGREE 1
 INTERCEPT -.1387619E+02
 REGRESSION COEFFICIENTS
 .8567857E+01

ANALYSIS OF VARIANCE FOR 1 DEGREE POLYNOMIAL				
SOURCE OF VARIATION	DEGREE OF FREEDOM	SUM OF SQUARES	MEAN SQUARE	F VALUE
DUE TO REGRESSION	1	20554.28929	20554.28929	135.5656
DEVIATION ABOUT REGRESSION	13	1971.04405	151.61877	
TOTAL	14	22525.33333		

POLYNOMIAL REGRESSION OF DEGREE 2
 INTERCEPT .1507253E+02
 REGRESSION COEFFICIENTS
 -.1649337E+01 .6385747E+00

ANALYSIS OF VARIANCE FOR 2 DEGREE POLYNOMIAL				
SOURCE OF VARIATION	DEGREE OF FREEDOM	SUM OF SQUARES	MEAN SQUARE	F VALUE
DUE TO REGRESSION	2	22236.50780	11118.25390	461.9364
DEVIATION ABOUT REGRESSION	12	288.82553	24.06879	
TOTAL	14	22525.33333		

POLYNOMIAL REGRESSION OF DEGREE 3
 INTERCEPT .1852381E+02
 REGRESSION COEFFICIENTS
 -.3885340E+01 .9769356E+00 -.1409837E-01

ANALYSIS OF VARIANCE FOR 3 DEGREE POLYNOMIAL				
SOURCE OF VARIATION	DEGREE OF FREEDOM	SUM OF SQUARES	MEAN SQUARE	F VALUE
DUE TO REGRESSION	3	22247.89365	7415.96455	294.0300
DEVIATION ABOUT REGRESSION	11	277.43969	25.22179	
TOTAL	14	22525.33333		

POLYNOMIAL REGRESSION OF DEGREE 4
 NO IMPROVEMENT
 POLYNOMIAL REGRESSION OF DEGREE 3

TABLE OF RESIDUALS				
OBSERVATION NO.	X VALUE	Y VALUE	Y ESTIMATE	RESIDUAL
1	1.00000	10.00000	15.60131	-5.60131
2	2.00000	16.00000	14.54809	1.45191
3	3.00000	20.00000	15.27956	4.72044
4	4.00000	23.00000	17.71113	5.28887
5	5.00000	25.00000	21.75821	3.24179
6	6.00000	26.00000	27.33621	-1.33621
7	7.00000	30.00000	34.36054	-4.36054
8	8.00000	36.00000	42.74661	-6.74661
9	9.00000	48.00000	52.40983	-4.40983
10	10.00000	62.00000	63.26560	-1.26560
11	11.00000	78.00000	75.22935	2.77065
12	12.00000	94.00000	88.21648	5.78352
13	13.00000	107.00000	102.14239	4.85761
14	14.00000	118.00000	116.92250	1.07750
15	15.00000	127.00000	132.47222	-5.47222

REGRESSIONS (HELP=99) : 0

STAMP (HELP=99) : 99

PRINCIPAL MODULES

1 = EDIT
2 = PLOTS
3 = STATS
4 = CORRELATION
5 = REGRESSIONS
6 = NON LINEAR OPTIMIZATION
99 = HELP
0 = STOP

STAMP (HELP=99) : 6

ENTER FUNCTION USING

V1 TO V5, K1 TO K9, Z1 TO Z9

E.G.: (K1*Z1+Z2*V2-V1)**2

VI ARE INDEXED VARIABLES FROM YOUR FILE

KI ARE REAL CONSTANT VALUES

ZI ARE SEARCH VARIABLES

YOU WILL BE PROMPTED TO IDENTIFY

VI, KI, ZI LATER ON.

(V1-Z1-(1.-Z1)*(EXP(Z2*V2/Z3)-1.)/(EXP(Z2*V2/Z3)+EXP(Z2)-2.))**2

ENTER NAMES OF INDEXED VARIABLES V

ENTER NAME (V1) : NLY

ENTER NAME (V2) : NLX

ENTER ESTIMATES FOR SEARCH VARIABLES Z

Z1 =.3

Z2 =1

Z3 =100

VALUE OF OBJECTIVE FUNCTION F = .29276191E+00

NEED A NEW GUESS (Y/N) ?N

VALUE OF OBJECTIVE FUNCTION F = .29276191E+00

VALUES OF FIRST ESTIMATES ARE :

Z1 = .30000000E+00

Z2 = .10000000E+01

Z3 = .10000000E+03

YOU MAY NOW USE ONE OF THE OPTIMIZATION ALGORITHMS

NONLIN OPTIONS

1 = EDITING FUNCTION
2 = POWELL METHOD
3 = SIMPLEX METHOD
4 = ROSENBROCK METHOD
99 = HELP
0 = EXIT

NONLIN (HELP=99) : 2

POWELL ALGORITHM

MAXIMUM NUMBER OF ITERATIONS = 40

PRINTING OPTION = 2

ESCALE FACTOR = 1.

TOLERANCES ARE : .01500000 .05000000 5.00000000

DO YOU ACCEPT THESE DEFAULT VALUES (Y/N)Y

POWELL-BOTM OPTIMIZATION ROUTINE

PARAMETERS

N = 3 MAXIT = 40 ESCALE = 1.00

INITIAL GUESSES (Z) ACCURACY REQUIRED FOR VARIABLES (E)

3.00000000E-01 1.50000000E-02

1.00000000E+00 5.00000000E-02

1.00000000E+02 5.00000000E+00

```

ITERATION      1          21 FUNCTION VALUES      F = .23253187E-01
.30000000E-01    .16367218E+01    .10098786E+03
ITERATION      2          32 FUNCTION VALUES      F = .12704995E-01
.12608660E+00    .23420329E+01    .11508342E+03
ITERATION      3          46 FUNCTION VALUES      F = .11064164E-01
.14465958E+00    .29022605E+01    .11736274E+03
ITERATION      4          55 FUNCTION VALUES      F = .11059149E-01
.14344050E+00    .28981023E+01    .11706335E+03
C TO CONTINUE : C
  END OF MINIMIZATION PROCESS
  OPTIMUM VALUE OF F = .11059149E-01
  VALUES OF THE VARIABLES
  Z(1) = .14344050E+00
  Z(2) = .28981023E+01
  Z(3) = .11706335E+03
C TO CONTINUE : C
NONLIN          (HELP=99) : 99
NONLIN OPTIONS
-----
  1 = EDITING FUNCTION
  2 = POWELL METHOD
  3 = SIMPLEX METHOD
  4 = ROSENBROCK METHOD
  99 = HELP
  0 = EXIT
NONLIN          (HELP=99) : 3
  SIMPLEX ALGORITHM
  MAXIMUM NUMBER OF ITERATIONS = 1000
  NUMBER OF SIGNIFICANT DIGITS = 5
  VARIANCE CONVERGENCE = 0.2
  STEP LENGTH MODIFIER = 1.0
DO YOU ACCEPT THESE DEFAULT VALUES (Y/N)Y
"Z" VARIABLES TO BE BOUNDED (Y/N) ?Y
THE BOUNDARY CODE VALUES ARE :
  0 = NO BOUNDS
  1 = UPPER BOUND ONLY
  2 = LOWER BOUND ONLY
  3 = BOTH UPPER AND LOWER BOUNDS
ENTER FOR EACH VARIABLE A BOUNDARY CODE : 0 1 3
UPPER BOUND FOR Z2 : 3
LOW. AND UP. BOUNDS FOR Z3 : 80 120
ITERATION      0          1 FUNCTION VALUES      F = .29276191E+00
.30000000E+00    .10000000E+01    .10000000E+03
ITERATION      10         21 FUNCTION VALUES      F = .24608148E-01
.12155350E+00    .34864540E+01    .10605281E+03
ITERATION      20         40 FUNCTION VALUES      F = .11468238E-01
.13501238E+00    .26704574E+01    .11441082E+03
ITERATION      30         59 FUNCTION VALUES      F = .11067242E-01
.14623668E+00    .29142227E+01    .11732727E+03
ITERATION      40         77 FUNCTION VALUES      F = .11057677E-01
.14383445E+00    .28979878E+01    .11705614E+03
ITERATION      41         80 FUNCTION VALUES      F = .11057558E-01
.14426217E+00    .29003973E+01    .11702954E+03
  NB. OF SIGNIFICANT DIGITS CLOSING CRITERION
  SIMPLEX RESTARTED CONVERGENCE 2.00000E-01 NB FUNC CALLS 82
ITERATION      41         83 FUNCTION VALUES      F = .11057395E-01
.14416217E+00    .29003973E+01    .11702954E+03
ITERATION      50        104 FUNCTION VALUES      F = .11757275E-01
.15441326E+00    .33082066E+01    .11780918E+03
ITERATION      60        123 FUNCTION VALUES      F = .11079626E-01
.14229632E+00    .28990032E+01    .11721387E+03

```

```

ITERATION 70          141 FUNCTION VALUES          F = .11058419E-01
.14343639E+00      .28859213E+01      .11700623E+03
ITERATION 72          146 FUNCTION VALUES          F = .11057505E-01
.14402208E+00      .29060216E+01      .11699885E+03
NB. OF SIGNIFICANT DIGITS CLOSING CRITERION
SIMPLEX RESTARTED CONVERGENCE 2.00000E-01 NB FUNC CALLS 147
ITERATION 72          148 FUNCTION VALUES          F = .11057450E-01
.14412208E+00      .29060216E+01      .11699885E+03
ITERATION 80          167 FUNCTION VALUES          F = .13044023E-01
.15015989E+00      .31731898E+01      .11308295E+03
ITERATION 90          185 FUNCTION VALUES          F = .11134380E-01
.14955742E+00      .30054079E+01      .11825511E+03
ITERATION 100         202 FUNCTION VALUES          F = .11060418E-01
.14376090E+00      .28928878E+01      .11717375E+03
ITERATION 110         219 FUNCTION VALUES          F = .11057387E-01
.14371400E+00      .28990531E+01      .11697222E+03
ITERATION 111         222 FUNCTION VALUES          F = .11057306E-01
.14391411E+00      .28994883E+01      .11701587E+03
NB. OF SIGNIFICANT DIGITS CLOSING CRITERION
C TO CONTINUE : C
END OF MINIMIZATION PROCESS
OPTIMUM VALUE OF F = .11057306E-01
VALUES OF THE VARIABLES
Z(1) = .14391411E+00
Z(2) = .28994883E+01
Z(3) = .11701587E+03
C TO CONTINUE : C
NONLIN (HELP=99) : 99
NONLIN OPTIONS
-----
1 = EDITING FUNCTION
2 = POWELL METHOD
3 = SIMPLEX METHOD
4 = ROSENBROCK METHOD
99 = HELP
0 = EXIT
NONLIN (HELP=99) : 4
ROSENBROCK ALGORITHM
MAX. NUMBER OF FUNCTION EVALUATIONS = 1000
MAX. NUMBER OF AXES ROTATIONS = 20
NUMBER OF SUCCESSIVE FAILURES BEFORE TERMINATION = 5
CODE OF STEP SIZE TO BE USED AFTER ROTATION = 1
TOLERANCE FOR FUNCTION BETWEEN 2 EVALUATIONS = 0.01
SCALING FACTOR FOR STEP SIZE INCREASE (.GE.1) = 1.1
SCALING FACTOR FOR STEP SIZE REDUCTION (.LE.1) = 0.9
INITIAL STEP SIZES FOR THE VARIABLES ARE :
.01500000 .05000000 5.00000000
DO YOU ACCEPT THESE DEFAULT VALUES (Y/N)N
ENTER 10 NEW VALUES : 1000 20 5 1 .0001 1.1 .9 .005 .01 1
ROSENBROCK MINIMIZATION PROCEDURE
PARAMETERS
MAXK = 1000 MKAT = 20 MCYC = 5 NSTEP = 1
ALPHA = 1.10 BETA = .90 EPSY = 1.0000E-04
STAGE NUMBER 1          84 FUNCTION VALUES          F = .14248758E-01
.90789046E-01      .22109994E+01      .11029741E+03
CLOSING CRITERION IS:
TOLERANCE ON FUNCTION MINIMUM VALUE
STAGE NUMBER 2          156 FUNCTION VALUES          F = .13315771E-01
.90620722E-01      .23315371E+01      .11057446E+03
C TO CONTINUE : C

```

```
END OF MINIMIZATION PROCESS
OPTIMUM VALUE OF F = .13315771E-01
VALUES OF THE VARIABLES
Z(1) = .90620722E-01
Z(2) = .23315371E+01
Z(3) = .11057446E+03
C TO CONTINUE : C
NONLIN (HELP=99) : 0
STAMP (HELP=99) : 0
```

APPENDIX B

EDITING FUNCTIONS FOR NON-LINEAR OPTIMIZATION

EDITING FUNCTIONS FOR NON-LINEAR OPTIMIZATION

Example 1

This first example uses data which illustrate the multiple linear regression option in the sample run presented in Appendix A. The objective function is the sum of the weighted squares of residuals:

$$F = \sum_{i=1} \frac{[\text{MLR3} - (a_0 + a_1 \text{MLR1} + a_2 \text{MLR2})]^2}{\text{MLRR}}$$

Since indexed variables are used, the program automatically performs the summation of all the following F_i values:

$$F_i = \frac{[\text{MLR3}(i) - [a_0 + a_1 \text{MLR1}(i) + a_2 \text{MLR2}(i)]]^2}{\text{MLRR}(i)}$$

The calculation of this function using the function-editing option of STAMP is shown on the following page.

PRINCIPAL MODULES

```

-----
1  = EDIT
2  = PLOTS
3  = STATS
4  = CORRELATION
5  = REGRESSIONS
6  = NON LINEAR OPTIMIZATION
99 = HELP
0  = STOP

STAMP          (HELP=99) : 6
ENTER FUNCTION USING
V1 TO V5, K1 TO K9, Z1 TO Z9
E.G.: (K1*Z1+Z2*V2-V1)**2
VI ARE INDEXED VARIABLES FROM YOUR FILE
KI ARE REAL CONSTANT VALUES
ZI ARE SEARCH VARIABLES
YOU WILL BE PROMPTED TO IDENTIFY
VI, KI, ZI LATER ON.
(V1-(Z1*V2+Z2*V3))**2/V4
ENTER NAMES OF INDEXED VARIABLES V
ENTER NAME ( V1 ) : MLR3
ENTER NAME ( V2 ) : MLR1
ENTER NAME ( V3 ) : MLR2
ENTER NAME ( V4 ) : MLRR
ENTER ESTIMATES FOR SEARCH VARIABLES Z
Z1 =.1
Z2 =.5
VALUE OF OBJECTIVE FUNCTION F =      .20673293E+02
NEED A NEW GUESS (Y/N) ?N
VALUE OF OBJECTIVE FUNCTION F =      .20673293E+02
VALUES OF FIRST ESTIMATES ARE :
      Z1 =      .10000000E+00
      Z2 =      .50000000E+00
YOU MAY NOW USE ONE OF THE OPTIMIZATION ALGORITHMS
NONLIN OPTIONS
-----
1  = EDITING FUNCTION
2  = POWELL METHOD
3  = SIMPLEX METHOD
4  = ROSENBROCK METHOD
99 = HELP
0  = EXIT

NONLIN          (HELP=99) : 0
STAMP           (HELP=99) : 0

```

Example 2

This second example illustrates the optimization of a function that does not use indexed variables:

$$F = -3803.84 - 138.08 X_1 - 232.92 X_2 + 123.08 X_1^2 + 203.64 X_2^2 + 182.25 X_1 X_2$$

This function would use 67 characters if written in standard FORTRAN. However, using the real constant feature of the function-editing option of STAMP it requires only 41 characters. The programming of this function using the function-editing option of STAMP is shown on the following page.

PRINCIPAL MODULES

```

-----
1  = EDIT
2  = PLOTS
3  = STATS
4  = CORRELATION
5  = REGRESSIONS
6  = NON LINEAR OPTIMIZATION
99 = HELP
0  = STOP

```

STAMP (HELP=99) : 6

ENTER FUNCTION USING

V1 TO V5, K1 TO K9, Z1 TO Z9

E.G.: (K1*Z1+Z2*V2-V1)**2

V1 ARE INDEXED VARIABLES FROM YOUR FILE

K1 ARE REAL CONSTANT VALUES

Z1 ARE SEARCH VARIABLES

YOU WILL BE PROMTED TO IDENTIFY

V1, K1, Z1 LATER ON.

K1-K2*Z1+K3*Z2+K4*Z1*Z1+K5*Z2*Z2+K6*Z1*Z2

ENTER REAL CONSTANT K VALUES

K1 =-3803.84

K2 =138.08

K3 =-232.992

K4 =123.08

K5 =203.64

K6 =182.25

ENTER ESTIMATES FOR SEARCH VARIABLES Z

Z1 =.2

Z2 =.5

VALUE OF OBJECTIVE FUNCTION F = -.38738938E+04

NEED A NEW GUESS (Y/N) ?N

VALUE OF OBJECTIVE FUNCTION F = -.38738938E+04

VALUES OF FIRST ESTIMATES ARE :

Z1 = .20000000E+00

Z2 = .50000000E+00

YOU MAY NOW USE ONE OF THE OPTIMIZATION ALGORITHMS

NONLIN OPTIONS

```

-----
1  = EDITING FUNCTION
2  = POWELL METHOD
3  = SIMPLEX METHOD
4  = ROSENBROCK METHOD
99 = HELP
0  = EXIT

```

NONLIN (HELP=99) : 0

STAMP (HELP=99) : 0

Example 3

This example illustrates a more complex problem involving implicit FORTRAN functions and indexed variables. Based on the well-known model of Lynch, the program is used here to find the best values of the parameters of the efficiency curve of a hydrocyclone. Observed efficiency values and sieve sizes have been entered in the STAMP data files and named respectively NLY and NLX. The model equation is given here:

$$Y = Rf + (1 - Rf) \frac{e^{\frac{\alpha X}{d}} - 1}{e^{\frac{\alpha X}{d}} + e^{\alpha} - 2} \quad (1)$$

where Y is the predicted efficiency curve
X are the sieve sizes
Rf, α and d are the parameters to be found.

The solution is obtained by minimizing F (the sum of squares of residuals) with respect to Rf, α and d (the search variables):

$$F = \sum_i \left[NLY_i - Rf - (1 - Rf) \frac{e^{\frac{\alpha NLX_i}{d}} - 1}{e^{\frac{\alpha NLX_i}{d}} + e^{\alpha} - 2} \right]^2 \quad (2)$$

Since one might need the values of the efficiency curve predicted by the model, the CREATE option of the EDIT module of STAMP is used for this purpose.

The independent index variable is defined as such, but the previous search variables are here defined as real parameters using the values obtained by the optimization algorithm. The dependent index variable, of course, is not used any longer.

The edited function is:

$$Rf + (1 - Rf) \frac{e^{\frac{\alpha NLX}{d}} - 1}{e^{\frac{\alpha NLX}{d}} + e^{\alpha} - 2} \quad (6)$$

The resultant indexed variable is the efficiency curve predicted by the model. It may be given a name and compared to the observed efficiency curve using the EXAMINE option of the EDIT module.

The printout of the edition of F-function; the calculation of predicted efficiency values and the comparison between the predicted and observed values are shown on the following page.

PRINCIPAL MODULES

```

-----
1  = EDIT
2  = PLOTS
3  = STATS
4  = CORRELATION
5  = REGRESSIONS
6  = NON LINEAR OPTIMIZATION
99 = HELP
0  = STOP

STAMP                (HELP=99) : 6
ENTER FUNCTION USING
V1 TO V5, K1 TO K9, Z1 TO Z9
E.G.: (K1*Z1+Z2*V2-V1)**2
VI ARE INDEXED VARIABLES FROM YOUR FILE
KI ARE REAL CONSTANT VALUES
ZI ARE SEARCH VARIABLES
YOU WILL BE PROMPTED TO IDENTIFY
VI, KI, ZI LATER ON.
(V1-Z1-(1.-Z1)*(EXP(Z2*V2/Z3)-1.)/(EXP(Z2*V2/Z3)+EXP(Z2)-2.))**2
ENTER NAMES OF INDEXED VARIABLES V
ENTER NAME ( V1 ) : NLV
ENTER NAME ( V2 ) : NLX
ENTER ESTIMATES FOR SEARCH VARIABLES Z
Z1 =.3
Z2 =1
Z3 =100
VALUE OF OBJECTIVE FUNCTION F = .29276191E+00
NEED A NEW GUESS (Y/N) ?N
VALUE OF OBJECTIVE FUNCTION F = .29276191E+00
VALUES OF FIRST ESTIMATES ARE :
      Z1 = .30000000E+00
      Z2 = .10000000E+01
      Z3 = .10000000E+03
YOU MAY NOW USE ONE OF THE OPTIMIZATION ALGORITHMS
NONLIN OPTIONS

```

```

-----
1  = EDITING FUNCTION
2  = POWELL METHOD
3  = SIMPLEX METHOD
4  = ROSENBROCK METHOD
99 = HELP
0  = EXIT

NONLIN                (HELP=99) : 0
STAMP                (HELP=99) : 99
PRINCIPAL MODULES

```

```

-----
1  = EDIT
2  = PLOTS
3  = STATS
4  = CORRELATION
5  = REGRESSIONS
6  = NON LINEAR OPTIMIZATION
99 = HELP
0  = STOP

STAMP                (HELP=99) : 1
EDIT                (HELP=99) : 2
CREATE A VAR        (HELP=99) : 2
ENTER STOP TO EXIT
ENTER NAME ( IND ) : NLX

```

```

ENTER NAME ( IND ) : STOP
ENTER O TO EXIT
ENTER A PARAMETER : .14344
ENTER A PARAMETER : 2.8981
ENTER A PARAMETER : 117.06
ENTER A PARAMETER : 0
      V1 = NLX
      K1 = .14344E+00
      K2 = .28981E+01
      K3 = .11706E+03
ENTER FUNCTION
K1+(1.-K1)*(EXP(K2*V1/K3)-1.)/(EXP(K2*V1/K3)+EXP(K2)-2.)
.172E+00 .188E+00 .212E+00 .261E+00 .344E+00 .504E+00 .734E+00
.926E+00 .992E+00 .100E+01
INSERT VARIABLE IN FILE Y/N?Y
ENTER A NAME FOR THE VARIABLE (4 CHARACTERS): NLY*
ENTER A DESCRIPTOR (16 CHAR.) FOR THIS NAME :NLY*
DESCRIPTOR : PRED. EFF. CURVE
CREATE A VAR (HELP=99) : 0
EDIT (HELP=99) : 4
EXAMINE VAR (HELP=99) : 2
HOW MANY VARIABLES WOULD YOU LIKE TO LIST (MAX = 5, STOP = 0)?3
ENTER NAME ( ) : NLX
ENTER NAME ( ) : NLY
ENTER NAME ( ) : NLY*
SEQUENCE NLX NLY NLY*
NUMBER ----
1 .190E+02 .200E+00 .172E+00
2 .270E+02 .220E+00 .188E+00
3 .370E+02 .180E+00 .212E+00
4 .530E+02 .240E+00 .261E+00
5 .740E+02 .300E+00 .344E+00
6 .105E+03 .550E+00 .504E+00
7 .148E+03 .750E+00 .734E+00
8 .210E+03 .880E+00 .926E+00
9 .305E+03 .960E+00 .992E+00
10 .420E+03 .980E+00 .100E+01
EXAMINE VAR (HELP=99) : 0
EDIT (HELP=99) : 0
STAMP (HELP=99) : 0

```

APPENDIX C

SAMPLE OF A CONTROL CARD DECK

```

.PROC,STAMP84,IDY=#ID,$UMODE$=INTER,ZPFN,ZID.
.* *****
.*
.*      STAMP84 PROCEDURE:
.*      IDY      = DATA FILES IDENT
.*      UMODE    = BATCH : DATA ARE TO BE EDITED IN BATCH MODE
.*              CLEAN : CLEAN UP DATA FILES PRIOR EXECUTION
.*      ZPFN     = PFN OF THE BATCH DATA FILE WHEN UMODE = BATCH
.*      ZID      = ID OF THE BATCH DATA FILE WHEN UMODE = BATCH
.*
DAYFILE(OFF)
.*
.*      ATTACH DATA FILES
.*
ATTACH,TAPE25,IDY_STAMP,ID=INDEX.
ATTACH,TAPE26,IDY_STAMP,ID=DATA.
.*
.*      JUST IN CASE... TAKE COPIES|
.*
REQUEST,D1,*PF.
REQUEST,D2,*PF.
COPY,TAPE25,D1.
COPY,TAPE26,D2.
REWIND,D1,D2,TAPE25,TAPE26.
CATCYC,D1,IDY_SAVESTAMP,ID=INDEX,RP=999.
CATCYC,D2,IDY_SAVESTAMP,ID=DATA,RP=999.
SKIP,OK.
EXIT.
.*
.*      DATA FILES DO NOT EXIST YET. USE TESTSTAMP FILES TO INITIALIZE THEM
.*
ATTACH,TAPE35,TESTSTAMP,ID=INDEX,MR=1.
ATTACH,TAPE36,TESTSTAMP,ID=DATA,MR=1.
REWIND,TAPE35,TAPE36.
REQUEST,TAPE25,*PF.
REQUEST,TAPE26,*PF.
COPYRMS,TAPE35,TAPE25.
COPYRMS,TAPE36,TAPE26.
REWIND,TAPE25,TAPE26.
CATALOG,TAPE25,IDY_STAMP,ID=INDEX,RP=60.
CATALOG,TAPE26,IDY_STAMP,ID=DATA,RP=60.
RETURN,TAPE35,TAPE36,TXT.
ENDIF,OK.
IFE,$UMODE$.NE.$INTER$,DOINTER.
ATTACH,LIB,STAMP84,ID=EDTLIB.
LIBRARY(LIB)
IFE,$UMODE$.NE.$CLEAN$,DOCLEAN.
REMARK.
REMARK.      DATA ARE EDITED IN BATCH MODE
REMARK.
ATTACH,TAPE10,ZPFN,ID=ZID.
ADSTAMP.
LIBRARY.
RETURN,TAPE10,LIB.
SKIP,DOINTER.
ENDIF,DOCLEAN.

```

```

REMARK.
REMARK.      CLEANING UP STAMP DATA FILES
REMARK.
REQUEST,TAPE35,*PF.
REQUEST,TAPE36,*PF.
CLEANF.
RETURN,TAPE25,TAPE26.
BATCH,TAPE35,RENAME,TAPE25.
BATCH,TAPE36,RENAME,TAPE26.
CATCYC,TAPE25,IDY_STAMP,ID=INDEX,RP=999,KP=2.
CATCYC,TAPE26,IDY_STAMP,ID=DATA,RP=999,KP=2.
LIBRARY.
RETURN,LIB.
ENDIF,DOINTER.
REMARK.
REMARK.      STAMP EXECUTION STARTS NOW
REMARK.
ATTACH,PSTAMP,STAMP84,ID=SEGLIB,MR=1.
PSTAMP.
.*
.*      SAVE NEW FILES AND PURGE COPIES
.*
EXTEND,TAPE25.
EXTEND,TAPE26.
PURGE,D1.
PURGE,D2.
RETURN,TAPE25,TAPE26,PSTAMP,TAPE20,D1,D2.
DAYFILE(ON)
REVERT.
EXIT,U.
REMARK.
REMARK. ***
REMARK. ***   PROBLEM ENCOUNTERED DURING EXECUTION,
REMARK. ***   REFER TO DAYFILE FOR ERROR MESSAGES.
REMARK. ***   IF PROBLEMS ARISE IN ACCESSING IDY DATA FILES,
REMARK. ***   USE IDY_SAVE INSTEAD, AND PURGE OLD FILES.
REMARK. ***   UNDER THAT EVENT, LAST MODIFICATIONS TO DATA FILE
REMARK. ***   ARE UNFORTUNATELY LOST.
REMARK. ***
RETURN,D1,D2,TAPE25,TAPE26,PSTAMP,TAPE20.
COPDF,XX.
REWIND,XX.
COPY,XX.
RETURN,XX.
DAYFILE,ON.
REVERT.

```

```

.DATA,DATASEG.
EDF      TREE      EDFUNC-(TABLE2,EXTFNC,BRAKET)
ROSE     TREE      ROSENB-NBROCK
SPLX     TREE      SIMPLX-(BOUND,LUBND,SMPLX-(DISPL,FUNC))
POW      TREE      POWELL-BOTM
BRAN     TREE      COMPRS-(EDF,POW,SPLX,ROSE)
NLIN     TREE      NONLIN-FUNCT-BUILD-ERROR-FONCT2-BASICO-ARGUM2-BRAN
POLY     TREE      POLYRG-(ORDER,GDATA,MULTR)
POLYRG   INCLUDE   MINV
MREG7    INCLUDE   GMPRD
REG4A    TREE      LREG4A-LREG4H
REGSIM   TREE      SPOCLR-(REG4A,LREG4B,LREG4C,LREG4D,LREG4E,LREG4F,LREG4G)
MLR4     TREE      MREG4-(SMPY,TPRD,MINV,GMPRD)
MLR5     TREE      MREG5-(MATA,SSUB)
MLR6     TREE      MREG6-DCPY
MLR7     TREE      MREG7-GMTRA
MLRINT   TREE      INTMLR-LOC-(MREG1,EXPVAR,MLR4,MLR5,MLR6,MLR7)
STEP     TREE      STPRG-STOUT
REGSR    TREE      SR-(CORRE2,STEP)
SELREG   TREE      DOREG-(REGSIM,REGSR,MODEL,INSTR,MLRINT,POLY)
THESIS   TREE      PTHESIS-FONCT-ARGUM-OPERAT
EXP       TREE      BLDEXP-ERREUR-REPACK-(TABLE,EXTFTN,THESIS)
EDITE    TREE      EDIT-MODIFY-CREATE-INSERT-FRMEQU-EXP
ARBRE    TREE      STATS-(EDITE,PLOTS,DSTATS,CORRE,SELREG)
EXA      TREE      EXAVAR-LIST-LISDATA-PUTVAR-ARBRE
          TREE      STAMP-(FIXFIT,INTRO,GETVAR-HOLD-(NLIN,EXA))
          COMMON   IO,WORK,FILINFO,ANSWERS,V1V5,ARG,DIM
          END      STAMP

```

