No. 2035

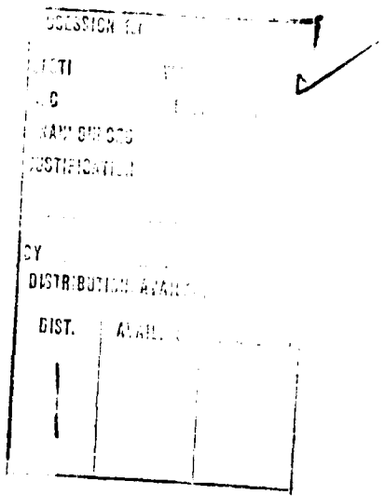
TECHNICAL REPORT

DA-MRCA: A FORTRAN IV PROGRAM FOR MULTIPLE LINEAR REGRESSION

by

K. Abt
G. Gemmill
T. Herring
R. Shade
Computation and Analysis Laboratory

U. S. NAVAL WEAPONS LABORATORY
DAHLGREN, VIRGINIA



U. S. Naval Weapons Laboratory
Dahlgren, Virginia

DA-MRCA: A FORTRAN IV PROGRAM FOR MULTIPLE LINEAR REGRESSION

by

K. Abt
G. Gemmill
T. Herring
R. Shade

Computation and Analysis Laboratory

NWL Report No. 2035

Foundational Research Project No. 29Y

March 1966

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

ABSTRACT

This report contains the documentation of a multiple linear regression program for up to 50 independent variables, written in FORTRAN IV for the IBM 7030 (STRETCH) computer. The program incorporates part of the results obtained from an effort to explore the present limitations of high speed computation in the area of linear statistical models. DA-MRCA includes options for both forward and backward automatic ranking of the independent variables by order of prediction power for the dependent variable. The report contains the description of these options, along with an outline of the applicability of the program which includes, in a convenient form, non-orthogonal analysis of variance. Justifications are given for extensive checks made on the accuracy of the matrix inversions. The resulting internal decisions and their effects on the computational flow are described in detail. Also, a failure analysis is given in which causes for failures to obtain acceptable inverses and possible consequences of corrective measures are discussed.

FOREWORD

The DA-MRCA program (Dahlgren Multiple Regression Comprehensive Analysis) documented in this report is partially based on the TV-MRCA program (Tennessee Valley Authority Multiple Regression Comprehensive Analysis) of the Tennessee Valley Authority. The TV-MRCA program became available to the authors through the SHARE Program Library. Although much larger in scope and applicability, DA-MRCA still contains some computational details from its nucleus routine, TV-MRCA. (In order to reflect this fact the initials "MRCA" have been retained for the present program.) TV-MRCA included, for a regression model containing up to 23 independent variables, the bases for the features described at the following places of the present report: Paragraph C of Section VI.2.a.(1); paragraphs A-F of Section VI.2.a.(2) (excluding all references to ANOVA tables, the final comprehensive analysis, IVOR, and BIVOR); paragraphs A, B, and I of Section VI.2.a.(3); and Section VI.2.a.(4) (excluding the option for selected input design points). These features were applicable, in TV-MRCA, to the main run and to hand selected reruns. The first additions to and revisions of the coding of the TV-MRCA program were performed by Mr. R. Scanlon, Mr. D. Green, and Mrs. Julia Gray, members of the former Scientific Programming and Analysis Branch, Computation Division.

The work reported was done in the Mathematical Statistics Branch, Operations Research Division, and the Operations Sciences Branch, Computer Programming Division, with Foundational Research Funds No. 29Y/R0110101/WR-6-7042 ("Computer Programs for Statistical Analyses").

The flow charts contained in the present documentation were drawn by Messrs. Thomas B. Yancey and John S. Darling and the report was typed by Miss Judy D. Merryman.

The work on this report was completed on 26 March 1966.

APPROVED FOR RELEASE:

Bernard Smith

BERNARD SMITH
Technical Director

I. INTRODUCTION

The need for a capable computer routine to solve extensive multiple regression problems in the application of statistical methods to naval ordnance research studies and other investigations at the Naval Weapons Laboratory led to the development of the present DA-MRCA program. Connected with this development was an effort to explore the present limitations of high speed computation in the area of linear statistical models. The program incorporates part of the results obtained from this research.

DA-MRCA has served, during all stages of its development, in the solution of actual statistical problems and, also, in research studies to develop more advanced and/or specialized computer routines (to be documented) for statistical analyses. After years of additions to and revisions of the program it is felt that DA-MRCA has reached a desired format and that its documentation is appropriate at this time.

The DA-MRCA program is written in FORTRAN IV for the IBM 7030 (STRETCH) computer and performs all the usual phases of a multiple linear regression analysis, that is, an analysis based upon the model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_\nu x_\nu + \cdots + \beta_N x_N + e \qquad (I\text{-}1)$$

where

y = "dependent" (random) variable

x_ν = "independent" (non-random) variables, $\nu = 1,\ldots,N$

β_ν = regression coefficients, $\nu = 1,\ldots,N$

β_0 = a constant

e = "residual", or "error" term: a random variable with expectation zero and variance σ^2, usually assumed to be normally distributed.

The upper limit for the number of independent variables to be included in the model is N=50. The main results of the analysis (based on a set of observed x and y values and obtained by the principle of least squares) are the estimates of the regression coefficients, β_ν, the constant, β_0, and the residual variance, σ^2, i.e., a prediction formula for the dependent variable and a measure of its accuracy. Furthermore, the following features are included in the program: Computation of predicted values of the dependent variable at selected

input design points and/or "synthetic" design points; computation of prediction standard deviations for the construction of confidence or tolerance limits at selected input design points and/or synthetic design points; a listing of the prediction errors, $\hat{e}$; a bar-chart and a Chi-square test on the normality of these errors; computation of the standard deviations of the regression coefficients; printout of the full inverse of the matrix of the normal equations; computation of various other pertinent statistics, an analysis-of-variance table, and a final comprehensive printout. For more details about these features see Chapter VI. (It should be noted that DA-MRCA is not capable of handling more than one dependent variable at a time. Neither can the program obtain weighted least squares solutions nor can it fit regression models through the origin.)

Since the theoretical aspects of the normal phases of multiple regression analysis form a well established part of mathematical statistics (see, for example, Anderson and Bancroft [1952]), these aspects need not be discussed in this report.

In addition to the "usual" features, the program has three options for the identification of the significant independent variables. These options are discussed in more detail in Chapter III. In the first option, the model is re-evaluated on the basis of a "hand" selected subset of $N' \leq N$ independent variables. This option can be used to test the null hypothesis on any specified subset of N-N' regression coefficients, β_ν. In the other two options the independent variables are automatically ranked by order of prediction power for the dependent variable. The first of these options employs the "IVOR" routine ("Independent Variable Ordering by Regression Sums of Squares"). This routine uses a forward or "build-up" technique to rank the independent variables in descending order of importance. The second ranking option employs "BIVOR" ("Backward Independent Variable Ordering by Regression Sums of Squares"). This routine uses a reverse ordering technique by which the independent variables are ranked in ascending order of importance. In Chapter III, it is shown that the disturbing effects of possibly existing "compounds" (to be defined) upon the ranking of the independent variables can be avoided only by application of the BIVOR technique. Therefore, the BIVOR option is recommended whenever feasible. There are, however, situations in which the IVOR technique has its advantages, as also discussed in Chapter III.

Essentially all of the "usual" features which were listed previously are also applied, or can optionally be applied, in the "reruns" of these three options for the identification of the significant independent variables.

Also built into the program are extensive checks on the accuracy of the computations. The elements of the calculated identity matrix are checked for their deviations from either 1 or 0, and internal decisions are made with respect to the acceptance of the matrix inversions according to accuracy requirements imposed by the program user. The details of these checks are discussed in Sections VI.1.b. and VI.2.

A preprocessor program for DA-MRCA, MTRAN, has been developed for possible transformations of observed x and y values if such are necessary. This program, however, is not described at length in the present report but is covered in a separate documentation (Herring [1966]). For a discussion of variable transformations, see Sections II.2. and VII.2.a.

The various chapters of this report are directed at different types of readers. Chapter II is mainly for the reader who wants to be informed about the possible applications of the program. No specialized statistical, mathematical or programming knowledge is required for understanding this chapter, except for Section II.3, where some knowledge of analysis of variance is necessary. (As in Chapter II, programming knowledge is not required for reading Chapters III through VII.) Chapter III is written mainly for the analyst seeking information about the theory, techniques, and use of the three model re-evaluation options of the program, especially IVOR and BIVOR. (These two procedures are introduced with this report.) Chapters IV and V define the terms used and explain the input preparation for the program, respectively, and are, therefore, essential for any program user. Chapter VI is written for the analyst who wants information on the computations and the meaning of the printouts. Program running time formulae and an example problem are also given in this chapter. Chapter VII can be of assistance to the program user in case of a failure to obtain a problem solution. Chapter VIII is written for the programmer and for the programming-oriented analyst. This chapter contains the FORTRAN IV documentation of DA-MRCA (including flow charts) and is essential for program changes and/or conversions.

The reader will notice some repetition in reading the report as a whole. However, the report is intended not only as a complete description of DA-MRCA, but also as a direct work aid in which case the program user would generally refer only to a specific chapter or section at a time. Each section contains all the necessary information, often given in the form of references to other sections.

II. APPLICABILITY OF THE PROGRAM

In this chapter the various types of problems to which the DA-MRCA program can be applied are discussed. Some general statements about the applicability are followed by sections on specific types of application.

II.1 General Applicability

The DA-MRCA program is applicable to all problems in which a preconceived linear mathematical model of the form

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_\nu x_\nu + \cdots + \beta_N x_N \qquad \text{(II-1)}$$

is to be evaluated on the basis of $n \geq N+1$ given sets of values, $\{y; x_1, x_2, \ldots, x_N\}$, by use of the principle of least squares. Essentially this evaluation consists of solving for the unknown coefficients, $\beta_\nu (\nu = 0, 1, \ldots, N)$ and attaching a measure of importance to the individual variables, x_ν, thereby characterizing their "prediction power" for Y. In the narrower sense of multiple linear regression ($n > N+1$) the n observations, y, of the "dependent" variable (random) are expressed in the terms of the multiple regression model (I-1),

$$y = Y + e = \beta_0 + \sum_{\nu=1}^{N} \beta_\nu x_\nu + e,$$

where the x_ν are the "independent variables" ($\nu = 1, \ldots, N$) and where e is a random variable with expectation zero and variance σ^2. (Note that the regression model (I-1) is obtained by merely adding the random variable e to the mathematical model (II-1).) Although e is usually assumed to be normally distributed, it does not have to be unless statistical hypotheses about the β_ν are to be tested, or confidence intervals are to be constructed.

The i^{th} set of observations, $\{y; x_1, x_2, \ldots, x_N\}_i$, is defined by the coordinates of the dependent variable and the N independent variables and is called the i^{th} "data point." The numerical data of a given regression problem is comprised of n such data points ($i = 1, \ldots, n$). The i^{th} set of coordinates of the N independent variables, $\{x_1, x_2, \ldots x_N\}_i$, is called the i^{th} "input design point." In general, there is no restriction concerning the relative position of the input design points except, naturally, in the case of linear dependencies in the matrix of the normal equations. (See Section VII.2.b.) For example, the design points do not have to define a complete rectangular grid in the N-dimensional space, a situation in which orthogonal polynomials are often used. The application of these does require such (orthogonal) grids.

The x_ν values, in the theory of multiple regression, are assumed to be non-random, that is, they are determined at the will of the experimenter. However, in a more general interpretation, they may also be values which have been measured, or observed, without appreciable error. Sometimes multiple regression is applied in such a broad sense that the only requirement for a given variable being used as an "independent" variable, is the assumption of a cause-effect relationship between the variable and the "dependent" variable, y. All errors originating from the "independent" variables x_ν are then attributed, by definition, to the variability of y, and the x_ν are again considered as non-random variables. According to the definition of the model (I-1), the y values for a given design point are assumed to be randomly and independently sampled from a distribution (usually normal) with expectation

$$Y = \beta_0 + \sum_{\nu=1}^{N} \beta_\nu x_\nu$$

and variance σ^2.

With the above, the general linear multiple regression problem, to which DA-MRCA is applicable, consists of fitting a least squares surface of the form (II-1) to n observations y_i at n input design points (not necessarily all distinct), where these points are located in the N-dimensional space defined by the N independent variables. Specifically, the program serves to identify those independent variables which explain a significant portion of the variability in the numerical values of y, or, in other words, which have significant prediction power for y. One possibility to arrive at this identification is by application of the automatic ranking procedures IVOR and/or BIVOR. IVOR and BIVOR each provide for the ranking of all N independent variables simultaneously, or for ranking independent variables within specified groups. A second possibility to identify the significant independent variables is to apply the option for "hand selecting" a specified subset of independent variables to be deleted from the original model, and then test the contribution of these deleted independent variables to the fit. Also possible is the computation of statistics necessary for the construction of confidence intervals for the true response values Y at the input design points and/or "synthetic" design points located within the original experimental space.

By definition, the least squares fit for the model (I-1) reduces to a "perfect fit" when the number $n_N (n_N \leq n)$ of distinct input design points in the N-dimensional space is equal to N+1. When $n_N = n (= N+1)$, i.e., when there is exactly one value y_i at each distinct design point (the surface being a perfect fit to each individual value y_i, $i=1,2,\ldots,n$), the fit is called a "zero-error perfect fit." This "non-statistical"

or "deterministic" use of multiple regression is also possible with DA-MRCA, as was implied in the statements about the model (II-1) at the beginning of this section. The application of the program in this case is discussed, in more detail, in Section II.4.

The linearity of the mathematical model (II-1) depends only on the linearity of the unknown parameters, i.e., on that of the β_ν's. The general linear model, consequently, can be conceived to be of various forms, each of which can be fitted by DA-MRCA. For example, each x_ν can be a (non-linear) function of one or more other variables. Some of the more common equations of linear form are discussed in Section II.2. There are also many equations that, although non-linear in their parameters, can be made linear by an appropriate transformation. The use of DA-MRCA in fitting this type of equation is also discussed in the next section (II.2).

In order to solve a regression problem a decision must be made as to which independent variables should be included in the model and in which functional form the chosen independent variables should be included in the model. Helpful in this decision may be theoretical considerations, previous experience with the variables, a plot of the data, or some other means. Of particular help can be the use of the program's ranking methods IVOR and BIVOR. These methods allow the analyst to start with a possibly very elaborate model (a polynomial, in general) in which all terms having in reality little or no prediction power for the dependent variable, y, will automatically be identified.

The use and application of IVOR and BIVOR are explained in detail, together with the discussion of the theory of these ranking procedures, in Section III.2. There it is shown that the BIVOR option should be used, whenever possible, for the automatic ranking of the N independent variables.

II.2 Specific Linear Models and Linearization

The most straightforward application of the general linear model (II-1),

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_\nu x_\nu + \cdots + \beta_N x_N,$$

occurs when all N variables, x_ν, represent the first powers of original observed independent variables. In the example case given in Section VI.5, where the dependence of y = Ballistic Limit (of projectile) upon Thickness and Hardness (of target plate) is analyzed, such a straightforward model would include only the two original independent variables, Thickness (x_1, say) and Hardness (x_2, say), and would, therefore, have the form:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 .$$

As indicated before, however, the x_ν can also represent functions of the form

$$x_\nu = f_\nu\{z_{\nu_1}, z_{\nu_2}, \ldots, z_{\nu_j}, \ldots\}, \qquad \text{(II-2)}$$

where these functions do not contain parameters to be estimated and where the z_{ν_j} are variables (assumed to be non-random) whose observed numerical values completely specify the numerical value of x_ν. The simplest example of such functions are the polynomial terms $x_\nu = z^\nu$ of a single original independent variable, z. A model containing only these terms would appear as

$$Y = \beta_0 + \beta_1 z + \beta_2 z^2 + \cdots + \beta_\nu z^\nu + \cdots + \beta_N z^N ,$$

that is, as the equation of an N^{th} degree polynomial in one variable. More generally, the x_ν can represent polynomial terms in several original independent variables, z_j. This implies the applicability of DA-MRCA in the important area of multivariate polynomial fitting with up to N=50 polynomial terms, including the linear terms. The data handling in this case is very simple because the numerical values of the polynomial terms can be automatically generated by the program. The program user merely specifies which polynomial terms are to be included in the model and writes as input only the numerical values of the original independent variables, z_j. From these, the values of the terms of higher than first order are automatically generated and internally used as input for the generation of the matrix of the normal equations. As is true for any type of independent variable, x_ν, the use of the options for hand selected reruns or for IVOR and/or BIVOR will provide the analyst with the desired information concerning the necessary degree of the polynomial needed in the fit. This enables the program user to maximize the "goodness of fit", provided that he starts with a polynomial equation of high enough degree in all original independent variables. IVOR and BIVOR will automatically rank the polynomial terms according to their prediction power for y and thus provide the analyst with a basis for choosing a "significant model." To illustrate this with the example of Section VI.5, the analyst might have assumed that the polynomial in $x_1 = z_1$ = Thickness and $x_2 = z_2$ = Hardness would not have to be of higher than the second degree in order to predict the Ballistic Limit, y, sufficiently well. Accordingly, he would enter the program with the model

$$Y = \beta_0 + \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_1^2 + \beta_4 z_1 z_2 + \beta_5 z_2^2 .$$

Numerical input would be (besides y) only $x_1=z_1$ and $x_2=z_2$, whereas $x_3=z_1^2$, $x_4=z_1z_2$, and $x_5=z_2^2$ would be generated by the program. The application of BIVOR, say, might yield as the "significant model" (using the symbols $\hat{Y}$ and b_ν for the estimated parameters):

$$\hat{Y} = b_0 + b_1 z_1 + b_4 z_2 .$$

Here, it is implied that BIVOR ranked the variables z_2, z_1^2, and z_2^2 as the least important ones and that their contribution to the fit was found to be nonsignificant according to a prechosen significance level.

As indicated before, both IVOR and BIVOR contain an option for grouping the independent variables such that the ranking process takes place within only one group at a time. (For more details see Sections VI.1.d and VI.1.e.) This grouping can be applied to the case of polynomial terms such that terms of equal degree, for example, will be ranked exclusively among themselves. The reader is referred to Section VII.2.a for an important application of this feature in connection with using transformed variables to increase the computational accuracy when fitting polynomials.

Although polynomial terms are the most frequently occurring type of functions, f_ν, in formula (II-2), functions other than polynomials can as well be represented by the x_ν. Examples are $x_\nu=z_{\nu_1}\sin(z_{\nu_2})$, $x_\nu=\sqrt{z_{\nu_1}z_{\nu_2}}$, $x_\nu=\log z_\nu$, etc. In particular, such functions will occur when linearization of the given (non-linear) model must be achieved by transformations.

Although the method of least squares may also be applied to non-linear models, the normal equations which result are non-linear in the parameters and generally must be solved by iterative methods. DA-MRCA is not capable of fitting such equations, but some of the non-linear equations can be evaluated after performing the appropriate transformation that leads to the necessary linear form. Suppose, for example, the analyst wishes to consider the non-linear equation

$$Y^* = \beta_0^*(z_1)^{\beta_1}$$

as the model. (The asterisks are used for distinction of the terms of the non-linear model from those of the linear model.) A simple transformation to either common or natural logarithms will result in the linear equation

$$\log Y^* = \log \beta_0^* + \beta_1 \log z_1$$

which is identical to the general linear model if one lets log $Y^*=Y$ of the linear model, log $\beta_0^*=\beta_0$ and log $z_1=x_1$. In this case, therefore, the logarithms of the values of both the dependent and independent variables must be used as input to the program. The resulting least squares equation can be retransformed into the original form by substituting the antilog of the estimated coefficient $\widehat{\log \beta_j^*}$ for b_j^* in the original equation as expressed in estimated terms:

$$\hat{Y}^* = b_0^* z_1^{b_1}.$$

Another example of a non-linear model that can be linearized by a logarithmic transformation is

$$Y^* = \beta_0^*(\beta_1^*)^{z_1}(\beta_2^*)^{z_2}.$$

This will lead to

$$\log Y^* = \log \beta_0^* + (\log \beta_1^*) z_1 + (\log \beta_2^*) z_2.$$

With $\log Y^* = Y$, $z_1 = x_1$, $z_2 = x_2$, in this case, the logarithms of only the values of the dependent variable have to be used as input.

It should be noted that, whenever a transformation is used to linearize an equation, it is the sum of squares of deviations on the transformed variables that is minimized and not the sum of squares on the original variables. This has consequences in the use of the results from DA-MRCA: point and interval estimation must be done based on the calculations for the transformed variables. Only after the predicted values and/or confidence limits have been computed, will they be re-transformed into the original scale of the non-linear model. As a result one obtains, for example, non-symmetric confidence limits about the $\hat{Y}$ values.

Often it is necessary to apply a transformation only on the dependent variable in order to achieve a normal (or near-normal) distribution for y as is desired in many cases. (The built-in Chi-square test on the normality of the residuals, $\hat{e}$, may give an indication for the necessity and type of such a transformation. See Section VI.1.c.) Another reason for transforming y only could be to stabilize the variance which might be a function of the coordinates, x_ν, of the design points. It is a known fact, however, that in many cases in which a transformation of the y values is appropriate for either of these two reasons, it is also necessary for the other one. In addition to this, experience has shown that when the experimental data indicates the necessity of a transformation for normalizing the y values and/or for stabilizing their variance, often this is the only transformation which also linearizes the functional relationship between Y and the x's. For example, in the model $Y^* = \beta_0^*(\beta_1^*)^{z_1}$, the

observations y* of the dependent variable will usually not be distributed normally, but the values of $y = \log y^* = \log \beta_0^* + (\log \beta_1^*) z_1 + e$ often will be.

Because of the importance of the various transformations it is repeated here that the preprocessing program MTRAN ("DA-MRCA Transformation", see Herring [1966]) is available for use in conjunction with DA-MRCA. This program can perform the following transformations on the values of the dependent variable, the independent variable(s), or on the values of both types of variables:

$\ln (A+x)$ *)

$\ln [B+\ln (C+x)]$ *)

$\sqrt{x}$

$\frac{1}{D+x}$ *)

$\mathrm{Sin}^{-1} x$

$2\ \mathrm{Sin}^{-1} \sqrt{x}$

$\sin x$

$\cos x$

$\frac{x}{E}$ *)

$\frac{x-\bar{x}}{R_x}$ **)

*) The constants A, B, C, D, E are to be specified by the analyst.

**) This transformation is only for the independent variables. The purpose is to increase the matrix inversion accuracy. For details see Section VII.2.a.

II.3 Non-Orthogonal Analysis of Variance and Covariance

DA-MRCA, being a program for general multiple linear regression, can naturally also be applied to analysis of variance and covariance models, in particular to data classifications with incomplete and/or unbalanced data (non-orthogonal ANOVA). For the general discussion of the multiple regression treatment of non-orthogonal analysis of variance, see Brownlee [1960].

As an example of the application of DA-MRCA to non-orthogonal analysis of variance, a 2x3 crossed classification with qualitative factors and with unequal (and non-proportional) cell numbers is treated.

The two factors of the example are denoted as $\mathcal{A}$ and $\mathcal{B}$, and the analysis of variance model is:

$$y_{\alpha\beta\rho} = Y_{\alpha\beta} + e_{\alpha\beta\rho} = \mu + a_\alpha + b_\beta + ab_{\alpha\beta} + e_{\alpha\beta\rho}.$$

The various terms have the following meaning:

$y_{\alpha\beta\rho}$ = ρth observation in cell "$\alpha\beta$" of the response variable (random), where

$$\rho = 1,\ldots,R_{\alpha\beta}$$
$$\alpha = 1,\ldots,A$$
$$\beta = 1,\ldots,B$$

with $R_{\alpha\beta}$ being the number of observations in cell "$\alpha\beta$" and with A and B being the numbers of levels in factors $\mathcal{A}$ and $\mathcal{B}$, respectively (A=2 and B=3 in the present example);

$Y_{\alpha\beta}$ = expected or true value of the response variable y in cell "$\alpha\beta$";

μ = general constant;

a_α = constant for level α of factor $\mathcal{A}$;

b_β = constant for level β of factor $\mathcal{B}$;

$ab_{\alpha\beta}$ = interaction constant for level combination $\alpha\beta$;

$e_{\alpha\beta\rho}$ = error term, assumed to be normally independently distributed with expectation zero and variance σ^2.

In the multiple regression approach to this case of only qualitative factors the model constants (in the example: μ, a_α, b_β, and $ab_{\alpha\beta}$) become the regression coefficients of auxiliary independent variables which take on only the values 1 and 0, as will be demonstrated

below. For the inversion of the matrix of the normal equations, linear restrictions have to be imposed on the estimates of the various sets of constants, reducing the number of constants in each set to the number of degrees of freedom available in each corresponding factorial effect. For example, there are A main effect constants a_α in factor $\mathcal{A}$, but only A-1 degrees of freedom are available in the main effect of $\mathcal{A}$. Since in non-orthogonal analysis of variance for qualitative factors, the estimates of only the contrasts between model constants are meaningful rather than the estimates of the constants themselves (see, for example, Graybill [1961], Chapter 13), the choice of the type of linear restrictions imposed on the estimates of the model constants is arbitrary. For the ease of computation, a good choice is to let the last constant in each set be equal to zero. Applied to the present example, this means:

$$\hat{a}_A = \hat{b}_B = \hat{ab}_{\alpha B} = \hat{ab}_{A\beta} = 0; \quad \alpha = 1,\ldots,A; \quad \beta = 1,\ldots,B.$$

The model of the example can be written (using the notation for the estimates which are in reality only to be found later by least squares):

$$\hat{Y}_{\alpha\beta} = \hat{\mu}\cdot x_0 + \hat{a}_1 x_1 + \hat{b}_1 x_2 + \hat{b}_2 x_3 + \hat{ab}_{11} x_4 + \hat{ab}_{12} x_5 .$$

In this equation, x_0 is a dummy variable always taking the value 1 and the x_i, $i=1,\ldots,5$, are the above mentioned auxiliary variables.

Each of the 6 cells then leads to an equation of the above form for each of the corresponding $R_{\alpha\beta}$ observations, giving altogether

$$\sum_{\alpha=1}^{2}\sum_{\beta=1}^{3} R_{\alpha\beta} = R..$$

input design points for the multiple regression approach:

$$\hat{Y}_{11} = \hat{\mu}\cdot 1 + \hat{a}_1\cdot 1 + \hat{b}_1\cdot 1 + \hat{b}_2\cdot 0 + \hat{ab}_{11}\cdot 1 + \hat{ab}_{12}\cdot 0$$

$$\hat{Y}_{12} = \hat{\mu}\cdot 1 + \hat{a}_1\cdot 1 + \hat{b}_1\cdot 0 + \hat{b}_2\cdot 1 + \hat{ab}_{11}\cdot 0 + \hat{ab}_{12}\cdot 1$$

$$\hat{Y}_{13} = \hat{\mu}\cdot 1 + \hat{a}_1\cdot 1 + \hat{b}_1\cdot 0 + \hat{b}_2\cdot 0 + \hat{ab}_{11}\cdot 0 + \hat{ab}_{12}\cdot 0$$

$$\hat{Y}_{21} = \hat{\mu}\cdot 1 + \hat{a}_2\cdot 0 + \hat{b}_1\cdot 1 + \hat{b}_2\cdot 0 + \hat{ab}_{11}\cdot 0 + \hat{ab}_{12}\cdot 0$$

$$\hat{Y}_{22} = \hat{\mu}\cdot 1 + \hat{a}_2\cdot 0 + \hat{b}_1\cdot 0 + \hat{b}_2\cdot 1 + \hat{ab}_{11}\cdot 0 + \hat{ab}_{12}\cdot 0$$

$$\hat{Y}_{23} = \hat{\mu}\cdot 1 + \hat{a}_2\cdot 0 + \hat{b}_1\cdot 0 + \hat{b}_2\cdot 0 + \hat{ab}_{11}\cdot 0 + \hat{ab}_{12}\cdot 0$$

In this example, the numerical values of the auxiliary "independent" variables associated with the interaction terms, x_4 and x_5, can be seen to be the products of the values of the auxiliary independent variables associated with the two appropriate main effects, x_1 and x_2, and x_1 and

$x_{\cdot\cdot}$, respectively. This "product rule" applies correspondingly also to all crossed classification models containing higher order interactions, which simplifies greatly the input writing for non-orthogonal analysis of variance and covariance for qualitative factors: only the 1's and 0's of the auxiliary variables for the main effects need be input. The numerical values of the interaction variables are generated by the program as products according to the specifications put on the appropriate control card. (For details, see Section V.2, Card Type 3.)

With the design matrix thus generated, the least squares procedure yields the model estimates, or "regression coefficients", $\hat{\mu}$, $\hat{a}_{\cdot}$, $\hat{b}_1$, $\hat{b}_{\cdot}$, $\hat{ab}_{11}$, and $\hat{ab}_{\cdot\cdot}$. Also, the sum of squares between cells or "total regression" sum of squares is given. By the hand re-evaluation option of DA-MRCA, null-hypotheses concerning the various factorial effects can be tested. However, it is not recommended to test a null hypothesis on the main effects $\mathcal{A}$ or $\mathcal{B}$ as long as the interaction $\mathcal{AB}$ is present in the model. The reason is that the additional regression sum of squares due to $\mathcal{A}$ or $\mathcal{B}$, or, more specifically, due to the auxiliary variables x_1, or $x_{\cdot}$ and $x_{\cdot}$, associated with $\mathcal{A}$ or $\mathcal{B}$, respectively, is dependent upon the arbitrary restrictions imposed on the model constants as long as $x_{\cdot}$ and $x_{\cdot}$ are present in the model. (See Scheffé [1959], p. 117.) The additional regression sums of squares due to $\mathcal{A}$ or $\mathcal{B}$ become independent of the arbitrary restrictions only when the auxiliary variables $x_{\cdot}$ and $x_{\cdot}$ of the interaction $\mathcal{AB}$ are deleted from the model. Therefore, the recommended sequence of testing in the present example is to first delete simultaneously $x_{\cdot}$ and $x_{\cdot}$ (thereby obtaining the additional regression sum of squares due to $\mathcal{AB}$), and then, to delete the independent variables associated with both $\mathcal{AB}$ and $\mathcal{A}$ or both $\mathcal{AB}$ and $\mathcal{B}$, provided the interaction $\mathcal{AB}$ is not significant. This type of procedure will be referred to as testing under "restricted admissibility", i.e., initially only $\mathcal{AB}$ is "admissible" for testing but $\mathcal{A}$ and $\mathcal{B}$ are not.

In order to illustrate the application of DA-MRCA to non-orthogonal analysis of covariance one merely would have to add covariates to the above ANOVA model of the 2x3 crossed classification example. The covariates become part of the model for all calculations and remain part of it during the testing of any specified null hypothesis concerning the factorial effects.

Since the DA-MRCA program can handle up to N=50 independent variables, 50 is also the upper limit for the number of degrees of freedom for factorial effects to be included in non-orthogonal analysis of variance models. In non-orthogonal analysis of covariance this upper limit of the degrees of freedom for factorial effects is reduced by the number of covariates included in the model.

Since, in general, individual factorial effects will have more than one degree of freedom, the automatic ranking procedures IVOR and BIVOR generally cannot be applied for the ranking by significance of factorial effects. In cases of only single-degree-of-freedom qualitative effects, however, this application is possible. For testing under "restricted admissibility" as discussed before, the single-degree-

of-freedom effects must be grouped, in DA-MRCA, according to their order i.e., main effects first, then 2-factor interactions, then 3-factor interactions, etc. Since the ranking is done within only one group at a time, this application of BIVOR (or IVOR) guarantees the restricted admissibility of the effects for testing, although in an overstrict manner. For example, in a 2x2x2 factorial classification, the one-degree of freedom effects would be grouped as follows. Group 1: $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$; Group 2: $\mathcal{AB}$, $\mathcal{AC}$, $\mathcal{BC}$; Group 3: $\mathcal{ABC}$. BIVOR would delete $\mathcal{ABC}$ first, then rank $\mathcal{AB}$, $\mathcal{AC}$, and $\mathcal{BC}$, and finally (after deletion of both the third and second group) rank $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$.

Note: Work is presently in progress on the documentation of NOVACOM, a FORTRAN IV program for "Non-Orthogonal Variance and Covariance Analysis by Multiple Regression" which is able to automatically rank multiple-degree-of-freedom factorial effects under restricted admissibility. NOVACOM is based on the ideas that were indicated in this section and, in addition, on some of the suggestions contained in Abt [1965].

II.4 "Non-Statistical" Applications of DA-MRCA

As already mentioned in Section 1 of this chapter, DA-MRCA also provides for the possibility of "zero-error perfect fits." These were defined to be "perfect fits" ($n_{\mathcal{A}}=N+1$) in which there is exactly one y value at each of the $n_{\mathcal{A}}$ distinct design points. Since in these cases the "error", or the residual variance, is zero, the essential element of statistics is absent. Consequently, there is no possibility to apply statistical tests or to perform interval estimation.

The least squares method degenerates to the solution of a system of N+1 linear equations of rank N+1, having as a solution the perfect fit. Such a zero-error perfect fit has one of its many applications as an interpolation formula. Since IVOR and BIVOR are independent of the existence of an error term, they both can be applied in the case where the pre-conceived model (i.e., the model with the N independent variables of the "main run") is a zero-error perfect fit. The subsequent independent variable selections by IVOR or BIVOR will give (least squares) interpolation fits of monotonically changing overall accuracy. From these the analyst can choose the model which satisfies his accuracy requirements with respect to the prediction of the original values of the response variable, Y. This technique is sometimes very useful when a closed expression of sufficient accuracy is to be found for the entries of a table of values.

III. THE IDENTIFICATION OF SIGNIFICANT INDEPENDENT VARIABLES

III.1 Testing A Specified Null Hypothesis by the Main Theorem

The testing of a linear hypothesis concerning the contribution of any specified subset of N-N' independent variables to the regression sum of squares due to N independent variables is made possible by a model re-evaluation option of the program. The test is based on what may be called the Main Theorem of Multiple Regression. The content of this theorem, see, for example, Anderson and Bancroft [1952], p. 172, is as follows:

In the general linear model (I-1),

$$y = \sum_{i=0}^{N} \beta_i x_i + e,$$

the residuals, e, are assumed to be normally independently distributed with expectation zero and variance σ^2. Then, under $H_0\{\beta_{i_1} = \beta_{i_2} = \cdots = \beta_{i_{N-N'}} = 0\}$, where $\{\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_{N-N'}}\}$ are the regression coefficients of the N-N' independent variables whose contribution to the regression sum of squares is to be tested, the variance ratio

$$F_e = \frac{SS_{N-N'}}{N-N'} \Big/ \frac{ATSS - ASSR_N}{n-N-1} \qquad \text{(III-1)}$$

is distributed as F with N-N' and n-N-1 degrees of freedom. The terms in this formula are defined as follows:

$ASSR_N$ = "total" regression sum of squares (adjusted for the mean), with N degrees of freedom, due to all N independent variables;

$SS_{N-N'}$ = $ASSR_N - ASSR_{N'}$ = "additional regression sum of squares", with N-N' degrees of freedom, due to the specified subset of N-N' independent variables, where $ASSR_{N'}$ is the regression sum of squares (adjusted for the mean) due to the N' independent variables left in the model after deleting the N-N' independent variables whose contribution to the fit is to be tested;

ATSS = $\sum_{i=1}^{n} (y_i - \bar{y})^2$ = total sum of squares (of y) adjusted for the mean, with n-1 degrees of freedom;

n = total number of observed y values.

When using the model re-evaluation option, the analyst merely specifies the N-N' independent variables, whose contribution to the regression sum of squares is to be tested, by indicating the complementary N' independent variables for which the program will make a "rerun." The specified set of the N' independent variables in a particular rerun of this option is called a "Hand Selection" of independent variables in order to distinguish it from a set automatically arrived at in any rerun of IVOR or BIVOR.

The F ratios (III-1) are computed and listed for all specified reruns in a "final comprehensive analysis table."

III.2 Ranking by IVOR and BIVOR

The subroutines IVOR and BIVOR for the automatic ranking of the independent variables by order of importance are also based on the Main Theorem. The routines may serve to separate the non-significant independent variables from the significant ones (or to find a "significant model") according to the F ratio (III-1) which is computed at each step. IVOR and BIVOR are particularly useful when the analyst knows nothing about the relative importance of the N IV's, or when the program user wants to confirm earlier results with new sets of input data.

The ranking of the independent variables in IVOR and BIVOR is done according to their prediction power for the dependent variable. This prediction power is measured by the additional regression sum of squares, $SS_{N-N'}$, (from the Main Theorem) which is due to the independent variables in question. It is possible to use, as ranking criterion, the additional regression sum of squares, or its complementary value, $ASSR_{N'}$, since the associated degrees of freedom are equal for each independent variable to be ranked. Therefore, the F test of the Main Theorem, within each step, has equal power with respect to degrees of freedom for each independent variable to be ranked.

The rankings proceed as follows:

In IVOR, a forward ranking process is executed, which, at the first step, searches among all N independent variables for the one which yields the largest value $ASSR_{N'} = ASSR_1$. This is the one independent variable among the N which, when it is the only one included in the model, explains the largest portion of the total regression sum of squares, $ASSR_N$. In the second step, IVOR searches for that pair of independent variables, consisting of the independent variable ranked most important in the first step, plus one of the remaining N-1 independent variables, which yields the largest value $ASSR_{N'} = ASSR_2$. This is continued through step number N-1, at the end of which the first N-1 most important independent variables will have been ranked. The

least important independent variable (Number N) is, thereby, determined automatically. Obviously, this ranking procedure results in a descending order of importance of the independent variables.

In BIVOR, a reverse ranking process is executed, which, at the first step, searches among all N independent variables for the one which yields the smallest value $SS_{N-N'} = SS_{N-(N-1)} = SS_1$. This is the independent variable among the N which, when deleted from the model, gives the smallest additional regression sum of squares. In the second step, BIVOR searches for that pair of independent variables, consisting of the independent variable ranked least important in the first step plus one of the remaining N-1 independent variables, which yields the smallest value $SS_{N-N'} = SS_{N-(N-2)} = SS_2$. This is continued through step number N-1, at the end of which the N-1 least important independent variables will have been ranked. The most important independent variable (Number N) is, thereby, determined automatically. As can be seen, the BIVOR ranking procedure results in an ascending order of importance of the independent variables.

In both IVOR and BIVOR the independent variables can optionally be grouped such that the ranking process is performed within only one group at a time. For details and for an application of the grouping feature as a device to save computing time, see Sections VI.1.d and VI.1.e; for other applications see Sections II.3 and VII.2.a.

As indicated earlier, the ranking of independent variables by their prediction power in both IVOR and BIVOR is mainly a means of identifying those IV's (independent variables) which have a significant prediction power for the dependent variable. In addition to this, the rankings give the experimenter an indication of the relative importance of the IV's, and these rankings sometimes are valuable in their own right. Generally, however, the goal to be achieved with such rankings is to determine a "significant model" containing a minimum number of IV's with maximum prediction power for the dependent variable. It is emphasized that, for this goal, the rankings as done by IVOR and BIVOR are not ideal but are feasible and considered to be adequate. (For a discussion of the "ideal method" see Section III.3.)

It is important to note that an independent variable which, by itself, has a large prediction power for y might not appear to have such in the ranking by IVOR or BIVOR. This could happen, for example, for one of two correlated (possibly highly) independent variables when both of them individually have considerable prediction power for y. Both IVOR and BIVOR would put the one independent variable of the two which has the higher (possibly only slightly) prediction power into the group of important independent variables and might rank the second one as being unimportant. Accordingly, this second independent variable may then appear to have little or no prediction power. It must be

recalled, however, that the prediction power of an independent variable, as defined here, is the additional prediction power in excess of that of the other independent variables already contained in the model. By itself, the second independent variable may be very important, but in combination with the first one it loses all its significance. Thus the ranking order, as established by IVOR or BIVOR, must be viewed under the aspect of the strictly prediction-power-oriented character of the ranking processes.

One might expect that IVOR and BIVOR will yield the same ranking order of the independent variables. However, this is, in general, not the case. One reason for this difference is the possible existence, in the data of a regression problem, of a so-called "compound" which has been defined in Abt [1965]. In brief, a "compound" is comprised of a set of $\tilde{N} \leq N$ independent variables plus the dependent variable when the error variance σ^2 associated with all $\tilde{N}$ independent variables is smaller, by orders of magnitude, than the error variance associated with any subset of $\tilde{N}$-1 independent variables, i.e., after any single independent variable has been excluded from the set of $\tilde{N}$ independent variables comprising the compound together with y.

The effect of the existence of a compound upon the ranking of independent variables is such that in the forward procedure (as executed by IVOR) an independent variable which does not belong to the compound might be ranked as most important and possibly as significant, whereas in the reverse procedure (as executed by BIVOR), this same independent variable might be ranked as least important and possibly as non-significant. The explanation is that in reverse ranking (BIVOR) the unity of the compound with its associated small error variance is preserved, as it should be, until the latest possible step of the procedure, whereas in forward ranking (IVOR) this unity could not be reached before the $\tilde{N}^{th}$ step, and possibly not until the very last step. A numerical example in which the latter actually happens is also given in Abt [1965].

Only when both ranking procedures result in equal, or nearly equal, orderings will the analyst know that there are no compounds (or no compounds of any consequence) present among the independent variables. The only protection against the disturbing effects of compounds upon the ranking is the application of the BIVOR routine. It is, therefore, strongly recommended to always use the BIVOR option for the automatic ranking of independent variables. Moreover, BIVOR is always an economical choice since a BIVOR ranking is at least 4 times faster than a full IVOR ranking. (For computational details and problem running time formulae, see Chapter VI.)

There are, however, two situations in which IVOR becomes a desirable option. A less important third situation is discussed in Section VII.2.b, where IVOR is shown to be advantageous in finding a

"perfect fit." The first situation arises when a large series of multiple regression problems of equal structure (with the same independent variables contained in the model for each problem) have to be processed and when the following two conditions hold true: (a) the sum of the BIVOR running times would be excessive; (b) one is only interested in a screening-type investigation as to the first few most important independent variables in each problem. For this situation IVOR has a cut-off option to search only for the first "IQ" most important variables, where IQ is a control card input number. (See Card Type 4, Section V.2.) That is, IVOR ceases ranking after step number IQ and, therefore, does not rank the N-IQ least important independent variables. Naturally, this application of the IQ-option of IVOR implies the risk of not detecting the effects of possibly existing compounds upon the ranking order. However, this is the price for saving computing time. (For IQ much smaller than N the running time of IVOR is considerably shorter than that of BIVOR; see time formulae in Section VI.4.)

The second situation in which IVOR becomes desirable also calls for the cut-off option of IVOR. The situation arises when, in a given problem with many independent variables, the significant IV's are to be found, but the final model is to be kept to a minimum number of independent variables in order to obtain small standard deviations for interval estimation purposes. In such a situation, the analyst should apply both BIVOR and IVOR, the latter with an IQ, say, in the vicinity of what is considered to be the maximum number of independent variables to be included in the final model. If there are no compounds, it is possible that the first IQ most important independent variables (or a subset of them), as ranked by IVOR, account for a higher portion of the total regression sum of squares than do the corresponding number of the most important independent variables in BIVOR. However, this evidence can be obtained only by comparing the results from both IVOR and BIVOR. This fact serves to re-emphasize the importance of the BIVOR routine, which should be applied for the ranking of the independent variables--alone or together with the IQ-option of IVOR--whenever the available computer time allows its use.

III.3 Comparison of IVOR and BIVOR with Other Techniques

The rankings of the independent variables as done in IVOR and BIVOR correspond to "forward" and "reverse" ranking, respectively, as discussed in Abt [1965]. The IVOR ranking proceeds in the same general forward direction as the "Stepwise Multiple Regression" technique by Efroymson [1960], but is otherwise different from that technique, as is obvious from reading Sections III.2 and VI.1.d.

Only after the DA-MRCA program was completed in its present form, a paper by Hamaker [1962] came to the attention of the authors in which two computational methods are discussed for the successive inclusion and deletion of independent variables: "forward selection" and "backward elimination", respectively. These two methods are

based on analyses of successive residuals, and, therefore, do not immediately seem to imply results which could be identical with those of IVOR and BIVOR, respectively. However, the numerical results of examples exhibited in the paper certainly suggest this both with respect to the ranking orders of the independent variables and the associated additional regression sums of squares. No attempt has been made to prove the general equality of the results of IVOR and "forward selection" or of those of BIVOR and "backward elimination."

As mentioned in Section III.2, IVOR and BIVOR are not ideal but are considered adequate for the purpose of ranking independent variables by order of importance and, thereby, finding a "significant model."

Naturally, the ideal method for determining the "significant model" would be to find the most important IV as in the first step of IVOR, but then to deviate from IVOR as follows. In the second step all $\frac{1}{2}N(N-1)$ possible pairs of IV's would be included in the model, and the one with the largest prediction power would be selected as the most important pair. Correspondingly, in the third step the most important triple of IV's would be found, etc. Since the most important pair of IV's would not necessarily contain the most important single IV found in the first step (and correspondingly for the triple versus the pair, and so on) a unique ranking would not necessarily result from this procedure. The significant model, however, would be found at the step where the F value (III-1) is non-significant for the first time, and the procedure could be stopped at this point. This "ideal" technique may be feasible for small values of N, but for larger N, such as IVOR and BIVOR are capable of handling, the indicated technique is infeasible with even the largest computer equipment available at the present time. In order to illustrate this, the following comparison of estimated minimum computer times (in seconds, on the IBM 7030 STRETCH) for the "ideal" technique to the actual running times of BIVOR, according to formula (VI-23) in Section VI.4, is given.

N	8	16	32
"Ideal" technique for finding significant model	24	25400	6.9×10^3
BIVOR	6	13	71
Ratio	4	~1950	~10^3

This table shows, for example, that with N=16 independent variables in the model, the estimated minimum computer time on the IBM 7030 for the "ideal" technique is 25400 seconds, which is approximately

1950 times the number of seconds BIVOR would need to rank the 16 IV's. For N=32, the figure is 6.9 billion seconds, whereas BIVOR needs a mere 71 seconds. The times for the "ideal" technique are based on the assumption that all 2^N-1 combinations of the IV's are examined. Naturally, these times would be, on the average, much smaller if the procedure were stopped after the significant model was found. However, the analyst could not predict at which step this would happen, and he probably would have to consider the times based on the 2^N-1 combinations. The result would be only the significant model, with no indication as to the relative importance of either the IV's contained in the significant model or of those not contained in the significant model.

Nevertheless, when N is sufficiently small, the program user can apply the "ideal" technique by using the option for hand selections of independent variables. The number of hand selected reruns is restricted, in one regression problem, to 999. (See Section V.2, Card Type 2, columns 5-7.) Therefore, N=9 is the upper limit for the number of independent variables contained in a model which is to be analyzed by the "ideal" technique: $2^9-1 = 511$. However, the analyst has to specify each combination of independent variables required by the "ideal" technique on a rerun card (see Section V.2, Card Type 10). In other words, the technique cannot be executed automatically by DA-MRCA.

Gorman and Toman [1966] have recently suggested a modification of the "ideal" technique by applying fractional factorial plans to sample the 2^N-1 possible combinations of IV's in order to reduce the computational effort required for the "ideal" technique.

IV. DEFINITIONS FOR INPUT, COMPUTATIONS, AND PRINTOUT

In this chapter the definitions of technical terms which are used in the following chapters are listed alphabetically. (Some of these terms have already been used in the previous chapters.) This list of definitions includes such familiar terms as, for example, "independent variable" and "data matrix." However, since such terms are often used in the literature with varying shades of meaning, the authors decided to include these in the list because a clear definition was considered necessary for the present purpose.

In the wording of each definition all the terms which are defined elsewhere in the list are marked by a dashed underline. The definitions are as follows:

A - The symbol used for the matrix of the normal equations.

Accepted Run - A run which passes all 5 tests concerning the feasibility and accuracy of the solution of the normal equations associated with the regression model for the given run. The five tests are those on the determinant, R^2, s^2, the $c_{\nu\nu}$, and the $i_{\nu\nu}$. For details see paragraphs B, D, E, F, and H of Section VI.2.a.(2).

Additional Regression Sum of Squares - In the Main Theorem the regression sum of squares, $SS_{N-N'}$, due to the addition of a specified subset of N-N' independent variables to the model containing the N' independent variables.

ASSR - "Adjusted (for the mean) Sum of Squares due to Regression." For the algebraic formulation of ASSR see Section VI.3.a. The term is used, in the report, in two applications:

(1) $ASSR_K$ = ASSR value due to K independent variables,

(2) $ASSR(x_1, x_2, \ldots)$ = ASSR value due to the set $(x_1, x_2, \ldots)$ of independent variables.

BIVOR - "Backward Independent Variable Ordering by Regression sums of squares." BIVOR is an optional subroutine which ranks the independent variables in ascending order of importance according to their contribution to the total regression sum of squares. See Section III.2 and Section VI.1.e for further explanation.

Calculated Identity Matrix - See definition of "identity matrix."

Card Type - One of the ten types of cards which constitute the problem deck. Each type of card is punched according to the input explanation and format given in Section V.2.

$c_{\nu\nu'}$ - The element in the $(\nu+1)^{th}$ row and $(\nu'+1)^{th}$ column of the inverse, A^{-1}, of the matrix of the normal equations. $(\nu, \nu' = 0,1,2,\ldots,K)$.

Coding - A term sometimes used for the transformation of the coordinates of the OCIV's to increase the computational accuracy, where the specific transformation recommended is $v=(x-\bar{x})/R_x$. See Section VII.2.a for further discussion.

Coordinate - The numerical value of an independent variable or of the dependent variable specifying, for the corresponding variable, the location of the design point or the data point. The observed numerical values of the dependent variable and the OCIV's are sometimes referred to as "observed coordinates", in contrast to the computed coordinates of the GCIV's.

Data Matrix - The $n \times (K+1)$ matrix consisting of the n data points. With K=N or K=N'<N, the data matrix is defined for the main run or for any rerun, respectively. The data matrix is printed only for the main run, see Section VI.3.a.

Data Point - A point specified by its K+1 coordinates in the (K+1)-dimensional space which is defined by the K independent variables and the dependent variable. With K=N or K=N'<N, a data point is defined for the main run or for any rerun, respectively. The number of data points (not necessarily all distinct) in a given regression problem is called n. As can be seen, a data point is defined by the coordinates of a design point and the coordinate of the dependent variable. Since several data points can be based on a common design point, one has $n \geq n_x$, where n_x is the number of distinct input design points in the space defined by the K independent variables.

Dependent Variable - The response variable, y (random), for which a numerical value y_i, $i = 1,2,\ldots,n$, is observed at each one of the n observed (not necessarily all distinct) input design points.

Design Matrix - The $n \times (K+1)$ matrix, denoted by X, of the n coordinates of the K independent variables, augmented by a column vector of n 1's for the constant, $x_0=1$. With K=N or K=N'<N, the design matrix is defined for the main run or for any rerun, respectively. Each row of the design matrix represents an input design point, not necessarily all different.

Design Point - A point specified by its K coordinates in the K-dimensional space which is defined by the K independent variables. With K=N or K=N'<N, a design point is defined for the main run or for any rerun, respectively. The symbol used for a design point is $(x_1, x_2, \ldots, x_v, \ldots, x_K)_i$.

Distinct Design Point - A design point specified by a unique combination of K coordinates. With K=N or K=N'<N, a distinct design point is defined for the main run or for any rerun, respectively. The number of distinct input design points in a given run with K independent variables is called n_x. In case of a rerun (K=N'<N)

the number n_K is defined only for an independent variable selection containing a specific set of K=N' IV's. It should be noted that n_K is equal for all those IVS's in a given regression problem which contain the same OCIV's.

$E_{\nu\nu'}$ - The element in the $(\nu+1)^{th}$ row and the $(\nu'+1)^{th}$ column of the matrix of the normal equations. Algebraically,

$$E_{\nu\nu'} = \sum_{i=1}^{n} x_{\nu i} x_{\nu' i}. \quad (\nu,\nu' = 0,1,2,\ldots,K).$$

$E_{\nu y}$ - The element in the $(\nu+1)^{th}$ row and the $(N+2)^{th}$ column of the summation matrix. $(\nu = 0,1,\ldots,N)$. Algebraically,

$$E_{\nu y} = \sum_{i=1}^{n} x_{\nu i} y_i.$$

E_{yy} - The total sum of squares of y, unadjusted for the mean. Algebraically,

$$E_{yy} = \sum_{i=1}^{n} y_i^2.$$

E_{yy} is the lower right hand corner element of the summation matrix.

GCIV - "Generated Concomitant Independent Variable." A GCIV is an independent variable which is generated from powers and/or cross-products of OCIV's. A GCIV may also be called a "product term."

Generated Independent Variable - See GCIV.

Hand Selected Rerun - The desired regression computations which are performed for a model containing a specified subset of $N' \leq N$ independent variables, where the particular set of N' independent variables is indicated on a punched card (Card Type 10, see Section V.2) in the problem deck.

I_e - The symbol used for the calculated identity matrix.

Identity Matrix - The (K+1) x (K+1) matrix, denoted by I_e, resulting from multiplying the inverse of the matrix A of the normal equations by the matrix A itself (in this sequence): $I_e = A^{-1}A$. With K=N or $K=N' \leq N$, the identity matrix is defined for the main run or for any rerun, respectively. The identity matrix is computed in each run in order to check the accuracy of A^{-1}. For details see Sections VI.1.b and VI.2.a.(2).

Independent Variable - One of the non-random variables, x_ν, in the linear regression model, whose prediction capacity for the dependent variable, y, is being investigated by a regression analysis. See also the definitions of OCIV and GCIV. For further discussion see Sections II.1 and II.2.

Independent Variable Selection - A subset of N' of the N independent variables originally input for a given regression problem. In the corresponding rerun, the regression computations are performed for the model containing these N' independent variables. Independent variable selections may be done "by hand" (see Section III.1 and Card Type 10 in Chapter V) or automatically by IVOR and/or BIVOR. Not every independent variable selection will necessarily lead to all desired computations of a rerun.

Input Design Point - A design point specified by its K observed or measured coordinates in the K-dimensional space defined by the K independent variables (both OCIV's and GCIV's) for which an observed or measured value of the dependent variable exists. With K=N or K=N'<N, an input design point is defined for the main run or for any rerun, respectively. The number of distinct input design points for any run (with K independent variables) is called n_K. An input design point, as the name suggests, is part of the data input for the program. However, the actual input writing is done, in DA-MRCA, only for the coordinates of the OCIV's, whereas the coordinates of the GCIV's may automatically be computed by the program.

$i_{\nu\nu'}$ - The element in the $(\nu+1)^{th}$ row and $(\nu'+1)^{th}$ column of the calculated identity matrix. $(\nu,\nu' = 0,1,2,\ldots,K)$.

IV - "Independent Variable" (see definition).

IVOR - "Independent Variable Ordering by Regression sums of squares." IVOR is an optional subroutine which ranks the independent variables in descending order of importance according to their contribution to the total regression sum of squares. See Section III.2 and Section VI.1.d for further explanation.

IVS - "Independent Variable Selection" (see definition).

K - The number of independent variables in a given run. In the main run, K=N; in a rerun, K=N'<N, i.e., K equals the number of the independent variables contained in the specific independent variable selection of the given rerun.

Leftmost Group - In IVOR and BIVOR, the first group of independent variables, according to the input and generation sequence, as designated by Card Type 4 and Card Type 5, respectively. (See Section V.2.) The leftmost group in IVOR is the first group of independent variables to be ranked, whereas in BIVOR the leftmost group is the last group of independent variables to be ranked.

Leftmost IV - At a given step of IVOR and/or BIVOR, the first (according to the input and generation sequence) unranked independent variable in a given group of independent variables.

Main Run - The regression computations which are performed for the model containing all N independent variables originally input for a given regression problem.

Main Theorem - The theorem of multiple regression on which all hypothesis testing and ranking of independent variables are based in DA-MRCA. See Section III.1 for a full discussion.

Matrix of the Normal Equations - The (K+1) x (K+1) symmetric matrix denoted by A and formed by pre-multiplying the design matrix, X, by its transpose, X'. For the full algebraic representation of A see Section VI.3.a. With K=N or K=N'<N, the matrix of the normal equations is defined for the main run or for any rerun, respectively.

n - The number of data points input in one regression problem. ($n \leq 7000$).

N - The number of independent variables (OCIV's and GCIV's) contained in the original regression model, i.e., in the model of the main run. ($N \leq 50$).

N' - The number of independent variables (OCIV's and GCIV's) contained in the model of a rerun.

n_x - The number of distinct design points in a given run with a specific set of K independent variables contained in the regression model.

Non-Obvious Linear Dependency - A linear dependency among two or more rows (columns) of the matrix of the normal equations when the dependency is not obvious in the sense of the "obvious linear dependency" (see definition).

Non-Zero Error Perfect Fit - A perfect fit in the case where the number of data points, n, is larger than the number, n_K, of distinct design points input: $n>n_K(=K+1)$. This term is used only when a distinction from a zero error perfect fit appears to be necessary.

Obvious Linear Dependency - A linear dependency among two or more rows (columns) of the matrix of the normal equations when the cause for the dependency can immediately be recognized from the number and/or constellation of the n_K distinct design points. See Section VII.2.b. for more details.

OCIV - "Original Concomitant Independent Variable." An OCIV is an independent variable which has physically been observed or measured for each value of the dependent variable. (The auxiliary variables used for the main effects in the multiple regression approach to analysis of variance, see Section II.3, are also considered as OCIV's with respect to the method of input into the program.) The term OCIV is used to differentiate this type of independent variable from a GCIV. The adjective "concomitant" stems from the concept of analysis of covariance to which DA-MRCA can also be applied. To distinguish OCIV's from GCIV's, the OCIV's are sometimes given the symbols z_j, $j = 1,\ldots,IR$, where IR is the number of OCIV's.

Original Independent Variable - See OCIV.

Perfect Fit - The least squares fit in the case where the number of distinct design points input, n_K, equals the number of independent variables in the model, plus 1: $n_K=K+1$. See also the definition for "zero error perfect fit" and for "non-zero error perfect fit."

Powersum - A term sometimes used in the discussion of GCIV's where it stands for the sum of the exponents of all OCIV's which are contained in the GCIV. For example, the powersum of the GCIV $x_1^2x_2x_3^3$ is 6.

Predicted Value (= Prediction) - The value ($\hat{Y}$) of the dependent variable as computed by evaluating the regression line (least squares fit) for a given model at an input design point or a synthetic design point.

Prediction Error - The deviation ($\hat{\epsilon}$) of the input value (y) of the dependent variable from the predicted value ($\hat{Y}$) of the dependent variable for any input design point in a given run.

Prediction Power - A term used for a characteristic of an individual independent variable or a group of IV's with respect to the dependent variable. The prediction power is measured by the additional regression sum of squares due to the individual IV or the group of IV's. See also Chapter III.

Prediction Standard Deviation for Individual Observations - The estimate ($s'_{(p)}$) of the standard deviation of a prediction in a given run at a specified design point. The prediction standard deviation may be computed for a selected input design point or a synthetic design point and is used in the computation of confidence limits for individual future observations (tolerance limits) of the dependent variable. (See Section VI.3.)

Prediction Standard Deviation for the Prediction Line - The estimate ($s_{(p)}$) of the standard deviation for the prediction line (regression equation) in a given run at a specified design point. The prediction standard deviation for the prediction line may be computed for a selected input design point or a synthetic design point and is used for the computation of confidence limits for the prediction line. (See Section VI.3.)

Problem Deck - The deck of punched cards which constitute the program input for one regression problem. The problem deck consists of cards of Types 1-10, see Section V.1.

Product Term - A synonym for GCIV.

Program Deck - The deck of punched cards containing the input-output requirements (see Section VIII.3) and the program instructions which are coded in FORTRAN IV for the IBM 7030 Computer. The program deck and the problem deck together constitute the total card input for a regression problem.

Program Variable - A program input parameter whose value is to be specified by the program user for each regression problem.

Ranking of Independent Variables - A process automatically executed by IVOR or BIVOR, sometimes also referred to as "ordering" of IV's.

Regression Problem - The totality of all phases of the regression analysis to be performed on one set of n data points as specified by one problem deck. A regression problem might include, therefore, the main run and several reruns, IVOR and BIVOR, the Chi-square test on normality of residuals in all runs, and other optional features.

Regression Sum of Squares Adjusted for the Mean = ASSR. See definition of ASSR.

Rejected Run - A run which is not an accepted run, i.e., a run which fails one of the 5 tests mentioned in the definition of an accepted run.

Rerun - The desired regression computations which are performed for a model containing a specified subset of $N' \leq N$ independent variables, i.e., the computations performed for a specified independent variable selection. A rerun can be specified automatically or "by hand."

Restricted Admissibility - A term used in connection with the ranking procedures IVOR and BIVOR. When ranking polynomial terms, or auxiliary variables in non-orthogonal analysis of variance, it is sometimes not advisable to consider all unranked IV's at a given step for ranking at that step. See Sections II.3 and VII.2.a for more details. Restricted admissibility can be effected by the grouping of IV's in IVOR and BIVOR, see Sections VI.1.d and VI.1.e.

Rightmost Group - In IVOR and BIVOR, the last group of independent variables, according to the input and generation sequence, as designated by Card Type 4 and Card Type 5, respectively. (See Section V.2.) The rightmost group in IVOR is the last group of independent variables to be ranked, whereas in BIVOR the rightmost group is the first group of independent variables to be ranked.

Rightmost IV - At a given step of IVOR and/or BIVOR, the last (according to the input and generation sequence) unranked independent variable in a given group of independent variables.

Run - The totality of all desired phases of the regression analysis to be performed on a model including a specified set of K independent variables. With $K=N$ or $K=N' \leq N$, the main run or any rerun is included in this definition.

Selected Input Design Point - An input design point selected by the program user, for which the prediction and the prediction standard deviation for the prediction line or for individual observations are to be computed.

Significant Model - A regression model containing all independent variables which contribute significantly to the total regression sum of squares due to the N independent variables in a regression problem, as determined by reruns and the associated F ratios for regression on deleted independent variables.

$SS_{N-N'}$ - See definition of "additional regression sum of squares."

Step (of IVOR or BIVOR) - All calculations which lead to the determination of an independent variable to be included in or to be deleted from the regression model in IVOR or BIVOR, respectively.

Summation Matrix - The (N+2) x (N+2) symmetric matrix composed of the (N+1) x (N+1) matrix (A) of the normal equations of the main run, the constants, $E_{\nu y}$, of the normal equations ($\nu = 0,1,\ldots,N$), and the sum of squares, E_{yy}, of the observations of the dependent variable. For the algebraic representation of the summation matrix see Section VI.3.a. The summation matrix is defined and printed only for the main run.

Synthetic Design Point - A point in the K-dimensional space defined by the K independent variables of a given run at which no value of the dependent variable has been observed. With K=N or K=N$'$<N, a synthetic design point is defined for the main run or for any rerun, respectively. The K coordinates of a synthetic design point are specified by the analyst. The concept is employed in an optional subroutine which computes predictions and prediction standard deviations for the prediction line or for individual observations at specified synthetic design points.

Total Regression Sum of Squares - A term sometimes used for the ASSR value of the main run, i.e., $ASSR_N$. (The main run contains the "totality" of all N independent variables originally considered in the regression problem, hence this name for $ASSR_N$.)

$x_{\nu i}$ - The symbol used for the numerical value (coordinate) of independent variable x_ν for the i^{th} data point. ($i = 1,\ldots,n$; $x_{0i} = 1$; $\nu = 1,2,\ldots,N$ in the main run.)

y_i - The symbol used for the numerical value (coordinate) of the dependent variable for the i^{th} data point. ($i = 1,\ldots,n$.)

Zero Error Perfect Fit - A perfect fit in the case where the number of data points, n, equals the number, n_K, of distinct design points input: $n=n_K(=K+1)$. The zero error perfect fit leaves no degrees of freedom for the error variance, hence the name. For further discussion see Section II.4.

V. INPUT PREPARATION

In this chapter the preparation of input for the DA-MRCA program is described. The various sections of the chapter give the problem deck setup (Section V.1), the preparation of the problem deck (Section V.2), and an example problem deck (Section V.3).

V.1 Problem Deck Setup

The problem deck for the general case is listed below by card type. There are ten card types required for the general case, and they are designated in order of input and by card name. For specific cases more than one punched card of a particular card type may be necessary. The names of these card types are followed with an "(S)" to denote the plural possibility. The explanation of each card type and the instructions for the preparation of the problem deck are given in the next section.

CARD TYPE 1 - PROBLEM IDENTIFICATION CARD

CARD TYPE 2 - PROBLEM CONTROL CARD

CARD TYPE 3 - PRODUCT TERM DESCRIPTION CARD(S) (Optional)*

CARD TYPE 4 - IVOR CONTROL CARD (Optional)*

CARD TYPE 5 - BIVOR CONTROL CARD (Optional)*

CARD TYPE 6 - SELECTED INPUT DESIGN POINT CARD(S) (Optional)*

CARD TYPE 7 - SYNTHETIC DESIGN POINT CARD(S) (Optional)*

CARD TYPE 8 - DATA INPUT CARDS

CARD TYPE 9 - DATA TERMINATION CARD

CARD TYPE 10 - RERUN CARD(S) (Optional)*

NOTE: The cards whose names are marked with asterisks (*) control optional features of the program and are omitted when the corresponding options are not desired.

The problem deck, as listed above, is stacked behind the program deck and constitutes the input for one regression problem. The information contained on the DATA INPUT and DATA TERMINATION CARDS (Card Types 8 and 9) may be placed on magnetic tape and the remainder

of the problem deck prepared on cards. Problem decks for additional regression problems are stacked consecutively behind the program deck. Each problem deck may contain a different combination of the optional cards. If a multiple problem case utilizes tape data of the types previously specified, the tape data must be ordered in the same manner as it would be presented as parts of the problem decks. Also, for the case of tape input, the tape identification number must be punched on the REEL CARD (third card of the program deck) starting in column 18. No identification number is necessary for card input.

V.2 Preparation of Problem Deck

In this section, instructions for the preparation of the problem deck are given. These instructions consist of: (a) the columns in which the punched entries are to be made; (b) the input formats; (c) the symbolic names of the program variables (when applicable); and (d) explanations of the punched entries associated with each program variable.

To facilitate the reading of the input instructions for the program user, who may be unfamiliar with the FORTRAN language, an explanation of the various format specifications used to describe the input-output data of DA-MRCA follows. Each format specification contains a letter indicating the type of information which must be input; also, the format specification contains integers which control the number of input fields to be used, the number of columns in each field, and the regulation of the assumed decimal point if the decimal point is not entered on the input card.

Format Specification A - This specification is of the form Aw, where A indicates that the input can be alphanumeric (alphabetical or numerical) and the w indicates the number of columns in the field. By writing a repetition number in front of the A, the same format specification can be applied to several successive fields, e.g., 10A8 means ten eight-column fields of alphanumeric information.

Format Specification I - This specification is of the form Iw, where the I indicates that the input must be an integer and the w indicates the number of columns in the field. Decimal points are not permitted and all input entries must be right adjusted, i.e., all entries are punched in the column or columns furthermost to the right within the field.

Format Specification X - This specification is of the form wX, which means that a field of w columns is to be left blank.

Format Specification E (Exponential) - This specification is of the form Ew.d, where the E indicates that the input value describes a real number of the scientific notation, for example, a number of the form 2.30×10^4. (The actual FORTRAN representation is 2.30E+04.) The

w indicates the number of columns in the field. The d indicates the number of digits to the right of the assumed decimal point if an actual decimal point is not punched. A repetition number written in front of the E applies the same format specification to a corresponding number of successive fields. In DA-MRCA the E format is used for the input of the two program variables TOLI1 and TOLI2 (Card Type 2 of the problem deck, see below) and, if specified, for the input of the coordinates of the OCIV's, the dependent variable, and the coordinates of the synthetic design points. The exponential part of the input number is generally of the form E±ee; however, other forms, such as E±e, ±ee and ±e, are permissible. Positive exponents can also be expressed as Ee or Eee. Example: The input values +5879E+03, .5879E+3, +58.79+01 and 5879.-1 would all read as 587.9 if the input format specification E9.4 is used.

Format Specification F - This specification is of the form Fw.d, where the F indicates that the input value describes a real number without an exponent notation; the w indicates the number of columns in the field and the d specifies the number of digits in the fractional portion of the number. (The d-specification is overridden by a punched decimal point.) A repetition number written in front of the F applies the same format specification to a corresponding number of successive fields. In DA-MRCA the F format is used, if specified, for the input of the coordinates of the OCIV's, the dependent variable, and the coordinates of the synthetic design points. Example: The input value of 16897 would be read as 1689.7 if the input format specification of F5.1 is used.

The instructions for the input preparation follow below.

CARD TYPE 1 - PROBLEM IDENTIFICATION CARD

Column	Format	Program Variable	Explanation
1-80	10A8	PGLB	Regression Problem Identification Card. (Any columns may be used.)

CARD TYPE 2 - PROBLEM CONTROL CARD

Column	Format	Program Variable	Explanation
1-2	I2	IR	Enter the number of original concomitant independent variables (OCIV's) whose coordinates will be input on DATA INPUT CARDS (Card Type 8).
3-4	I2	IS	Enter the number of generated concomitant independent variables (GCIV's) to be computed from the IR OCIV's (see Card Type 3). IR + IS = N ≤ 50.

CARD TYPE 2 (Cont'd)

Column	Format	Program Variable	Explanation
5-7	I3	NR	Enter the number of hand selected reruns (see Card Type 10). Punch a 0 if only automatic reruns are desired as selected by IVOR and/or BIVOR. $0 \leq NR \leq 999$.
8-10	I3	MVP	Enter the number of synthetic design points to be read from Card Type 7 - SYNTHETIC DESIGN POINT CARD(S) - for which the computations indicated in column 14 of the present card will be performed. $0 \leq MVP \leq 999$.
11-13	I3	NDR	Enter the number of selected input design points for which the computations indicated in column 14 will be performed. The selected input design points are denoted on Card Type 6 - SELECTED INPUT DESIGN POINT CARD(S). $0 \leq NDR \leq 999$.
14	I1	MVPL	0 = Predictions and prediction standard deviations for individual observations will be computed for selected input design points and/or synthetic design points for the main run and each hand selected rerun. (The standard deviations can be used to construct tolerance limits for individual observations, see Section VI.3.b.(2).) 1 = Predictions and prediction standard deviations for the prediction line will be computed for selected input design points and/or synthetic design points for the main run and each hand selected rerun. (The standard deviations can be used to construct confidence limits for the prediction line, see Section VI.3.b.(2).)
15	I1	NPE	0 = Predictions and prediction errors will not be printed and the test for normality of the prediction errors will not be performed for hand selected reruns and IVOR and/or BIVOR reruns.

CARD TYPE 2 (Cont'd)

Column	Format	Program Variable	Explanation
			1 = Predictions and prediction errors will be printed and the test for normality of the prediction errors will be performed for hand selected reruns and IVOR and/or BIVOR reruns.
16	I1	NDPO	0 = The coordinates of the data points will be printed (in the data matrix) in the format 9F13.6 and the predictions and the prediction errors will be printed in the format 2F15.6. 1 = The coordinates of the data points will be printed (in the data matrix) in the format 7E17.8 and the predictions and the prediction errors will be printed in the format 2E15.6. 2 = The coordinates of the data points will not be printed but the predictions and the prediction errors will be printed in the format 2F15.6.
17	I1	TAPE	0 = The coordinates of the OCIV's and the dependent variable and also the data termination indicator will be input on cards. 1 = The above will be input on magnetic tape. (The tape identification number must be entered on the REEL CARD of the program deck starting in column 18.)
18	I1	IVORGO	0 = IVOR and BIVOR will not be used. 1 = IVOR will be used. 2 = BIVOR will be used. 3 = IVOR and BIVOR will be used.
19-20	I2	NFD	Enter the number of data fields to be read from each DATA INPUT CARD (input record, if tape is used) as indicated by the input reading format (see columns 41-80). If no entry is given or if a zero is entered, seven data fields will be assumed.

CARD TYPE 2 (Cont'd)

Column	Format	Program Variable	Explanation
21	I1	IBID	0 = In BIVOR, the identity matrix will be computed for all reruns and accuracy checks will be performed on all identity matrices (see columns 23-40). 1 = In BIVOR, the identity computations and accuracy checks will be terminated with the first rerun in which an identity matrix has been computed which satisfies the accuracy criteria imposed by the value of I(1) (see columns 23-31). This option is a time-saving device which may be advantageously applied in cases with a large number of independent variables. See also Section VI.2.d., paragraph C.
22	--	----	Leave blank.
23-31	E9.5	TOLI1	Enter the value of I(1). This value will be used as the accuracy criterion for controlling the printout of the identity matrix for the main run and each rerun. If $\lvert i_{\nu\nu'} - L \rvert \geq I(1)$, where L=1 when $\nu=\nu'$ and L=0 when $\nu\neq\nu'$, the identity matrix will be printed. For further discussion and for the choice of I(1) see Section VI.1.b. Notice that, according to the format specification, this entry does not have to be right adjusted. The same applies to the next two entries (TOLI2 and FORM).
32-40	E9.5	TOLI2	Enter the value of I(2), where $I(2) \geq I(1)$. I(2) will be used as the accuracy criterion which determines acceptance or rejection of the regression computations for the main run or any rerun. If $\lvert i_{\nu\nu} - 1 \rvert \geq I(2)$, the run will be rejected. (NOTE: I(2) applies only to the elements of the main diagonal of the identity matrix.) For further discussion and for the choice of I(2) see Section VI.1.b.

CARD TYPE 2 (Cont'd)

Column	Format	Program Variable	Explanation
41-80	5A8	FORM	Enter the format specifications by which each Card Type 8 - DATA INPUT CARD (data input record, if tape input is used) is to be read. These format specifications do not include the first two columns of each DATA INPUT CARD which must be left blank. All coordinates of a data point may be read in the same manner by using a simple format specification such as 7F10.4 (see Card Type 8). However, if necessary or convenient, more complex format specifications may be entered whereby the various coordinates of a data point may occupy a varying number of columns. For example, if a record format of F12.5, 5F10.0, F8.4 were entered, the dependent variable, the first five OCIV's, and the sixth OCIV would constitute the input record and will be read by these formats, respectively. (NOTE: The commas must be entered to separate the individual formats.) If, in this example, more than six OCIV's were required to represent a data point, the additional OCIV's would constitute another input record and would be read by the same format specifications which means, the seventh OCIV would be read by F12.5, the eighth, ninth, tenth, eleventh, and twelfth OCIV's would be read by 5F10.0 and the thirteenth OCIV by F8.4, etc. If MFD = 0 (columns 19-20) the format 7F10.4 is assumed and no entry is necessary in columns 41-80.

CARD TYPE 3 - PRODUCT TERM DESCRIPTION CARD(S) (Optional)

This card is used to input the description of the IS product terms (OCIV's) which are to be generated from the values of the IR original concomitant independent variables (OCIV's). (See columns 1-4 of Card Type 2.) The OCIV's are powers and/or cross-products of the OCIV's and are generated as additional independent variables. A product term description designates the independent variables (OCIV's or OCIV's) which are to be used as multiplicative factors in the

CARD TYPE 3 (Cont'd)

generation of a GCIV. Any OCIV may be used as a factor in the generation of any GCIV and any previously generated GCIV may be used as a factor in the generation of a subsequent GCIV. A product term description consists of the subscripts of the independent variables which are to be used as factors in generating the GCIV. The following example case (IR = 2, IS = 7, N = 9) illustrates the procedure for writing product term descriptions. (This is the case of the example problem discussed in Sections V.3 and VI.5.)

IV	OCIV	GCIV	Product Term Description
x_1	z_1		Not applicable
x_2	z_2		Not applicable
x_3		$z_1 z_2$	1 2
x_4		z_1^2	1 1
x_5		z_2^2	2 2
x_6		$z_1^2 z_2$	1 1 2 or 1 3 or 2 4
x_7		$z_1 z_2^2$	1 2 2 or 1 5 or 2 3
x_8		z_1^3	1 1 1 or 1 4
x_9		z_2^3	2 2 2 or 2 5

As many as ten factors may be designated for each product term description and four product term descriptions may be punched on each card of this Card Type. If no product terms are to be generated (IS = 0), this card must be omitted from the input deck.

Column	Format	Program Variable	Explanation
			The description of the first product term occupying up to 20 columns is entered in columns 1-20 using two column fields to designate the factors:
1-2	I2	IN(1,1)	Enter the subscript of the independent variable to be used as the first factor in the product term.

CARD TYPE 3 (Cont'd)

Column	Format	Program Variable	Explanation
3-4	I2	IN(1,2)	Enter the subscript of the independent variable to be used as the second factor in the product term.
.	.	.	.
.	.	.	.
.	.	.	.
19-20	I2	IN(1,10)	Enter the subscript of the independent variable to be used as the tenth factor in the product term. (The description of the product term $z_1 z_2$ would be a 1 in column 2 and a 2 in column 4.) The descriptions of the second, third, and fourth product terms occupying up to 20 columns each are entered in columns 21-40, 41-60 and 61-80, respectively, in the same manner as the first product term description.

If more than four product terms are desired (IS > 4), cards in the same format are added as needed.

CARD TYPE 4 - IVOR CONTROL CARD (Optional)

The information which is input on this card determines the conditions under which IVOR will consider the independent variables for ranking. The independent variables can be divided into groups of consecutive independent variables, according to the sequence of input and generation, whereupon IVOR ranks the variables within these groups starting with the first group (see IVOR explanation in Section VI.1.d). The input parameters of IVOR are the number of variables to be ordered, the number of groups into which the variables are to be divided and the number of variables in each group. If IVORGO = 1 or 3 (see column 18, Card Type 2), this card must be included in the input deck. If IVORGO = 0 or 2, this card must be omitted from the input deck.

CARD TYPE 4 (Cont'd)

Column	Format	Program Variable	Explanation
1-2	I2	IQ	Enter the number of independent variables to be ordered by IVOR. If all N independent variables are to be ordered, enter 0 or leave blank. Otherwise $IQ \leq \sum_{j=1}^{M_I} N_j$ where M_I is the number of groups and N_j is the number of independent variables in the jth group.
3-5	I3	MI	Enter the number (M_I) of groups into which the set of independent variables is to be divided for ordering within groups. $1 \leq M_I \leq 25$.
6-8	I3	NJ(1)	Enter the number (N_1) of independent variables in the first group.
9-11	I3	NJ(2)	Enter the number (N_2) of independent variables in the second group.
. . .	. . .	. . .	. . .
78-80	I3	NJ(25)	Enter the number (N_{25}) of independent variables in the twenty-fifth group (if $M_I = 25$).

In order to consider all independent variables as one group, put MI = M_I = 1 and NJ(1) = N_1 = IR + IS = N. If only a subset of the N independent variables is to be considered, specify this by

$$\sum_{j=1}^{M_I} N_j < N;$$

however, the independent variables excluded will be the rightmost independent variables according to the input and generation sequence.

CARD TYPE 5 - BIVOR CONTROL CARD (Optional)

The information which is input on this card indicates the conditions under which BIVOR will consider the independent variables for ranking. As for IVOR, the independent variables can be divided into groups of consecutive independent variables, according to the sequence of input and generation. (The number of independent variables in the respective groups of IVOR and BIVOR may be entirely different.) BIVOR will do the ordering within each group starting with the last group (see BIVOR explanation in Section VI.1.e). If IVORGO = 2 or 3 (see column 18, Card Type 2), this card must be included in the input deck. If IVORGO = 0 or 1, this card must be omitted from the input deck.

Column	Format	Program Variable	Explanation
1-2	I2	MB	Enter the number (M_B) of groups into which the independent variables are to be divided for ordering within groups. $1 \leq M_B \leq 25$.
3-5	I3	LOT(1)	Enter the number (N_1) of independent variables in the first group, which will be the last group of IV's ordered. (N_q is the number of independent variables in the q^{th} group.)
6-8	I3	LOT(2)	Enter the number (N_2) of independent variables in the second group, which will be the next to last group of IV's ordered.
.	.	.	.
.	.	.	.
.	.	.	.
75-77	I3	LOT(25)	Enter the number (N_{25}) of independent variables in the twenty-fifth group (if $M_B = 25$) which will be the first group of IV's ordered.

In order to consider all independent variables as one group, put MB = M_B = 1 and LOT(1) = N_1 = IR + IS = N. If only a subset of the N independent variables is to be considered, specify this by

$$\sum_{q=1}^{M_B} N_q < N;$$

however, the independent variables excluded will be the rightmost independent variables according to the input and generation sequence.

CARD TYPE 5 (Cont'd)

NOTE: The program variable "LOT" is also used in connection with Card Type 10 - RERUN CARD - where it represents a different input parameter. The reader who is interested in more details about the variable LOT is referred to Chapter VIII.

CARD TYPE 6 - SELECTED INPUT DESIGN POINT CARD(S) (Optional)

The input design points for which the predictions and prediction standard deviations will be computed (see column 14, Card Type 2) are indicated on this card; these design points are denoted as selected input design points. Entries made on this card refer to the design points according to their order of input, i.e., if the computations are desired for the design point that was input first, a 1 is entered on this card, if the computations are desired for the design point that was input third, a 3 is entered on this card, etc. The computations are performed for the main run and all hand selected reruns. There must be exactly NDR entries (see columns 11-13, Card Type 2) on this card and they must be in numerically ascending order. If NDR = 0, this card must be omitted from the input deck. NDR $\leq$ 999.

Column	Format	Program Variable	Explanation
1-4	I4	IKEEPR(1)	Enter the number corresponding to the input order of the first selected input design point.
5-8	I4	IKEEPR(2)	Enter the number corresponding to the input order of the second selected input design point.
.	.	.	.
.	.	.	.
.	.	.	.
77-80	I4	IKEEPR(20)	Enter the number corresponding to the input order of the twentieth selected input design point.

IKEEPR(i) < IKEEPR(i + 1) for i = 1,2,...,(NDR-1). Additional cards are used if NDR > 20 and are continued in the same format.

CARD TYPE 7 - SYNTHETIC DESIGN POINT CARD(S) (Optional)

The synthetic design points for which the predictions and the prediction standard deviations (see column 14, Card Type 2) will be computed are specified on this card. A synthetic design point is specified by coordinates of the IR OCIV's and the IS GCIV's at which no actual experimentation was performed or no observation was made. (The coordinates of the GCIV's are not input on this card because they are generated from the coordinates of the OCIV's by the instructions given on Card Type 3.) By employing the feature of synthetic design points it is possible to obtain predictions and prediction standard deviations for arbitrarily chosen values of the independent variables. For example, the feature can advantageously be used for interpolation. The computations are performed for the main run and all hand selected reruns. The number of synthetic design points input must equal MVP (see columns 8-10, Card Type 2). The synthetic coordinates of the IR OCIV's are input with the same format that is used for the DATA INPUT CARDS, which is the format entered in columns 41-80 of Card Type 2, ignoring columns 1 and 2; however, the first field of the format (starting with column 3 of the first card of Card Type 7) is left blank since it corresponds to the first field of the DATA INPUT CARDS which is reserved for observations of the dependent variable. Anything punched in this field will be ignored by the program.

An explanation of the preparation of this control card is given below for the assumed format of 7F10.4. If MVP = 0, this card must be omitted from the input deck. MVP ≤ 999.

Column	Format	Explanation
1-2	2X	Leave blank.
3-12	10X	Leave blank.
13-22	F10.4	Enter "synthetic" z_{11}, the value of the first OCIV for the first synthetic design point.
23-32	F10.4	Enter "synthetic" z_{21}, the value of the second OCIV for the first synthetic design point.
.	.	.
.	.	.
.	.	.
63-72	F10.4	Enter "synthetic" z_{61}, the value of the sixth OCIV for the first synthetic design point.

Under the assumed format, 7F10.4, which is used here as an example, and if $6 < IR \leq 13$, a second card would be needed to complete the representation of the first synthetic design point. This second

CARD TYPE 7 (Cont'd)

card would be read with the same format (7F10.4) with the exception that columns 3-12 are used for the synthetic value of the seventh OCIV (syn z_{71}). If IR ≥ 13, additional cards would be necessary in order to completely represent the first synthetic design point, and the same format would be applied. Succeeding synthetic design points are input on successive cards in a similar manner.

CARD TYPE 8 - DATA INPUT CARDS

These cards are used to input the observed coordinates, $(y; z_1, z_2, \ldots, z_{IR})_i$, of the n data points, where IR is the number of OCIV's and i = 1,2,...,n. The numerical values are entered on the cards according to the format which has been specified in columns 41-80 of Card Type 2, ignoring columns 1 and 2. If more than one card is required to represent each data point, the additional cards (containing OCIV's only) will be read by the same format specification. An explanation of the preparation of these cards is given below for the assumed format 7F10.4 for data input.

Column	Format	Explanation
1-2	2X	Leave blank.
3-12	F10.4	Enter y_1, the observed coordinate of the dependent variable for the first data point.
13-22	F10.4	Enter z_{11}, the observed coordinate of the first OCIV for the first data point.
23-32	F10.4	Enter z_{21}, the observed coordinate of the second OCIV for the first data point.
.	.	.
.	.	.
.	.	.
63-72	F10.4	Enter z_{61}, the observed coordinate of the sixth OCIV for the first data point.

Under the assumed format, 7F10.4, which is used here as an example, and if 6 ≤ IR ≤ 13, a second card would be needed to complete the representation of the first data point. This second card would be read with the same format (7F10.4) with the exception that columns 3-12 are used for z_{71}, the observed coordinate of the seventh OCIV of the first data point. If IR ≥ 13, additional cards would be necessary in order to completely represent the first data point, and the additional cards would be written in the same format as the second card. The

CARD TYPE 8 (Cont'd)

coordinates $(y; z_1, z_2, \ldots, z_{IR})_i$ of the succeeding data points, where i = 2,3,...,n, are input on successive cards in a similar manner. The GCIV coordinates are generated using the OCIV coordinates which are input on these cards. The DATA INPUT CARDS and the SYNTHETIC DESIGN POINT CARD(S) are identical in format; however, the first field of the DATA INPUT CARDS contains the coordinates of the dependent variable and the first field of the SYNTHETIC DESIGN POINT CARD(S) is left blank. The program limitation on the number, n, of data points is: $n \leq 7000$.

CARD TYPE 9 - DATA TERMINATION CARD

Column	Format	Program Variable	Explanation
1-2	I2	M1	Enter any non-zero value.

If the information on Card Type 8 is on tape, the information on Card Type 9 must be on tape and must have a record length given by the format in columns 41-80 of Card Type 2 (or the assumed format, 7F10.4) plus 2 columns.

CARD TYPE 10 - RERUN CARD(S) (Optional)

This control card provides the capability of deleting any combination of independent variables (OCIV's or GCIV's) from the original model and, thereby, repeating the regression computations for a specified independent variable selection of $N' < N$ IV's. If all desired phases are executed, this repetition is called a rerun. A rerun card must be included in the input deck for each rerun that is desired and, therefore, NR (see columns 5-7, Card Type 2) rerun cards are needed. Each column of a rerun card represents an independent variable (OCIV or GCIV) in the original model for the main run. If a 1 is entered in the column, the corresponding independent variable is excluded from the model. If a 0 is entered in the column, the corresponding independent variable is included in the model. This card must be omitted from the input deck if NR = 0. $NR \leq 999$.

Column	Format	Program Variable	Explanation
1	I1	Lot (1)	Enter a zero; this column represents the constant which must be retained in the regression model for all runs.

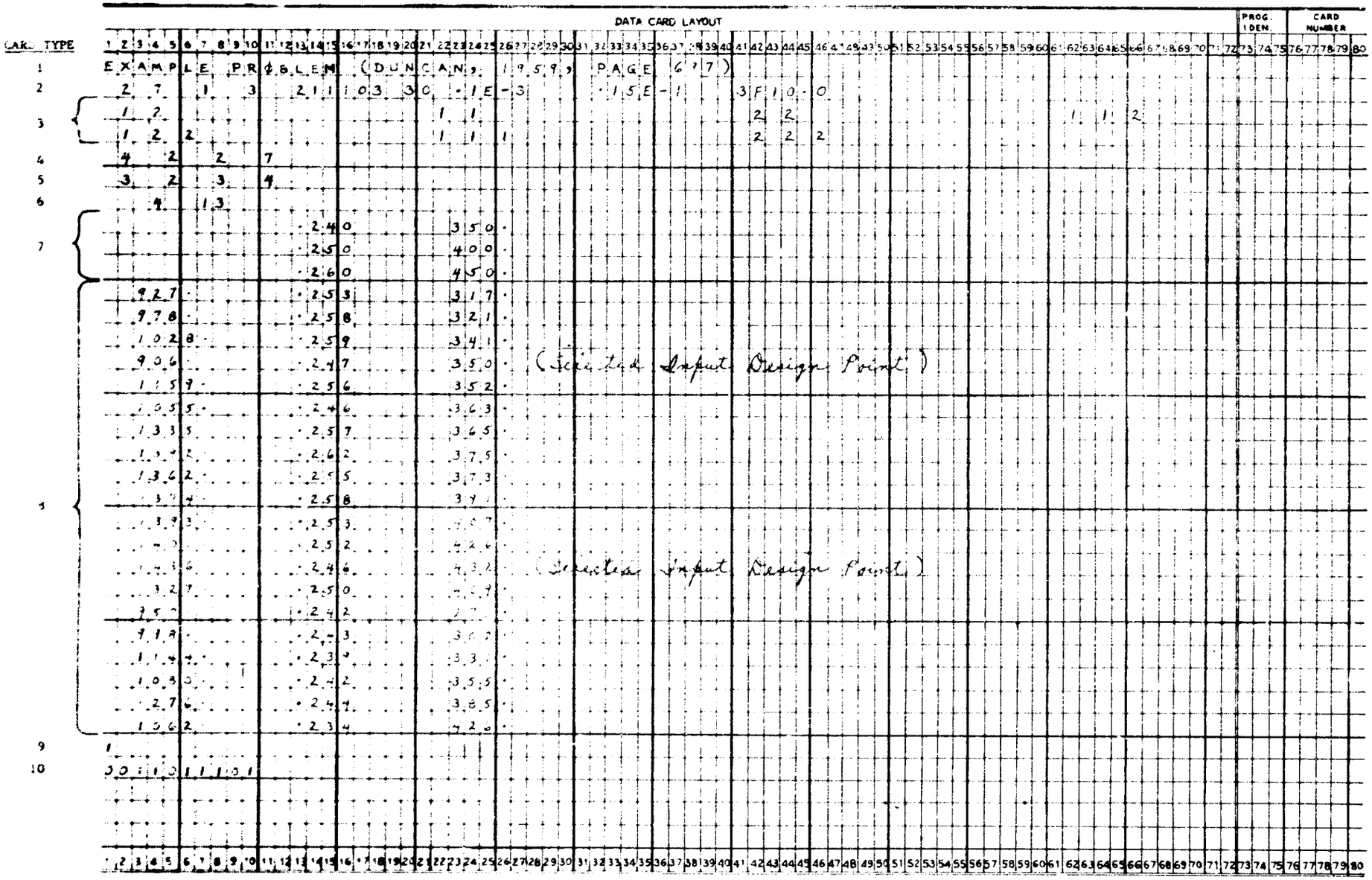
CARD TYPE 10 (Cont'd)

Column	Format	Program Variable	Explanation
2	I1	Lot (2)	This column represents the first independent variable; enter a zero if it is to be retained in the model or enter a one if it is to be deleted from the model.
3	I1	Lot (3)	This column represents the second independent variable; enter a zero if it is to be retained in the model or enter a one if it is to be deleted from the model.
.	.	.	.
.	.	.	.
.	.	.	.
51	I1	Lot (51)	This column represents the fiftieth (if N=50) independent variable; enter a zero if it is to be retained in the model or enter a one if it is to be deleted from the model.

Subsequent rerun cards are written in the same format.

V.3 Example Problem Deck

A card layout of the problem deck for the example problem which is discussed in Section VI.5 is given on the following page. An explanation for each card of the problem deck is also provided.


DATA CARD LAYOUT
PROG. IDEN.
CARD NUMBER
CARD TYPE
EXAMPLE PROBLEM (DUNCAN, 1959 PAGE 617)
(Selected Input Design Point)
(Selected Input Design Point)

Card Type	Column	Explanation
1	1-80	Identification of the problem.
2	1-2	IR=2; two OCIV's (z_1 and z_2) are input.
	3-4	IS=7; seven GCIV's are to be generated.
	5-7	NR=1; one hand selected rerun is to be executed.
	8-10	MVP=3; three synthetic design points are to be input.
	11-13	NDR=2; two selected input design points will be specified.
	14	MVPL=1; predictions and prediction standard deviations for the prediction line will be computed for the 3 synthetic design points and the 2 selected input design points for the main run and the hand selected rerun.
	15	NPE=1; prediction and prediction errors will be computed and printed and the Chi-square test for normality of the prediction errors will be performed for all reruns.
	16	NDPO=1; the coordinates of the data points will be printed in the format 7E17.8 and the predictions and the prediction errors will be printed in the format 2E15.6.
	17	TAPE=0; DATA INPUT and DATA TERMINATION are on cards.
	18	IVORGO=3; both IVOR and BIVOR will be used.
	19-20	NFD=3; there are three data fields on each DATA INPUT CARD.
	21	IBID=0; the identity matrices will be computed for all BIVOR reruns and the accuracy checks will be performed on all identity matrices from BIVOR reruns.
	23-31	I(1)=.1E-3=.0001 = accuracy criterion for printout of identity matrices.
	32-40	I(2)=.15E-1=.015 = accuracy criterion for rejection/ acceptance of runs.
	41-80	FORM=3F10.0; input format by which each DATA INPUT CARD is to be read is three ten-column fields in the F format starting with column 3.

Card Type	Column	Explanation
3 (first card)	1-20	IN(1,1)=1, IN(1,2)=2; the first GCIV (third independent variable) is $z_1 z_2 = x_3$.
	21-40	IN(2,1)=1, IN(2,2)=1; the second GCIV (fourth independent variable) is $z_1 z_1 = z_1^2 = x_4$.
	41-60	IN(3,1)=2, IN(3,2)=2; the third GCIV (fifth independent variable) is $z_2 z_2 = z_2^2 = x_5$.
	61-80	IN(4,1)=1, IN(4,2)=1, IN(4,3)=2; the fourth GCIV (sixth independent variable) is $z_1 z_1 z_2 = z_1^2 z_2 = x_6$.
3 (second card)	1-20	IN(5,1)=1, IN(5,2)=2, IN(5,3)=2; the fifth GCIV (seventh independent variable) is $z_1 z_2 z_2 = z_1 z_2^2 = x_7$.
	21-40	IN(6,1)=1, IN(6,2)=1, IN(6,3)=1; the sixth GCIV (eighth independent variable) is $z_1 z_1 z_1 = z_1^3 = x_8$.
	41-60	IN(7,1)=2, IN(7,2)=2, IN(7,3)=2; the seventh GCIV (ninth independent variable) is $z_2 z_2 z_2 = z_2^3 = x_9$.
4	1-2	IQ=4; IVOR will terminate after four independent variables have been ordered.
	3-5	MI=2; the independent variables are to be divided into two groups for ordering by IVOR.
	6-8	NJ(1)=2; the first two independent variables (x_1,x_2) are to be considered as the first group.
	9-11	NJ(2)=7; the next seven independent variables (x_3,x_4,x_5,x_6,x_7,x_8,x_9) are to be considered as the second group.
5	1-2	MB=3; the independent variables are to be divided into three groups for ordering by BIVOR.
	3-5	LOT(1)=2; the first two independent variables (x_1,x_2) are to be considered as the first group in BIVOR.

Card Type	Column	Explanation
5	6-8	LOT(2)=3; the next three independent variables (x_3, x_4, x_5) are to be considered as the second group.
	9-11	LOT(3)=4; the next four independent variables (x_6, x_7, x_8, x_9) are to be considered as the third group.
6	1-4	IKEEPR(1)=4; the fourth input design point (according to order of input) is to be used as a selected input design point for the calculations specified in column 14, Card Type 2.
	5-8	IKEEPR(2)=13; the thirteenth input design point (according to order of input) is to be used as a selected input design point for the calculations specified in column 14, Card Type 2.
7 (first card)	13-22	The value of the first OCIV for the first synthetic design point is entered (syn z_{11} = .240).
	23-32	The value of the second OCIV for the first synthetic design point is entered (syn z_{21} = 350).
7 (second card)	13-22	The value of the first OCIV for the second synthetic design point is entered (syn z_{12} = .250).
	23-32	The value of the second OCIV for the second synthetic design point is entered (syn z_{22} = 400)
7 (third card)	13-22	The value of the first OCIV for the third synthetic design point is entered (syn z_{13} = .260).
	23-32	The value of the second OCIV for the third synthetic design point is entered (syn z_{23} = 450).
8 (first card)	3-12	The observed coordinate of the dependent variable for the first data point is entered (y_1 = 927).
	13-22	The observed coordinate of the first OCIV for the first data point is entered (z_{11} = .253).
	23-32	The observed coordinate of the second OCIV for the first data point is entered (z_{21} = 317).
8 (second card thru twentieth card)		These cards are written in the same format as the preceding card.

Card Type	Column	Explanation
9	1-2	A non-zero value is entered for the purpose of indicating termination of data.
10	1	Lot (1)=0; the constant term must always be retained in the model.
	2	Lot (2)=0; the first independent variable (x_1) is included in the model for this rerun.
	3	Lot (3)=1; the second independent variable (x_2) is excluded from the model for this rerun.
	4	Lot (4)=1; the third independent variable (x_3) is excluded from the model for this rerun.
	5	Lot (5)=0; the fourth independent variable (x_4) is included in the model for this rerun.
	6	Lot (6)=1; the fifth independent variable (x_5) is excluded from the model for this rerun.
	7	Lot (7)=1; the sixth independent variable (x_6) is excluded from the model for this rerun.
	8	Lot (8)=1; the seventh independent variable (x_7) is excluded from the model for this rerun.
	9	Lot (9)=0; the eighth independent variable (x_8) is included in the model for this rerun.
	10	Lot (10)=1; the ninth independent variable (x_9) is excluded from the model for this rerun.

VI. COMPUTATION AND PRINTOUT

VI.1 Some Basic Computational Features

In this section some basic computational features will be discussed which merit being set aside from the description of the computational details given in Section VI.2. The discussion of these features may also provide a better understanding of the DA-MRCA program as a whole.

VI.1.a Matrix Inversion

The inverse of the matrix of the normal equations and the solution vector are obtained, in any given run, by the Gaussian elimination method with the largest element as pivot. In the following, the algorithm is outlined for the interested reader who prefers a discussion in general algebraic terms rather than interpreting those parts of the program listing (Section VIII.4) which represent this inversion procedure. The proof for the validity of the algorithm is omitted since it appears to be beyond the scope and intent of the present report. A proof is given, for example, in Cohen [1959]. The inversion subroutine was adopted without change from the nucleus program (TV-MRCA) of DA-MRCA.

The algorithm is described in terms of the main run, that is, as applied to the (N+1) x (N+1) matrix of the normal equations augmented by the right-hand vector of the N+1 elements $E_{\nu y}$. However, the algorithm is identically applied also to all reruns with $N' \leq N$ independent variables contained in the model.

The procedure (as discussed for the main run) consists of N+1 cycles, after each of which all (N+1)(N+2) elements involved will have changed. The elements of the matrix of the i^{th} cycle are denoted by the superscript i attached to the elements $E_{\nu\nu'}$ and $E_{\nu y}$: ${}^{i}E_{\nu\nu'}$, ${}^{i}E_{\nu y}$. By definition, i=0 indicates the original elements ${}^{0}E_{\nu\nu'} = E_{\nu\nu'}$, ${}^{0}E_{\nu y} = E_{\nu y}$; $\nu,\nu' = 0,1,\ldots,N$. At the end of cycle number N+1, the elements equal those of the inverse matrix A^{-1} and of the regression coefficients, respectively: ${}^{N+1}E_{\nu\nu'} = c_{\nu\nu'}$ and ${}^{N+1}E_{\nu y} = b_{\nu}$. The algorithm is as follows:

1st Cycle (i=1)

(1) The square matrix A of the normal equations with rank N+1 is searched for the element with largest absolute value, which is found on the main diagonal. This element is called the pivot element and is denoted by ${}^{0}E_{pp}$. Row p is called the pivot row; this row cannot be used as the pivot row in any one of the remaining N cycles.

All subsequent steps (Nos. (2) - (5)) of the 1st cycle are exactly like steps (2) - (5) of the i^{th} cycle as described below.

ith Cycle

(1) The square matrix of rank (N+1)-(i-1) = N+2-i, obtained from the matrix at the end of cycle No. i-1 by deleting all i-1 rows and columns corresponding to the pivot elements used previously, is searched for the element with largest absolute value, which is found on the main diagonal. This element is the pivot element of the i^{th} cycle and is denoted by ${}^{i-1}E_{pp}$. The corresponding row cannot be used as the pivot row in any one of the remaining N+1-i cycles.

(2)

$$ {}^{i}E_{p\nu'} = \frac{{}^{i-1}E^{*}_{p\nu'}}{{}^{i-1}E_{pp}}, \qquad \text{for } \nu' = 0,1,\ldots,N, $$

with

$$ {}^{i-1}E^{*}_{p\nu'} = \begin{cases} {}^{i-1}E_{p\nu'} & \text{if } \nu' \neq p \\ 1 & \text{if } \nu' = p \end{cases} $$

(3)

$$ {}^{i}E_{py} = \frac{{}^{i-1}E_{py}}{{}^{i-1}E_{pp}} $$

(4)

$$ {}^{i}E_{\nu\nu'} = {}^{i-1}E^{*}_{\nu\nu'} - {}^{i-1}E_{\nu p}\,{}^{i}E_{p\nu'} $$

$$ \text{for} \begin{cases} \nu = 0,1,\ldots,p-1,p+1,\ldots,N \\ \nu' = 0,1,\ldots,N \end{cases} $$

with

$$ {}^{i-1}E^{*}_{\nu\nu'} = \begin{cases} {}^{i-1}E_{\nu\nu'} & \text{if } \nu' \neq p \\ 0 & \text{if } \nu' = p \end{cases} $$

(5)

$$ {}^{i}E_{\nu y} = {}^{i-1}E_{\nu y} - {}^{i-1}E_{\nu p}\,{}^{i}E_{py} $$

$$ \text{for} \quad \nu = 0,1,\ldots,p-1,p+1,\ldots N. $$

$(N+1)^{th}$ Cycle

The computations are as in (1) - (5) of the i^{th} cycle with i=N+1. The results are:

$$\left.\begin{aligned} {}^{N+1}E_{\nu\nu'} &= c_{\nu\nu'} \\ {}^{N+1}E_{\nu y} &= b_{\nu} \end{aligned}\right\} \quad \text{for } \nu,\nu' = 0,1,2,\ldots,N.$$

The determinant, Δ, of the matrix A equals the product of the N+1 pivot elements of the N+1 cycles:

$$\Delta = \prod_{i=0}^{N} {}^{i}E_{pp}.$$

VI.1.b. Checks on the Accuracy of the Inverse Matrix

VI.1.b.(1) Introductory Remarks

The accuracy of the inverse, A^{-1}, of the matrix of the normal equations of a given run with $K(\leq N)$ independent variables, which is obtained in DA-MRCA by the modified Gaussian elimination process as described in the previous section, depends upon the natural limitation of the computer accuracy. For example, in the IBM 7030, 13 digit accuracy is present when single precision is used as in DA-MRCA. The limited computer accuracy causes the propagation of errors. Some contributing factors to the amount of these errors, as contained in the elements of A^{-1}, are:

(a) the rank of the matrix A;

(b) the underlying type of regression problem (for example, polynomial regression vs. ordinary linear regression with original independent variables only);

(c) the ranges of the values of the independent variables (for example, $|x_\nu| > 1$ vs. $|x_\nu| < 1$);

(d) the relative position of the n_K distinct input design points.

In general (an exception is discussed in Section VI.1.b.(3)), the only practical way to check on the amount of the propagated errors contained in the elements of the inverse A^{-1} is to calculate the product

$$I_c = A^{-1}A, \qquad \text{(VI-1)}$$

that is, to form a "calculated identity matrix", I_c, and to compare it with the exact identity (or unit) matrix, I. This is done in the present program for each run (in BIVOR, however, only when specified, see column 21 of Card Type 2, Section V.2). The checks on I_c, as

described further below, not only serve to reject unacceptably inaccurate inverses but also to identify cases in which the matrix of the normal equations contains "obvious" or "non-obvious" linear dependencies. These topics are further discussed, along with corrective measures to be taken in such rejection cases, in Chapter VII.

When I_c is calculated according to (VI-1), it is possible that the errors contained in the elements of A^{-1} are drastically magnified such that the off-diagonal elements of I_c are far from zero. This may even be true under the (unrealistic) assumption that the elements of A^{-1} are obtained without computational errors, except for the truncation errors due to the natural limitation of the computer accuracy, i.e., 13-digit accuracy as present on the IBM 7030 with single precision. In fact, the derivations in Section VI.1.b.(2) below are based on this assumption that the elements of A^{-1} are free from error, except truncation error. The main diagonal elements of I_c (which should all be 1) are the only elements of I_c which will never be affected by this type of magnifying process. Therefore, the accuracy check on I_c is restricted, in DA-MRCA, to the main diagonal. If the largest deviation from 1 in the main diagonal of I_c exceeds the input value of I(2) specified by the program user, the inverse is automatically rejected by the program as being unacceptably inaccurate. (The deviations from zero of the off-diagonal elements of I_c are also checked, but only for the purpose of deciding whether or not the matrix I_c is to be printed for visual inspection.)

The justification for the above statements is given in the next section and is based on the regression model (I-1) as used in DA-MRCA. If the model

$$y = \bar{y} + \sum_{\nu=1}^{N} \beta_\nu (x_\nu - \bar{x}_\nu) + e, \qquad \text{(VI-2)}$$

i.e., the "adjusted" regression model, were used, the elements ($\tilde{E}_{\nu\nu'}$, say) of the matrix of the normal equations would also be adjusted for the averages, e.g., $\tilde{E}_{\nu\nu'} = E_{\nu\nu'} - \frac{E_{\nu 0} E_{0\nu'}}{n}$, and a different situation (not necessarily an improved one) would arise with respect to the error magnifying process when calculating an identity matrix. See the remarks in Section VII.2.a concerning the effects of the transformation (VII-1), $v = \frac{x - \bar{x}}{R_x}$.

VI.1.b.(2) Justification for the Rejection Criterion

In this section a justification is given for the rejection criterion (as described before) which involves only the main diagonal elements of $I_c = A^{-1}A$. The justification is given under the simplifying assumption that the elements, $c_{\nu\nu'}$, of A^{-1} are free from error, except truncation error. It will be shown that even these

truncation errors in the $c_{\nu\nu'}$ are sometimes sufficient to cause large deviations from zero in the off-diagonal elements of I_c. Naturally, these deviations are even larger when the $c_{\nu\nu'}$ also contain propagated errors, as is almost always the case in reality.

All errors will be derived in terms of their approximate "orders of magnitude." For this purpose the following definition is introduced:

> Definition: The "order of magnitude" of a number, z, is defined, for the derivations of this section, to be the nearest power of ten to which z can be rounded. The symbol "$\approx$" is used to indicate that the number or algebraic term located to the right of the symbol is the order of magnitude of the term located to the left of the symbol. The symbol "$\approx$" is also applied to matrices, and its meaning shall then be that the matrix to the right of the symbol is the matrix of the orders of magnitude of the corresponding elements of the matrix to the left of the symbol.

For example, for z = 677232:

$$z = 677232 \doteq .7 \times 10^6 \approx 1 \times 10^6 = 10^6.$$

Another example is:

$$z = -0.0434 \doteq -0.4 \times 10^{-1} \approx -.1 \times 10^{-1} = -10^{-2}.$$

The approximate orders of magnitude of the truncation errors contained in the elements of A^{-1} and of the errors in the elements of I_c will be derived for the case of the main run, that is, for A being of rank (N+1) x (N+1). Naturally, the results are similarly valid for the matrix A of any rerun with $N' \leq N$ independent variables.

With $E_{\nu\nu'} = \sum_{i=1}^{n} x_{\nu i} x_{\nu' i}$, the matrix A of the normal equations for the main run is:

$$A = \begin{bmatrix} E_{00} & E_{01} & E_{02} & \cdots & E_{0\nu'} & \cdots & E_{0N} \\ E_{10} & E_{11} & E_{12} & \cdots & E_{1\nu'} & \cdots & E_{1N} \\ E_{20} & E_{21} & E_{22} & \cdots & E_{2\nu'} & \cdots & E_{2N} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ E_{\nu 0} & E_{\nu 1} & E_{\nu 2} & \cdots & E_{\nu\nu'} & \cdots & E_{\nu N} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ E_{N0} & E_{N1} & E_{N2} & \cdots & E_{N\nu'} & \cdots & E_{NN} \end{bmatrix} \qquad \text{(VI-3)}$$

The elements of A^{-1} will be expressed following Cramer's Rule. This may be done because the specific characteristics of the inversion process of DA-MRCA and the associated error propagation are unimportant for the purpose of the present derivation. To repeat, the only purpose is to show the magnifying process of the truncation errors contained in A^{-1} which can take place when $I_c = A^{-1}A$ is formed.

To arrive at the justification desired, it will further be necessary to make use of a known result from the theory of determinants: The determinant of order k,

$$D = \begin{vmatrix} d_{11} & d_{12} & \cdots & d_{1k} \\ d_{21} & d_{22} & \cdots & d_{2k} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ d_{k1} & d_{k2} & \cdots & d_{kk} \end{vmatrix} ,$$

can be expressed in the following form:

$$D = \sum^{k!} (\pm d_{1\alpha} d_{2\beta} d_{3\gamma} \cdots d_{k\kappa}), \qquad \text{(VI-4)}$$

where the summation extends over all k! members which result from the k! possible permutations of the subscripts $\alpha\neq\beta\neq\gamma\neq \cdots \neq\kappa$, each subscript taking one of the values 1, 2, 3, ..., k.

Applying (VI-4) to the elements $c_{\nu\nu'}$ of A^{-1} and recalling that $E_{\nu\nu'} = E_{\nu'\nu}$, one has from (VI-3) according to Cramer's Rule:

$$\left.\begin{aligned} c_{\nu\nu'} &= \frac{1}{\Delta} \sum^{N!} (\pm E_{0i_0} E_{1i_1} E_{2i_2} \cdots E_{ji_j} \cdots E_{(\nu-1)i_{\nu-1}} E_{(\nu+1)i_{\nu+1}} \cdots E_{Ni_N}) \\ &\text{with} \\ &\quad i_j = 0,1,2, \ldots, (\nu'-1),(\nu'+1), \ldots, N; \quad i_j \neq i_{j'} , \end{aligned}\right\} \text{(VI-5)}$$

$$\left.\begin{aligned} &\text{where} \\ \Delta = \mathrm{Det}(A) &= \sum^{(N+1)!} (\pm E_{0i_0} E_{1i_1} E_{2i_2} \cdots E_{ji_j} \cdots E_{\nu i_\nu} \cdots E_{Ni_N}) \\ &\text{with} \\ &\quad i_j = 0,1,2, \ldots, N; \quad i_j \neq i_{j'} . \end{aligned}\right\} \text{(VI-6)}$$

The sum in (VI-5) consists of an even number (N!) of members with alternating signs. Accordingly, the truncation error of this sum (or that of $c_{\nu\nu'}\Delta$) should have an approximate order of magnitude equal to that of the truncation error of the absolutely largest one of the N! members. However, the largest member cannot generally be defined, but

an upper bound for it can be determined by the application of Schwartz's inequality. This upper bound equals

$$U = \left(\prod_{\substack{j=0 \\ j \neq \nu, \nu'}}^{N} E_{jj} \right) \sqrt{E_{\nu\nu} E_{\nu'\nu'}} \,. \qquad \text{(VI-7)}$$

It will be demonstrated later that it is not unrealistic to use this upper bound for the largest of the N! members in $c_{\nu\nu'} \Delta$ because the use of U and of the value

$$U' = \left(\prod_{\substack{j=0 \\ j \neq \nu, \nu'}}^{N} E_{jj} \right) E_{\nu'\nu} \qquad \text{(VI-8)}$$

both lead to the same approximate results. But U' is indeed one of the N! members in the sum of (VI-5).

In order to illustrate the derivation of formulae (VI-7) and (VI-8), the term $c_{\nu\nu'} \Delta$ is evaluated for the example case of $\nu=2$, $\nu'=3$, and N=3: For this example, in none of the members in the sum (VI-5) is there an E-term having as its first subscript $\nu=2$ or as its second subscript $\nu'=3$. Disregarding the signs, the 6 members of the sum are:

1. $E_{00}E_{11}E_{32}$, 2. $E_{00}E_{12}E_{31}$, 3. $E_{01}E_{10}E_{32}$, 4. $E_{01}E_{12}E_{30}$,

5. $E_{02}E_{11}E_{30}$, 6. $E_{02}E_{10}E_{31}$.

The first of these members is the one which was generally denoted as U' in (VI-8). Recalling that

$$E_{\nu'\nu} = \sum_{i=1}^{n} x_{\nu i} x_{\nu' i},$$

or shorter,

$$E_{\nu'\nu} = \Sigma\, x_{\nu'} x_{\nu},$$

Schwartz's inequality shows that

$$E_{\nu\nu} = \Sigma x_{\nu'} x_{\nu} \leq \sqrt{\Sigma\, x_{\nu'}^2 \cdot \Sigma\, x_{\nu}^2} = \sqrt{E_{\nu'\nu'} E_{\nu\nu}}.$$

Therefore, and according to (VI-7), the value $U = E_{00}E_{11}\sqrt{E_{22}E_{33}}$ is also an upper bound for $U' = E_{00}E_{11}E_{32}$. (As indicated before for the general case, both values $E_{00}E_{11}E_{32}$ and $E_{00}E_{11}\sqrt{E_{22}E_{33}}$ will lead to the same results with respect to the approximate orders of magnitude of the truncation error of c_{23}.) To show the validity of the upper bound U for one more member of the six, take the fourth: $E_{01}E_{12}E_{30}$. Here one has

$$E_{01}E_{12}E_{30} = \Sigma x_0x_1 \Sigma x_1x_2 \Sigma x_3x_0 < \sqrt{\Sigma x_0^2 \Sigma x_1^2 \Sigma x_1^2 \Sigma x_2^2 \Sigma x_3^2 \Sigma x_3^2}$$

$$= \Sigma x_0^2 \Sigma x_1^2 \sqrt{\Sigma x_2^2 \Sigma x_3^2} = E_{00}E_{11}\sqrt{E_{22}E_{33}} .$$

Continuing the main derivation, the truncation error of $c_{\nu\nu'}$ will be called $\delta(c_{\nu\nu'})$ and expressed as $10^{-H}c_{\nu\nu'}$, where the exponent H (> 0) is left unspecified for the time being. Therefore, replacing the sum in (VI-5) by the term (VI-7) (which substitution, according to the argument used, is possible only under the simultaneous multiplication of both sides of (VI-5) with 10^{-H}) one gets:

$$\delta(c_{\nu\nu'}) = 10^{-H}c_{\nu\nu'} \approx \frac{10^{-H}}{\Delta}\left(\prod_{\substack{j=0 \\ j\neq\nu,\nu'}}^{N} E_{jj}\right)\sqrt{E_{\nu\nu}E_{\nu'\nu'}} . \qquad \text{(VI-9)}$$

Here, it is sufficient to replace Δ by its approximate order of magnitude. This can be set equal to the order of magnitude of the product of the main diagonal elements of Δ,

$$\prod_{j=0}^{N} E_{jj},$$

which is the largest member in the sum of the (N+1)! members in (VI-6). In doing so, therefore, one actually replaces Δ by an upper bound which results in a lower bound for the order of magnitude of the truncation error of $c_{\nu\nu'}$:

$$\delta(c_{\nu\nu'}) \approx \frac{10^{-H}}{\sqrt{E_{\nu\nu}E_{\nu'\nu'}}} . \qquad \text{(VI-10)}$$

However, if the lower bounds of the truncation errors are able to cause the large deviations in the off-diagonal elements of I_e (as will be shown), these deviations are in reality even larger for the true truncation errors in the $c_{\nu\nu'}$.

The element $i_{\nu\nu'}$ of $I_e = A^{-1}A$ is obtained as

$$i_{\nu\nu'} = \sum_{\nu^*=0}^{N} c_{\nu\nu^*}E_{\nu^*\nu'} . \qquad \text{(VI-11)}$$

Defining

$$c_{\nu\nu^*} = \bar{c}_{\nu\nu^*} + \delta(c_{\nu\nu^*})$$

where $\tilde{c}_{\nu\nu*}$ is the true value of the inverse element (i.e., a value free from truncation error and any other error) and where $\delta(c_{\nu\nu*})$ is the truncation error of $c_{\nu\nu*}$ as defined before, one has from (VI-11):

$$i_{\nu\nu'} = \sum_{\nu*=0}^{N} \{\tilde{c}_{\nu\nu*} + \delta(c_{\nu\nu*})\}E_{\nu*\nu'}$$

$$= \begin{cases} 1 + \sum_{\nu*=0}^{N} \{\delta(c_{\nu\nu*})\}E_{\nu*\nu} & \text{if } \nu'=\nu \\ 0 + \sum_{\nu*=0}^{N} \{\delta(c_{\nu\nu*})\}E_{\nu*\nu'} & \text{if } \nu'\neq\nu \end{cases}$$

This leads to the definition of the error of $i_{\nu\nu'}$ caused by the truncation error of $c_{\nu\nu'}$:

$$\delta(i_{\nu\nu'}) = \sum_{\nu*=0}^{N} \{\delta(c_{\nu\nu*})\}E_{\nu*\nu'} . \qquad \text{(VI-12)}$$

(Notice that this derivation implied the assumption of no additional truncation errors being introduced when forming $i_{\nu\nu'}$.)

Inserting (VI-10) into (VI-12) one has:

$$\delta(i_{\nu\nu'}) \approx \sum_{\nu*=0}^{N} \cdot \frac{10^{-M}}{\sqrt{E_{\nu\nu}E_{\nu*\nu*}}} E_{\nu*\nu'} . \qquad \text{(VI-13)}$$

At this point it is necessary to introduce another approximation. Since only orders of magnitude are considered, it appears sufficient to put, in general,

$$E_{\nu\nu'} = \sum_{i=1}^{n} x_{\nu i}x_{\nu' i} \approx n\,\bar{x}_{\nu}\bar{x}_{\nu'} . \qquad \text{(VI-14)}$$

Substituting these orders of magnitude in (VI-13), one gets

$$\delta(i_{\nu\nu'}) \approx \sum_{\nu*=0}^{N} \frac{\bar{x}_{\nu'}}{\bar{x}_{\nu}} 10^{-M} = (N+1)10^{-M}\frac{\bar{x}_{\nu'}}{\bar{x}_{\nu}} . \qquad \text{(VI-15)}$$

An identical result is obtained when the term (VI-8), U', rather than (VI-7), U, is used to replace the sum in (VI-5). In this case one has, instead of (VI-10), for the truncation error of $c_{\nu\nu'}$:

$$\delta'(c_{\nu\nu'}) \approx \frac{10^{-n}E_{\nu^*\nu}}{E_{\nu\nu}E_{\nu^*\nu'}} .$$

This leads to the error of $i_{\nu\nu'}$, corresponding to (VI-13):

$$\delta'(i_{\nu\nu'}) \approx \sum_{\nu^*=0}^{N} 10^{-n} \frac{E_{\nu^*\nu}E_{\nu^*\nu'}}{E_{\nu\nu}E_{\nu^*\nu^*}} .$$

Using again the approximation (VI-14), one has $\delta'(i_{\nu\nu'}) \approx \delta(i_{\nu\nu'})$, which was to be shown.

Finally, using (VI-15), the matrix $\delta(I_c)$ of the approximate orders of magnitude of the errors in I_c, caused by the truncation errors $\delta(c_{\nu\nu'})$ only, is obtained:

$$\delta(I_c) \approx (N+1)10^{-n} \begin{bmatrix} 1 & \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_\nu & \cdots & \bar{x}_{\nu'} & \cdots & \bar{x}_N \\ \frac{1}{\bar{x}_1} & 1 & \frac{\bar{x}_2}{\bar{x}_1} & \cdots & \frac{\bar{x}_\nu}{\bar{x}_1} & \cdots & \frac{\bar{x}_{\nu'}}{\bar{x}_1} & \cdots & \frac{\bar{x}_N}{\bar{x}_1} \\ \frac{1}{\bar{x}_2} & \frac{\bar{x}_1}{\bar{x}_2} & 1 & \cdots & \frac{\bar{x}_\nu}{\bar{x}_2} & \cdots & \frac{\bar{x}_{\nu'}}{\bar{x}_2} & \cdots & \frac{\bar{x}_N}{\bar{x}_2} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \frac{1}{\bar{x}_\nu} & \frac{\bar{x}_1}{\bar{x}_\nu} & \frac{\bar{x}_2}{\bar{x}_\nu} & \cdots & 1 & \cdots & \frac{\bar{x}_{\nu'}}{\bar{x}_\nu} & \cdots & \frac{\bar{x}_N}{\bar{x}_\nu} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \frac{1}{\bar{x}_{\nu'}} & \frac{\bar{x}_1}{\bar{x}_{\nu'}} & \frac{\bar{x}_2}{\bar{x}_{\nu'}} & \cdots & \frac{\bar{x}_\nu}{\bar{x}_{\nu'}} & \cdots & 1 & \cdots & \frac{\bar{x}_N}{\bar{x}_{\nu'}} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \frac{1}{\bar{x}_N} & \frac{\bar{x}_1}{\bar{x}_N} & \frac{\bar{x}_2}{\bar{x}_N} & \cdots & \frac{\bar{x}_\nu}{\bar{x}_N} & \cdots & \frac{\bar{x}_{\nu'}}{\bar{x}_N} & \cdots & 1 \end{bmatrix} \qquad \text{(VI-16)}$$

The formulation (VI-16) shows the following: (a) the orders of magnitude of the errors in the main diagonal elements of I_c (caused only by truncation errors in the $c_{\nu\nu'}$) are approximately $(N+1)10^{-H}$ and are, therefore, independent of the numerical values of the independent variables; (b) the orders of magnitude of the errors in the off-diagonal elements of I_c are approximately the orders of magnitude of the ratios, multiplied by $(N+1)10^{-H}$, of the averages of the independent variables as given in the matrix and are, therefore, dependent upon the numerical values of these independent variables; (c) the approximate orders of magnitude of the errors of the off-diagonal elements of I_c are reciprocal with respect to the main diagonal (apart from the factor $(N+1)10^{-H}$), viz.,

$$\delta(i_{\nu\nu'}) \approx (N+1)10^{-H}\frac{\bar{x}_{\nu'}}{\bar{x}_{\nu}} \quad \text{versus} \quad \delta(i_{\nu'\nu}) \approx (N+1)10^{-H}\frac{\bar{x}_{\nu}}{\bar{x}_{\nu'}} .$$

According to these findings, an off-diagonal element of I_c can appear to be so much in error that it is not even in the vicinity of zero. This is particularly likely to happen when one deals with polynomials. For example, in a polynomial in one independent variable x, the term

$$\bar{x}_{\nu} = \frac{1}{n}\sum_{i=1}^{n} x_i^{\nu}$$

can be rather large when $|x| > 1$ and the exponent ν is sufficiently large.

If the order of magnitude of $\bar{x}_{\nu}$ is called 10^{M}, then the error of $i_{0\nu}$ is, for example, according to (VI-15):

$$\delta(i_{0\nu}) \approx (N+1)10^{-H}\bar{x}_{\nu} \approx (N+1)10^{-H+M} .$$

If M is approximately equal to H, the apparent deviation of $i_{0\nu}$ from zero can be considerable, and it is obvious that this deviation can be large even if the matrix inversion was perfectly accurate within the natural limitations of the computer accuracy.

The following simple example was actually computed with DA-MRCA in order to illustrate what has been shown theoretically. The numbers displayed are taken from the program output. There is only one independent variable in the example, and its 5 distinct numerical values were chosen extremely large in order to emphasize the effect. The x values are as follows (written in the exponential format):

.39062500E+14

.26435638E+15

.75493321E+15

.39721133E+16

.10000000E+17

The matrix A is accordingly:

$$A = \begin{bmatrix} .50000000E+01 & .15030465E+17 \\ .15030465E+17 & .11641902E+33 \end{bmatrix}$$

From this, DA-MRCA computed A^{-1}:

$$A^{-1} = \begin{bmatrix} .32685427E-00 & -.42199048E-16 \\ -.42199048E-16 & .14037838E-31 \end{bmatrix}$$

and, finally:

$$I_c = A^{-1}A = \begin{bmatrix} .10000000E+01 & .64000000E+02 \\ -.78886091E-30 & .10000000E+01 \end{bmatrix}$$

The deviation of i_{01} from zero is 64, that is, the apparent error of i_{01} has an order of magnitude 10^2. According to (VI-15), the error of i_{01} should have an approximate order of magnitude equal to that of $(N+1) \times 10^{-H}\bar{x}_1$. The average of the 5 levels of $x_1 = x$ is $\bar{x} = .30060931E+16$. Therefore, the error of i_{01} should have an approximate order of magnitude of $2(10^{-H})(.3)10^{+16} \approx 10^{16-H}$. With H=14 for the IBM 7030 (single precision), the apparent order of magnitude of the error of i_{01} equals the one theoretically predicted: $\delta(i_{01}) \approx 10^{16-14} = 10^2$. Equally interesting is the apparent order of magnitude of the error of $i_{10} = -.78886091E-30$ which is 10^{-30} if one neglects the negative sign. According to (VI-15), the approximate order of magnitude of the deviation of i_{10} from zero should be that of $2(10^{-H})\frac{1}{\bar{x}}$ which is $10^{-H}10^{-15} = 10^{-29}$ with H=14. This approximation, therefore, is almost as good as the one for $\delta(i_{01})$. Finally, the errors in the main diagonal elements of I_c should have orders of magnitude equal to that of $2(10^{-14})$ which cannot be observed since only 8 digits are printed by the program. Obviously, in this case, the good agreement between the predicted and apparent orders of

magnitude of the errors in I_c is due to the small rank of the matrix A. It can be assumed that propagation errors are practically absent when a matrix of rank 2, as in this example, is inverted. In this case, therefore, the apparent errors in the elements of I_c should essentially be the magnified truncation errors of the $c_{\nu\nu'}$, the approximate orders of magnitude of which are given by (VI-16).

It should be noted that the errors of the off-diagonal elements of I_c might appear to be large not only when the x values are very large (and of equal sign) as in the above example, but also when the x values are very small (and of equal sign). If the latter is the case, the deviations from zero of the elements in the lower half of I_c will be very large.

The only way to guarantee that the errors of all elements of I_c will be of equal order of magnitude (i.e., $(N+1) \times 10^{-n}$) would be to apply a standardizing transformation to the x values, such as $v = \frac{x-\bar{x}}{R_x}$ which is discussed in Section VII.2.a. With $R_x = \max(x) - \min(x)$, this transformation results in average values of the independent variables which have an approximate order of magnitude 1, and this, as can be seen from (VI-16), leads to the uniformity of the orders of magnitude of the errors in all elements of I_c. Only in this case, therefore, would it make sense to check the accuracy of <u>all</u> elements of I_c, or, preferably, of all elements of the residual matrix $I_c - I$. For this situation a measure like the Euclidean norm could be used to check the accuracy of $I_c - I$ and, thereby, the accuracy of the inverse matrix.

However, as is shown in Section VII.2.a, the transformation $\frac{x-\bar{x}}{R_x}$ can be very undesirable for the program user in certain situations. It is essentially for this reason that in DA-MRCA the accuracy checks on the identity matrix are restricted to its main-diagonal. Since all $(N+1)^2$ elements of A^{-1} are involved in this check, it is felt that by this check the program user is sufficiently protected from inaccurate or fictitious inverses.

In connection with the results of the present section, the reader is referred to an example case of a 5th order polynomial which is also given in Section VII.2.a. In this example, the off-diagonal elements of I_c deviate from zero to a much larger extent than indicated by (VI-16), which is in accordance with the assumptions leading to (VI-16). The deviations practically vanish when the x values are "coded," i.e., when the transformation $v = \frac{x-\bar{x}}{R_x}$ is applied.

VI.1.b.(3) The Choice of I(1) and I(2)

Restating from Section VI.1.b.(1), the program rejects an inverse as unacceptably inaccurate when the largest deviation from 1 in the main diagonal of I_c exceeds a value, I(2), specified by

the program user. As to the choice of I(2), extensive studies have been made by the authors. One method which was applied to find a direct relation between the maximum deviation of the $i_{\nu\nu}$ from 1 and the accuracy of the inverse, was the computation of perfect fit regression cases. In these cases the regression sum of squares, as computed by using the elements of the inverse, via the regression coefficients:

$$ASSR_N = \sum_{\nu=0}^{N} b_\nu E_{\nu y} - \frac{1}{n} E_{oy}^2 ,$$

can be compared with its hand-computed equivalent. (This is the exceptional case, mentioned in Section VI.1.b.(1), in which the accuracy of the inverse can independently be checked.) The results from the calculated example cases confirmed the experience gained by the authors in many problems previously solved with DA-MRCA: The chosen value of I(2) should lie between 0.001 and 0.01, depending upon the rank of A. With this choice the analyst can be confident that inaccurate or fictitious inverses will be rejected by the program and that, in general, sufficiently accurate inverses will not be rejected.

Since the analyst might sometimes wish to visually inspect the whole calculated identity matrix, DA-MRCA provides for the possibility of printing it. The decision of whether or not to print I_c is made by the program: only when none of the elements of I_c-I is in error by more than a value, I(1), specified by the program user, will I_c not be printed. The reason for this device is twofold:

(a) If I_c is not printed, the user knows at once that all errors are smaller than I(1).

(b) If the user is not interested in the inspection of I_c, he can possibly choose I(1) so large (but not larger than I(2)) that in most cases I_c will, in fact, not be printed, whereby printout and printing time of the whole regression problem will be reduced. If he chooses I(1) = I(2), he will get a printout of I_c only in rejection cases.

Occasionally the program user wants every identity matrix printed. He can achieve this by putting I(1) = 0. Otherwise, the choice of the value of I(1) must be left to the user. For the purpose of acquainting the user with the program, concerning the behavior of I_c, the experience of the authors showed that a value of I(1) in the vicinity of 10^{-4} should be chosen.

VI.1.c Chi-Square Test on Normality of Residuals

Significance tests based on the main theorem of multiple regression (Section III.1) and the construction of confidence intervals require normality of the distribution of the residuals e in the model (I-1). The only way to test the hypothesis of normality is to examine the distribution of the "estimated" residuals, $\hat{e}_i = \hat{y}_i - \hat{Y}_i$, $i = 1,\ldots,n$. This is done in the present program by the Chi-square test. One should, however, remember that the F test (III-1) of the main theorem is rather robust with respect to the form of the distribution of the residuals. Therefore, unless striking evidence of non-normality is shown by either the bar chart of the frequency distribution of the $\hat{e}_i$ or the computed Chi-square value, or both, the analyst would not be too concerned about the hypothesis testing aspects. For interval estimation, however, normality as demonstrated by the $\hat{e}_i$ is essential.

Both the bar chart and the Chi-square value (if it can be computed) should, therefore, be considered merely as aids to determine whether a transformation of the observed values of the dependent variable, y, would be necessary or helpful to achieve normality or approximate normality of the residuals. Also, the possible significance of the computed Chi-square value should not be taken too literally. The Chi-square test for normality is only an approximation, and the number of degrees of freedom, m-K-3, obtained by subtracting the number, K+2, of parameters estimated (K+1 regression coefficients plus the standard deviation in case of a model containing K IV's) from m-1, where m is the final number of intervals, certainly is a safe lower limit.

The fixed number of 30 initial intervals into which the observed range of the residuals is partitioned also deserves some discussion. As outlined in more detail in Section VI.2.a.(3), the Chi-square subroutine automatically arrives at a new partitioning of the range into m ≤ 30 intervals by combining subsets of the 30 initial intervals into m new intervals such that each one of the m has an expected number of more than 5 observations. The initial number of 30 intervals was chosen as a compromise to avoid the extremes of: (1) having, in most runs, few expected residuals (little more than five) in each of the final m intervals, and (2) having, in most runs, too small a number m such that the degrees of freedom of Chi-square, m-K-3, would be non-positive.

VI.1.d IVOR

In this section the basic steps of the computational procedure of IVOR ("Independent Variable Ordering by Regression sums of squares") are explained. The principles of this ranking method and its

applications, along with those of BIVOR, were already discussed in Section III.2; whereas the computational details in the subroutine IVOR (including the relevant checks for the acceptability of a rerun and internal decisions based on these checks) are given in Section VI.2.c.

The N independent variables (OCIV's and GCIV's, or OCIV's only) in the preconceived model of a regression problem are optionally divided into M_I consecutive groups according to the IV input sequence, with N_j independent variables in the respective groups, $j = 1,\ldots,M_I$. The primary purpose of the grouping option is to allow the possibility of ranking IV's under "restricted admissibility." (This type of ranking has several applications as discussed in Sections II.3 and VII.2.a.) Another use of the grouping feature is as a device to save computing time; see the remarks at the end of this section (VI.1.d). Not all N IV's in a given regression problem need be included in the grouping. If the total number,

$$\sum_{j=1}^{M_I} N_j,$$

of the independent variables in the M_I groups is less than N, the last (or rightmost)

$$N - \sum_{j=1}^{M_I} N_j$$

independent variables are excluded from the IVOR ordering. If the user does not want to use the grouping at all, he should put all IV's in one group, i.e., let $M_I = 1$ and $N_1 = N$. (See input preparation for Card Type 4, Section V.2.)

IVOR starts the ordering within the first (or leftmost) group of N_1 IV's and, after having completed the ordering within that group, proceeds to the second group and further to the right until the ordering is completed within all M_I groups.

For the present description only, the IV's of group j, $j = 1,\ldots,M_I$, are denoted by $x^{(j)}$, $h = 1,\ldots,N_j$. With this notation, the first N_1 steps of IVOR are:

First Step. Each of the N_1 IV's of the first group ($x^{(1)}$; $h = 1,\ldots,N_1$) is included in the model, one at a time, as the only independent variable in the model. For each IV the ASSR value (Regression Sum of Squares Adjusted for the mean) is computed. Among these N_1 ASSR values the maximum is found and the independent variable whose inclusion in the model led to the maximum is denoted as $x^{(1)}_{\{1\}}$. Accordingly, $x^{(1)}_{\{1\}}$ is considered as the most important IV in the first group.

Second Step. Each of the N_1-1 IV's of the first group which have not yet been ranked ($x_h^{(1)}$; h = 1,...,N_1 but $\neq$ (1)) is included in the model, one at a time, together with $x_{\{1\}}^{(1)}$, the IV ranked most important in the first step. That is, in the second step, the model always contains two IV's, of which one is $x_{\{1\}}^{(1)}$. Then the N_1-1 ASSR values due to the N_1-1 sets of two IV's are computed and the maximum is found. The independent variable which, in union with $x_{\{1\}}^{(1)}$, led to this maximum ASSR value is denoted as $x_{\{2\}}^{(1)}$ and is considered as the second most important IV in the first group.

Third Step. Each of the N_1-2 IV's of the first group which have not yet been ranked ($x_h^{(1)}$; h = 1,...,N_1 but $\neq$ (1) and (2)) is included in the model, one at a time, together with $x_{\{1\}}^{(1)}$ and $x_{\{2\}}^{(1)}$. Then the N_1-2 ASSR values due to the N_1-2 sets of three IV's are computed and the maximum is found. The independent variable which, together with $x_{\{1\}}^{(1)}$ and $x_{\{2\}}^{(1)}$, led to this maximum is denoted as $x_{\{3\}}^{(1)}$ and considered as the third most important IV in the first group.

Step 4 to Step N_1. The procedure is continued, corresponding to Steps 1-3, until $x_{\{N_1-1\}}^{(1)}$ is found in Step N_1-1. In step N_1, the remaining IV in the first group is, naturally, considered to be the least important one and is denoted as $x_{\{N_1\}}^{(1)}$.

The remaining steps of IVOR are as follows:

Step N_1+1. Each of the N_2 IV's of the second group ($x_h^{(2)}$; h = 1,...,N_2) is included in the model, one at a time, together with all N_1 IV's of the first group. Then the N_2 ASSR values are computed, each one due to N_1+1 IV's. Among these N_2 ASSR values the maximum is found and the independent variable of the second group whose inclusion in the model led to this maximum, is denoted as $x_{\{1\}}^{(2)}$. This IV is considered as the most important independent variable in the second group.

Steps (N_1+2) to (N_1+N_2) follow correspondingly.

The procedure is continued with the third group, fourth group, etc., until all independent variables in all groups have been ranked.

The procedure thus described may be called the "standard" IVOR procedure. However, since the number of matrix inversions and relevant computations performed by the "standard" IVOR routine may result in excessive computer time, an input parameter, IQ (columns 1 and 2, Card Type 4), is available for possible use in limiting the number of IV's to be ordered by IVOR. If IQ > 0, only the IQ most important independent variables will be found, i.e., ordered by IVOR

under this option, and the N-IQ least important IV's will not be ordered at all. IQ must fulfill the inequality

$$IQ \leq \sum_{j=1}^{M_I} N_j,$$

but can otherwise be chosen freely, such that, for example, the ordering may cease after some IV's of a given group and all IV's in the previous group(s) have been ordered. For example, with

$$IQ = \sum_{j=1}^{j^*} N_j + 3,$$

where $N_{j^*+1} > 3$, say, IVOR will first order the $\sum_{j=1}^{j^*} N_j$ independent variables in the first j* groups as described above. Then it will find, among all N_{j^*+1} IV's of group j*+1, the three most important ones in the usual manner and cease ordering. The last $N_{j^*+1}-3$ IV's in group j*+1 and all IV's in the subsequent groups will be left unordered.

Two remarks should be made with respect to the grouping feature in the IVOR procedure.

The first concerns its use as a means to rank IV's under restricted admissibility. Namely, the sequence in which the IV's, especially GCIV's, are input to the program is critical when the grouping option is exercised for this purpose. Since the allocation of the IV's to the various groups is performed according to the input sequence, it is necessary to input first all those IV's which would be admissible for ranking at the first step of IVOR and, therefore, would define the first group. In general, these would be the OCIV's, that is, IV's with a powersum of 1. In general, all IV's with a powersum of 2 would follow next, that is, all GCIV's representing terms of second order; etc. In other words, the GCIV's would have to be specified in the sequence indicated in the example given for Card Type 3 (see Section V.2).

The second remark concerns the use of the grouping feature as another device (along with the IQ feature) to save computing time. One such time saving effect is achieved by specifying

$$\sum_{j=1}^{M_I} N_j < N,$$

provided the user is willing to save time by not ranking the

$$N - \sum_{j=1}^{M_I} N_j$$

rightmost IV's. Also, the user can group the IV's by some preconceived scale of importance which, in case of GCIV's being present, may or may not be the grouping required for ranking under restricted admissibility. Computing time is saved because the IVOR ordering always takes place within only one group at a time, which leads to fewer matrix inversions and relevant computations than would be necessary when the IV's were not grouped. Again, the user has to specify the input order of IV's such that this grouping by preconceived importance is possible. When choosing time saving devices in IVOR, the user should clearly distinguish between the consequences of using IQ and the grouping feature.

The program user should be aware that whenever he applies the grouping feature (with $M_I > 1$), IVOR will give a ranking of independent variables, by prediction power for the dependent variable, within only the designated groups of IV's. This ranking may be called "sub-ranking", in contrast to the ranking when all IV's are considered to be in one group ($M_I = 1$). (See also the discussion of the ranking results for the example problem in Section VI.5.)

VI.1.e BIVOR

The computational procedure of BIVOR ("Backward Independent Variable Ordering by Regression sums of squares") is based on principles similar to those of IVOR which were discussed in the last section. In the present section, therefore, the essential steps of BIVOR are given while reference is often made to Section VI.1.d.

The optional grouping of IV's is done in the same manner as in IVOR; however, the number, M_B, of groups in BIVOR and the numbers, N_q, of IV's in the groups ($q=1,\ldots,M_B$) may be different from M_I and the N_j of IVOR, respectively, when both options, IVOR and BIVOR, are exercised. Also in BIVOR, the

$$N - \sum_{q=1}^{M_B} N_q$$

rightmost IV's may be excluded from the ordering. As to the use of the grouping feature in BIVOR, see the remarks at the end of the present section.

For the following description it will be assumed that

$$\sum_{q=1}^{M_B} N_q = N,$$

which does not affect the general validity of the description. BIVOR starts the ordering within the last (or rightmost) group of N_{M_B} IV's and, after having completed the ordering within that group, proceeds to the next to last group and further to the left until the ordering is completed within all M_B groups. In more detail, the first $N_{M_B}=N_M$ steps of BIVOR are as follows. (For clarity and for the rest of the present section only, the subscript "B" (for BIVOR) will be eliminated from all terms such that M_B becomes M and N_{M_B} becomes N_M.)

First step. All

$$\sum_{q=1}^{M} N_q = N$$

independent variables are included in the model and the corresponding matrix of the normal equations is inverted. Then the N_M additional regression sums of squares, $SS_{N-(N-1)} = SS_1$, which are due to each of the N_M IV's contained in the last group, are computed. Their values are obtained by computing $[b_v^{(1)}]^2/c_{vv}^{(1)}$, (see Hader and Grandage [1958], p. 126), where the $b_v^{(1)}$ are the regression coefficients of the N_M IV's in the last group, and the $c_{vv}^{(1)}$ are the corresponding main diagonal elements of the inverse matrix. Of these N_M SS_1 values the minimum is found and the IV whose deletion led to it is denoted as $x_{(1)}^{(M)}$. Accordingly, this independent variable is ranked as the least important one in the last group. Notice that this IV which was ranked first, as the least important one, received the subscript "(1)." In IVOR it was the most important IV which received the subscript "(1)." This convention is correspondingly applied in the following steps of BIVOR.

Second Step. The IV found least important in the first step, $x_{(1)}^{(M)}$, is deleted from the model and the matrix of the normal equations corresponding to the N-1 IV's remaining in the model is inverted.

In order to find the minimum of the N_M-1 values $SS_{N-(N-2)} = SS_2$, due to the least important IV found in the first step plus any one of the N_M-1 IV's not yet ranked in the last group, the following relation is used. By the additivity property of additional regression sums of squares one has $SS_2 = SS_1^{(1)} + SS_1^{(2)}$, where $SS_1^{(1)}$ is due to the least important IV in the last group, $x_{(1)}^{(M)}$, and $SS_1^{(2)}$ is the additional regression sum of squares (after $x_{(1)}^{(M)}$ is deleted from the model) due to any one of the N_M-1 IV's not yet ranked in the last group.

Since $SS_1^{(1)}$ is a constant in the search for the minimum of SS_2, only the N_M-1 $SS_1^{(2)}$ values need be searched for the minimum. These values are obtained in the program by computing the terms $[b_v^{(2)}]^2/c_{vv}^{(2)}$, where the $b_v^{(2)}$ are the regression coefficients (at the second step) of the N_M-1 IV's and the $c_{vv}^{(2)}$ are the corresponding main diagonal elements of the inverse matrix. Of these N_M-1 values the minimum is found and the IV whose deletion led to it is denoted as $x_{(2)}^{(M)}$. Accordingly, this IV is ranked as the next-to-least important one in the last group.

Step 3 to Step N_M. The procedure is continued, corresponding to the first two steps, until $x_{(N_M-1)}^{(M)}$ is found in Step N_M-1. In Step N_M, the remaining IV in the last group is, naturally, considered to be the most important one and is denoted as $x_{(N_M)}^{(M)}$.

The remaining $N-N_M$ steps of BIVOR are as follows:

Step N_M+1. All $N-N_M$ IV's are included in the model and the corresponding matrix of the normal equations is inverted. The minimum of the additional regression sums of squares, $SS_{N-(N-N_M-1)} = SS_{N_M+1}$, is found by searching for the minimum of the values $[b_v^{(N_M+1)}]^2/c_{vv}^{(N_M+1)}$. Here, the $b_v^{(N_M+1)}$ are the regression coefficients (at Step N_M+1) of each of the N_{M-1} IV's of Group M-1 and the $c_{vv}^{(N_M+1)}$ are the corresponding main diagonal elements of the inverse matrix. The IV whose deletion (from Group M-1) led to the minimum is denoted as $x_{(1)}^{(M-1)}$ and is ranked as the least important one in Group M-1.

Steps (N_M+2) to (N_M+N_{M-1}) follow correspondingly.

The procedure is continued through the remaining M-2 groups until all independent variables in all groups have been ranked.

The additional regression sums of squares as computed in BIVOR deserve some more discussion. The quantity

$$\frac{b_v^2}{c_{vv}}$$

equals the familiar numerator in the F statistic to test the hypothesis $\beta_v = 0$ in a model containing, say, N' IV's:

$$F = \frac{b_v^2}{c_{vv}} \Big/ s^2. \qquad \text{(VI-17)}$$

In other words, the quantities b_v^2/c_{vv} used in BIVOR to find the least important IV in a given group at a given step (with a model containing N' IV's), are equal to the quantities used to test, in the familiar manner and one at a time, the significance of the N' regression coefficients. However, because of the correlations that generally exist among all the N' IV's, one would not obtain a meaningful ordering

of IV's if the F values (VI-17) of the IV's were computed and ranked according to their magnitudes. Therefore, at a given step of BIVOR, the least significant of these quantities is selected and the corresponding IV is deleted from the model, whereupon at the next step, again the smallest of the b_v^2/c_{vv} quantities is found and again the corresponding IV is deleted from the model, and so on. This process then leads to the BIVOR ranking of independent variables by prediction power for the dependent variable, as described.

Because of the possible existence of compounds (see Section III.2) the minimum values of the b_v^2/c_{vv} quantities can vary considerably from one step of BIVOR to the next. In fact, once a significant model has been found based on the BIVOR ordering and on the main theorem F value, (III-1), independent variables ranked as "more important" could very well have b_v^2/c_{vv} values which are much smaller than the one corresponding, for example, to the "least important" IV of the significant model. This would appear as if less significant IV's were ranked as being more important than the more significant IV's. However, this conclusion is wrong, and the right conclusion should be that a compound is present.

As in IVOR, the grouping feature in BIVOR can be used as a means to rank IV's under restricted admissibility. This grouping is done in much the same way as was discussed in the last section (VI.1.d) and has the same possible consequences with respect to "subranking" as were mentioned there.

In BIVOR, the grouping option is, besides its application to ranking under restricted admissibility, the only device available to save computing time. The fact that not all N IV's of the preconceived model need be included in the grouping makes the time saving possible. With

$$\sum_{q=1}^{M} N_q < N$$

the last

$$N - \sum_{q=1}^{M} N_q$$

independent variables will be excluded from the BIVOR ordering.

VI.2 Computational Details

In this section the computational details of DA-MRCA are described for one regression problem. The intention is not to give a description of the details contained in the flow charts (Section VIII.2) or in the program listing (Section VIII.4), but rather to describe the more important computations and decisions made by the program, inasmuch as they are not discussed in previous sections. Also, justifications are given for some of these details where considered to be helpful in understanding the program. Along with the description, all possible statements are quoted which may result from computational decisions and appear as printout. Whenever mention is made that the "program stops", this refers to the one regression problem being processed, if not otherwise stated. In this case, should there be more than one regression problem to be processed by DA-MRCA, the program would go to the next problem.

Generally, the order in which the computational details are described is the order in which they are performed by the program. In some places this order is not kept for the purpose of a better understanding of the description.

References to subroutine names are not made since in some instances the same type of computation is executed, at different places, by different subroutines. The interested reader is referred to the flow charts in Section VIII.2.

The computation and use of the "Analysis of Variance Tables" and of the "Final Comprehensive Analysis Table" are not discussed in this section. This is done only in Section VI.3.b.

VI.2.a Main Run

In this section the computational details of the main run are given. However, most of these computations are correspondingly performed for any rerun. (See Sections VI.2.b - VI.2.d.)

VI.2.a.(1) Initial Operations

The operations described in this section are performed only once per regression problem, i.e., they are performed for the main run but are not repeated if reruns are included in the regression problem.

A. If the total number, IR+IS=N, of independent variables input is ≤ 0 or > 51, the program stops and the statement "CARD TYPE 2 IS INCORRECT" is printed. Otherwise ($0 < N < 51$) the program continues.

B. If the number, n, of data points input is ≤ 1 or > 7000, the program stops and the statement "TOO FEW OR TOO MANY DATA POINTS" is printed. Otherwise ($1 < n \leq 7000$) the program continues.

C. The summation matrix is computed. However, only the elements of the main diagonal and those above the main diagonal are actually computed. Since the summation matrix is symmetrical, the elements below the main diagonal are merely copied from those above the diagonal.

VI.2.a.(2) Matrix Inversion and Accuracy Checks

The operations described in the following paragraphs A - I are performed for the main run and, in general, for any rerun. The computations are expressed in terms of K independent variables contained in the model, where K=N defines the main run and K=N' < N defines a rerun with N' IV's contained in the model.

A. The inverse of the (K+1)x(K+1) matrix A, i.e., the inverse, A^{-1}, of the matrix of the normal equations, is computed. (The computational procedures involved in the matrix inversion, the computation of the determinant and the solution of the normal equations are explained in detail in Section VI.1.a.)

B. The determinant of A is tested and if found to be non-positive, the statement "MATRIX FAILED TO INVERT" is printed. For this case, and in the main run only, the averages of the N IV's and of the dependent variable are computed and printed and the program goes to reruns (if any). Also, if the determinant is non-positive for the main run, there will be no final comprehensive analysis for any type of reruns (HAND selected, IVOR, or BIVOR), and the following statement is made at the end of the printout of the regression problem: "NO FINAL COMPREHENSIVE PRINTOUT SINCE MATRIX FOR MAIN RUN COULD NOT BE INVERTED." - In case of a hand selected rerun, the program goes to the next hand selected rerun (if any). In case of an IVOR or BIVOR rerun, see Sections VI.2.c or VI.2.d, respectively. - If the determinant is positive, its value is printed, along with the inverse matrix and the solution to the normal equations (regression coefficients).

C. The following values are computed:

(a) The error sum of squares,

$$SSE = E_{yy} - \sum_{\nu=0}^{K} b_\nu E_{\nu y}$$

(where $E_{yy} = \sum_{i=1}^{n} y_i^2$ and $E_{\nu y} = \sum_{i=1}^{n} x_{\nu i} y_i$, with $x_{0i} \equiv 1$);

(b) The total sum of squares adjusted for the mean, $ATSS = E_{yy} - \frac{1}{n} E_{0y}^2$;

(c) The regression sum of squares (due to K independent variables) adjusted for the mean,

$$ASSR_K = \sum_{\nu=0}^{K} b_\nu E_{\nu y} - \frac{1}{n} E_{0y}^2;$$

(d) The square of the correlation coefficient, i.e., the coefficient of determination,

$$R^2 = \frac{ASSR_K}{ATSS}.$$

D. R^2 is tested and if found to be negative, the statement "SQUARE OF CORRELATION COEFFICIENT IS NEGATIVE" is printed. For this case, and in the main run only, the operations concerning the averages and the final comprehensive analysis are performed as described in paragraph B above. - In case of a hand selected rerun, the program goes to the next one (if any). In case of IVOR or BIVOR, see Section VI.2.c or VI.2.d, respectively. - If $R^2 \geq 0$, the correlation coefficient (R) is computed and printed.

E. The residual variance (s^2) is computed by dividing SSE by n-K-1. The residual variance is then tested and if found to be negative, the statement "VARIANCE IS NEGATIVE" is printed. For this case, and in the main run only, the operations concerning the averages and the final comprehensive analysis are performed as described in paragraph B above. - In case of a hand selected rerun, the program goes to the next one (if any). In case of IVOR or BIVOR, see Section VI.2.c or VI.2 d, respectively. - If s^2 is found to be non-negative, the square root of the residual variance (s) is computed and printed. (If the quantity n-K-1=0, s is set equal to zero and the F value of the ANOVA table is printed as all nines. This is the case of the "zero error perfect fit.")

F. The elements of the main diagonal of the inverse matrix (the $c_{\nu\nu}$) are tested. The first element found to be negative (if any) results in the statement "AN ELEMENT OF THE MAIN DIAGONAL OF THE INVERSE MATRIX IS NEGATIVE." For this case, and in the main run

only, the operations concerning the averages and the final comprehensive analysis are performed as described in paragraph B above. In case of a hand selected term, the program goes to the next one (if any). In case of IVOR or BIVOR, see Section VI.2.c or VI.2.d, respectively. - If there are no negative elements on the main diagonal, the standard deviations of the regression coefficients are computed:

$$\sqrt{\hat{V}[b_\nu]} = s\sqrt{c_{\nu\nu}}, \quad \text{where } \nu = 0,1,\ldots,K.$$

G. The elements of the calculated identity matrix (I_c), the $i_{\nu\nu'}$ ($\nu,\nu' = 0,1,\ldots,K$) are obtained by forming the product of the inverse matrix (A^{-1}) and the matrix of the normal equations (A), in this order. The identity matrix is used for checking the accuracy of the inversion process. The specifics of this use and their justifications are discussed in Section VI.1.b.

H. The absolute values of the deviations from 1 of the main diagonal elements of I_c are tested against I(1). The first deviation found to be $\geq$ I(1) (if any) is tested to determine if it is also $\geq$ I(2). If it is, the identity matrix is printed with the statement "DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MATRIX LARGER THAN I(2) = ... RUN REJECTED." In the blank the input value of I(2) is printed. In this case the program goes directly to the operations described in Section VI.2.a.(3). If the first deviation which is $\geq$ I(1) is not $\geq$ I(2), the testing is continued on the remaining diagonal elements. If any of the deviations of the main diagonal elements are $\geq$ I(1) but none of these deviations is $\geq$ I(2), the identity matrix is printed with the statement "DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MATRIX LARGER THAN I(1) = ... BUT LESS THAN I(2) = ... RUN ACCEPTED." In the blanks the input values of I(1) and I(2) are printed.

I. If all deviations (absolute) of the main diagonal elements are $<$ I(1), the absolute values of the off-diagonal elements are tested. The first time that an off-diagonal element (absolute) is $>$ I(1), the identity matrix is printed with the statement "DEVIATIONS OF ALL MAIN DIAGONAL ELEMENTS IN THE IDENTITY MATRIX SMALLER THAN I(1)=.... DEVIATION OF AN OFF-DIAGONAL ELEMENT LARGER THAN I(1). RUN ACCEPTED." If all off-diagonal elements also have absolute values $<$ I(1), the identity matrix is <u>not</u> printed, but the statement "DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLER THAN I(1) = ... RUN ACCEPTED" <u>is</u> printed.

VI.2.a.(3) Predicted Values, Prediction Errors, Normality Test, and Averages

The following operations A - I are always performed for the main run and are optionally performed for reruns. As in the last section, the operations are expressed in terms of K independent variables contained in the model.

A. The n predicted values (the $\hat{Y}_i$) are computed by evaluating the obtained regression equation for each of the n input design points.

B. The prediction errors, $\hat{e}_i = y_i - \hat{Y}_i$, are computed for each input design point by subtracting the predicted value from the observed value of the dependent variable. The normality test described later in this section is performed on these prediction errors. Some general aspects of the test are discussed in Section VI.1.c.

C. The sum of squares of the prediction errors,

$$\sum_{i=1}^{n} (y_i - \hat{Y}_i)^2,$$

is computed. This sum of squares should equal the error sum of squares, SSE, given in Section VI.2.a.(2), paragraph C, and is identified, when printed, as the "CHECK ERROR SUM OF SQUARES." The check error sum of squares is computed as an additional check on the computational accuracy. Since the values $\hat{e}_i = y_i - \hat{Y}_i$ are already computed, this check is inexpensive. However, no sensing is built into the program to compare the two error sums of squares.

D. The maximum and minimum of the n prediction errors are found and the range (= the maximum prediction error minus the minimum prediction error) is computed. The range is then divided by 30 to give the common length (D) of the 30 intervals used in the prediction error frequency distribution. The upper bounds of each of the 30 intervals are computed by adding D, 2D, 3D, . . ., 30D, respectively, to the minimum prediction error. Thereby, the maximum prediction error becomes the upper bound of the last interval.

Each prediction error is then assigned to its proper interval, i.e., to the interval with the smallest upper bound which is not exceeded by the prediction error. A count is then made of the number (f'_j) of prediction errors observed in each of the 30 intervals. The f'_j are used in the bar chart of the printout, see the following paragraph (E).

E. The quantity $\frac{n}{5} - (K+3)$ is computed and checked. If this quantity is ≥ 0, the bar chart is printed, along with the

statement "CHI SQUARE COULD NOT BE COMPUTED." For this case, and in the main run only, the program goes to the operations described in paragraph I below. - In case of a hand selected rerun, the program goes to the operations described in Section VI.2.a.(4), should the option for selected and/or synthetic design points be exercised. - This check is a joint consequence of (1), the restriction that $\hat{\varphi}_j$, the expected number of observations in an interval, should be greater than 5 and (2), the definition of the degrees of freedom for the Chi-square statistic as the number of intervals, for which $\hat{\varphi}_j > 5$, minus K+3. The circumflex on $\hat{\varphi}_j$ is used to express the fact that these expected frequencies are based on the estimates of the mean and the standard deviation of the distribution of the prediction errors. If the quantity $\frac{n}{5}$ - (K+3) is ≤ 0, the degrees of freedom for Chi-square could never be >0 and further computations would be meaningless. The restriction on $\hat{\varphi}_j$ and the degrees of freedom for Chi-square are more fully discussed in the following paragraph F.

F. If the quantity $\frac{n}{5}$ - (K+3) is > 0, an attempt is made to compute the Chi-square statistic. The expected frequency distribution is formed. This distribution gives the number of prediction errors that would be expected in each of the 30 intervals if the sample of n prediction errors was actually from a normal distribution having a mean and standard deviation equal to those of the observed prediction errors. Since the expected frequency in each interval is computed by a system subroutine which uses the standardized normal distribution function, the 30 upper bounds must be standardized by dividing each upper bound by s. (The average of the observed prediction errors is zero and, consequently, is not subtracted in standardizing the upper bound.) The expected frequency in each of the 30 intervals is obtained by multiplying the number of data points, n, by the probability, obtained from the standard normal tables, that an observation will be in a given interval. The expected frequencies in each of the 30 intervals are then examined and, if necessary, some of the intervals are combined in order that each of the resulting m intervals has an expected frequency of more than 5. If, for example, the expected frequency in the first of the 30 intervals is ≤ 5, the frequency is added to that of the next interval. This procedure is continued until the first time a new interval results which does have an expected frequency of more than 5. Succeeding intervals are similarly tested and, if necessary, combined. If the last interval, or intervals, does not have an expected frequency of more than 5, it is combined with the last interval which did have a frequency of more than 5. In this way m "new" intervals are formed, each of which has an expected frequency, $\hat{\varphi}_j$, greater than 5.

G. The number (f_j) of observed prediction errors is counted for each of the m intervals and the contribution to Chi-square is computed for each interval. The contribution for the

j^{th} interval is

$$\frac{1}{\hat{\varphi}_j}(f_j - \hat{\varphi}_j)^2,$$

where f_j and $\hat{\varphi}_j$ are as defined above. These contributions to Chi-square are then printed for each of the m intervals, along with the observed and expected number of observations in that interval.

H. The quantity m-K-3 is computed. If m-K-3 $\leq$ 0, the statement "CHI SQUARE COULD NOT BE COMPUTED" is printed. In this case the program continues as described in paragraph E above.

If m-K-3 > 0, the Chi-square statistic is computed by summing the individual contributions over the m intervals.

I. Only in the main run are the averages of the N independent variables and of the dependent variable computed and printed.

VI.2.a.(4) Predicted Values and Prediction Standard Deviations at Selected Input and/or Synthetic Design Points

If the run (main run or hand selected rerun) passed all tests in paragraphs B, D, E, and F of Section VI.2.a.(2), and if selected input and/or synthetic design points are present (see columns 8-13, Card Type 2, Section V.2), the coordinates of the OCIV's of these points are printed and the corresponding predicted values and prediction standard deviations for either individual observations or for the prediction line are computed and printed. If the run did not pass the four tests mentioned above, predicted values and prediction standard deviations cannot be obtained for either selected input or synthetic design points.

VI.2.b Hand Selected Reruns

In order to execute a hand selected rerun (if any are specified) the program deletes the proper rows and columns from the summation matrix according to the specified independent variable selection of K = N' < N IV's. The operations described in Section VI.2.a.(2) are then performed for this IVS (with the exceptions mentioned there). If NPE=1, (column 15, Card Type 2), the operations of paragraph A - H of Section VI.2.a.(3) are also performed for this IVS.

Predictions and prediction standard deviations for selected input and/or synthetic design points are computed only when

the option is exercised and when the hand selected IVS passed all tests described in paragraphs B, D, E, and F of Section VI.2.a.(2).

VI.2.c IVOR

In this section the computational details which are performed to arrive at an IVOR ordering of independent variables are described. (The IVOR ordering is explained in Section VI.1.d.)

If

$$\sum_{j=1}^{M_I} N_j = N,$$

only the first N-1 steps of IVOR are performed since the main run has already been performed. There is no possibility in IVOR to call, in each IVOR rerun, for predictions and prediction standard deviations at selected input and/or synthetic design points. As indicated before, if the main run fails any of the tests performed on the determinant, R^2, s^2, and the $c_{\nu\nu}$'s (as described in paragraphs B, D, E, and F of Section VI.2.a.(2)), there will be no IVOR Final Comprehensive Analysis.

At any given step of IVOR (where "step" is as defined in Section VI.1.d) the following operations are performed:

A. The established IVOR model of the preceding step is augmented by one independent variable at a time. There may be left, say, H IV's not yet ordered within the group in which IVOR is presently operating. Each of the H IV's is added, one at a time, to the IVOR model of the preceding step by deleting one less row and column from the summation matrix than in the previous step. Each of the H corresponding matrices of the normal equations (A) is then inverted and its determinant computed.

B. The procedure to decide whether or not to accept any of the H independent variable selections for further consideration at this step depends upon whether the main run was accepted or rejected. ("Acceptance" is defined as passing all 5 tests described in paragraphs B, D, E, F, and H of Section VI.2.a.(2). "Rejection" is defined as failing one or more of these tests.)

(B.a) If the main run was accepted: The determinant is checked for each of the H IVS's and if found to be non-positive, this IVS is excluded from further consideration at this step. For all IVS's with non-positive determinants the statement "MATRIX FAILED TO INVERT, IVS ....." is printed, where the blank is filled by the identification of the IVS. For all IVS's whose determinant is found to be positive the ASSR value is computed. Should all H determinants be non-positive the statement "NO VALID ASSR'S WERE COMPUTED" is printed and the IVOR ordering is terminated.

(B.b) If the main run was rejected: R^2 and s^2 are computed for each one of the H IVS's. R^2 and s^2 are then tested to determine if either of them is negative, and the determinant is tested to determine if it is non-positive. If a failure occurs, the statements concerning the determinant, R^2, and s^2 as given in paragraphs B, D, and E of Section VI.2.a.(2) are printed along with the IVS identification. These IVS's are excluded from further consideration at this step. Then the operations described in paragraphs F, G, H, and I of Section VI.2.a.(2) are performed for each one of the H or the remaining IVS's. If for a given IVS an element of the main diagonal of the inverse matrix is found to be negative, the appropriate statement is printed and this IVS is excluded from further consideration at this step. If an IVS has to be excluded from further consideration because an element of the main diagonal of the identity matrix has an absolute deviation from 1 greater than I(2), the appropriate statement is printed together with the identification of the IVS. (The other possible statements concerning the elements of the identity matrix are printed only when the IVS is later chosen as the established IVOR model of this step.) If none of the H IVS's could be accepted, IVOR stops and prints "NO VALID ASSR'S WERE COMPUTED."

C. If, in either case of paragraph B (above), only one IVS of the H considered led to a valid ASSR value, this IVS represents the established IVOR model at this step. In other words, the individual IV whose inclusion led to the only valid ASSR value is ordered as the independent variable with the maximum contribution to the "total" regression sum of squares at this step. For this IVS, all pertinent printouts are given. Also computed and printed for this IVS, provided the option is exercised for reruns, are the predicted values, the prediction errors and the normality test as described in Section VI.2.a.(3). IVOR then goes to the next step (if there is any).

D. If more than one IVS in paragraph B (above) led to a valid ASSR value, these values are compared among themselves as follows. The valid ASSR value corresponding to the IVS with the leftmost IV added to the model of the preceding step is denoted as $ASSR^{(1)}$. Then for each of the remaining valid ASSR values (the $ASSR^{(i)}$'s, say) the following quantities are computed:

$$\Delta_i = \left| \frac{ASSR^{(i)} - ASSR^{(1)}}{ASSR^{(1)}} \right|$$

(D.a) If none of the quantities Δ_i exceeds the fixed value $.5 \times 10^{-8}$, all of the ASSR's are considered to be equal and a "perfect fit" is considered to have been reached. (When a perfect fit is being reached, each IV contributes the same additional regression

sum of squares towards the ASSR value of this perfect fit.) The left-most IV is then defined as the most important IV ordered at this step, and a complete printout (as discussed in paragraph C above) is given for the corresponding IVS, along with the statement "PERFECT FIT. IVS = ..." The IVOR subroutine then stops completely.

(D.b) If one or more of the quantities Δ_i exceeds the value $.5 \times 10^{-8}$, the maximum ASSR value is found and the IV which led to the maximum is considered as the most important IV at this step. A complete printout (as in paragraph C above) is given for the corresponding IVS, and the IVOR subroutine goes to the next step (if there is any).

VI.2.d BIVOR

In this section the computational details which are performed to arrive at a BIVOR ordering of independent variables are described. (The BIVOR ordering is explained in Section VI.1.e.)

If $\sum_{q=1}^{M_\theta} N_q < N$, BIVOR deletes the last $(N - \sum_{q=1}^{M_\theta} N_q)$ independent variables from the model of the main run by deleting the corresponding rows and columns from the summation matrix. BIVOR then starts the ordering by inverting the matrix with $\sum_{q=1}^{M_\theta} N_q$ independent variables contained in the model.

There is no possibility in BIVOR to call, in each BIVOR rerun, for predictions and prediction standard deviations at selected input and/or synthetic design points. As indicated before, if the main run failed any of the tests performed on the determinant, R^2, s^2, and the c_{vv}'s (as described in paragraphs B, D, E, and F of Section VI.2.a.(2)), there will be no BIVOR Final Comprehensive Analysis.

The operations at any given step of BIVOR (where "step" is as defined in Section VI.1.e) are dependent upon whether or not the preceding step led to an accepted BIVOR rerun.

A. If the main run was rejected and all preceding steps of BIVOR (if any) led to rejected reruns, the operations are as follow:

(A.a) From the BIVOR model (which was rejected) of the preceding step the rightmost IV is deleted by deleting the corresponding row and column from the matrix of the normal equations (A) of the preceding step. Then the elements of the inverse matrix A^{-1}, the determinant of A, R^2, and s^2 are computed. These values are subjected to the respective tests described in paragraphs B, D, E, and F of Section VI.2.a.(2). If the new IVS fails any of these tests, again the rightmost IV is deleted from the model for the next step and the checks are repeated for the new model. - If the new IVS passes all 4 tests, the operations of the next paragraph (A.b) are performed.

(A.b) The identity matrix, $I_c = A^{-1}A$, is computed for the present step's IVS (which passed the four checks mentioned in the last paragraph). Then the checks as described in paragraphs H and I, Section VI.2.a.(2), are performed on the elements of I_c. The first time a main diagonal element of I_c has an absolute deviation from 1 which is greater than I(2), the IVS of the present step will be rejected. However, in this case this IVS will be given a complete printout, including the predicted values, prediction errors and normality test (Section VI.2.a.(3)). The reason for this treatment is that the value of I(2) is, after all, an optional input value chosen by the program user and that the IVS rejected on the grounds of I(2) may be marginal in its accuracy but essentially acceptable. By having the printout for this run, the analyst is given additional information as to the possibility of reconsidering the regression problem with some of the input parameters changed. There is, in this case, a certain danger of misinterpretation of the printout. Although at each individual BIVOR rerun the statement is printed that this run is rejected, it could appear, from the final comprehensive analysis (if this is printed), as if the series of deletions from the right was a genuine BIVOR ordering of independent variables. This will occur most likely when the value of I(2) was chosen too small. - Also in this case (of the BIVOR IVS failing only the I_c test) the subroutine goes to the next step by deleting the rightmost IV from the model.

If the IVS of the present step is accepted, the operations of the next paragraph are performed.

(A.c) If the IVS of the present step was accepted, i.e., passed all five checks described in paragraphs (A.a) and (A.b) above, the additional regression sums of squares ($= b_v^2/c_{vv}$) are computed for all IV's not yet ordered in the group in which BIVOR is presently operating. If there are more than one of these additional regression sums of squares, the minimum is found and the IV which led to it is ranked as the least important one at this step. Since the accepted

IVS of this step represents the first accepted rerun of BIVOR, it is given the complete printout, including predictions, prediction errors, and the normality test. BIVOR then goes to the next step (if any), as described for this case in the next paragraph (B).

B. If the main run and/or the IVS of any previous step has been accepted, BIVOR goes to the next step by computing the additional regression sums of squares for all IV's which have not yet been ordered in the group in which BIVOR is presently operating. The values are compared and the IV which led to the minimum additional regression sum of squares is deleted from the model. The matrix A of this new IVS is inverted and the determinant, R^2, s^2, and I_c are computed and the corresponding tests are performed as described in paragraphs B, D, E, F, H, and I of Section VI.2.a.(2). (If the option described in paragraph C below is chosen, the tests on the elements of I_c are terminated with that rerun in which all absolute deviations of the matrix elements are $<$ I(1) for the first time.) - This BIVOR rerun is given a full printout, including the predicted values, prediction errors, and normality test if this option is exercised for reruns. The BIVOR ordering is terminated when an IVS arrived at contains only one independent variable.

C. If the option to discontinue the identity matrix checks in BIVOR is used (i.e., IBID = 1 on Card Type 2), then the identity matrix is printed for the first BIVOR rerun in which the absolute values of all deviations are $<$ I(1), together with the statement "DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLER THAN I(1) = ... RUN ACCEPTED. NO IDENTITY MATRIX CHECKS WILL BE MADE ON SUBSEQUENT BIVOR RUNS." Accordingly, for ensuing reruns in a BIVOR sequence the identity matrix is not computed and no checking is done. The purpose of this option in BIVOR is to save computer time. Since each subsequent BIVOR IVS contains only a subset of the independent variables contained in the model of the rerun in which the checking ceased, the assumption is made that, in the great majority of cases, in all subsequent BIVOR runs all absolute deviations of the elements of the identity matrix would be $<$ I(1).

VI.3 Printout

In this section the general formulation of the printout is given, supplemented by comments when considered necessary for clarification. (The comments are contained in Section VI.3.b.)

VI.3.a Formulation of Printout

This section contains the algebraic formulation of the printout of DA-MRCA. The printout for one regression problem is divided into four parts:

(1) Basic Information

(2) Main Body

(3) Analysis of Variance Tables

(4) Final Comprehensive Analysis Table.

The "Basic Information" part is printed only once per regression problem and contains

(A) a printout of the problem parameters input on Card Types 1 - 6,

(B) the data matrix, and

(C) the summation matrix.

The second part, the "Main Body" printout, contains

(A) all information pertaining to the matrix inversion,

(B) various statistics

(C) predicted values, prediction errors, normality test, and averages, and

(D) predicted values and prediction standard deviations at selected input and/or synthetic design points (optional).

The main body is printed for the main run and for each rerun, except for specific options which are not called or cannot be called for a rerun. The third part contains the "Analysis of Variance Tables" for the main run and for all reruns. The "Final Comprehensive Analysis Table" is printed as the fourth and last part and contains information for hand selected reruns and for IVOR and/or BIVOR, should any of these options be exercised. All wording which is shown in capital letters is actually printed by the program; all comments or general formulations printed in lower case letters and put in parentheses are either not printed at all by the program or not printed in this form.

The comments on the printout formulation are given in the next section (VI.3.b).

(Identification of problem as given on Card Type 1)

```
IR IS  NR MVP NDR MVPL NPE NDPØ TAPE IVØRGØ NFD IBID    TØLI1      TØLI2    FØRM - INPUT DATA DESCRIPTION -CARD TYPE 2
XX XX XXX XXX XXX   X   X    X    X     X    XX   X  ±.XXXE±XX ±.XXXE±XX
```

```
PRØDUCT TERM DESCRIPTIØNS -CARD TYPE 3
 XX XX XX XX XX XX XX XX XX XX/ XX XX XX XX XX XX XX XX XX XX/ XX XX ........
```

```
IQ  MI MJ(I),I=1,2,...,MI   -CARD TYPE 4
XX  XX   XX XX XX XX ....
```

```
MB LØT(I),I=1,2,...,MB      -CARD TYPE 5
XX   XX XX XX XX .......
```

```
NUMBERS ØF SELECTED INPUT DESIGN PØINTS -CARD TYPE 6
XXXX XXXX XXXX XXXX XXXX .........
```

(Columns occupied by program variables in this printout do not all agree with those specified in the input specification, Section V.2. See comment in Section VI.3.b.(1), paragraph A.)

(Identification of problem as given on Card Type 1)

1	x_{11}	x_{21}	x_{31}	$\cdots$	$x_{\nu 1}$	$\cdots$	x_{N1}	y_1
2	x_{12}	x_{22}	x_{32}	$\cdots$	$x_{\nu 2}$	$\cdots$	x_{N2}	y_2
3	x_{13}	x_{23}	x_{33}	$\cdots$	$x_{\nu 3}$	$\cdots$	x_{N3}	y_3
.	.	.	.		.		.	.
.	.	.	.		.		.	.
.	.	.	.		.		.	.
i	x_{1i}	x_{2i}	x_{3i}	$\cdots$	$x_{\nu i}$	$\cdots$	x_{Ni}	y_i
.	.	.	.		.		.	.
.	.	.	.		.		.	.
.	.	.	.		.		.	.
n	x_{1n}	x_{2n}	x_{3n}	$\cdots$	$x_{\nu n}$	$\cdots$	x_{Nn}	y_n

SUMMATION MATRIX

E_{00}	E_{01}	E_{02}	$\cdots$	$E_{0\nu'}$	$\cdots$	E_{0N}	E_{0y}
E_{10}	E_{11}	E_{12}	$\cdots$	$E_{1\nu'}$	$\cdots$	E_{1N}	E_{1y}
E_{20}	E_{21}	E_{22}	$\cdots$	$E_{2\nu'}$	$\cdots$	E_{2N}	E_{2y}
.	.	.		.		.	.
.	.	.		.		.	.
.	.	.		.		.	.
$E_{\nu 0}$	$E_{\nu 1}$	$E_{\nu 2}$	$\cdots$	$E_{\nu\nu'}$	$\cdots$	$E_{\nu N}$	$E_{\nu y}$
.	.	.		.		.	.
.	.	.		.		.	.
.	.	.		.		.	.
E_{N0}	E_{N1}	E_{N2}	$\cdots$	$E_{N\nu'}$	$\cdots$	E_{NN}	E_{Ny}
E_{y0}	E_{y1}	E_{y2}	$\cdots$	$E_{y\nu'}$	$\cdots$	E_{yN}	E_{yy}

(where $E_{\nu\nu'} = \sum_{i=1}^{n} x_{\nu i} x_{\nu' i}$; $E_{\nu y} = \sum_{i=1}^{n} x_{\nu i} y_i$; $x_{0i} \equiv 1$; $E_{yy} = \sum_{i=1}^{n} y_i^2$)

(Identification of problem as given on Card Type 1)

(INDEPENDENT VARIABLE SELECTION $\left\{\begin{array}{l}\text{HAND}\\ \text{IVOR}\\ \text{BIVOR}\end{array}\right\}$ 0--------)

MATRIX INVERSION (number) . . . EVALUATION TIME = (number of seconds)

DETERMINANT = (value)

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

$$\begin{array}{ccccccc}
c_{00} & c_{01} & c_{02} & \cdots & c_{0v'} & \cdots & c_{0k} \quad b_0 \\
c_{10} & c_{11} & c_{12} & \cdots & c_{1v'} & \cdots & c_{1k} \quad b_1 \\
c_{20} & c_{21} & c_{22} & \cdots & c_{2v'} & \cdots & c_{2k} \quad b_2 \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
c_{v0} & c_{v1} & c_{v2} & \cdots & c_{vv'} & \cdots & c_{vk} \quad b_v \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
c_{k0} & c_{k1} & c_{k2} & \cdots & c_{kv'} & \cdots & c_{kk} \quad b_k
\end{array}$$

IDENTITY MATRIX

$$\begin{matrix} i_{00} & i_{01} & i_{02} & \cdots & i_{0\nu} & \cdots & i_{0K} \\ i_{10} & i_{11} & i_{12} & \cdots & i_{1\nu} & \cdots & i_{1K} \\ i_{20} & i_{21} & i_{22} & \cdots & i_{2\nu} & \cdots & i_{2K} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ i_{\nu 0} & i_{\nu 1} & i_{\nu 2} & \cdots & i_{\nu\nu} & \cdots & i_{\nu K} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ i_{K0} & i_{K1} & i_{K2} & \cdots & i_{K\nu} & \cdots & i_{KK} \end{matrix}$$

(Printout of appropriate comment regarding the magnitude of the elements of the calculated identity matrix, see Section VI.2.a.(2).)

STANDARD DEVIATION OF COEFFICIENTS

1. $s\sqrt{c_{00}}$ 2. $s\sqrt{c_{11}}$ $\cdots$ ν+1. $s\sqrt{c_{\nu\nu}}$ $\cdots$ K+1. $s\sqrt{c_{KK}}$ (where $s = \sqrt{\text{RESIDUAL VARIANCE}}$ as formulated below.)

$$SSE = E_{yy} - \sum_{\nu=0}^{K} b_\nu E_{\nu y}$$

RESIDUAL OR ERROR SUM OF SQUARES.

$$ATSS = E_{yy} - (E_{0y})^2/n$$

TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.

$$ASSR_K = \sum_{\nu=0}^{K} b_\nu E_{\nu y} - \frac{(E_{0y})^2}{n}$$

REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.

$$R = \sqrt{\frac{ASSR_K}{ATSS}} = \sqrt{\frac{\sum_{\nu=0}^{K} b_\nu E_{\nu y} - \frac{(E_{0y})^2}{n}}{E_{yy} - \frac{(E_{0y})^2}{n}}}$$

CORRELATION COEFFICIENT (R).

$$s = \sqrt{\frac{ATSS - ASSR_K}{n-K-1}} = \sqrt{\frac{E_{yy} - \sum_{\nu=0}^{K} b_\nu E_{\nu y}}{n-K-1}}$$

SQUARE ROOT OF RESIDUAL VARIANCE.

(Identification of problem as given on Card Type 1)

(INDEPENDENT VARIABLE SELECTION $\begin{pmatrix} \text{HAND} \\ \text{IVOR} \\ \text{BIVOR} \end{pmatrix}$ 0--------)

ITEM NUMBER   PREDICTED VALUE AND PREDICTION ERROR

1 $\hat{Y}_1 = \sum_{\nu=0}^{K} b_\nu x_{\nu 1}$ $\quad \hat{e}_1 = y_1 - \hat{Y}_1$ $\quad$ 2 $\hat{Y}_2 = \sum_{\nu=0}^{K} b_\nu x_{\nu 2}$ $\quad \hat{e}_2 = y_2 - \hat{Y}_2$ $\quad$ 3 $\hat{Y}_3 = \sum_{\nu=0}^{K} b_\nu x_{\nu 3}$ $\quad \hat{e}_3 = y_3 - \hat{Y}_3$

- -

i $\hat{Y}_i = \sum_{\nu=0}^{K} b_\nu x_{\nu i}$ $\quad \hat{e}_i = y_i - \hat{Y}_i$ -

- -

- - - - - - - - - - - - - - - - - - - n $\hat{Y}_n = \sum_{\nu=0}^{K} b_\nu x_{\nu n}$ $\quad \hat{e}_n = y_n - \hat{Y}_n$

CHECK ERROR SUM OF SQUARES

$$\sum_{i=1}^{n} \left(y_i - \sum_{\nu=0}^{K} b_\nu x_{\nu i}\right)^2$$

(Identification of problem as given on Card Type 1)

(INDEPENDENT VARIABLE SELECTION $\left\langle \begin{matrix} \text{HAND} \\ \text{IVOR} \\ \text{BIVOR} \end{matrix} \right\rangle$ 0--------)

PREDICTION ERROR FREQUENCY DISTRIBUTION

RANGE = $\hat{e}_{MAX} - \hat{e}_{MIN}$ (= 30D)

UPPER BOUND	FREQUENCY	BAR CHART
$\hat{e}_{MIN}$ + 1D	f'_1	I
$\hat{e}_{MIN}$ + 2D	f'_2	I
.	.	.
.	.	.
.	.	.
$\hat{e}_{MIN}$ + j'D	$f'_{j'}$	I
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
$\hat{e}_{MIN}$+ 30D	f'_{30}	I

(number of X's = observed frequency of prediction errors in each interval)

CHI	OBS FR	EXPD FR
$\frac{(f_1 - \hat{\varphi}_1)^2}{\hat{\varphi}_1}$	f_1	$\hat{\varphi}_1$
.	.	.
.	.	.
$\frac{(f_j - \hat{\varphi}_j)^2}{\hat{\varphi}_j}$	f_j	$\hat{\varphi}_j$
.	.	.
.	.	.
$\frac{(f_n - \hat{\varphi}_n)^2}{\hat{\varphi}_n}$	f_n	$\hat{\varphi}_n$

CHI SQUARE = $\sum_{j=1}^{m} \frac{(f_j - \hat{\varphi}_j)^2}{\hat{\varphi}_j}$ DEGREES OF FREEDOM = m-K-3

AVERAGES OF INDEPENDENT VARIABLES AND DEPENDENT VARIABLE

1 $\frac{E_{O1}}{n}$ 2 $\frac{E_{O2}}{n}$ ---------- V $\frac{E_{OV}}{n}$ ---------- N $\frac{E_{ON}}{n}$ N+1 $\frac{E_{Oy}}{n}$

(Identification of problem as given on Card Type 1)

(INDEPENDENT VARIABLE SELECTION $\left\{\begin{array}{l}\text{HAND}\\ \text{IVOR}\\ \text{BIVOR}\end{array}\right\}$ 0--------)

SELECTED INPUT DESIGN POINTS

(1) $x_{1'(1)}$ $x_{2'(1)}$ - - -

(2) $x_{1'(2)}$ $x_{2'(2)}$ - - -

. . . .

. . . .

(q) $x_{1'(q)}$ $x_{2'(q)}$ - - -

. . . .

. . . .

(Q) $x_{1'(Q)}$ $x_{2'(Q)}$ - - -

SYNTHETIC DESIGN POINTS

(Q+1) $x_{1'(Q+1)}$ $x_{2'(Q+1)}$ - - -

(Q+2) $x_{1'(Q+2)}$ $x_{2'(Q+2)}$ - - -

. . . .

. . . .

(q') $x_{1'(q')}$ $x_{2'(q')}$ - - -

. . . .

. . . .

(Q') $x_{1'(Q')}$ $x_{2'(Q')}$ - - -

ITEM NUMBER, PREDICTED VALUE, AND PREDICTION STANDARD DEVIATION FOR THE PREDICTION LINE
PREDICTION STANDARD DEVIATION FOR INDIVIDUAL OBSERVATIONS

(p) $\hat{Y}_{(p)} = \sum_{\nu=0}^{K} b_\nu x_{\nu(p)}$

$$s_{(p)} = s\sqrt{\sum_{\nu=0}^{K}\sum_{\nu'=0}^{K} c_{\nu\nu'} x_{\nu(p)} x_{\nu'(p)}}$$

$$s'_{(p)} = s\sqrt{1 + \sum_{\nu=0}^{K}\sum_{\nu'=0}^{K} c_{\nu\nu'} x_{\nu(p)} x_{\nu'(p)}}$$

(where (p) can be either (q) or (q'))

RUN (number) TOOK ..... SECONDS

(Identification of problem as given on Card Type 1)

MAIN RUN
(or)
INDEPENDENT VARIABLE SELECTION $\left\langle \begin{matrix} \text{HAND} \\ \text{IVOR} \\ \text{BIVOR} \end{matrix} \right\rangle$ 0--------

	DF	SS	MS	F
REGRESSION	K	$ASSR_K$	$\frac{1}{K} ASSR_K$	$\frac{\frac{1}{K} ASSR_K}{\frac{1}{n-K-1} SSE}$
REGRESSION				
REGRESSION				
ERROR	n-K-1	SSE	$\frac{1}{n-K-1} SSE$	
CORRELATION	$\sqrt{\frac{ASSR_K}{ATSS}}$			

$Y = b_0 + b_1X(1) + b_2X(2) + \cdots + b_\nu X(\nu) + \cdots + b_N X(N)$ (in main run, for example)

(Identification of problem as given on Card Type 1)

DEGREES OF FREEDOM OF ERROR VARIANCE = n-N-1

COEFFICIENT OF DETERMINATION	NO. (DF) OF DELETED VARIABLES	F FOR REGRESSION ON DELETED VARIABLES	INDEPENDENT VARIABLE SELECTION
$\frac{ASSR_N}{ATSS}$ - MAIN RUN			
$\frac{ASSR_{N'}}{ATSS}$	N-N'	$\frac{\frac{1}{N-N'} SS_{N-N'}}{\frac{1}{n-N-1}[ATSS - ASSR_N]}$	0 ----------
(etc.)			
*** IVOR FINAL COMPREHENSIVE ***			
$\frac{ASSR_1}{ATSS}$	N-1	$\frac{\frac{1}{N-1} SS_{N-1}}{\frac{1}{n-N-1}[ATSS - ASSR_N]}$	0 ----------
(etc.)			
*** BIVOR FINAL COMPREHENSIVE ***			
$\frac{ASSR_{N''}}{ATSS}$	N-N"	$\frac{\frac{1}{N-N''} SS_{N-N''}}{\frac{1}{n-N-1}[ATSS - ASSR_N]}$	0 ----------
(etc.)			

COMPREHENSIVE PRINTOUTS ---------- SECONDS

TOTAL PROBLEM RUNNING TIME (HRS./MIN./SEC.) = ... ≠ ..... ≠ .....

(Note: N' and N" are typical numbers of IV's in reruns.)

VI.3.b Comments on Printout

The comments in this section refer to the algebraic formulation of the printout as given in the previous section. The page numbers referenced are the page numbers of that printout. In some instances the possible use of the printed information is discussed inasmuch as this has not been done before.

VI.3.b.(1) Basic Information

A. Problem Parameters (page 87). The page is headed by the problem identification as given on Card Type 1. This identification is repeated, at the beginning of certain features, throughout the program output for ease in identifying the printout of a given regression problem when several problems have been run consecutively. Page 87 contains information given on input Card Types 2, 3, 4, 5, and 6, and identifies the problem parameters chosen for the regression problem. The columns occupied by the program variables in this printout do not all agree with those specified in the input specification, Section V.2. For clarity of reading, the entries are spaced across this page. The spaces filled by X's indicate digits are to be printed. In the Card Type 3 line, the individual product term descriptions are separated by slants. Zeros are printed in the spaces which are not needed to represent the product terms.

B. Data Matrix (page 88). The data matrix printout is optional (see column 16, Card Type 2) and can be either in the format 9F13.6 or 7E17.8, whichever is specified on Card Type 2. The data matrix is printed, if at all, for the main run only.

Each row of the data matrix is identified by its "data point number" ($i = 1,2,3,\ldots,n$) and consists of the N+1 coordinates of the N independent variables and the dependent variable.

The coordinates of the OCIV's are listed in the same order as punched on Card Type 8. If generated independent variables (GCIV's) are used, they follow the OCIV's, and their coordinates are listed in the same order as generated according to Card Type 3.

The data matrix is printed only once per regression problem (i.e., for the main run) but can easily be obtained for any rerun by deleting the column, or columns, that correspond to the independent variable(s) which are deleted in the rerun.

C. Summation Matrix (page 88). The summation matrix is printed only once per regression problem; its dimensions are N+2 by N+2. The (N+1)x(N+1) matrix consisting of the first N+1 rows and columns of the summation matrix is the matrix of the coefficients of the normal equations for the main run, or the matrix A. Both the matrix A and the summation matrix are symmetrical.

The summation matrix (and the matrix A) of any rerun can easily be obtained by deleting the row(s) and column(s), which correspond to the independent variable(s) to be deleted, from the summation matrix of the main run.

VI.3.b.(2) Main Body

The formulation of the printout of the main body is done in terms of K independent variables contained in the model. Accordingly, with K = N or K = N' < N this formulation is valid for the main run or any rerun, respectively. Wherever applicable, the K independent variables contained in the model are consecutively renumbered from 1 to K. If, for example, the first two independent variables of the main run are not included in a rerun, then the third IV of the main run becomes IV Number 1 of the rerun.

For reruns the main body is headed "INDEPENDENT VARIABLE SELECTION ( ) 0 --------." In the parentheses "HAND," or "IVOR," or "BIVOR," whichever applies, is printed. For the main run there is no identification printed at this place. The IVS is specifically identified by a series of N+1 0's and 1's, of which the first is always a 0. These N+1 digits represent the constant (the first 0) and the N independent variables, respectively, corresponding to their order of input. If a specific independent variable is contained in the IVS, a 0 is printed in the place corresponding to this IV; if it is not contained in the IVS, a 1 is printed. Thus, when IV Number ν (ν = 1,...,N) is contained in the IVS, digit number ν+1 from the left in this identification will be a 0. Because the constant (IV Number 0) is always contained in an IVS, the first digit is always printed as a 0. The IV's not contained in an IVS (which are, accordingly, represented by 1's), are often referred to as "deleted" IV's, that is, as IV's "deleted from the model." - The IVS identification is repeated at various other places of the printout, when appropriate.

A. Matrix Inversion (pages 89 and 90). The MATRIX INVERSION EVALUATION TIME includes the time required to invert the matrix, compute the determinant and solve the set of the normal equations. The main run is numbered 0, the first rerun 1, the second rerun 2, etc. The printouts of the matrix inversion evaluation time and of other running times were originally included for a time study which resulted in the time formulae given in Section VI.4. The running time printouts have been left in the program as a convenience for the user.

The DETERMINANT of the matrix A may be printed in the F format or the E format depending upon the magnitude of the value of the determinant. If the determinant is negative or equal to zero the statement "MATRIX FAILED TO INVERT" is printed. (See Section VI.2.a.(2).)

The elements of A^{-1}, i.e., of the INVERSE OF MATRIX A, are denoted as $c_{\nu\nu'}$ ($\nu = 0,1,\ldots,K$; $\nu' = 0,1,\ldots,K$). The inverse matrix should be symmetrical, i.e., $c_{\nu\nu'} = c_{\nu'\nu}$, but is sometimes not because of computational inaccuracies. Its dimensions are (K+1) by (K+1).

For further statements concerning the failure of the matrix inversion see paragraphs D, E, and F of Section VI.2.a.(2).

The SOLUTION TO SIMULTANEOUS EQUATIONS is the vector of the K+1 regression coefficients b_ν, $\nu = 0,1,\ldots,K$, with

$$b_\nu = \sum_{\nu'=0}^{K} c_{\nu\nu'} E_{\nu'y}.$$

The elements of the calculated IDENTITY MATRIX (I_c) are obtained by multiplying the inverse matrix A^{-1} by the matrix A, i.e., $I_c = A^{-1}A$. The dimensions of I_c are K+1 by K+1.

For possible printouts regarding the magnitude of the elements of the calculated identity matrix see paragraphs H and I of Section VI.2.a.(2) and Section VI.2.d. When the statement "DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLER THAN I(1) = ... RUN ACCEPTED" is made, the identity matrix is not printed.

B. Various Statistics (page 90). The STANDARD DEVIATION OF (regression) COEFFICIENTS,

$$\sqrt{\mathbb{V}[b_\nu]} = s\sqrt{c_{\nu\nu}},$$

are always consecutively numbered as described at the beginning of this section (VI.3.b.(2)). No. 1 is always the standard deviation of b_0. In the main run, the standard deviation identified by the number 3, for example, is the standard deviation of the second regression coefficient, b_2. In a rerun, the standard deviation numbered 2, for example, may be the standard deviation of the regression coefficient of IV No. 3 if IV's No. 1 and No. 2 (in the original model) have been deleted for this IVS.

The 5 other statistics are denoted elsewhere in the printout formulation and at various places of the report, as follows:

RESIDUAL OR ERROR SUM OF SQUARES = SSE

TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN = ATSS

REGRESSION SUM OF SQUARES (due to K IV's) ADJUSTED FOR THE MEAN = $ASSR_K$

CORRELATION COEFFICIENT = R

SQUARE ROOT OF RESIDUAL VARIANCE = s

Notice that, besides SSE, $ASSR_K$, and R, also the standard deviation, s, is redefined in each run (with K independent variables contained in the model) and is the basis, in that run, for the computation of the standard deviations of the regression coefficients, the normality test and the prediction standard deviations at selected input and/or synthetic design points.

C. Predicted Values, Prediction Errors, Normality Test, and Averages (pages 91 and 92). For each of the n input design points the PREDICTED VALUE ($\hat{Y}_i$) is printed, and similarly the PREDICTION ERROR ($\hat{e}_i$) as obtained by subtracting the predicted value from the actual observation of y. The number of the input design point is also printed and is referred to, in the heading of this printout, as ITEM NUMBER.

The CHECK ERROR SUM OF SQUARES,

$$\sum_{i=1}^{n} \left[y_i - \sum_{\nu=0}^{K} b_\nu x_\nu \right]^2,$$

should equal the Residual or Error Sum of Squares (SSE). Any discrepancy between the two is an indication of computer inaccuracy. (See paragraph C of Section VI.2.a.(3).)

The printout format for the predicted values and for the prediction errors is affected by the value of NDPO (column 16, Card Type 2). If NDPO≠1, these values are printed in the format 2F15.6; if NDPO=1, they are printed in the format 2E15.6.

The features of the PREDICTION ERROR FREQUENCY DISTRIBUTION are explained in detail in paragraphs D and E of Section VI.2.a.(3). The bar chart gives a graphical representation of the distribution of the prediction errors. Each prediction error is represented by an X. Should the number of prediction errors in any interval be greater than 60 (thereby exceeding the space provided for the X's), an asterisk is printed at the end of the 60 X's. For the purpose of easier reading, the bar chart is printed to the right of a column of "I"s, one "I" for each of the 30 intervals.

The entries for the CHI-square contribution, the OBServed FRequencies and the EXPecteD FRequencies are discussed, together with the establishing of the m new intervals, in paragraphs F and G of Section VI.2.a.(3). In paragraphs E and H of that section the checks are discussed which lead to the possible printout "CHISQUARE COULD NOT BE COMPUTED."

The AVERAGES OF INDEPENDENT VARIABLES AND DEPENDENT VARIABLE are printed only once per regression problem and are numbered, accordingly, from 1 to N+1, such that the average of the dependent variable is numbered N+1.

D. Predictions at Selected Input and/or Synthetic Design Points (page 93). Predicted values and standard deviations at selected input design points and/or synthetic design points are optionally computed and printed for the main run and hand selected reruns only (see Card Type 2, columns 8-13). They cannot be obtained for IVOR or BIVOR reruns.

The coordinates of the OCIV's for the SELECTED INPUT DESIGN POINTS and/or the SYNTHETIC DESIGN POINTS are printed for ease in identifying which points were selected and/or specified, respectively. In the general formulation, the selected input design points are renumbered 1, ..., q, ..., Q; whereas the synthetic design points are consecutively numbered Q+1, ..., q', ..., Q'. The coordinates are renumbered 1', 2', ... in order to indicate that these are the coordinates of the OCIV's contained in the IVS of the run.

For each of the design points, selected or synthetic, the PREDICTED VALUE, $\hat{Y}_{(p)}$, and the PREDICTION STANDARD DEVIATION FOR THE PREDICTION LINE, $s_{(p)}$, or the PREDICTION STANDARD DEVIATION FOR INDIVIDUAL OBSERVATIONS, $s'_{(p)}$, are printed. The index "(p)" refers to the number ("(q)" or "(q')") of the point in the set of the OCIV coordinates printed previously and is given under the heading ITEM NUMBER.

Either $s_{(p)}$ or $s'_{(p)}$, but not both, can be obtained in a given problem. (See Card Type 2, column 14.) Should, however, both standard deviations be desired, the one that is not printed can obviously be obtained as follows:

If $s_{(p)}$ is printed: $s'_{(p)} = \sqrt{(s_{(p)})^2 + s^2}$

If $s'_{(p)}$ is printed: $s_{(p)} = \sqrt{(s'_{(p)})^2 - s^2}$

(Note: The standard deviations $s_{(p)}$ and $s'_{(p)}$, as given in the printout formulation, are actually computed by the program in the "adjusted" form, i.e., for example,

$$s_{(p)} = s\sqrt{\frac{1}{n} + \sum_{\nu=1}^{K}\sum_{\nu'=1}^{K} c_{\nu\nu'}(x_{\nu(p)} - \bar{x}_\nu)(x_{\nu'(p)} - \bar{x}_{\nu'})},$$

where

$$\bar{x}_\nu = \frac{1}{n}E_{o\nu} = \frac{1}{n}\sum_{i=1}^{n} x_{\nu i}.)$$

The standard deviations will be useful if one wants to construct $(1-\alpha)$% confidence limits, $L_{1-\alpha,(p)}$, for the prediction line, i.e.,

$$L_{1-\alpha,(p)} = \hat{Y}_{(p)} \pm s_{(p)}t_{1-\frac{\alpha}{2},\,n-K-1},$$

or $(1-\alpha)$% "tolerance" limits, $L'_{1-\alpha,(p)}$, for individual future observations, i.e.,

$$L'_{1-\alpha,(p)} = \hat{Y}_{(p)} \pm s'_{(p)}t_{1-\frac{\alpha}{2},\,n-K-1}.$$

The synthetic design point feature can also be useful just for obtaining the predicted values of the regression equation for design points other than those originally input. In other words, the feature can be advantageously applied for interpolation.

At the end of the "Main Body," the computer time required to perform all of the calculations for this run is printed: "RUN (number) TOOK ..... SECONDS." The main run is identified as run 0, the first rerun as run 1, etc.

VI.3.b.(3) Analysis of Variance Tables

For each run (main run or rerun) an analysis of variance table (page 94) is printed. The essential statistics of the run are given in analysis of variance form, including, at the bottom, the estimated regression equation for that run. The terms contained in these tables are taken from the results of the computations previously performed. The definitions of the terms are given in the "Various Statistics" part of the Main Body, see paragraph B of Section VI.3.b.(2). The two mean squares ("MS") and the F value are computed specifically for this table.

It must be emphasized that each analysis of variance table has its own error term based on n-K-1 degrees of freedom. The two blank rows, each headed by the word "REGRESSION," are available for convenience in case the user wishes to calculate (by hand) a main

theorem F value (III-1) for testing a specific hypothesis. For an example of this, see the corresponding printout of the Example Problem, Section VI.5.

The subscripts of the independent variables in the regression equation are the original numbers of the IV's as input for the main run. (This is different from the Main Body in which the K IV's in the IVS are renumbered from 1 to K.) For example, if IV Number ν is not included in the IVS, the term with $X(\nu)$ is not present in this printout of the regression equation.

VI.3.b.(4) Final Comprehensive Analysis Table

The Final Comprehensive Analysis Table (page 95) gives the F values (III-1) of the main theorem FOR REGRESSION ON DELETED VARIABLES for each rerun, together with the COEFFICIENT OF DETERMINATION, the NUMBER ("NO." = DF = DEGREES OF FREEDOM) OF DELETED VARIABLES and the identification of the INDEPENDENT VARIABLE SELECTION. Although implied by the application of the main theorem, it is emphasized that all F values are based on the error term of the main run with n-N-1 degrees of freedom. The table is also a very convenient means to show the order in which the independent variables are ranked by IVOR and/or BIVOR if these options are exercised. There is a certain danger of misinterpretation of the BIVOR final comprehensive analysis when a BIVOR independent variable selection is rejected only on the grounds of failing the identity matrix checks. In this case the rightmost IV is deleted from the model, which might appear as a genuine BIVOR ordering of this independent variable if one judges from the final comprehensive analysis table only. For more details see paragraph (A.b) of Section VI.2.d.

Should the Final Comprehensive Analysis not be printed (but reruns are present), the statement "NO FINAL COMPREHENSIVE PRINTOUT SINCE MATRIX FOR MAIN RUN COULD NOT BE INVERTED" is given.

VI.4 Running Time Formulae

The formulae of this section give the approximate times (in seconds) which are required by the IBM 7030 STRETCH computer to execute the various parts and options of the DA-MRCA program. In these formulae the time, T (in seconds), is expressed in terms of the input parameters N, N', IQ, and n, where

N = number of IV's contained in the model of the main run,

N' = number of IV's contained in the model of any (hand selected) rerun,

IQ = number of IV's to be ordered by IVOR, and

n = number of data points input.

The formulae are based upon the results of a time study in which a series of regression problems was actually computed by the program. In this study, each regression problem represented a unique combination of the values of, at the most, three of the input parameters N, N', IQ, and n; and from each problem the time(s) required for the computations were recorded. The ranges of the four parameters were taken, in the time study, as they are likely to occur in actual regression problems. N and N' were varied over the full range, that is, up to the capacity of the program which is N=50 independent variables. IQ took the values 2, 4, 8, and 16; and the numbers of data points, n, were 60, 120, 240, and 480.

Then DA-MRCA was used to fit polynomials in N, N', IQ, n (as applicable) to the responses, T, i.e., to the actual running times observed. (In terms of the present report, T was the "dependent" variable and N, N', IQ, and n were the "OCIV's.") As a matter of fact, both IVOR and BIVOR were employed to evaluate the most efficient polynomials for the prediction of the running times.

The coefficients in these polynomials (i.e., the "regression" coefficients) were rounded such that the formulae give, in general, a safe upper limit for the running times.

Little is known about extrapolation with respect to n, the number of data points. However, since 4 points have been used within the range of the study ($0 < n \leq 480$), thus allowing a 3rd order polynomial in n to be fitted, some extrapolation should be permissible.

The formulae are as follows:

a. Time (in seconds) for the main run, excluding the option for predicted values and prediction standard deviations at selected input and/or synthetic design points:

$$T_1 = 2 + \frac{nN}{1000}\left[8 - \frac{5n}{1000}\right] \qquad \text{(VI-18)}$$

b. Time (in seconds) for one hand selected rerun with N' IV's contained in the model, excluding the options for (1) predicted values, prediction errors, and the normality test, and (2) predicted values and prediction standard deviations at selected input and/or synthetic design points:

$$T_2 = \frac{7(N')^2}{1000} \qquad \text{(VI-19)}$$

(T_2 = 17 seconds for N' = 49)

c. Time (in seconds) for the option for predicted values, prediction errors, and the normality test for one hand selected rerun:

$$T_3 = \frac{0.7\ nN'}{1000} \qquad \text{(VI-20)}$$

d. Time (in seconds) for the Final Comprehensive Analysis computations for M hand selected reruns:

$$T_4 = \frac{M}{2} \qquad \text{(VI-21)}$$

e. Time (in seconds) for one IVOR sequence in which only the first IQ most important IV's out of N are ordered, including the computations for the IVOR Final Comprehensive Analysis and excluding the main run and the option for predicted values, prediction errors, and the normality test:

$$T_5 = 2 + \frac{8(IQ)^2N}{1000} \qquad \text{(VI-22)}$$

(T_5 = 1002 seconds for IQ=N=50)

f. Time (in seconds) for one BIVOR sequence in which all N IV's are ordered, including the computations for the BIVOR Final Comprehensive Analysis and excluding the main run and the option for predicted values, prediction errors, and the normality test:

$$T_6 = 5 + \frac{2N^3}{1000} \qquad \text{(VI-23)}$$

(T_6 = 255 seconds for N=50)

g. Time (in seconds) for the option for predicted values, prediction errors, and the normality test in one IVOR sequence in which only the first IQ most important IV's out of N are ordered:

$$T_7 = (IQ+1)\left[1 + \frac{0.35n\ (N+1)}{1000}\right] \qquad \text{(VI-24)}$$

h. Time (in seconds) for the option for predicted values, prediction errors, and the normality test in one BIVOR sequence:

$$T_8 = (N+1)\left[1 + \frac{0.35n\ (N+1)}{1000}\right] \qquad \text{(VI-25)}$$

Some discussion of these formulae seems to be appropriate.

T_1, T_2, and T_3 each contain a constant term which, although of lesser importance, was not considered small enough to be neglected.

In T_1 the term $\frac{5n}{1000}$ should probably be subtracted from 8 only if

n is smaller than 500 and be disregarded otherwise. (T_1 as given in (VI-18) has its maximum at n=800.) Since the polynomial was fitted only for the range $0 < n \leq 480$, this rule seems to give some safe margin for extrapolation beyond n=480, and the formula would read, for these larger values of n, as:

$$T_1 = 2 + \frac{8nN}{1000} .$$

For obvious reasons, only T_1, T_3, T_7, and T_8 depend upon n, the number of data points input, while the other 4 time formulae do not contain n. For T_2, T_5, and T_6, the maximum numerical values are given, in order to indicate the speed of the program with respect to reruns.

The comparison of T_5 and T_6 shows that a full IVOR sequence (with IQ=N) takes approximately 4 times the time of a full BIVOR sequence. Naturally, T_5 is strictly valid only for $IQ \leq 16$; however, it can be assumed that it is approximately valid also for the whole range, i.e., $IQ \leq 50$.

T_5 and T_6 were obtained without the grouping of IV's in IVOR and BIVOR. This means that, if grouping is applied in these options, the running times will be less than given by T_5 and/or T_6.

Obviously, T_7 and T_8 are identical for IQ=N.

No formulae have been evaluated for the option to compute predicted values and prediction standard deviations at selected input and/or synthetic design points.

The actual running times of the various parts of the example problem in Section VI.5 may serve as examples of the application of the formulae. In the example problem, the parameters take the following values:

$N = 9$

$N' = 3$ (in M=1 hand selected rerun)

$IQ = 4$

$n = 20$

This gives the following times:

a.
$$T_1 = 2 + \frac{(20)(9)}{1000}\left[8 - \frac{(5)(20)}{1000}\right] = 3.42$$

(The actual time for "RUN 0", including predicted values and prediction standard deviations, was 4.03 seconds.)

b.
$$T_2 = \frac{(7)(3^2)}{1000} = 0.06$$

c.
$$T_3 = \frac{(0.7)(20)(3)}{1000} = 0.04$$

($T_2 + T_3 = 0.10$, but "RUN 1" included predicted values and prediction standard deviations and actually took 2.10 seconds.)

d.
$$T_4 = \frac{1}{2} = 0.50$$

e.
$$T_5 = 2 + \frac{(8)(4^2)(9)}{1000} = 3.15$$

f.
$$T_6 = 5 + \frac{(2)(9^3)}{1000} = 6.46$$

g.
$$T_7 = (4+1)\left[1 + \frac{(0.35)(20)(9+1)}{1000}\right] = 5.35$$

h.
$$T_8 = (9+1)\left[1 + \frac{(0.35)(20)(9+1)}{1000}\right] = 10.70$$

This gives a total of

$$\sum_{j=1}^{8} T_j = 29.68 \text{ seconds.}$$

The actual "TOTAL PROBLEM RUNNING TIME" was 29 seconds. The latter time included the predicted values and prediction standard deviations at 2 selected input design points and 3 synthetic design points in the main run and in the only hand selected rerun, which seems to compensate for the time saving in IVOR and BIVOR due to the grouping feature as applied here but not considered in the time formulae.

VI.5 Example Problem

The example regression problem contained in this section is given in order to illustrate the various capabilities of the DA-MRCA program and to exhibit a sample of the program output.

The data of the example problem, as listed in the table below, was taken from Duncan [1959], p. 697. This was done in preference to fabrication of artificial variables and data, and the example was selected as a representation of a typical regression problem. (Naturally, no attempt is made to find a practical solution to any aspect of the general ballistic problem.)

There are n=20 data points in the problem. Each one consists of (a) the coordinate of the dependent variable, y = "Ballistic Limit", which is a measure in ft./sec. of the projectile velocity required to penetrate armor plate; (b) the coordinate of the first OCIV, x_1 = thickness of plate in inches; and (c) the coordinate of the second OCIV x_2 = Brinnell hardness number of the plate material.

y	x_1	x_2
Ballistic Limit in Feet/Sec.	Thickness in Inches	Brinnell Hardness No.
927	.253	317
978	.258	321
1,028	.259	341
906	.247	350
1,159	.256	352
1,055	.246	363
1,335	.257	365
1,392	.262	375
1,362	.255	373
1,374	.258	391
1,393	.253	407
1,401	.252	426
1,436	.246	432
1,327	.250	469
950	.242	275
998	.243	302
1,144	.239	331
1,080	.242	355
1,276	.244	385
1,062	.234	426

The input preparation for the example problem, based on this data, is exemplified in Section V.3.

The GCIV's generated are x_1x_2, x_1^3, x_2^2, $x_1^2x_2$, $x_1x_2^2$, x_1^3, and x_2^3. Both ranking options, IVOR and BIVOR, are exercised. There are $M_1=2$ groups of IV's specified in IVOR: the two OCIV's x_1 and x_2, are in the first group and the 7 GCIV's are in the second group. Only IQ=4 IV's are to be ranked. Under the restriction due to grouping, these 4 IV's will include the two OCIV's of the first group (to be ranked among themselves) and the two most important GCIV's of the second group. In BIVOR, there are $M_2=3$ groups: the two OCIV's are in the first group, the three GCIV's of second order are in the second group, and the four GCIV's of third order are in the third group. For the other specifications see Section V.3.

Pertinent comments in handwriting are added to the computer printout exhibited. Due to space limitations the printout is not complete, some printout having been deleted. Whenever this applies, an appropriate comment is made.

The two IVOR analysis of variance tables exhibited are used to show the type of hypothesis testing which can be conveniently achieved with these tables. The example null hypothesis is that fitting x_1 (plate thickness) in addition to x_2 (Brinnell hardness) does not significantly reduce the error sum of squares. This hypothesis is rejected at the 0.05 level of significance, which implies that including x_1 in the model in addition to x_2 does improve the fit significantly.

On the page where the final comprehensive analysis table is printed some interpretation is given of the rankings of the IV's resulting from IVOR and BIVOR. The IVS column is repeated in handwriting in order to clearly identify the IV's additionally included (symbol "0") and deleted (symbol "1") in consecutive steps of IVOR and BIVOR, respectively.

If the analyst wants to determine a "significant model" from each of these rankings, he may choose a significance level for the F value ("for regression on deleted variables") and determine the model accordingly. The analyst must be aware that such a model may depend upon the grouping of the IV's. For example, in the IVOR ranking of the present example, any significant model including any IV of the second group must necessarily also include the two OCIV's. It could be imagined that without grouping, one of the two OCIV's might not have been considered part of the significant model.

With P=0.05, say, as the chosen significance level, the "significant models" from the two rankings are determined as follows. The last and first significant F value in IVOR and BIVOR, respectively, is F_c = 3.384 with 7 and 10 degrees of freedom. (The tabled F value for 7 and 10 degrees of freedom at the 0.05 significance level is 3.14.) This leads to a "significant model" from IVOR which includes x_8, x_1, and $x_1^2x_2$, with an associated coefficient of determination (R^2) equal to 0.76. The "significant model" from BIVOR includes x_2, x_1, and x_1x_2, with R^2 = 0.75. Thus the two "significant models" differ only in their least important IV's, which might be due to the different groupings used in IVOR and BIVOR. (Because of the grouping in BIVOR, $x_1^2x_2$ had to be deleted in one of the first four steps.)

For a comparison of the actual times used by DA-MRCA to compute (and print) the various parts of the problem, with the times predicted by the formulae given in Section VI.4, see the end of that section.

```
DA-MRCA ... OUTPUT FROM PROGRAM VERSION  2/ 1/66


EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

IR IS  NR MVP NDR MVPL NPE NOPO IAPF IVORGO NFD IBID     TOL1     TOL2     FORM - INPUT DATA DESCRIPTION -CARD TYPE 2

 2  7  1  3  2  1  1  1  0  3  1  0  .100E-03  .500E-02  3F10.0

PRODUCT TERM DESCRIPTIONS -CARD TYPE 3

  1  2  0  0  0  0  0  0  0  0/  1  1  0  0  0  0  0  0  0/  2  2  0  0  0  0  0  0  0  0/  1  1  2  0  0  0  0  0  0  0/
  1  2  2  0  0  0  0  0  0  0/  1  1  1  0  0  0  0  0  0  0/  2  2  2  0  0  0  0  0  0  0/

IQ  MI NJ(I),I=1,2,....,MI   -CARD TYPE 4
 4   2    2  7

NR LOT(I),I=1,2,....,NR      -CARD TYPE 5
 3    2  1  4

NUMBERS OF SELECTED INPUT DESIGN POINTS -CARD TYPE 6
   4   13
```

(data matrix)

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

1	.25300000E-00	.31700000E+03	.80201300E+02	.64009000E-01	.10648900E+06	.20290853E+02	.25423717E+05
	.16194277E-01	.31855013E+08	.92700000E+03				
2	.25800000E-00	.32100000E+03	.82813000E+02	.66564000E-01	.10304100 +06	.21367044E+02	.26584578E+05
	.17173512E-01	.33076161E+08	.97800000E+03				
3	.25900000E-00	.34100000E+03	.88319000E+02	.67081000E-01	.11628100E+06	.22874621E+02	.30116779E+05
	.17373979E-01	.39651821E+08	.10280000E+04				
4	.24700000E-00	.35000000E+03	.86450000E+02	.61009000E-01	.12250009E+06	.21353150E+02	.30257500E+05
	.15069223E-01	.42875000E+08	.90600000E+03				
5	.25600000E-00	.35200000E+03	.90112000E+02	.65536000E-01	.12390400E+06	.23068672E+02	.31719424E+05
	.16777216E-01	.43614208E+08	.11590000E+04				
6	.24600000E-00	.36300000E+03	.89298000E+02	.60516000E-01	.13179900E+06	.21967308E+02	.32415174E+05
	.14886936E-01	.47832147E+08	.10550000E+04				
7	.25700000E-00	.36500000E+03	.93805000E+02	.66049000E-01	.13322500E+06	.24107885E+02	.34238825E+05
	.16974593E-01	.48627125E+08	.13350000E+04				
8	.26200000E-00	.37500000E+03	.98250000E+02	.68644000E-01	.14062500E+06	.25741500E+02	.36843750E+05
	.17984728E-01	.52734375E+08	.13920000E+04				
9	.25500000E-00	.37300000E+03	.95115000E+02	.65025000E-01	.13912900E+06	.24254325E+02	.35477895E+05
	.16581375E-01	.51895117E+08	.13620000E+04				
10	.25800000E-00	.39100000E+03	.10087800E+03	.66564000E-01	.15288100E+06	.26026524E+02	.39443298E+05
	.17173512E-01	.59776471E+08	.13740000E+04				
11	.25300000E-00	.40700000E+03	.10297100E+03	.64009000E-01	.16564900E+06	.26051663E+02	.41909197E+05
	.16194277E-01	.67419143E+08	.13930000E+04				
12	.25200000E-00	.42600000E+03	.10735200E+03	.63504000E-01	.18147600E+06	.27052704E+02	.45731952E+05
	.16003008E-01	.77308776E+08	.14010000E+04				
13	.24600000E-00	.43200000E+03	.10627200E+03	.60516000E-01	.18662400E+06	.26142912E+02	.45909504E+05
	.14886936E-01	.80621568E+08	.14360000E+04				
14	.25000000E-00	.46900000E+03	.11725000E+03	.62500000E-01	.21996100E+06	.29312500E+02	.54990250E+05
	.15625000E-01	.10316171E+09	.13270000E+04				
15	.24200000E-00	.27500000E+03	.66550000E+02	.58564000E-01	.75625000E+05	.16105100E+02	.18301250E+05
	.14172488E-01	.20796875E+08	.95000000E+03				
16	.24300000E-00	.30200000E+03	.73386000E+02	.59049000E-01	.91204000E+05	.17832798E+02	.22162572E+05
	.14348907E-01	.27543608E+08	.99800000E+03				
17	.23900000E-00	.33100000E+03	.79109000E+02	.57121000E-01	.10956100E+06	.18907051E+02	.26185079E+05
	.13651919E-01	.36264691E+08	.11440000E+04				
18	.24200000E-00	.35500000E+03	.85910000E+02	.58564000E-01	.12602500E+06	.20790220E+02	.30498050E+05
	.14172488E-01	.44738875E+08	.10800000E+04				
19	.24400000E-00	.38500000E+03	.93940000E+02	.59536000E-01	.14822500E+06	.22921360E+02	.36166900E+05
	.14526784E-01	.57066625E+08	.12750000E+04				
20	.23400000E-00	.42600000E+03	.99684000E+02	.54756000E-01	.18147600E+06	.23326056E+02	.42465384E+05
	.12812904E-01	.77308776E+08	.10620000E+04				

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

SUMMATION MATRIX

.20000000E+02	.49960000E+01	.73560000E+04	.18376700E+04	.12491160E+01	.27496700E+07	.45949425E+03
.68684108E+06	.31258406E-00	.10441681E+10	.23583000E+05			
.49960000E+01	.12491160E+01	.18376700E+04	.45949425E+03	.31258406E-00	.68684108E+06	.11499384E+03
.17171842E+06	.78290695E-01	.26075073E+09	.59002530E+04			
.73560000E+04	.18376700E+04	.27496700E+07	.68684108E+06	.45949425E+03	.10441681E+10	.17171842E+06
.26075073E+09	.11499384E+03	.40262395E+12	.87957870E+07			
.18376700E+04	.45949425E+03	.68684108E+06	.17171842E+06	.11499384E+03	.26075073E+09	.42969186E+05
.65172279E+08	.28803545E+02	.10050290E+12	.22007670E+07			
.12491160E+01	.31258406E-00	.45949425E+03	.11499384E+03	.78290695E-01	.17171842E+06	.28803545E+02
.42969186E+05	.19625829E-01	.65172279E+08	.14774740E+04			
.27496700E+07	.68684108E+06	.10441681E+10	.26075073E+09	.17171842E+06	.40262395E+12	.65172279E+08
.10050290E+12	.42969186E+05	.15755164E+15	.33312778E+10			
.45949425E+03	.11499384E+03	.17171842E+06	.42969186E+05	.28803545E+02	.65172279E+08	.10761467E+05
.16303348E+08	.72208612E+01	.25109382E+11	.55112051E+06			
.68684108E+06	.17171842E+06	.26075073E+09	.65172279E+08	.42969186E+05	.10050290E+12	.16303348E+08
.25109382E+11	.10761467E+05	.39308453E+14	.83340633E+09			
.31258406E-00	.78290695E-01	.11499384E+03	.28803545E+02	.19625829E-01	.42969186E+05	.72208612E+01
.10761467E+05	.49239705E-02	.16303348E+08	.37029002E+03			
.10441681E+10	.26075073E+09	.40262395E+12	.10050290E+12	.65172279E+08	.15755164E+15	.25109382E+11
.39308453E+14	.16303348E+08	.62526526E+17	.12804961E+13			
.23583000E+05	.59002530E+04	.87957870E+07	.22007670E+07	.14774740E+04	.33312778E+10	.55112051E+06
.83340633E+09	.37029002E+03	.12804961E+13	.28469483E+08			

MATRIX INVERSION @ ...EVALUATION TIME = 0.07732592 SECONDS.

DETERMINANT= 314.216085951 35

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

.16300843E+09	-.17809285E+10	-.12353752E+06	.90844457E+06	.64740701E+10	.28682493E+02	-.16363380E+07
-.12491560E+03	-.78490587E+10	.15468264E-02	.12821386E+07			
-.17808291E+10	.19723701E+11	.11640847E+07	-.86815990E+07	-.72713554E+11	-.22443965E+03	.15940940E+08
.96974629E+03	.89357029E+11	-.93435014E-02	-.11615967E+08			
-.12353783E+06	.11640985E+07	.22326959E+03	-.15457989E+04	-.35437946E+07	-.87810363E-01	.25415660E+04
.39049460E-00	.34954103E+07	-.73139220E-05	-.24858970E+04			
.90844650E+06	-.86815224E+07	-.15457986E+04	.10928633E+05	.26796158E+08	.53066431E+00	-.18621382E+05
-.23169072E+01	-.26637371E+08	.34142732E-04	.17033208E+05			
.64740669E+10	-.72713537E+11	-.35437798E+07	.26796064E+08	.27193126E+12	.53829591E+03	-.50063464E+08
-.23267904E+04	-.33894756E+12	.18196198E-01	.33453352E+08			
.28682702E+02	-.22446202E+03	-.87810461E-01	.53066510E+00	.53830483E+03	.61050227E-04	-.65068037E+00
-.28539800E-03	-.43374575E+03	.83516655E-08	.97098934E+00			
-.16363404E+07	.15940971E+08	.25415647E+04	-.18621377E+05	-.50063594E+08	-.65067878E+00	.33748210E+05
.25812176E+01	.50022677E+08	.17250618E-04	-.29518659E+05			
-.12491556E+03	.96975711E+03	.39049507E-00	-.23169110E+01	-.23268310E+04	-.28539801E-03	.25812251E+01
.14183919E-02	.20149175E+04	-.57659315E-07	-.29323499E+01			
-.78490533E+10	.89356991E+11	.34963911E+07	-.26637248E+08	-.33894750E+12	-.43373461E+03	.50022504E+08
.20148669E+04	.42879809E+12	-.40672760E-01	-.29537024E+08			
.15468508E-02	-.93437679E-02	-.73139402E-05	.34142867E-04	.18197167E-01	.83516692E-08	.17250370E-04
-.57659331E-07	-.40673933E-01	.52655683E-11	-.23700792E-03			

IDENTITY MATRIX

.99972858E+00	-.54007396E-04	-.92720032E-01	-.24132729E-01	-.13679266E-04	-.33227539E+02	-.62394142E-02
-.75288086E+01	-.41935127E-05	-.13767000E+05				
.29988885E-02	.10006655E+01	.86753845E+00	.15997696E-00	.19722059E-03	.42048438E+03	.58757782E-01
.99156250E+02	.51507726E-04	.17687600E+06				
.19630534E-06	.45147317E-07	.10000714E+01	.18242747E-04	.12087185E-07	.24101257E-01	.47320500E-05
.71010590E-02	.28644536E-08	.10832031E+02				
-.15024561E-05	-.29418152E-06	-.53477287E-03	.99985535E+00	-.99375029E-07	-.21905518E-00	-.34898520E-04
-.54336548E-01	-.24097972E-07	-.85351563E+02				
-.10468125E-01	-.22895634E-02	-.33441162E+01	-.97935538E+00	.99940837E+00	-.13127813E+04	-.20996094E-00
-.33529297E+03	-.14115497E-03	-.41445600E+06				
-.38937742E-10	-.63096195E-11	-.13358658E-07	-.33942342E-08	-.21707081E-11	.99999558E+00	-.81399776E-09
-.11287630E-05	-.53379523E-12	-.17242432E-02				
.29918738E-05	.61286846E-06	.11234283E-02	.29601157E-03	.18383435E-06	.43083191E-00	.10000632E+01
.95523834E-01	.43955879E-07	.16921484E+03				
.19258550E-09	.35072389E-10	.64610504E-07	.16734703E-07	.11397105E-10	.26583672E-04	.45765773E-08
.10000063E+01	.28315128E-11	.10452271E-01				
.15266895E-01	.25086403E-02	.42390747E+01	.86502075E+00	.89712441E-03	.15063750E+04	.25402832E-00
.32233594E+03	.10001847E+01	.77230400E+06				
-.18041124E-14	-.31225023E-15	-.63948846E-12	-.15987212E-12	-.11796120E-15	-.18917490E-09	-.41744386E-13
-.58207661E-10	-.27985086E-16	.99999990E+00				

DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MATRIX LARGER THAN I(1)=.00010000 BUT LESS THAN I(2)= .00500000 .RUN ACCEPTED.

STANDARD DEVIATION OF COEFFICIENTS

1	.11077912E+07	2	.12185605E+03	3	.12964856E+04	4	.90705959E+04	5	.45246237E+08	6	.67794825E+00
7	.15939631E+05	8	.32677666E+01	9	.56817134E+08	10	.19910196E-03				

75284.540023804	RESIDUAL OR ERROR SUM OF SQUARES.
660588.55000019	TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
585304.00997639	REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
0.94129386660320	CORRELATION COEFFICIENT (R).
86.766664119236	SQUARE ROOT OF RESIDUAL VARIANCE.

Circled elements in the main diagonal of the identity matrix have deviations greater than I(1), but all deviations in the main diagonal are less than I(2).

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 637)

ITEM NUMBER PREDICTED VALUE AND PREDICTION ERROR

1	.948574E+03	.784256E+02	2	.967460E+03	.105399E+02	3	.114553E+04	-.117530E+03
4	.104968E+04	-.143492E+03	5	.116537E+04	-.637481E+01	6	.112032E+04	-.653194E+02
7	.126658E+04	.684240E+02	8	.136664E+04	.253584E+02	9	.128518E+04	.768239E+02
10	.139247E+04	-.184661E+02	11	.141210E+04	-.190981E+02	12	.143897E+04	-.379654E+02
13	.137904E+04	.569580E+02	14	.133655E+04	-.954751E+01	15	.997260E+03	-.472596E+02
16	.921007E+03	.769935E+02	17	.111538E+04	.296169E+02	18	.108808E+04	-.804438E+01
19	.121045E+04	.655473E+02	20	.107636E+04	-.143621E+02			

CHECK ERROR SUM OF SQUARES
75284.635627836

The discrepancy between the "check error sum of squares" and the "residual or error sum of squares" is due to computational inaccuracy and is approximately 0.1, or 0.00013 % of the error sum of squares.

PREDICTION ERROR FREQUENCY DISTRIBUTION
RANGE = .222107E+03

UPPER BOUND	FREQUENCY	BAR CHART
-.13627841E+03	1	IX
-.12887474E+03	0	I
-.12147114E+03	0	I
-.11406756E+03	1	IX
-.10666398E+03	0	I
-.99260398E+02	0	I
-.91856816E+02	0	I
-.84453233E+02	0	I
-.77049651E+02	0	I
-.69646069E+02	0	I
-.62242487E+02	1	IX
-.54838904E+02	0	I
-.47435322E+02	0	I
-.40031740E+02	1	IX
-.32628158E+02	1	IX
-.25224575E+02	0	I
-.17820993E+02	2	IXX
-.10417411E+02	1	IX
-.30138286E+01	3	IXXX
.43897537E+01	0	I
.11793336E+02	1	IX
.19196919E+02	0	I
.26600500E+02	1	IX
.34004083E+02	1	IX
.41407665E+02	0	I
.48811247E+02	0	I
.56214829E+02	0	I
.63618412E+02	1	IX
.71021994E+02	2	IXX
.78425577E+02	3	IXXX

CHISQUARE COULD NOT BE COMPUTED

CHI OBS FR EXPD FR

No Chi-square calculations because the quantity $\frac{N}{3}-(N+3)$ $= \frac{N}{3}-(9+3)$ is negative.

AVERAGES OF INDEPENDENT VARIABLES AND DEPENDENT VARIABLE

1	.24980000E-00	2	.37780000E+03	3	.91853500E+02	4	.52455800E-01	5	.13748350E+06	6	.22974712E+02
7	.34342054E+05	8	.15529203E-01	9	.52208404E+02	10	.11791400E+04				

```
EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

SELECTED INPUT DESIGN POINTS...
(  1)  .247000E-00  .350000E+03
(  2)  .246000E-00  .432000E+03

SYNTHETIC DESIGN POINTS...
(  3)  .240000E-00  .350000E+03
(  4)  .250000E-00  .400000E+03
(  5)  .260000E-00  .450000E+03

ITEM NUMBER,PREDICTED VALUE,AND PREDICTION STANDARD DEVIATION FOR THE PREDICTION LINE
   1    .10496819E+04    .46193650E+02    2    .13790420E+04    .69003069E+02    3    .11024751E+04    .60061710E+02
   4    .13566178E+04    .43166473E+02    5    .12048451E+04    .36764085E+03

RUN    0 TOOK    4.03387500 SECONDS.
```

Input design point no. 4

Input design point no. 13

[redacted] PROBLEM (DUNCAN, 1959, PAGE 697)

[redacted]DENT VARIABLE SELECTION (HAND) [redacted]1101

MATRIX INVERSION 1 ...EVALUATION TIME = 0.00871296 SECONDS.

DETERMINANT= 0.6682972013E-20

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

.49183683E+08	-.59588120E+09	.24049318E+10	-.32333380E+10	.26641027E+06
-.59588120E+09	.72203513E+10	-.29144786E+11	.39199352E+11	-.32650762E+07
.24049318E+10	-.29144786E+11	.11765824E+12	-.15822969E+12	.13350945E+08
-.32333380E+10	.39199352E+11	-.15822969E+12	.21291953E+12	-.18136497E+08

IDENTITY MATRIX

.99994278E+00	-.12397766E-04	-.38146973E-05	-.71525574E-06
.61035156E-04	.10000916E+01	.30517578E-04	.47683716E-05
-.19531250E-02	-.30517578E-03	.99989319E+00	-.19073486E-04
.21972656E-02	.42724609E-03	.12207031E-03	.10000229E+01

DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MATRIX LARGER THAN I(1)=.00010000 BUT LESS THAN I(2)= .00500000 .RUN ACCEPTED.

STANDARD DEVIATION OF COEFFICIENTS

1 .13371973E+07 2 .16201832E+08 3 .65402769E+09 4 .87961109E+08

581687.73410034	RESIDUAL OR ERROR SUM OF SQUARES.
660588.55000019	TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
78900.815899849	REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
0.34560118761495	CORRELATION COEFFICIENT (R).
190.67113935064	SQUARE ROOT OF RESIDUAL VARIANCE.

The standard deviations of the coefficients are numbered consecutively. The four standard deviations in this rerun are $s\sqrt{C_{00}}$, $s\sqrt{C_{11}}$, $s\sqrt{C_{99}}$, and $s\sqrt{C_{88}}$, respectively.

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

INDEPENDENT VARIABLE SELECTION (HAND) 0011011101

ITEM NUMBER PREDICTED VALUE AND PREDICTION ERROR

1	.121913E+04	-.292132E+03	2	.124552E+04	-.267522E+03	3	.124711E+04	-.219114E+03
4	.116129E+04	-.255294E+03	5	.123832E+04	-.793242E+02	6	.115040E+04	-.954025E+02
7	.124256E+04	.924449E+02	8	.124256E+04	.149435E+03	9	.123294E+04	.129063E+03
10	.124552E+04	.128478E+03	11	.121913E+04	.173869E+03	12	.121093E+04	.190069E+03
13	.115040E+04	.285597E+03	14	.119248E+04	.134524E+03	15	.110725E+04	-.157248E+03
16	.111776E+04	-.119757E+03	17	.107836E+04	.656384E+02	18	.110725E+04	-.272477E+02
19	.112852E+04	.147475E+03	20	.104555E+04	.144481E+02			

CHECK ERROR SUM OF SQUARES
581687.78015081

INDEPENDENT VARIABLE SELECTION (HAND) 0011011101

PREDICTION ERROR FREQUENCY DISTRIBUTION
RANGE = .577730E+03

UPPER BOUND	FREQUENCY	BAR CHART
-.27287436E+03	1	IX
-.25361671E+03	2	IXX
-.23435906E+03	0	I
-.21510141E+03	1	IX
-.19584376E+03	0	I
-.17658611E+03	0	I
-.15732846E+03	0	I
-.13807081E+03	1	IX
-.11881316E+03	1	IX
-.99555512E+02	0	I
-.80297861E+02	1	IX
-.61040211E+02	1	IX
-.41782561E+02	0	I
-.22524910E+02	1	IX
-.32672600E+01	0	I
.15990390E+02	0	I
.35248041E+02	1	IX
.54505691E+02	0	I
.73763341E+02	1	IX
.93020992E+02	1	IX
.11227864E+03	0	I
.13153629E+03	2	IXX
.15079394E+03	3	IXXX
.17005159E+03	0	I
.18930924E+03	1	IX
.20856689E+03	1	IX
.22782454E+03	0	I
.24708219E+03	0	I
.26633984E+03	0	I
.28559750E+03	1	IX

CHI OBS FR EXPD FR

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

INDEPENDENT VARIABLE SELECTION (HAND) 0011011101

SELECTED INPUT DESIGN POINTS...
(1) .747000E-00
(2) .746000E-00

SYNTHETIC DESIGN POINTS...
(3) .740000E-00
(4) .750000E-00
(5) .760000E-00

Only x_1 is printed because x_2 is deleted from the model.

ITEM NUMBER,PREDICTED VALUE,AND PREDICTION STANDARD DEVIATION FOR THE PREDICTION LINE

1	.11612963E+04	.61593037E+02	2	.11504025E+04	.63352440E+02	3	.10874412E+04	.81738017E+02
4	.11924760E+04	.63222450E+02	5	.12472252E+04	.95355232E+02			

RUN [illegible] 2.09656553 [illegible]

```
BEGIN IVOR REGRESSION CALCULATIONS

EXAMPLE PROBLEM (DUNCAN, 1953, PAGE 697)

INDEPENDENT VARIABLE SELECTION (IVOR)   0101111111

MATRIX INVERSION      2 ...EVALUATION TIME =    0.00218141 SECONDS.

DETERMINANT=    88266.00000047

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS
    .31151945E+01    -.93338620E-02     .16275552E+03
   -.93339620E-02     .22638679E-04     .27634434E+01

DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLER THAN I(1)= .00010000 .RUN ACCEPTED.

STANDARD DEVIATION OF COEFFICIENTS
 1     .23663763E+01  2     .63820253E+00

    323560.39903455   RESIDUAL OR ERROR SUM OF SQUARES.
    660588.55000019   TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
    337028.45096564   REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
  0.71427890152727    CORRELATION COEFFICIENT (R).
    134.07296915290   SQUARE ROOT OF RESIDUAL VARIANCE.
```

The identity matrix is not printed because all deviations are smaller than I(1).

Predicted values, prediction errors and the Chi-square test for the first four terms follow, but are not shown here.

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

INDEPENDENT VARIABLE SELECTION (IVDR) 0001111111

MATRIX INVERSION 3 ...EVALUATION TIME = 0.00475598 SECONDS.

DETERMINANT= 983.9481440149

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

.98565853E+02	-.22303086E+03	-.76202959E-02	-.18064735E+04
-.22303086E+03	.89706353E+03	-.28700699E-02	.79205344E+04
-.76202959E-02	-.28700699E-02	.22667861E-04	.27381024E+01

DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLER THAN E(1)= .00310000 .RUN ACCEPTED.

STANDARD DEVIATION OF COEFFICIENTS

1 .33474921E+01 2 .36583434E+04 3 .58153806E+00

251626.91856887 RESIDUAL OR ERROR SUM OF SQUARES.
660588.55000014 TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
408962.03143132 REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
0.78489475211499 CORRELATION COEFFICIENT (R).
127.14420565315 SQUARE ROOT OF RESIDUAL VARIANCE.

— — — — — — — — — — —
— — — — — — — — — — —

Not shown are the predicted values, prediction errors and the Chi-square test for the second four terms and the entire main body for the third and fourth four terms

— — — — — — — — — — —
— — — — — — — — — — —

IVDR EXECUTION TIME 5.72419 SECONDS.

Total computer time required for the four four selected terms.

B E G I N   P I V O R   R E G R E S S I O N   C A L C U L A T I O N S

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 597)

INDEPENDENT VARIABLE SELECTION (BIVOR) 0300300010

MATRIX INVERSION   A ...EVALUATION TIME = 0.05925324 SECONDS.

DETERMINANT= 134735298297126

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

.19333159E+00	-.14516761E+09	-.59536081E+05	.42085536E+06	.26970755E+09	.20743072E+02	-.72068655E+06
-.88033890E+02	.80732017E-03	.74144959E+06				
-.14516760E+09	.11076603E+10	.43547147E+06	-.31306757E+07	-.20804473E+10	-.13405391E+03	.55167742E+07
.54986913E+03	-.84772445E-03	-.56607610E+07				
-.59536024E+05	.43547153E+06	.19476009E+03	-.13285994E+04	-.78002412E+06	-.84273701E-01	.21336836E+04
.37406547E-00	-.69072772E-05	-.22450528E+04				
.42085537E+06	-.31306759E+07	-.13285993E+04	.92739301E+04	.57403974E+07	.50372028E+00	-.15513933E+05
-.21917417E+01	.31616099E-06	.15198339E+05				
.26970752E+09	-.20804477E+10	-.78002335E+06	.57403965E+07	.40069659E+10	.19544623E+03	-.10522696E+08
-.73411953E+03	-.13953975E-01	.10105529E+08				
.20743081E+02	-.13405399E+03	-.84273727E-01	.50372050E+00	.19544641E+03	.60611488E-04	-.60008068E+00
-.28335988E-03	.83105234E-08	.96111150E+00				
-.72068650E+06	.55167740E+07	.21336837E+04	-.15513332E+05	-.10522697E+08	-.60008022E+00	.27912691E+05
.23461675E+01	.21995416E-04	-.26072993E+05				
-.88033948E+02	.54994960E+03	.37406557E-00	-.21917429E+01	-.73412048E+03	-.28335990E-03	.23461699E+01
.14008924E-02	-.57469194E-07	-.27935559E+01				
.80237351E-03	-.84774906E-03	-.69822877E-05	.31616173E-06	-.13953927E-01	.83105270E-08	.21995288E-04
-.57468209E-07	.52617103E-11	-.23980968E-03				

IDENTITY MATRIX

.99996967E+00	-.54659932E-05	-.11436462E-01	-.32601357E-02	-.19785948E-05	-.39331055E+01	-.68664551E-03
-.10919189E+01	-.17162503E+04					
.21469966E-01	.10000477E+01	.71088791E-01	.22473812E-01	.13380079E-04	.29370117E+02	.48158169E-02
.19422607E+01	.11702009E+05					
.89606747E-07	.14071702E-07	.10000280E+01	.77411532E-05	.48894435E-08	.13313293E-01	.20880252E-05
.26159554E-02	.39414063E+01					
-.75064400E-04	-.14400575E-06	-.25737766E-03	.59993739E+00	-.46333298E-07	-.99151611E-01	-.17505139E-04
-.21181497E-01	-.39125000E+02					
[illegible]	-.11257827E-03	-.15301514E-00	-.46913147E-01	.99996995E+00	-.68210939E+02	-.11142731E-01
[illegible]	-.25513200E+05					
[illegible]	[illegible]	-.64645360E-04	-.22046152E-04	-.14956925E-11	.99999715E+00	-.65574568E-09
[illegible]	-.13114756E-02					
[illegible]	[illegible]	[illegible]	.11893170E-03	.90003596E-07	.19430542E-00	.10000297E+01
[illegible]	.24195311E+02					
[illegible]	[illegible]	[illegible]	.12627736E-07	.23856652E-11	.15258789E-04	.29904186E-08
[illegible]	[illegible]					
[illegible]	[illegible]	[illegible]	[illegible]	.69388933E-17	.40017747E-10	-.79936058E-14
[illegible]	[illegible]					

[illegible] ELEMENTS IN THE IDENTITY MATRIX SMALLER THAN (EII)= .00010000
[illegible] THAN (EII), [illegible]

STANDARD DEVIATION OF COFFFICIENTS

1	.36863677E+06	2	.27839672E+07	3	.11700306E+04	4	.80738085E+04	5	.53070730E+07	6	.65271668E+00
7	.14007095E+05	8	.31469586E+01	9	.19231391E-03						

77319.165470123 RESIDUAL OR ERROR SUM OF SQUARES.
660588.55000019 TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
583269.38453007 REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
0.93965638616511 CORRELATION COEFFICIENT (R).
83.839221386761 SQUARE ROOT OF RESIDUAL VARIANCE.

— — — — — — — — — — —
— — — — — — — — — — —

Not shown are the predicted values, prediction errors and the Chi-square test for the first Bivor rerun and the entire main body of the remaining seven Bivor reruns.

— — — — — — — — — — —
— — — — — — — — — — —

BIVOR EXECUTION TIME 11.52400 SECONDS.

Total computer time required for all Bivor selected reruns.

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

MAIN RUN

(Analysis of Variance Table)

	DF	SS	MS	F
REGRESSION	9	585304.009976390	65033.778886265	8.638397587
REGRESSION				
REGRESSION				
ERROR	10	75284.540023804	7528.454002380	

CORRELATION .941293867

Y = 0.1282138588569OE+07 + -.11615967250284E+08 X(1) + -.2485897011610 4E+04 X(2) + 0.17033208382302E+05 X(3)
+ 0.3345335206583 9E+08 X(4) + 0.97098933633693E+00 X(5) + -.2951865945508 4E+05 X(6) + -.29234993064 79E+01 X(7)
+ -.29537023939548E+08 X(8) + -.23700792206804E-03 X(9)

INDEPENDENT VARIABLE SELECTION (HAND) 0011011101

(Analysis of Variance Table)

	DF	SS	MS	F
REGRESSION	3	78900.815399849	26300.271966616	0.723419675
REGRESSION				
REGRESSION				
ERROR	16	581687.734100340	36355.483381271	

CORRELATION .345601188

Y = 0.2664102682923 3E+06 + -.3265076249839 0E+07 X(1) + 0.1335094457823 0E+08 X(4) + -.1813649702150 0E+08 X(8)

BEGIN IVOR ANALYSIS OF VARIANCE TABLES

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

INDEPENDENT VARIABLE SELECTION (IVOR) 0101111111

		DF	SS	MS	F
REGRESSION	due to X_2	1	337028.450965640	337028.450965640	18.749259057
REGRESSION					
REGRESSION					
ERROR		18	323560.099034550	17975.561057475	

CORRELATION .714278802

Y = 0.162755519654325+03 + 0.27634433940883F+01 X(2)

INDEPENDENT VARIABLE SELECTION (IVOR) 0001111111

		DF	SS	MS	F
REGRESSION	due to X_1 and X_2	2	406962.031431320	203481.015715659	13.638862713
REGRESSION	due to X_2	1	337028.4509		
REGRESSION	due to X_1 (in addition to X_2)	1	69933.5805	69933.5805	9.289* $\left(= \frac{69933.5805}{7528.4540}\right)$
ERROR		17	253626.518568873	14919.206974640	

CORRELATION .784894752

Y = -.180647354869995+04 + 0.792053438123476+04 X(1) + 0.273410239333205+01 X(2)

* significant at .05 significance level — testing against s^2 from the main run

— — — — — — — — — —
— — — — — — — — — —

The analysis of variance tables for the third and fourth ivor regressions are not shown

— — — — — — — — — —
— — — — — — — — — —

BEGIN BIVOR ANALYSIS OF VARIANCE TABLES

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

INDEPENDENT VARIABLE SELECTION (BIVOR) 0C00000010

	DF	SS	MS	F
REGRESSION	8	583269.384530070	72908.673066258	10.372530521
REGRESSION				
REGRESSION				
ERROR	11	77319.165470123	7029.015042738	

CORRELATION .939656386

Y = 0.7414695907613?E+06 + -.5460761010802?E+07 X(1) + -.2245052761296?E+04 X(2) + 0.1519833860559?E+05 X(3)
+ 0.1010552855839?E+08 X(4) + 0.9411115013925?E+00 X(5) + -.2607293279429?E+05 X(6) + -.2793555783143?E+01 X(7)
+ -.2398096764744?E-03 X(9)

— — — — — — — — — —
— — — — — — — — — —

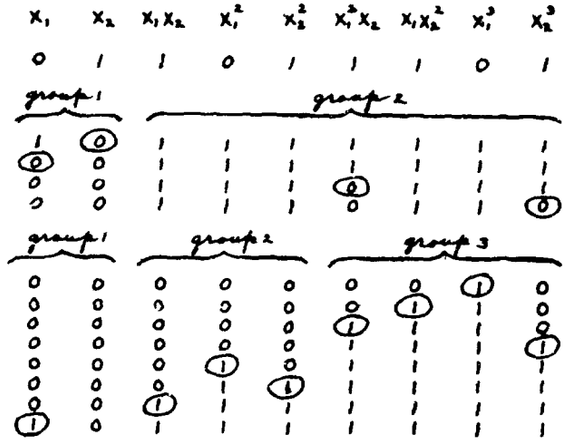
The analysis of variance tables for the last seven Bivor returns are not shown.

— — — — — — — — — —
— — — — — — — — — —

EXAMPLE PROBLEM (DUNCAN, 1959, PAGE 697)

DEGREES OF FREEDOM OF ERROR VARIANCE = 10 (Final Comprehensive Analysis Table)

COEFFICIENT OF DETERMINATION	NO (DF) OF DELETED VARIABLES	F FOR REGRESSION ON DELETED VARIABLES	INDEPENDENT VARIABLE SELECTION,
0.8860341 - MAIN RUN			
0.1194402	4	11.211	0011011101
*** IVOR FINAL COMPREHENSIVE ***			
0.5101942	4	4.122	0101111111
0.6160598	7	3.384*	0001111111
0.7604068	6	1.837	0001110111
0.8022130	5	1.471	0001110110
*** BIVOR FINAL COMPREHENSIVE ***			
0.4829541	1	0.270	0000000010
0.5745693	2	0.503	0000000110
0.6456336	3	1.182	0000001110
0.7056201	4	1.764	0000001111
0.7844139	5	1.783	0000101111
0.7447045	6	2.038	0000111111
0.6160598	7	3.384*	0001111111
0.5101942	8	4.122	0101111111



Encircled: IV's additionally included (0's) or deleted (1's).

Ivor ranking: $\underbrace{X_2, X_1}_{\text{group 1}}$; $\underbrace{X_1^2X_2, X_2^3}_{\text{group 2}}$

Bivor ranking: $\underbrace{X_2, X_1}_{\text{group 1}}$; $\underbrace{X_1X_2, X_2^2, X_1^2}_{\text{group 2}}$; $\underbrace{X_2^3, X_1^2X_2, X_1X_2^2, X_1^3}_{\text{group 3}}$

COMPREHENSIVE PRINTOUTS 5.599558638 SECONDS.

TOTAL PROBLEM - (UNITS, TIMEHRS./MIN./SEC.)=00:00:29

NWL REPORT NO. 2035

VII. FAILURE ANALYSIS

This chapter is concerned with failures which may occur in the use of the DA-MRCA program. In this context, a "failure" is defined in a very broad sense: It is meant to include all cases in which the user receives an output from the program which is principally different from what he expected to receive and what he was justified, from his own good judgment, to expect.

VII.1 Classification of Failures

The program user probably will encounter cases in which the desired results of the regression analysis cannot be obtained in specific runs. The program will indicate this failure (a) by stating, in some form, that the inverse of the matrix of the normal equations could not be obtained, or (b) by making a statement that the calculated identity matrix failed the accuracy check on the main-diagonal element deviations from 1. (For details about the statements, see Section VI.2.) Sometimes an inverse is obtained by the program although the user knows that the matrix is singular. This type of failure, however, should always become obvious by the accuracy checks on the identity matrix.

In this chapter the above indicated failures and their causes, as far as they are known to the authors, are analyzed and some corrective measures are discussed which the user might apply in order to obtain the desired problem solution. It can generally be stated that the failures are caused by inherent computer inaccuracies. The only exception is when no inverse is obtained because there are unknown linear dependencies among the rows or columns of the matrix of the normal equations.

The chart given on the following page represents a classification of possible failures and their causes. The chart should be self-explanatory; the causes as indicated in the appropriate boxes are defined and discussed, along with some corrective measures, in Section VII.2. The authors do not claim that the list of causes is complete; however, all causes known to the authors are given.

In the main area of failures, where the matrix is expected to invert and the calculated identity matrix is expected to pass the accuracy checks (first two rows of the chart), the analyst will be unable to readily identify the cause(s) of the program failure since he cannot be certain that theoretically there is a solution. However, by following the suggested corrective measures to be discussed, he may be able to obtain a solution and thereby to identify the cause(s) of the original failure.

The user of the program might ask why he should encounter the case in which the matrix is not expected to invert (last row of the chart) when in fact theoretically there is no solution but the program yields an inverse. (Such an inverse, however, will be identified as fictitious by the inaccurate identity matrix.) This case may indeed occur, for example, in the main run, when the analyst specifies a series of feasible independent variable selections (by hand) from an original set of N independent variables where N is larger than or equal to the number, n_N, of distinct input design points.

It is important to note that obtaining an inverse in such a situation constitutes, from the analyst's point of view, a failure with respect to what should be expected from the program. The event of obtaining this kind of fictitious inverse, therefore, has its proper place in the failure chart.

Failure Chart*

<table>
<tr><td colspan="2"></td><td>Matrix inverts but identity matrix fails accuracy check</td><td>Matrix does not invert</td></tr>
<tr><td rowspan="2">Analyst expects the matrix to invert and the identity matrix to pass accuracy check (since there are no obvious linear dependencies)</td><td>Theoretically there is a solution</td><td colspan="2">Cause of failure:
Limited computer accuracy</td></tr>
<tr><td>Theoretically there is no solution</td><td>Cause of failure:
Non-obvious linear dependencies plus truncation errors</td><td>Cause of failure:
Non-obvious linear dependencies</td></tr>
<tr><td>Analyst does not expect the matrix to invert (since there are obvious linear dependencies)</td><td>Theoretically there is no solution</td><td>Cause of failure:
truncation errors</td><td></td></tr>
</table>

* For the definitions of the terms used in the Failure Chart see the remaining sections of this chapter.

VII.2 Discussion of Failure Causes, Some Corrective Measures, and Examples

In this section the three failure causes, i.e., limited computer accuracy, linear dependencies, and truncation errors, will be discussed and some corrective measures and examples be given.

VII.2.a Limited Computer Accuracy

As is well known, no computer, large as it may be, is an "ideal computer," that is, a computer with absolute accuracy. The inaccuracy of the IBM 7030, for example, with its error in the fourteenth decimal digit (when using single precision as done in the present program), is large enough to effect the matrix inversion calculations to the extent that the inverses of large matrices might be worthless. Without presenting the details of the error propagation as present in the modified Gaussian elimination method used in the program, it can be stated that most errors are introduced by the subtraction of large numbers from other large numbers where these numbers differ only in the last few digits. These digits may well be beyond the last accurate one, i.e., beyond the thirteenth digit at the start of the calculations. One consequence of this may be, for example, the appearance of one or more negative elements in the main diagonal of the inverse, leading to the program statement that an inverse could not be obtained. Another consequence could be that, although the inverse can be obtained, the calculated identity matrix, I_c, deviates from the true identity matrix such that the accuracy checks on the main diagonal elements of I_c fail. This "limited computer accuracy" will cause failures most often in polynomial regression with high order terms contained in the model. At this point it must be recalled that the criterion by which the program accepts or rejects a run is dependent upon the analyst's choice. That is, the program user chooses the value of I(2) which will be the critical value not to be exceeded by the deviation (from 1) of any main diagonal element of the calculated identity matrix. (See Section VI.1.b.)

As a corrective measure to overcome the failures caused by the limited computer accuracy the following transformation of the independent variables is sometimes sufficient:

$$v = \frac{x-\bar{x}}{R_x} \quad . \qquad \text{(VII-1)}$$

This transformation, which is often also referred to as "coding" of the x's, is essentially a standardization, with centralization effected by the subtraction of the average, $\bar{x}$, from the original observation, x,

and with $|v| < 1$ effected through division by the range $R_x = x_{max} - x_{min}$. The transformation will be applied only to the "original" independent variables (OCIV's), and in polynomial regression, all higher order and cross-product terms (GCIV's) will be generated from the v variables. (As can easily be seen, if the GCIV's were also transformed, the matrix of the normal equations would have characteristics similar to those of a Hilbert matrix.) The transformation has the effect of keeping close to zero those elements in the matrix of the normal equations which, in polynomial regression, are sums of odd powers of the v values ($\Sigma v^3 \approx 0$, for example), or those elements which, in general multiple regression, are proportional to the covariance of two uncorrelated independent variables ($\Sigma v_1 v_2 \approx 0$, for example). The other elements of the matrix, for instance, the sums of the even powers in polynomial regression, are kept small by the transformation because of $|v| < 1$. The transformation, then, results in sufficiently large contrasts among the matrix elements of now smaller absolute value such that the subtractions mentioned before can be done with much higher accuracy.

It should be noted that the adjustment for the average x value as achieved in the v transformation leads to a much higher computational accuracy than can be achieved by starting with the regression model (VI-2) in which the independent variables are adjusted for their average values.

In case of polynomial regression the v transformation can become problematic to the program user who needs or wants prediction equations in the original x space. Only under a rather severe restriction (to be defined) will the regression sum of squares (ASSR) due to a group of independent variables in the v space be equal to the regression sum of squares due to the corresponding group of independent variables in the x space. Before defining the restriction, a very simple example is given in order to illustrate the situation. This example contains only one "original" independent variable, x. Imagine first that only its squared term (x^2) is included in the regression model. The regression sum of squares adjusted for the mean, ASSR, due to x^2 is:

$$ASSR(x^2) = \frac{[\Sigma\, x^2(y-\bar{y})]^2}{\Sigma\, (x^2-\overline{x^2})^2} .$$

Applying the v transformation to x, one gets for the corresponding regression sum of squares due to v^2:

$$ASSR(v^2) = \frac{[\Sigma\, v^2(y-\bar{y})]^2}{\Sigma\, (v^2-\overline{v^2})^2} .$$

Since $v = \frac{x-\bar{x}}{R_x}$, $ASSR(v^2)$ can be rewritten as

$$ASSR(v^2) = \frac{[\Sigma\ (x-\bar{x})^2\ (y-\bar{y})]^2}{\Sigma\ [(x-\bar{x})^2 - \overline{(x-\bar{x})^2}]^2}\ .$$

Now it can be shown that

$$ASSR(x^2) \neq ASSR(v^2).$$

For this it is sufficient to show that the two denominators are not proportional to each other. Indeed, one has

$$\Sigma\ (x^2-\overline{x^2})^2 = \Sigma\ x^4 - \frac{[\Sigma\ x^2]^2}{n}$$

$$\Sigma\ [(x-\bar{x})^2 - \overline{(x-\bar{x})^2}]^2 = \Sigma\ x^4 - \frac{[\Sigma\ x^2]^2}{n} + \delta,$$

where δ is not identically zero:

$$\delta = 4n\bar{x}\ [-\ (\bar{x})^3 + 2\bar{x}\overline{x^2} - \overline{x^3}] \neq 0.$$

Imagine next that only the linear terms, x or v, are included in the two models. It is easy to show that the two regression sums of squares are now equal:

$$ASSR(x) = \frac{[\Sigma\ (x-\bar{x})(y-\bar{y})]^2}{\Sigma\ (x-\bar{x})^2}$$

$$ASSR(v) = \frac{[\Sigma\ (v-\bar{v})(y-\bar{y})]^2}{\Sigma\ (v-\bar{v})^2}\ .$$

Since $\bar{v}=0$, one has

$$ASSR(v) = \frac{R_x^2\ [\Sigma\ (x-\bar{x})(y-\bar{y})]^2}{R_x^2\ \Sigma\ (x-\bar{x})^2} = ASSR(x).$$

Finally, the two regression sums of squares are again equal when both the linear and quadratic terms are included in the models:

$$ASSR(x,x^2) = ASSR(v,v^2).$$

The algebraic proof for this is omitted because of its length.

More generally, it can be demonstrated that the respective regression sums of squares in the x and v space are equal only when the polynomial regression models of order k, say, also include all terms of lower order than k:

$$ASSR(x,x^2,\ldots,x^{k-1},x^k) = ASSR(v,v^2,\ldots,v^{k-1},v^k).$$

This condition is generally also valid for polynomial regression models in more than one original independent variable. For example, in a case of two original independent variables, x_1 and x_2, and a model which is to include the cross-product term (x_1x_2 or v_1v_2), one has to include also the linear terms (x_1 and x_2, or v_1 and v_2, respectively) in order to have the regression sums of squares equal in the x and the v space:

$$ASSR(x_1,x_2,x_1x_2) = ASSR(v_1,v_2,v_1v_2).$$

This leads to the following conclusion. When the program user finds, for accuracy purposes, a need to apply the transformation (VII-1) and when he wants to keep, with respect to the regression sums of squares, the relations between corresponding terms of the two polynomial regression models undisturbed by the transformation, he must follow this Restriction: A polynomial regression model must contain all polynomial terms (including the linear terms) which can be separated as factors from the highest order terms contained in the model.

The program user can easily adhere to this restriction when linear hypotheses are to be tested by the option for hand selected reruns. When the user wants to automatically rank the transformed polynomial terms by IVOR or BIVOR, he can adhere to the restriction by application of the grouping feature as available in both routines. For this the polynomial terms should be grouped according to their powersum which is defined to be the sum of all exponents of the original independent variables contained in a term. For example, in a polynomial of second degree in two (transformed) independent variables v_1 and v_2, there would be two groups in IVOR and in BIVOR: v_1 and v_2 would form the first group with a powersum of 1 in each term, and v_1^2, v_1v_2, and v_2^2 would form the second group with a powersum of 2 in each term. Since the ranking begins in the first group in IVOR and in the last group in BIVOR, it can be seen that the above restriction is followed. It is, however, obvious that the restriction is being followed in an overstrict fashion: When in BIVOR, for example, v_2^2 and v_1v_2 have been found to be the least important terms in the last (second) group, v_1^2 is ranked automatically as the next least important term. In reality, at this step both v_1^2 and v_2 should be "admissible" for the determination of which term contributes less to the regression sum of squares when contained in the model. Note: In NOVACOM (see Section II.3) a BIVOR type ranking procedure can optionally be performed such that at each step all those polynomial terms become "admissible" for ranking which cannot be separated as factors from other terms contained in the model. Therefore, the terms become admissible in the

desired fashion, that is, according to the above restriction to be followed when the accuracy transformation

$$v = \frac{x-\bar{x}}{R_x}$$

is applied and when the models in the x and in the v space are to correspond to each other.

When the program user adheres to the restriction, he will in fact have a model (for example, a significant model) which corresponds, term by term, to the model in the original space. If it is desired and feasible, the program user can then retransform the values of the estimated regression coefficients into the values which the corresponding coefficients have in the original space. Naturally, the retransformation is very simple when product terms are not included in the model. In this case the regression coefficients of the original space are obtained by dividing the regression coefficients of the transformed space by the respective ranges R_x. In general, however, one would make use of the model obtained in the transformed space by transforming the coordinates of any design point of the original space for which one wants to compute the predicted value of the dependent variable and/or confidence limits.

Although the transformation (VII-1),

$$v = \frac{x-\bar{x}}{R_x} ,$$

seems to be the most effective one to increase the accuracy, division by a constant or subtraction of a constant sometimes is satisfactory. Division by a constant, that is the transformation $v' = \frac{x}{E}$, avoids the disadvantages which are characteristic of the transformation (VII-1): The retransformation of the model consists merely of dividing the regression coefficient obtained in the transformed space by E. In polynomial regression the retransformation consists of dividing the obtained regression coefficient of a polynomial term by the corresponding product of the E values used in the transformation of the original independent variables. For example, the regression coefficient obtained for the term

$$\frac{x_1}{E_1}\left(\frac{x_3}{E_3}\right)^2$$

is retransformed by dividing by $E_1E_3^2$.

The effect of the $\frac{x}{E}$ transformation, with respect to accuracy, is similar to that of the division by R_x in the v transformation: If the value of E is properly chosen, the absolute values of the transformed data can be made to lie between 0 and 1. This can sometimes be

achieved by choosing the proper power of ten for E, in which case the transformation can easily be executed by hand. However, this transformation is of little value if all or most of the untransformed OCIV coordinates are of equal sign. In this case the other simple transformation, i.e., the subtraction of a constant such that centralization is achieved, is sometimes sufficient. The constant G in this transformation, $v'' = x-G$, should be conveniently chosen close to the average of the x values, i.e., G should be a "working average." If it is appropriate to choose G as a whole number, this transformation also can easily be performed by hand. The transformation x-G has, however, the same type of side-effects with respect to the retransformation of a polynomial model as were shown to exist for the transformation $\frac{x-\bar{x}}{R_x}$.

The transformations

$$v = \frac{x-\bar{x}}{R_x} \text{ and } v' = \frac{x}{E} \text{ (but not } v'' = x-G)$$

can automatically be applied to the coordinates of the OCIV's by the preprocessor program MTRAN, as was mentioned in Section II.2. The output of MTRAN may be on cards or tape and represents the data input for DA-MRCA, i.e., the information usually punched on Card Type 8.

The following numerical example is given in order to illustrate the effects of the transformation $\frac{x-\bar{x}}{R_x}$. The problem contains one original independent variable x with 9 distinct levels. In the x space a polynomial of 5th degree was the highest that could be fitted by DA-MRCA, whereas, after application of the v transformation, a polynomial of 8th degree could be obtained. (Naturally in this example, this is the zero error perfect fit.) The printout shown is a reproduction of a part of the original printout of DA-MRCA for this example. The 9 data points are given below, where also the transformed (coded) x values are shown.

y	x	$v = \frac{x-\bar{x}}{R_x}$
9.5	47.30	-.45861017
0.6	47.47	-.45276825
43.7	54.65	-.20603285
49.9	54.83	-.19984729
48.3	61.90	+.04310804
65.5	64.20	+.1221458
96.4	68.43	+.26750667
128.5	70.63	+.34310804
149.1	76.40	+.5413898

EXAMPLE OF CODING EFFECT UPON COMPUTING ACCURACY, UNCODED DATA

INDEPENDENT VARIABLE SELECTION (HAND) C00000111 → Third rerun; model: $Y = \beta_0 + \sum_{v=1} \beta_v X^v$

MATRIX INVERSION 3 ...EVALUATION TIME = 0.02210582 SECONDS.

DETERMINANT= 0.1925274625E+33

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

.87124984E+09	-.71950633E+08	.23589329E+07	-.38387323E+05	.31013258E+03	-.99536155E+00	.13742652E+06
-.71950646E+08	.59423077E+07	-.19483425E+06	.31707873E+04	-.25618657E+02	.82227926E-01	-.11953279E+05
.23589338E+07	-.19483428E+06	.63885969E+04	-.10397710E+03	.84015133E+00	-.26968139E-02	.41194602E+03
-.38387345E+05	.31707885E+04	-.10397712E+03	.16923919E+01	-.13675775E-01	.43901217E-04	-.70317779E+01
.31013282E+03	-.25618671E+02	.84015166E+00	-.13675778E-01	.11051839E-03	-.35480577E-06	.59459991E-01
-.99536249E+00	.82227989E-01	-.26968155E-02	.43901234E-04	-.35480583E-06	.11301437E-08	-.19923119E-03

IDENTITY MATRIX

.99822998E+00	-.15234375E-00	-.90000000E+01	-.45600000E+03	-.34816000E+05	-.23592960E+07
.13732910E-03	.10097656E+01	.56250000E+00	.53000000E+02	.27520000E+04	.19660800E+06
-.46491623E-05	-.32806396E-03	.97607422E+00	-.19375000E+01	-.12800000E+03	-.81920000E+04
.67055225E-07	.44107437E-05	.28228760E-03	.10219727E+01	.13125000E+01	.76000000E+02
-.77125151E-09	-.47497451E-07	-.30994415E-05	-.22888184E-03	.98730469E+00	-.11875000E+01
.23339802E-11	.13460522E-09	.95460564E-08	.67055225E-06	.38146973E-04	.10033569E+01

DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MATRIX LARGER THAN I(2)= .01000000 .RUN REJECTED.

STANDARD DEVIATION OF COEFFICIENTS

1 .17992662E.06 2 .14899412E+05 3 .48722191E+03 4 .79300247E+01 5 .64082779E-01 6 .20573743E+03

111.47292301175 RESIDUAL OR ERROR SUM OF SQUARES.
20274.775555556 TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
20163.302632544 REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
0.99724715649801 CORRELATION COEFFICIENT (R).
0.09570676820 31 SQUARE ROOT OF RESIDUAL VARIANCE.

Note: This is the highest order polynomial (degree 5) for which the matrix of the normal equations could be inverted when the uncoded (untransformed) data was used. Although, with the choice of I(2)=.01, the run was rejected, the results are sufficiently close to those of the corresponding run with coded data. (See page 137)

EXAMPLE OF CODING EFFECT UPON COMPUTING ACCURACY. CODED DATA

MATRIX INVERSION 0 ...EVALUATION TIME = 0.04094003 SECONDS.

DETERMINANT= 0.19979109756-37

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

.98257836E+01	-.21289134E+03	.73867005E+03	.74267705E+04	-.31554249E+05	-.55436615E+05	.25422357E+06
.12724609E+04	-.96029049E+04	.52468371E+02				
-.21289138E+03	.93345390E+04	-.26534093E+05	-.22267866E+06	.10447991E+07	.16573487E+07	-.83122723E+07
-.38224088E+07	.18219970E+04	-.21424533E+03				
.73867005E+03	-.26536093E+05	.13466124E+06	.94378840E+06	-.50260371E+07	-.69783660E+07	.39549570E+08
.10184241E+08	-.86213304E+04	.24911650E+04				
.74267705E+04	-.22247866E+05	.94378840E+06	.74510158E+07	-.37015590E+08	-.58582339E+08	.29449993E+09
.13544784E+09	-.64589590E+03	.97566286E+04				
-.31554249E+05	.10447991E+07	-.50260371E+07	-.37015590E+08	.18954180E+09	.27426930E+09	-.14982034E+10
-.63532340E+09	.32734120E+11	-.70933321E+05				
-.55436615E+05	.16573487E+07	-.69783660E+07	-.58582339E+08	.27426930E+09	.43986180E+09	-.21873398E+10
-.10227595E+10	.48099638E+11	-.51902213E+05				
.25422357E+06	-.83122723E+07	.39549570E+08	.29449993E+09	-.14982034E+10	-.21873398E+10	.11864175E+11
.50776686E+10	-.25960369E+11	.51884104E+06				
.12724609E+04	-.38226088E+07	.18184241E+08	.13546784E+09	-.63532340E+09	-.10227595E+10	.50776686E+10
.23021330E+10	-.11194C39E+11	.90339650E+05				
-.96224C43E+04	.18219970E+04	-.86213304E+04	-.64589590E+09	.32734120E+10	.48099638E+10	-.25960369E+11
-.11194C39E+11	.58882787E+11	-.10441591E+07				

DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLER THAN I(1)= .00010000 .RUN ACCEPTED.

STANDARD DEVIATION OF COEFFICIENTS

1	.00000000E 00	2	.00000000E 00	3	.00000000E 00	4	.00000000E 00	5	.00000000E 00	6	.00000000E 00
7	.00000000E 00	8	.00000000E 00	9	.00000000E 00						

-.1921085A4E-05 RESIDUAL OR ERROR SUM OF SQUARES.
20274.77549994 TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
20274.77549574?7 REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
1.00000000004?4 CORRELATION COEFFICIENT (R).
.00000000000E 00 SQUARE ROOT OF RESIDUAL VARIANCE.

Main run; model: $Y = \beta_0 + \sum_{v=1}^{8} \beta_v V^v$, where $V = \frac{X - \bar{X}}{R_x}$

Note: With n=9 data points, this (main) run represents the zero-error perfect fit. The negative RESIDUAL SUM OF SQUARES is due to computational errors only.

EXAMPLE OF CODING EFFECT UPON COMPUTING ACCURACY. CODED DATA

Third rerun; model:
$Y = \beta_0 + \sum_{v=1}^{5} \beta_v V^v$
where $V = \frac{x - \bar{x}}{R_x}$

INDEPENDENT VARIABLE SELECTION (HAND) (000000111)

MATRIX INVERSION 3 ...EVALUATION TIME = 2.02499746 SECONDS.

DETERMINANT= 0.23315553468E-11

INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIONS

.14271061E+01	-.79243493E+01	-.49362193E+02	.14528440E+03	.22193133E+03	-.66977565E+03	.51065311E+02
-.79243493E+01	.85977307E+02	.34310059E+03	-.17884121E+04	-.16622066E+04	.61851823E+04	.19315036E+02
-.49362193E+02	.34310059E+03	.20165359E+04	-.80707301E+04	-.93509487E+04	.29173431E+05	.29602056E+03
.14528440E+03	-.17884121E+04	-.80707301E+04	.39695216E+05	.38958423E+05	-.13987238E+06	.15933383E+04
.22193133E+03	-.16622066E+04	-.93509487E+04	.38958423E+05	.43009451E+05	-.14046697E+06	-.68317505E+03
-.66977565E+03	.61851823E+04	.29173431E+05	-.13987238E+06	-.14046697E+06	.49609449E+06	-.41566304E+04

DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLER THAN I(1)= .00010000 .RUN ACCEPTED.

STANDARD DEVIATION OF COEFFICIENTS

1 .72826431E-01 2 .54526509E-02 3 .27375414E-03 4 .12145908E+04 5 .12774387E+04 6 .42938127E+04

111.49182775733 RESIDUAL OR ERROR SUM OF SQUARES.
20274.775955596 TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.
20163.283727709 REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN.
0.99724668899841 CORRELATION COEFFICIENT (R).
0.0962236331828 SQUARE ROOT OF RESIDUAL VARIANCE.

(See the note on page 135.)

VII.2.b Linear Dependencies

Linear dependencies among all or some of the rows (columns) of the matrix of the normal equations of a given run will cause this matrix to be singular and, therefore, fail to invert. Sometimes a fictitious inverse will be computed by the program because of the presence of truncation errors, see Section VII.2.c below. In some cases the analyst will be able to infer, from visual inspection of the number and the relative position of the n_K distinct input design points, as given in the design matrix, that linear dependencies are present. These will be referred to as "obvious" linear dependencies. They occur, for example, when the analyst includes as many or more independent variables in the regression model of a given run as there are distinct design points. For a discussion of some obvious linear dependencies see the end of this section.

In general, the linear dependencies will be "non-obvious" and, therefore, unknown to the analyst from visually inspecting the design matrix. It is in this sense that the linear dependencies are discussed here as a cause for a failure. The algebraic parts of the discussion are presented in terms of the main run; however, all conclusions are naturally equally valid for any rerun.

The matrix A of the normal equations of the main run can be expressed in terms of the design matrix X as follows:

$$A = X'X,$$

with

$$X = \begin{bmatrix} x_{01} & x_{11} & x_{21} & \cdots & x_{v1} & \cdots & x_{N1} \\ x_{02} & x_{12} & x_{22} & \cdots & x_{v2} & \cdots & x_{N2} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ x_{01} & x_{11} & x_{21} & \cdots & x_{v1} & \cdots & x_{N1} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ x_{0n} & x_{1n} & x_{2n} & \cdots & x_{vn} & \cdots & x_{Nn} \end{bmatrix},$$

where $x_{01}=1$. Since

$$\text{rank } [A] = \text{rank } [X],$$

X must be of rank N+1 in order that A is a non-singular matrix, assuming that $n_N \geq N+1$. By definition, X is of rank N+1 when no linear dependencies

exist among its N+1 columns. In other words, as soon as the coordinates $\{x_1, x_2, \cdots, x_\nu, \cdots, x_N\}_i$ of the n_N distinct input design points satisfy the identity

$$\sum_{\nu=0}^{N} a_\nu x_{\nu i} = 0 \quad \{i = 1,\ldots,n_N\} \qquad \text{(VII-2)}$$

with at least two coefficients, a_ν, being different from zero, the rank of X is smaller than N+1 and, thereby, A is singular. In a geometrical interpretation, the identity

$$\sum_{\nu=0}^{N} a_\nu x_{\nu i} = 0 \quad \{i\}$$

means that all n_N distinct design points are located on a hyperplane in the N-dimensional space defined by the N independent variables. (This hyperplane could have, at the most, N-1 dimensions.) Except for the cases of "obvious" linear dependencies, the analyst will not be able to determine, without further analysis, whether or not the n_N distinct input design points are located on a plane in the N-dimensional space. Should he want to determine this by analytical means, he would have to calculate the value of the determinant of the matrix consisting of any N+1 rows of X which represent distinct design points. This can be a considerable effort. In the present program, therefore, the detection of this general case of "non-obvious" linear dependencies is left to the built-in checks for the possibility of obtaining an inverse and to the checks on the accuracy of the calculated identity matrix. When "non-obvious" linear dependencies are present for a given independent variable selection and when a fictitious inverse is obtained, the main diagonal elements of the calculated identity matrix will deviate rather drastically from 1 and the run will clearly be rejected.

The only adequate corrective measure in the case of non-obvious linear dependencies is to delete one independent variable and to try to fit the reduced regression model. As discussed in Section VI.2.d, this deletion is performed automatically in the BIVOR option. IVOR, by nature, has an advantage over BIVOR in the handling of non-obvious linear dependencies and the identification of perfect fits. Since in BIVOR, indiscriminantly, the rightmost independent variable is deleted after a run was rejected, this deletion does not necessarily eliminate the unwanted non-obvious linear dependency. In fact, there could be many such deletions of rightmost IV's before a perfect fit is reached by BIVOR. IVOR, in contrast, will select, at each step, only those independent variables for possible inclusion into the model whose inclusion will not introduce linear dependencies. By this technique IVOR is capable of always finding the perfect fit with the maximum number of independent variables contained in the model.

Another remark regarding linear dependencies concerns the situation in which functions of the original independent variables are added to the model, as is the case, for example, in polynomial regression. Namely, it is wrong to assume that functional terms can always be added when there are no (non-obvious) linear dependencies caused by the original independent variables. The following simple example from polynomial regression may serve to illustrate this and the concept of the "non-obvious" linear dependency in general.

Example. Given the following $n_N=4$ design points in the plane of the two original independent variables x_1 and x_2,

x_1	-1	0	+2	+3
x_2	+1	-2	-2	+1

,

the regression model to be fitted is, say:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2,$$

which with 4 distinct design points should lead to a "zero error perfect fit." The inclusion of the term x_1^2 appears to be feasible, but it nevertheless leads to a non-obvious linear dependency: the 4 points $\{x_1, x_2, x_1^2\}$ are located on a plane in the 3-dimensional space. As can easily be verified, the 4 points satisfy the identity of form (VII-2), i.e., the 4 points are on a plane having this equation:

$$2 + 2x_1 + x_2 - x_1^2 \equiv 0.$$

It is, therefore, not possible to include x_1^2 in the regression model when x_1 and x_2 are included.

In the following, some "obvious" linear dependencies are discussed, two of which are derived from the general case, i.e., by specifying the coefficients, a_ν, in the identity

$$\sum_{\nu=0}^{N} a_\nu x_{\nu i} = 0.$$

All these cases can readily be identified from the design matrix X without further analysis. As has been the case previously in this section, the discussion of the obvious linear dependencies also will be presented in terms of the main run, i.e., for N independent variables.

Some "obvious" linear dependencies:

(1) In the identity (VII-2),

$$\sum_{\nu=0}^{N} a_\nu x_{\nu i} \equiv 0, \quad i = 1,2,\ldots,n_N,$$

all coefficients a_ν except a_0 and $a_{\nu*}$ are zero:

$$x_{\nu* i} = -\frac{a_0}{a_{\nu*}} = \text{constant}.$$

This means that the coordinate $x_{\nu* i}$ is equal for all n_N distinct design points. All independent variables, $x_{\nu*}$, satisfying this condition must be deleted from the model.

(2) In the identity (VII-2), all coefficients except $a_{\nu*}$ and $a_{\nu**}$ are zero:

$$a_{\nu*}x_{\nu* i} + a_{\nu**}x_{\nu** i} \equiv 0,$$

or

$$\frac{x_{\nu* i}}{x_{\nu** i}} \equiv -\frac{a_{\nu**}}{a_{\nu*}} = \text{constant}.$$

This is the case of proportionality for all n_N coordinates $x_{\nu* i}$ and $x_{\nu** i}$. One independent variable out of each pair $x_{\nu*}$, $x_{\nu**}$ satisfying this condition must be deleted from the model.

(3) $n_N \leq N$. This is the case of trying to fit too many independent variables for the number, n_N, of distinct input design points available. It will be met mostly in situations where functions of the original independent variables have been included in the regression model, as is the case in polynomial regression. The identity (VII-2) is automatically fulfilled by the $n_N \leq N$ design points since all n_N points are necessarily located on a "plane" in the N-dimensional space defined by the N independent variables. At least $N-n_N+1$ independent variable(s) must be deleted from the model in order to arrive at a solution.

(4) This case applies only when functions $x_\nu = f_\nu(z_1, z_2, \ldots, z_j, \ldots)$ of the original independent variables, z_j, are included in the model, as is the case, for example, in polynomial regression. It is related to case (3) ($n_N \leq N$) and defined as follows. Let the number of distinct values (coordinates) of the original independent variable z_j be L_j. The set of all functional terms $x_\nu = f_\nu$ of the model which contain z_j can be divided into groups such that a group consists of all those terms f_ν which contain one or more other variables z_{j*} ($j* \neq j$), all in an identical functional form. (The terms

f_ν in one of these groups need not contain any variable other than z_j.) Let the maximum number of terms f_ν in any group be M_j. Then an obvious linear dependency exists if $L_j \leq M_j$.

As a complex and probably unrealistic example, intended to illustrate the above definition, imagine that the model includes the following set of 9 terms all of which contain $z_j (z_j \neq z_{j*} \neq z_{j**})$:

$$z_{j*} \sin(z_j),\ z_{j*} \sin(2z_j),\ z_{j*} \sin(3z_j);\ \cos(z_j),\ \cos(2z_j);$$

$$z_{j**} \cos(z_j z_{j*});\ z_j^2 {}_{**} \cos(z_j z_{j*});\ z_j^3 {}_{**} \cos(z_j z_{j*});\ z_j^4 {}_{**} \cos(z_j z_{j*}).$$

The first three terms contain z_{j*} in an identical functional form, namely as a multiplier. The next two terms do not contain any other variable than z_j; and the last four terms each contain z_{j**} in a different functional form. This makes 6 groups with 3, 2, 1, 1, 1, 1 terms, respectively. Therefore, M_j equals 3. Should the number L_j of distinct values of z_j be smaller than or equal to 3, the inclusion of the first of the above groups (with 3 terms) in the model would lead to an obvious linear dependency.

In this case of $L_j \leq M_j$ the identity (VII-2) is again automatically fulfilled since the total number n_N of distinct design points will be located on a "plane" in the N-dimensional space defined by the N independent variables, as can readily be verified. For each original independent variable z_j for which $L_j \leq M_j$ is true, at least as many terms containing z_j per group must be deleted from the model such that, at the most, L_j-1 terms per group will remain. In the above example, deletion of $z_{j*} \sin(3z_j)$, say, would eliminate the obvious linear dependency if L_j is assumed to be exactly 3.

VII.2.c Truncation Errors

Truncation errors are, naturally, present in all computations performed. As indicated before, these errors become particularly important in one situation, i.e., when the matrix is singular (obvious or non-obvious linear dependencies being present) and, consequently, an inverse does not exist. In this situation the truncation errors sometimes lead to a fictitious inverse which, however, in all cases should be identified as such by the failure of the calculated identity matrix to pass the accuracy checks. This fictitious inverse is usually caused by an element of the main diagonal of the inverse which theoretically has the value zero but actually equals a small positive quantity stemming from a truncation error. One can, in fact, construct very simple cases with singular matrices for which the computer will obtain fictitious inverses.

There is no possibility whatsoever to avoid the "failures" which are caused by these errors when one deals with singular matrices. The analyst has to rely entirely upon the accuracy check on the calculated identity matrix in order to be protected from this type of a fictitious problem solution. In the experience of the authors no actual case occurred in which the inverse of a matrix known to be singular passed the identity matrix checks.

VIII. FORTRAN IV DOCUMENTATION OF DA-MRCA

In previous chapters of this report, references to problem variables have, in most instances, been made in terms of the general mathematical notation used. However, in the programming and coding phases of the DA-MRCA program, it has been necessary to redefine some of these variables in an acceptable FORTRAN IV variable notation. In addition, other variables have required initial definition due to the storage allocation conventions of the FORTRAN IV language.

Some of these FORTRAN variables have been defined in previous chapters of this report. For example, variable descriptions are provided in Chapter V (INPUT PREPARATION). However, if the reader has the desire or need to study and understand the FORTRAN formulation of the program, additional information is required to associate the mathematical concepts with the FORTRAN IV documentation.

This chapter, therefore, presents the FORTRAN IV documentation of the DA-MRCA program in the form of a glossary of program variables, flow charts, conversion notes, and a complete listing of the program.

VIII.1 Description of Program Variables

In this section are defined the program variables which are contained, (a) in COMMON storage, (b) in the MAIN PROGRAM, and (c) in program subroutines.

Input variables, indices of DO-loops, most variables defined in DATA statements, and most arguments in subroutines are not defined here.

VIII.1.a Variables in COMMON Storage

A - an array containing the matrix (A) of the normal equations; subroutine GAUSS changes this matrix to its inverse.

AKP - an array into which the array A is saved before subroutine GAUSS is called.

AVV - an array which contains averages of the independent variables and the dependent variable.

AW - an array which contains averages of independent variables in subroutine PREVAR and which contains the various regression sums of squares adjusted for the mean in subroutines IVØR and BIVØR.

B - an array containing the constants, E_{yy}, of the normal equations; subroutine GAUSS changes this vector to contain the solution of the normal equations (i.e., the regression coefficients).

BB - an array which is used to save the constants, E_{yy}, of the normal equations.

BSDEV - an array which contains the standard deviations of the regression coefficients.

DETERM - the determinant of A.

ERRØR - a variable which is used as an error return from subroutines ABT and GAUSS and which controls printout in subroutine REDUCM.

IBIDS - a variable which is used in conjunction with IBID to control the computation and checking of the identity matrix.

ICASE - a counter for the number of inverse matrices which are printed.

ISKIP - if the main run was rejected for any reason, ISKIP=2; otherwise ISKIP=1.

ITØTAL - initially set equal to the rank of the matrix of the normal equations, A, for the main run, this value is later used, in IVØR and BIVØR, as the upper limit on the number of independent variables at various steps of these subroutines.

JLIM - a variable which is set equal to IR+1, the number of OCIV's given as input, plus 1.

KMUM - a variable which indicates step size in the looping used to read the data input.

KNUM - a variable which is used by subroutine RDIT as the number of data fields per record and by subroutine BIVØR to indicate to subroutine CASSR that CASSR is being called from BIVØR.

M - the total number of data points (= n in previous chapters).

M1 - a variable which indicates when the data termination card has been read.

M4 - a variable which is used to control page headings in subroutine CMPR.

N - the number of independent variables present in the model at any step.

NN - the number of independent variables present in the model (at any step), plus 2.

NNL - a variable which is used to index the last row and/or last column of the summation matrix S containing the constants, E_{yy}, of the normal equations.

NNN - the rank of the matrix of the normal equations at any step.

NNNSAV - a variable which saves the rank of the matrix of the normal equations for the main run.

NNSAV - a variable which saves the main run value of the variable NN.

NNXA - a variable equal to the main run value of the variable NNN.

NØBS - this variable (EQUIVALENCED to IDGØ in subroutines ABT, IDENTM, and PRINTM) is used to indicate the acceptance or rejection of the identity matrix.

NPED - a variable which controls the predicted value and Chi-square computations.

RECM - the reciprocal of the number of observations M.

RSSMØ - this variable value equals the main run regression sum of squares adjusted for the mean. If the main run does not pass the four checks on the determinant of A, R^2, s^2, and the $c_{\nu\nu}$ (see paragraphs B, D, E, and F of Section VI.2.a.(2)), this value is negative indicating that no final comprehensive is to be printed.

S - the summation matrix; the first N+1 rows and N+1 columns represent the matrix of the normal equations; the $(N+2)^{th}$ row and column are the constants, $E_{\nu y}$ ($\nu = 0,1,\ldots,N$), and E_{yy}, of the normal equations.

SDEV - the square root of the residual variance.

SELECT - this variable indicates whether a rerun is a hand selected rerun, an IVØR rerun, or a BIVØR rerun, for printout purposes.

X - an array which contains the coordinates for each data point.

XD - an array which is used in subroutine PREVAR to contain the coordinates of the selected input or synthetic design points adjusted for the averages of the corresponding input coordinates.

YSDEV - an array containing the prediction standard deviations.

YY - an array containing the predicted values.

VIII.1.b Variables in the MAIN PROGRAM

IAPE - the actual logical tape number of the tape containing the coordinates of the data points.

INDX - a variable which is set equal to IR+1. The coordinates of the first IR independent variables, modified by the independent variable selection, are printed to identify the selected input and/or synthetic design points.

KØUNT - counter for the selected input and/or synthetic design points.

MMM - an index used in the coding to reverse the order of input items in the LØT array.

NSAV - saves the main run value of N.

XIT - the time which is computed by the various timing subroutines.

XYIT - used only as a required argument to the EØF function.

VIII.1.c Variables in Program Subroutines

(1) Variables in Subroutine ABT*

ASSR - the regression sum of squares adjusted for the mean.

ATSS - the total sum of squares adjusted for the mean.

CHI - an array whose j^{th} element contains a contribution to the Chi-square statistic if the j^{th} interval is the last of a group of intervals having a total of more than 5 expected prediction errors. Otherwise CHI(J) = -1.0.

CHISUM - The Chi-square statistic.

CMPFR - an array whose j^{th} element contains summed expected prediction errors if the j^{th} interval was the last of a group of intervals having a total of more than 5 expected prediction errors. Otherwise, the contents of CMPFR(J) are meaningless.

CØR - the correlation coefficient.

CØRSQ - the square of the correlation coefficient (i.e., the coefficient of determination).

*Note - T. Herring, who coded the program DA-MRCA, named this subroutine for a co-author of the report.

EDELTA - the interval size in the Chi-square computations.

ERANGE - the range of the prediction errors.

ESSØ - the main run value of the error sum of squares.

ESTEP - an array which contains the upper bounds of the 30 intervals, into which the range of the prediction errors is divided.

ES2 - the sum of squares of the prediction errors; the check error sum of squares.

EYYL - the minimum prediction error.

EYYU - the maximum prediction error.

FGRAPH - an array which contains the symbols for the prediction error frequency distribution bar chart.

FN - a floating point representation of the rank of the matrix of the normal equations.

FØUT - the F ratio for regression on deleted variables.

IDF - the degrees of freedom of Chi-square.

IFGRPH - the number of symbols which are to be printed on a line in the prediction error frequency distribution bar chart.

IFREQ - an array which contains the frequencies of occurrence of prediction errors in the intervals delimited by the ESTEP array.

IØBF - an array whose j^{th} element contains the summed observed frequencies of prediction errors for a group of intervals if the j^{th} interval was the last of a group of intervals containing a total of more than 5 expected prediction errors. Otherwise, the contents of IØBF(J) are meaningless.

LXMAX - the element number of the maximum prediction error.

LXMIN - the element number of the minimum prediction error.

NR0 - the number of independent variables for the main run.

NR1 - the number of data points minus the main run value of NNN, i.e., the degrees of freedom of the error variance.

NR2 - the number of independent variables which have been deleted from the model.

SDEVSQ - the residual variance.

SSE - the residual, or error, sum of squares.

SSR - the unadjusted regression sum of squares.

XIT - the time, in seconds, for the execution of subroutine GAUSS.

(2) Variables in Subroutine BIVØR

AMAX - the maximum ASSR value.

AMIN - the minimum ASSR value.

IDUM - a dummy argument to subroutine CASSR.

IMAX - the LØT array index of that independent variable which is to be deleted from the model.

ISEE - a variable value which ensures that the identity matrix will be checked only until an inverse is found whose associated identity matrix element deviations are all smaller than I(1).

ISTART - a variable value which is used to define the LØT array index of the leftmost independent variable of a group of independent variables.

IXMAX - the index of the maximum ASSR value in the AW array.

IXMIN - the index of the minimum ASSR value in the AW array.

JLØT - a variable used to index the regression coefficients and inverse matrix diagonal elements which are due to independent variables for which ASSR values are to be computed.

JSAVE - the index of the LØT array element which element is to be set equal to 1 if the matrix inversion is not accepted.

KASSR - a counter of the ASSR values which are computed at each step.

KGØ - a variable which indicates the failure of the matrix inversion in subroutine CASSR.

LAT - an array which holds the LØT array indices of the independent variables for which ASSR values are computed.

NØBS - a variable which indicates whether or not a matrix inversion is the first accepted inversion in subroutine BIVØR.

NQQ - a variable value equal to the number of independent variables in a group in the grouping feature for independent variables.

(3) Variables in Subroutine CASSR

ASSR - the regression sum of squares adjusted for the mean.

ATSS - the total sum of squares adjusted for the mean.

CØRSQ - the square of the correlation coefficient.

FNNN - a floating point representation of the rank of the matrix of the normal equations.

SDEVSQ - the residual variance.

SSE - the residual, or error, sum of squares.

SSR - the unadjusted regression sum of squares.

(4) Variables in Subroutine CHISQ

FØ - the actual number of prediction errors in a group of intervals in the search for a group of intervals having more than 5 expected prediction errors.

FØC - the computed (expected) number of prediction errors in a group of intervals in the search for a group of intervals having more than 5 expected prediction errors.

FØMRE - the program looks ahead each time it finds a group of intervals having more than 5 expected prediction errors to determine whether or not more than 5 expected prediction errors remain; if not, then the remaining frequencies are associated with the preceding group and FØMRE is the resulting difference between the observed frequency and the expected frequency.

FØT - the total number of (observed) prediction errors which have contributed to the Chi-square statistic.

JJ - the interval index of the interval which was the last of a group of intervals containing more than 5 expected prediction errors.

KØUNT - a variable which counts the number of groups of intervals having more than 5 expected prediction errors.

PRØBN - the area under the normal frequency function from $-\infty$ to the upper bound of any of the various intervals into which the range of prediction errors is divided.

PRØBØ - the area under the normal frequency function from $-\infty$ to the upper bound of the last interval which was the last of a group of intervals containing more than 5 expected prediction errors.

REMAIN - the remaining number of expected prediction errors.

(5) Variables in Subroutine CMPR

AN - a floating point representation of the total number of data points.

ESQUØT - the error sum of squares.

FQUØT - the F value for regression in the analysis of variance tables.

IRCT - the number of independent variables in the present model, plus 1.

IW - an index for elements in arrays which elements are used to define the variable output formats.

K - the number of independent variables in the present model, plus 2.

LAST - a variable used in the computation of index values for arrays which are used to complete the definition of the variable formats for the printing of the regression equation.

LL - a variable used to control the printing of page headings.

NØMR - an integer representation of the error degrees of freedom.

ØMR - the error degrees of freedom.

R - a floating point representation of the number of independent variables in the model; the degrees of freedom for regression.

RSQUØT - the mean square for regression.

(6) Variables in Subroutine FIX

LIT - an array which contains a BCD representation of the first NNNSAV (see Section VIII.1.a) elements of the LØT array and BCD zeroes for the remaining elements.

(7) Variables in Subroutine GAUSS

AMAX - the maximum element, of those elements searched, in the matrix of the normal equations at each step of the inversion process.

ICØLUM - the column number of the maximum of those elements in unpivoted rows.

IGØ - a variable which indicates when no pivot element could be found at a step of the inversion process.

INDEX - an array containing the row and column numbers of those elements which are used as pivot elements.

IPIVØT - an array which indicates those rows of A which have served as pivot rows.

IRØW - the row number of the pivot element.

PIVØT - a variable set equal to the value of the pivot element.

SWAP - a temporary storage location used to interchange rows and columns.

T - a variable which is equal to the successive elements of the A matrix which are in the same column as the pivot element.

(8) Variables in Subroutine IDENTM

AIDENT - the identity matrix.

SUM - a variable which is used to compute the individual elements of the identity matrix.

(9) Variables in Subroutine IVØR

AMAX - the maximum ASSR value.

AMIN - the minimum ASSR value.

IGØ2 - a variable which indicates the case of a perfect fit.

IMAX - the LØT array index of that independent variable which is to be included in the model.

ISTART - a variable which is used to define the LØT array index of the leftmost independent variable of a group of independent variables.

LXMAX - the index of the maximum ASSR value in the AW array.

LXMIN - the index of the minimum ASSR value in the AW array.

KASSR - a counter of the ASSR values which are computed at each step.

KGØ - a variable which indicates that a non-valid ASSR value was computed by the CASSR subroutine.

KØUNT - a counter of the number of independent variables which have been "actively" ordered, i.e., ordered in a group of IV's as long as there is more than one IV left in the group.

LAT - an array which holds the LØT array indices of the independent variables for which ASSR values are computed.

NUM - a variable which is used to determine when to cease "actively" ordering independent variables in a specified group of independent variables, i.e., when there is only one independent variable left in the group.

TØLSS - a tolerance which is used to establish equality of ASSR values and hence the perfect fit.

(10) Variables in Subroutine MAXMIN

The variables used by this subroutine have been amply defined by any one of its calling subroutines, and, therefore, these variables will not be further defined here.

(11) Variables in Subroutine PREVAR

JJJ - an index which is used to delete independent variables from the X array.

TEMXX - equals $\sum_{v=1}^{N} \sum_{v'=1}^{N} c_{vv'}(x_{v(i)} - \bar{x}_v)(x_{v'(i)} - \bar{x}_{v'})$

which is used in the computation of prediction standard deviations.

(12) Variables in Subroutine PRINTM

AIDENT - the identity matrix.

(13) Variables in Subroutine RDISK

ISTART - a variable which is always 1 more than the number of records read from tape or disk logical unit 10.

IWHICH - the number (record number) of a data point which is to be used as a selected input design point in prediction standard deviation calculations.

NUMBER - the number of records which must be read in order to position the storage device so that the IWHICHth data point can be read with the next READ statement.

SKIP - a variable which is used to skip records.

(14) Variables in Subroutine RDIT

INDEX - the number of an independent variable which is to be used as a factor in a product term, plus 1.

J1 - the Y array index of the last variable on each card of input.

KK - the Y array index of a product term.

MZ - if a data point requires more than one card or record to contain the coordinates of the OCIV's, then MZ is used as a dummy variable in reading those cards or records after the first card or record.

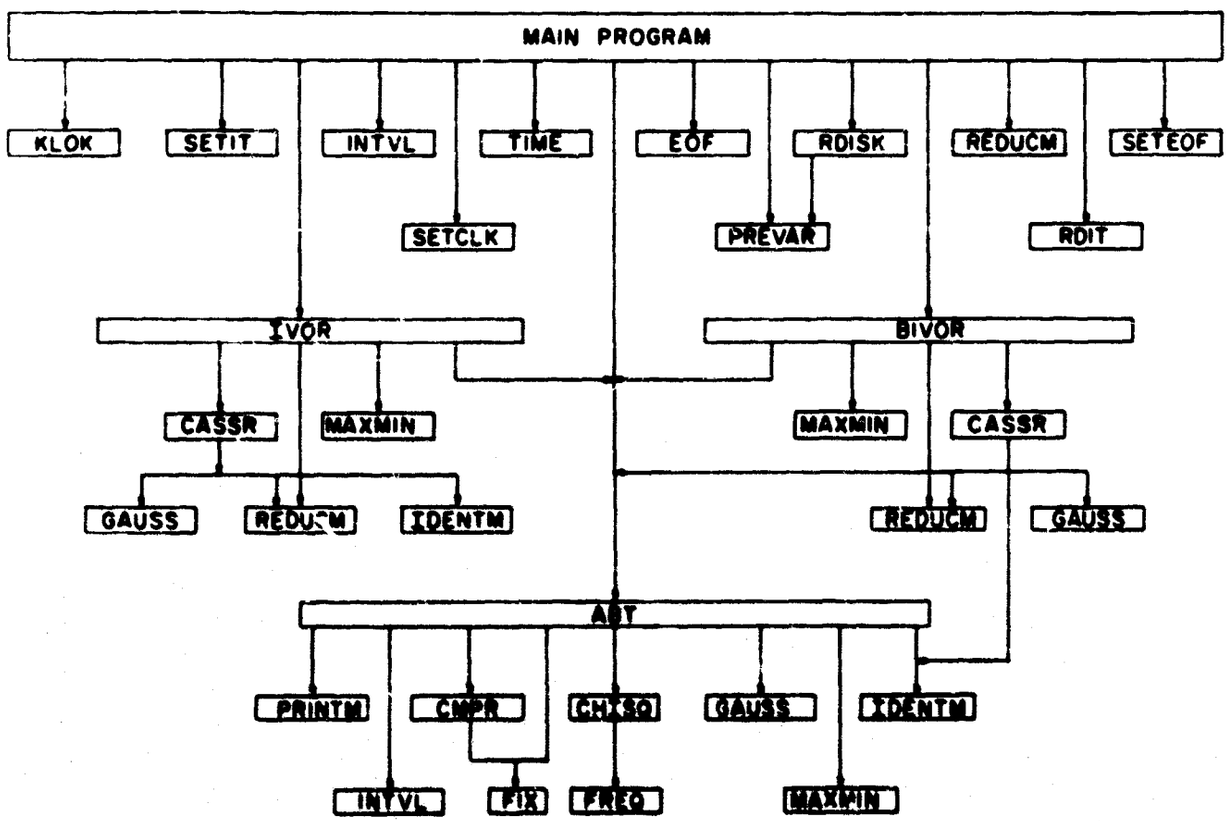
Y - an array which contains the coordinates of the dependent variable and those of the OCIV's as they are read and which later contains also the coordinates of the GCIV's.

(15) Variables in Subroutine REDUCM

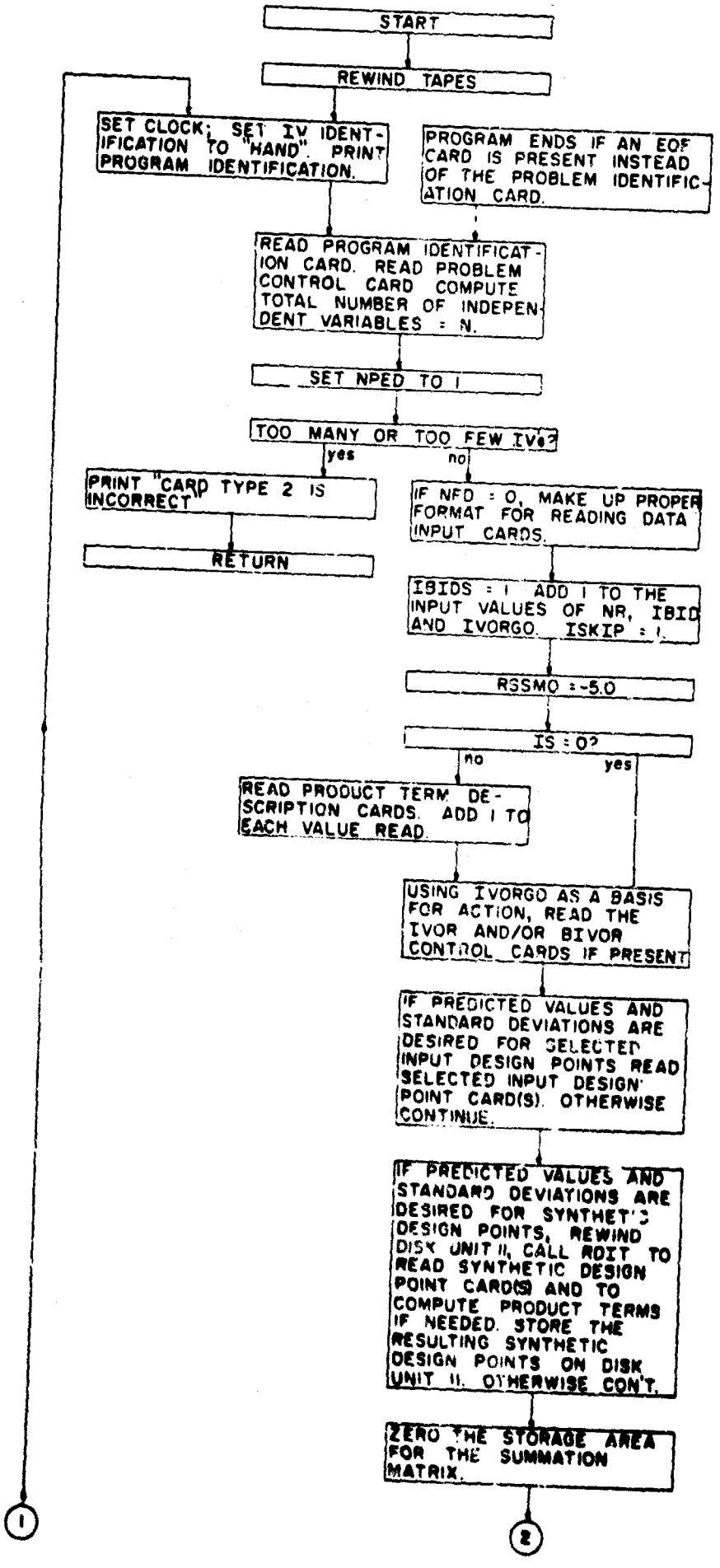
These variables are described in Section VIII.1.a and, therefore, will not be further described here.

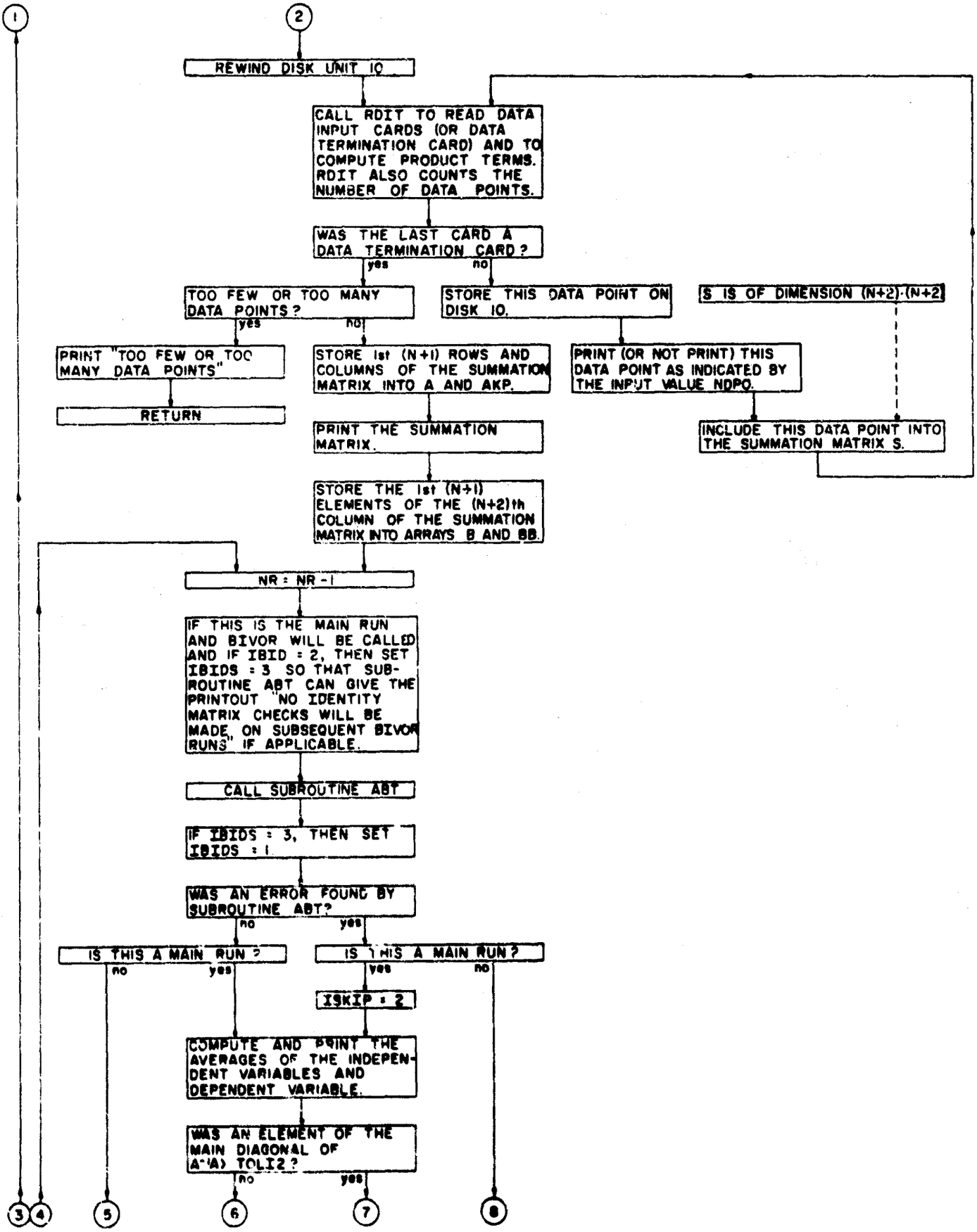
VIII.2 Flow Charts

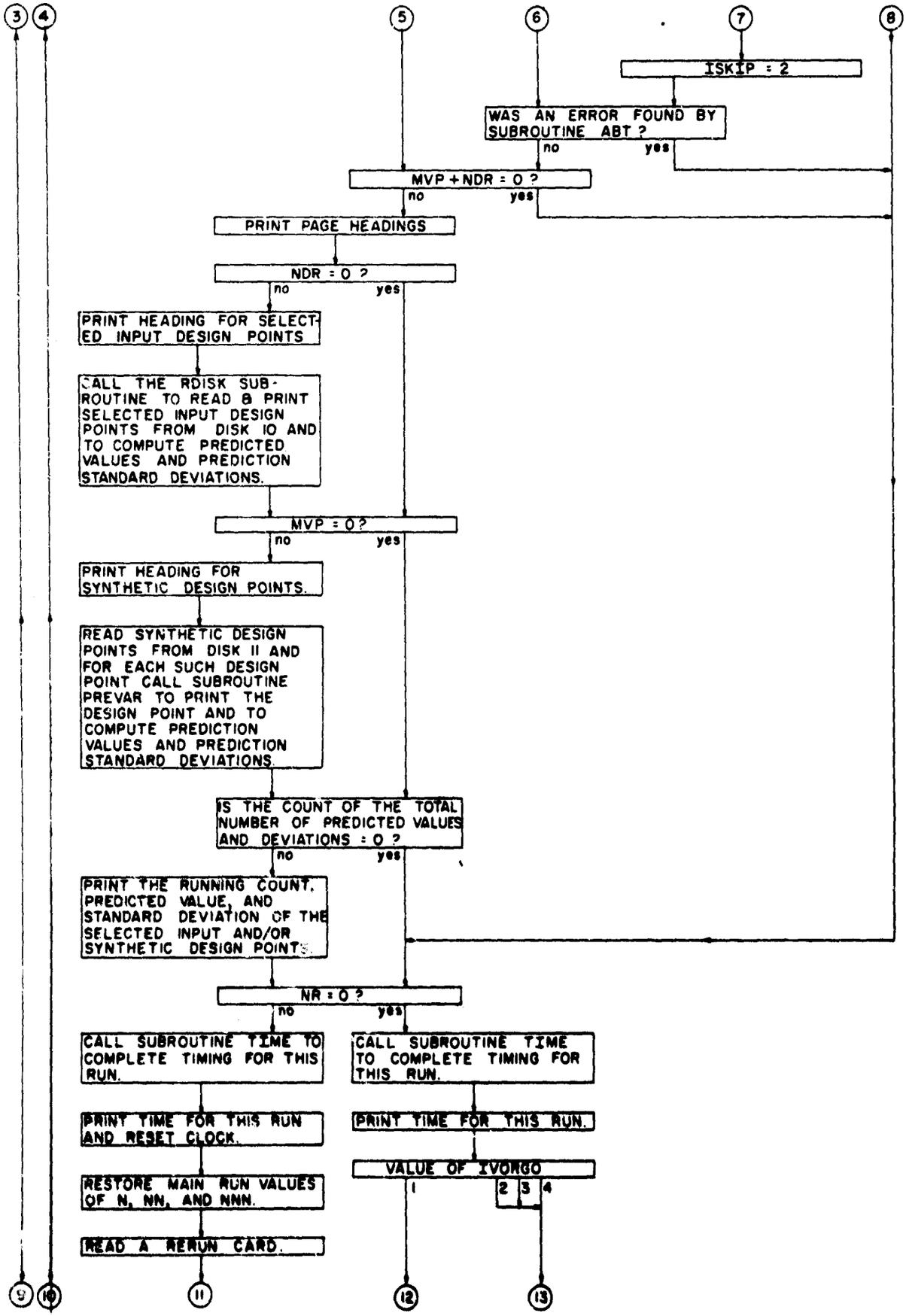
a. DA-MRCA SUBROUTINE FLOW CHART

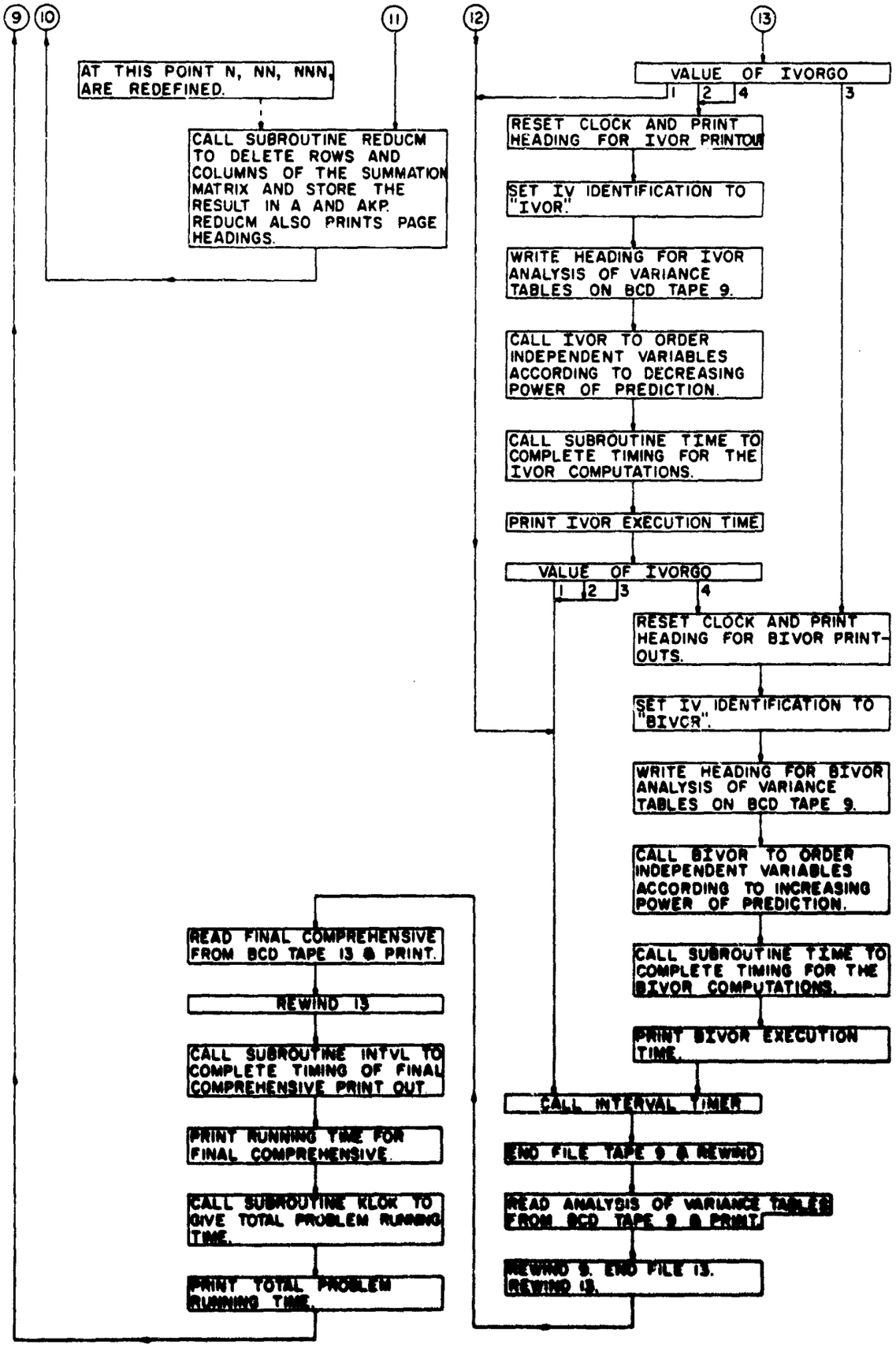


b. MAIN PROGRAM

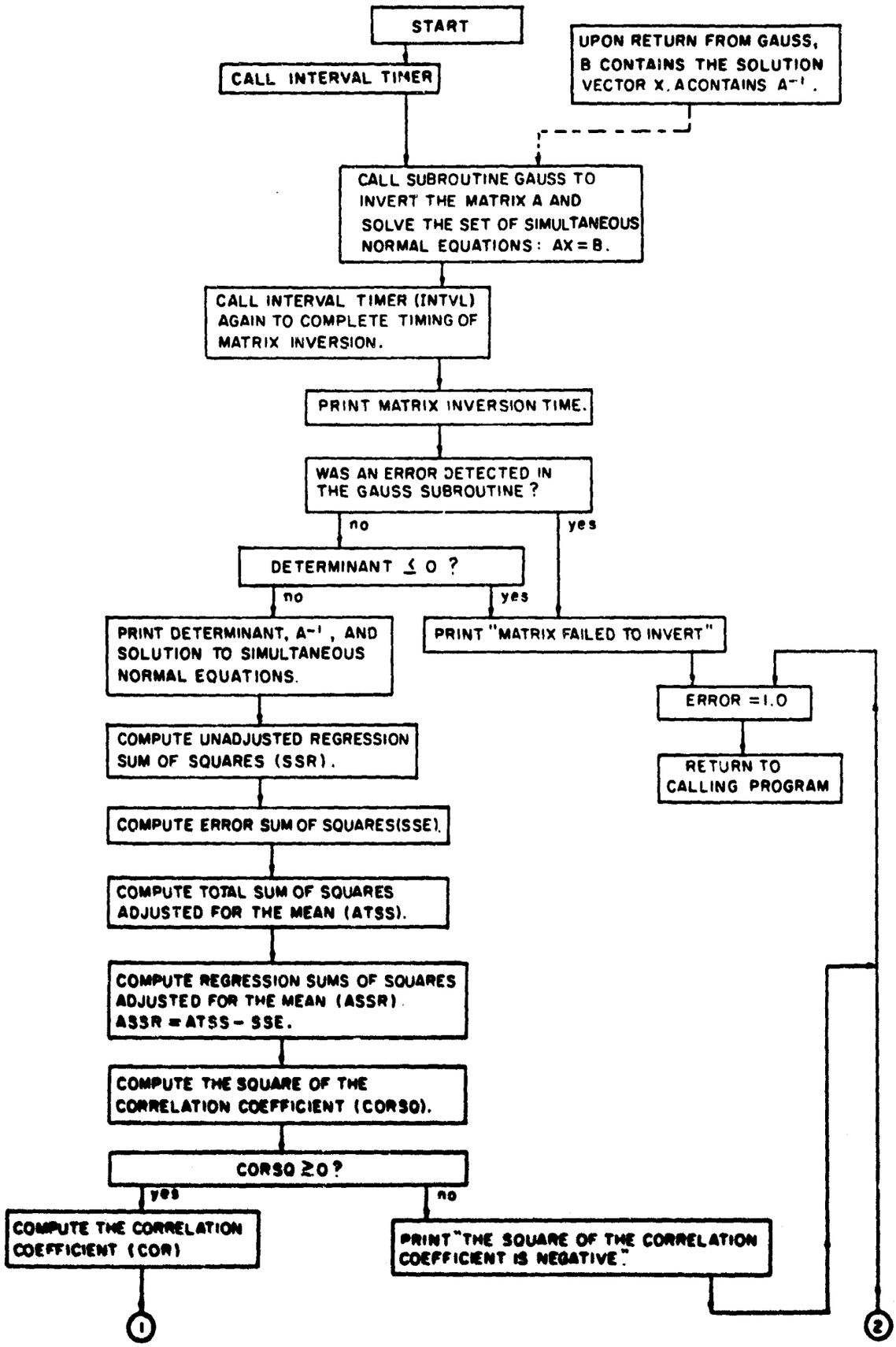


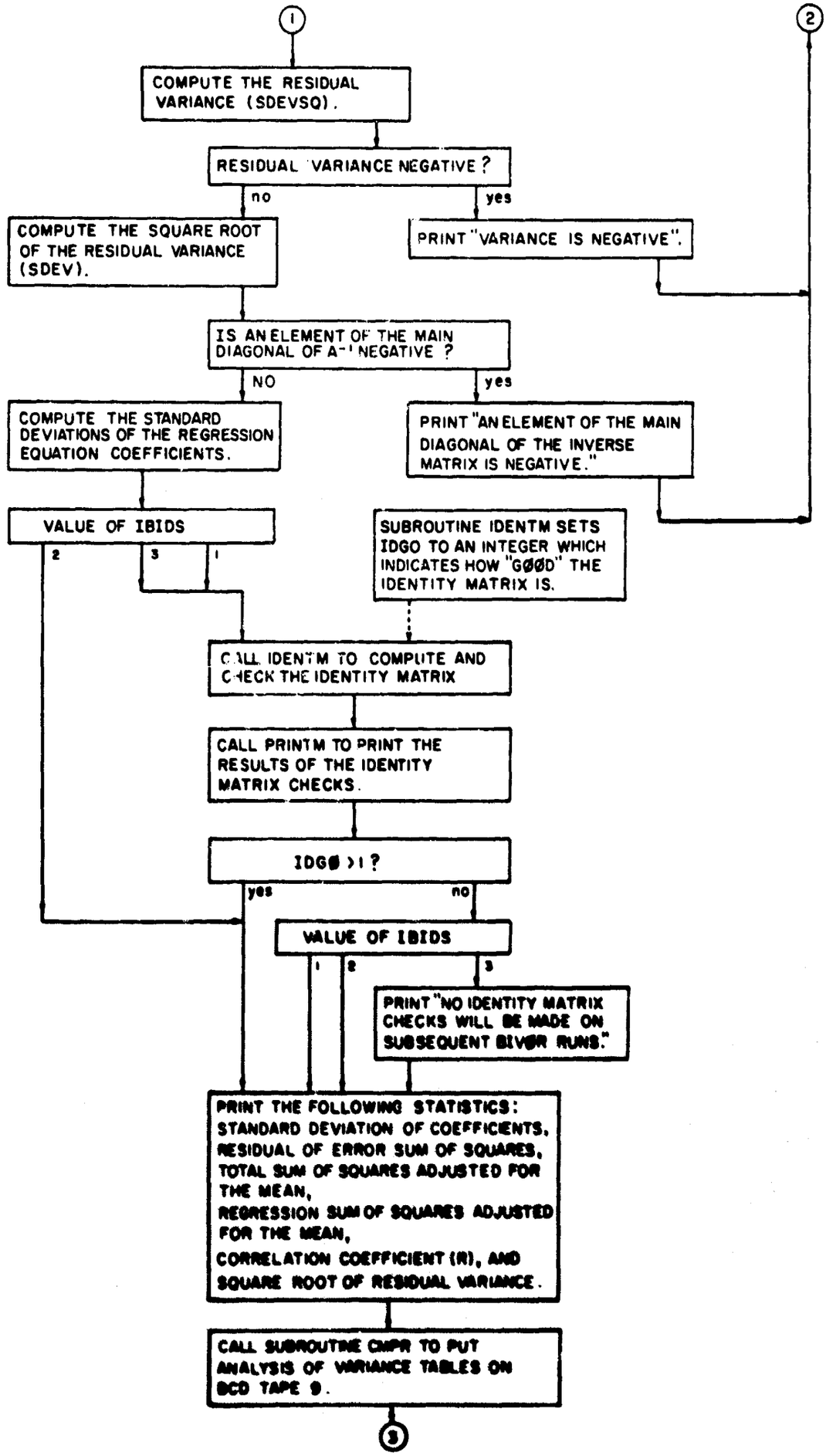

1
2
REWIND DISK UNIT IO
CALL RDIT TO READ DATA INPUT CARDS (OR DATA TERMINATION CARD) AND TO COMPUTE PRODUCT TERMS. RDIT ALSO COUNTS THE NUMBER OF DATA POINTS.
WAS THE LAST CARD A DATA TERMINATION CARD?
yes
no
TOO FEW OR TOO MANY DATA POINTS?
STORE THIS DATA POINT ON DISK IO.
S IS OF DIMENSION (N+2)·(N+2)
yes
no
PRINT "TOO FEW OR TOO MANY DATA POINTS"
STORE 1st (N+1) ROWS AND COLUMNS OF THE SUMMATION MATRIX INTO A AND AKP.
PRINT (OR NOT PRINT) THIS DATA POINT AS INDICATED BY THE INPUT VALUE NDPO.
RETURN
PRINT THE SUMMATION MATRIX.
INCLUDE THIS DATA POINT INTO THE SUMMATION MATRIX S.
STORE THE 1st (N+1) ELEMENTS OF THE (N+2)th COLUMN OF THE SUMMATION MATRIX INTO ARRAYS B AND BB.
NR = NR - 1
IF THIS IS THE MAIN RUN AND BIVOR WILL BE CALLED AND IF IBID = 2, THEN SET IBIDS = 3 SO THAT SUBROUTINE ABT CAN GIVE THE PRINTOUT "NO IDENTITY MATRIX CHECKS WILL BE MADE ON SUBSEQUENT BIVOR RUNS" IF APPLICABLE.
CALL SUBROUTINE ABT
IF IBIDS = 3, THEN SET IBIDS = 1
WAS AN ERROR FOUND BY SUBROUTINE ABT?
no
yes
IS THIS A MAIN RUN?
IS THIS A MAIN RUN?
no
yes
yes
no
ISKIP = 2
COMPUTE AND PRINT THE AVERAGES OF THE INDEPENDENT VARIABLES AND DEPENDENT VARIABLE.
WAS AN ELEMENT OF THE MAIN DIAGONAL OF A'A) TOLI2?
no
yes
3
4
5
6
7
8

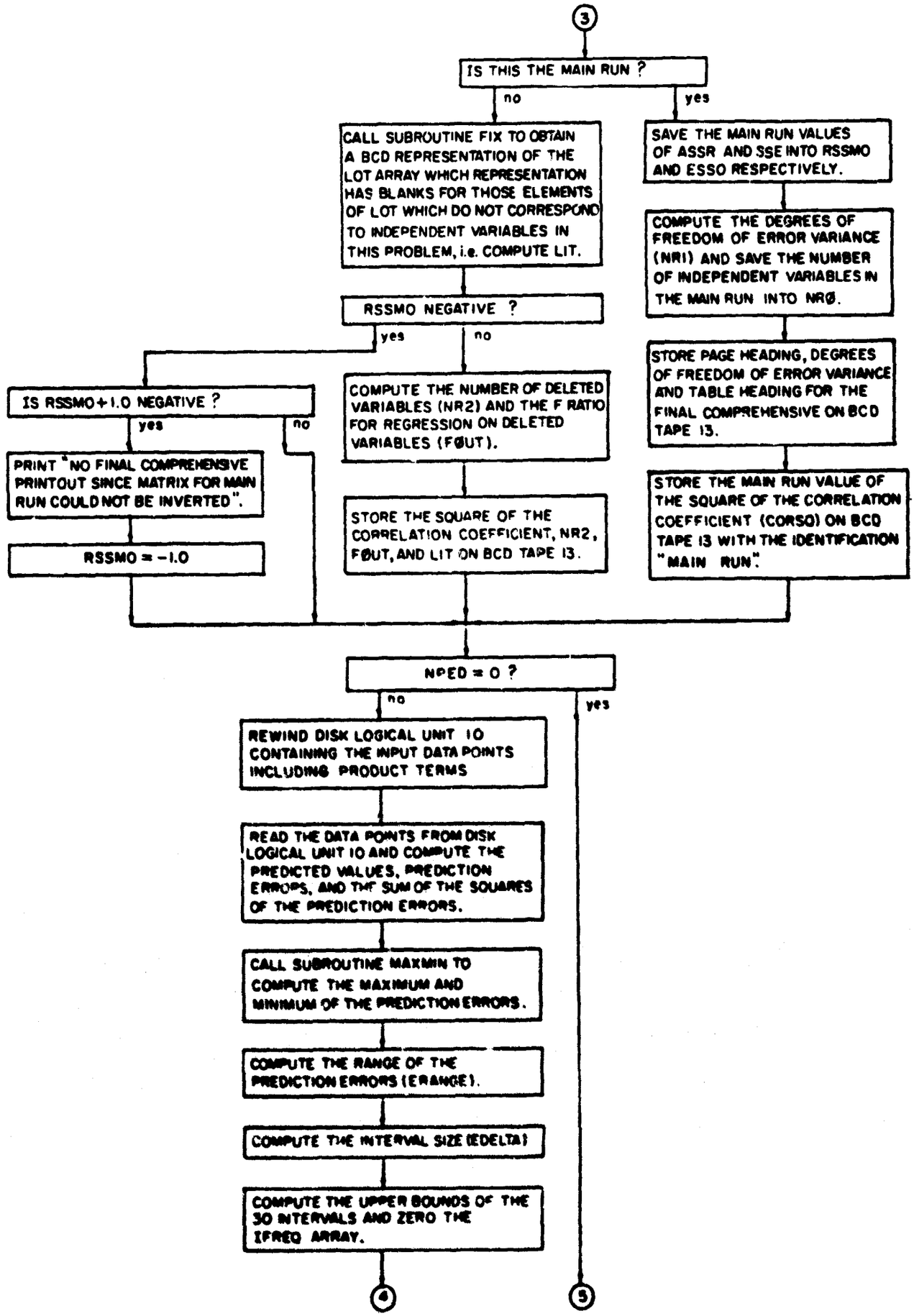

3
4
5
6
7
8
ISKIP = 2
WAS AN ERROR FOUND BY SUBROUTINE ABT ?
no
yes
MVP + NDR = 0 ?
no
yes
PRINT PAGE HEADINGS
NDR = 0 ?
no
yes
PRINT HEADING FOR SELECTED INPUT DESIGN POINTS
CALL THE RDISK SUBROUTINE TO READ & PRINT SELECTED INPUT DESIGN POINTS FROM DISK 10 AND TO COMPUTE PREDICTED VALUES AND PREDICTION STANDARD DEVIATIONS.
MVP = 0?
no
yes
PRINT HEADING FOR SYNTHETIC DESIGN POINTS.
READ SYNTHETIC DESIGN POINTS FROM DISK 11 AND FOR EACH SUCH DESIGN POINT CALL SUBROUTINE PREVAR TO PRINT THE DESIGN POINT AND TO COMPUTE PREDICTION VALUES AND PREDICTION STANDARD DEVIATIONS.
IS THE COUNT OF THE TOTAL NUMBER OF PREDICTED VALUES AND DEVIATIONS = 0 ?
no
yes
PRINT THE RUNNING COUNT, PREDICTED VALUE, AND STANDARD DEVIATION OF THE SELECTED INPUT AND/OR SYNTHETIC DESIGN POINTS.
NR = 0 ?
no
yes
CALL SUBROUTINE TIME TO COMPLETE TIMING FOR THIS RUN.
PRINT TIME FOR THIS RUN AND RESET CLOCK.
RESTORE MAIN RUN VALUES OF N, NN, AND NNN.
READ A RERUN CARD.
CALL SUBROUTINE TIME TO COMPLETE TIMING FOR THIS RUN.
PRINT TIME FOR THIS RUN.
VALUE OF IVORGO
1
2
3
4
9
10
11
12
13

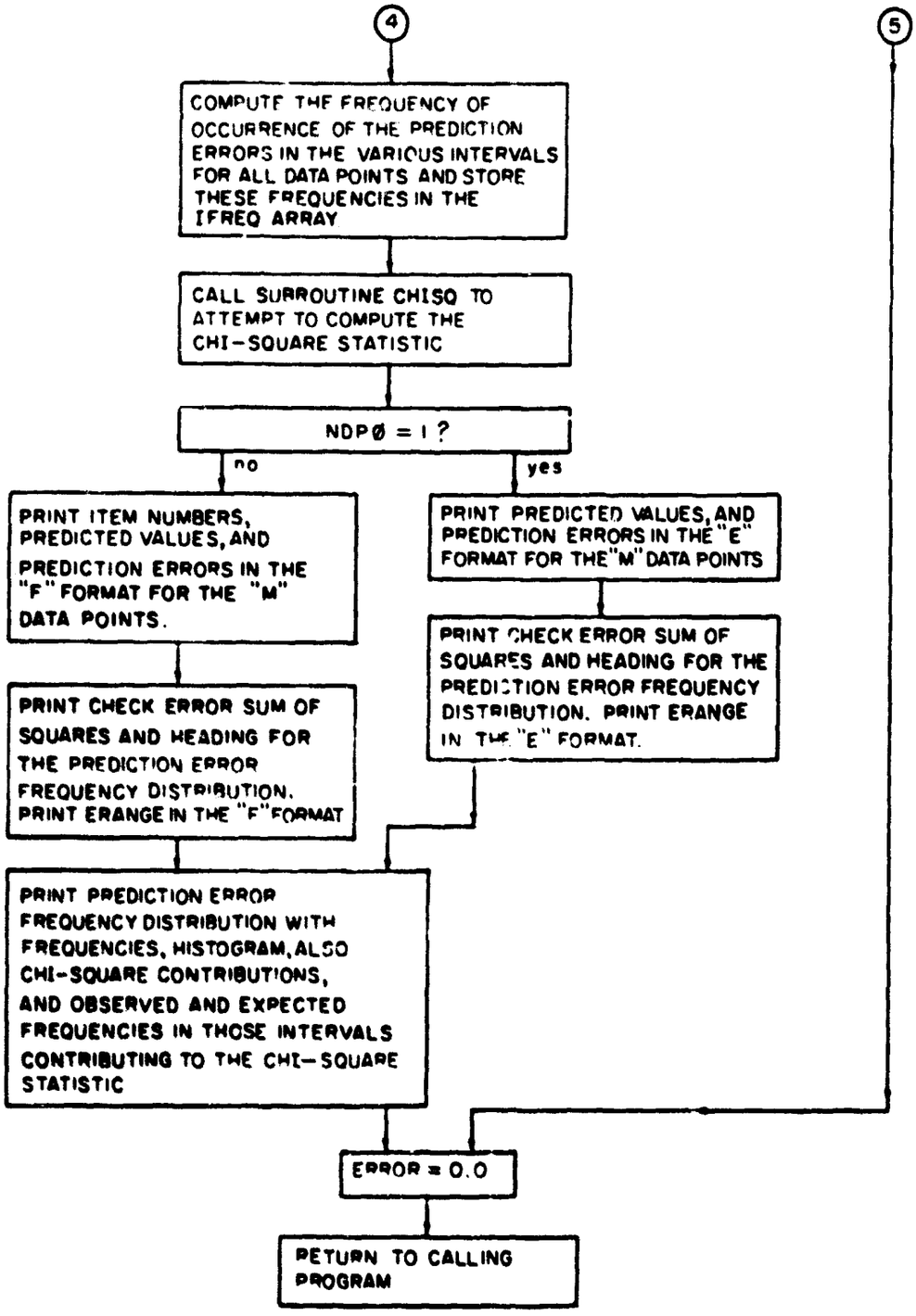

9
10
11
12
13
AT THIS POINT N, NN, NNN, ARE REDEFINED.
CALL SUBROUTINE REDUCM TO DELETE ROWS AND COLUMNS OF THE SUMMATION MATRIX AND STORE THE RESULT IN A AND AKP. REDUCM ALSO PRINTS PAGE HEADINGS.
VALUE OF IVORGO
1
2
4
3
RESET CLOCK AND PRINT HEADING FOR IVOR PRINTOUT
SET IV IDENTIFICATION TO "IVOR."
WRITE HEADING FOR IVOR ANALYSIS OF VARIANCE TABLES ON BCD TAPE 9.
CALL IVOR TO ORDER INDEPENDENT VARIABLES ACCORDING TO DECREASING POWER OF PREDICTION.
CALL SUBROUTINE TIME TO COMPLETE TIMING FOR THE IVOR COMPUTATIONS.
PRINT IVOR EXECUTION TIME.
VALUE OF IVORGO
1
2
3
4
RESET CLOCK AND PRINT HEADING FOR BIVOR PRINT-OUTS.
SET IV IDENTIFICATION TO "BIVOR".
WRITE HEADING FOR BIVOR ANALYSIS OF VARIANCE TABLES ON BCD TAPE 9.
CALL BIVOR TO ORDER INDEPENDENT VARIABLES ACCORDING TO INCREASING POWER OF PREDICTION.
CALL SUBROUTINE TIME TO COMPLETE TIMING FOR THE BIVOR COMPUTATIONS.
PRINT BIVOR EXECUTION TIME.
CALL INTERVAL TIMER
END FILE TAPE 9 & REWIND
READ ANALYSIS OF VARIANCE TABLES FROM BCD TAPE 9 & PRINT
REWIND 9. END FILE 13. REWIND 13.
READ FINAL COMPREHENSIVE FROM BCD TAPE 13 & PRINT.
REWIND 13
CALL SUBROUTINE INTVL TO COMPLETE TIMING OF FINAL COMPREHENSIVE PRINT OUT
PRINT RUNNING TIME FOR FINAL COMPREHENSIVE
CALL SUBROUTINE KLOK TO GIVE TOTAL PROBLEM RUNNING TIME.
PRINT TOTAL PROBLEM RUNNING TIME.

c. SUBROUTINE ABT

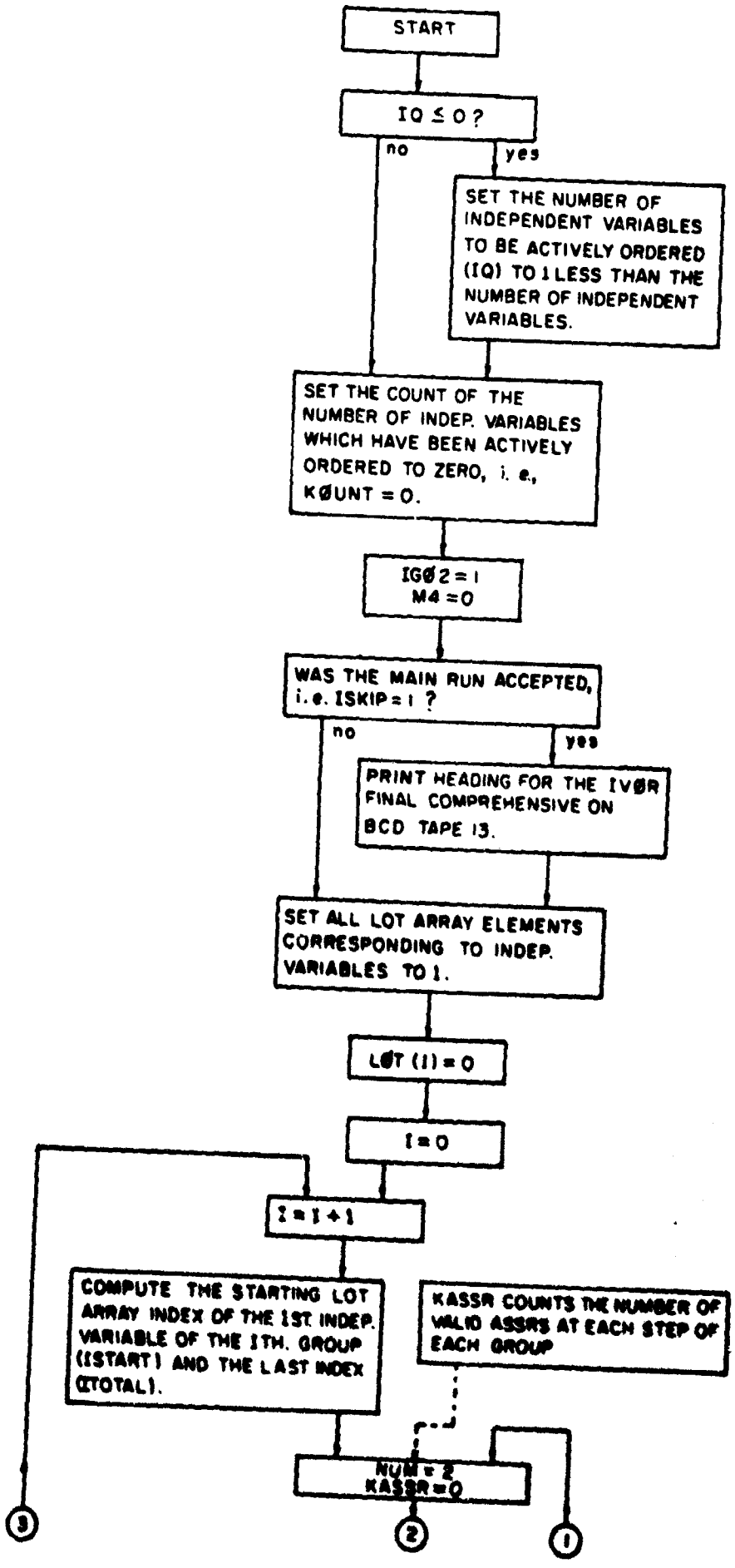


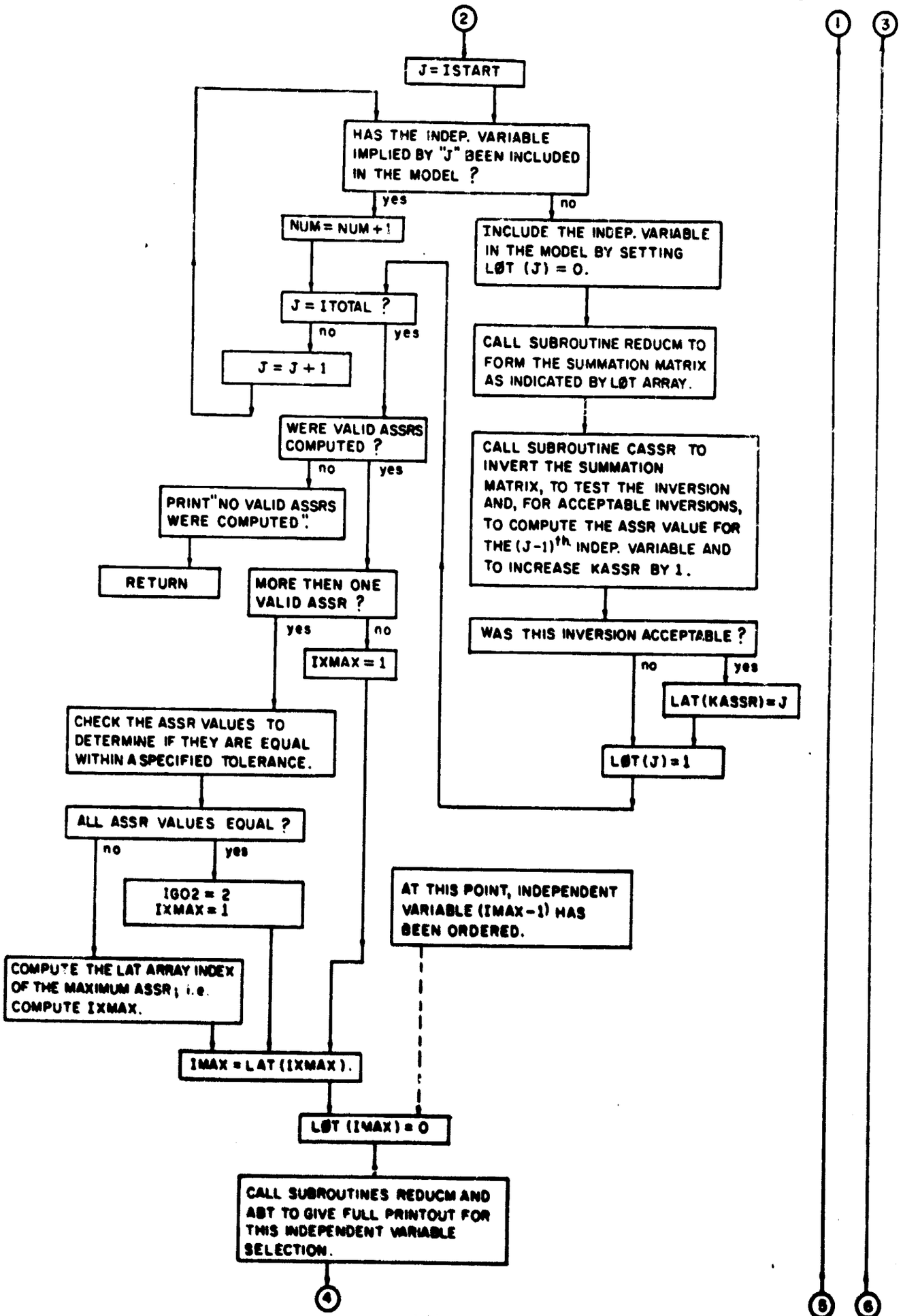

1
2
COMPUTE THE RESIDUAL VARIANCE (SDEVSQ).
RESIDUAL VARIANCE NEGATIVE ?
no
yes
COMPUTE THE SQUARE ROOT OF THE RESIDUAL VARIANCE (SDEV).
PRINT "VARIANCE IS NEGATIVE".
IS AN ELEMENT OF THE MAIN DIAGONAL OF A⁻¹ NEGATIVE ?
NO
yes
COMPUTE THE STANDARD DEVIATIONS OF THE REGRESSION EQUATION COEFFICIENTS.
PRINT "AN ELEMENT OF THE MAIN DIAGONAL OF THE INVERSE MATRIX IS NEGATIVE."
VALUE OF IBIDS
2
3
1
SUBROUTINE IDENTM SETS IDGO TO AN INTEGER WHICH INDICATES HOW "GØØD" THE IDENTITY MATRIX IS.
CALL IDENTM TO COMPUTE AND CHECK THE IDENTITY MATRIX
CALL PRINTM TO PRINT THE RESULTS OF THE IDENTITY MATRIX CHECKS.
IDGØ > 1 ?
yes
no
VALUE OF IBIDS
1
2
3
PRINT "NO IDENTITY MATRIX CHECKS WILL BE MADE ON SUBSEQUENT BIVØR RUNS."
PRINT THE FOLLOWING STATISTICS:
STANDARD DEVIATION OF COEFFICIENTS,
RESIDUAL OF ERROR SUM OF SQUARES,
TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN,
REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN,
CORRELATION COEFFICIENT (R), AND
SQUARE ROOT OF RESIDUAL VARIANCE.
CALL SUBROUTINE CMPR TO PUT ANALYSIS OF VARIANCE TABLES ON BCD TAPE 9.
3

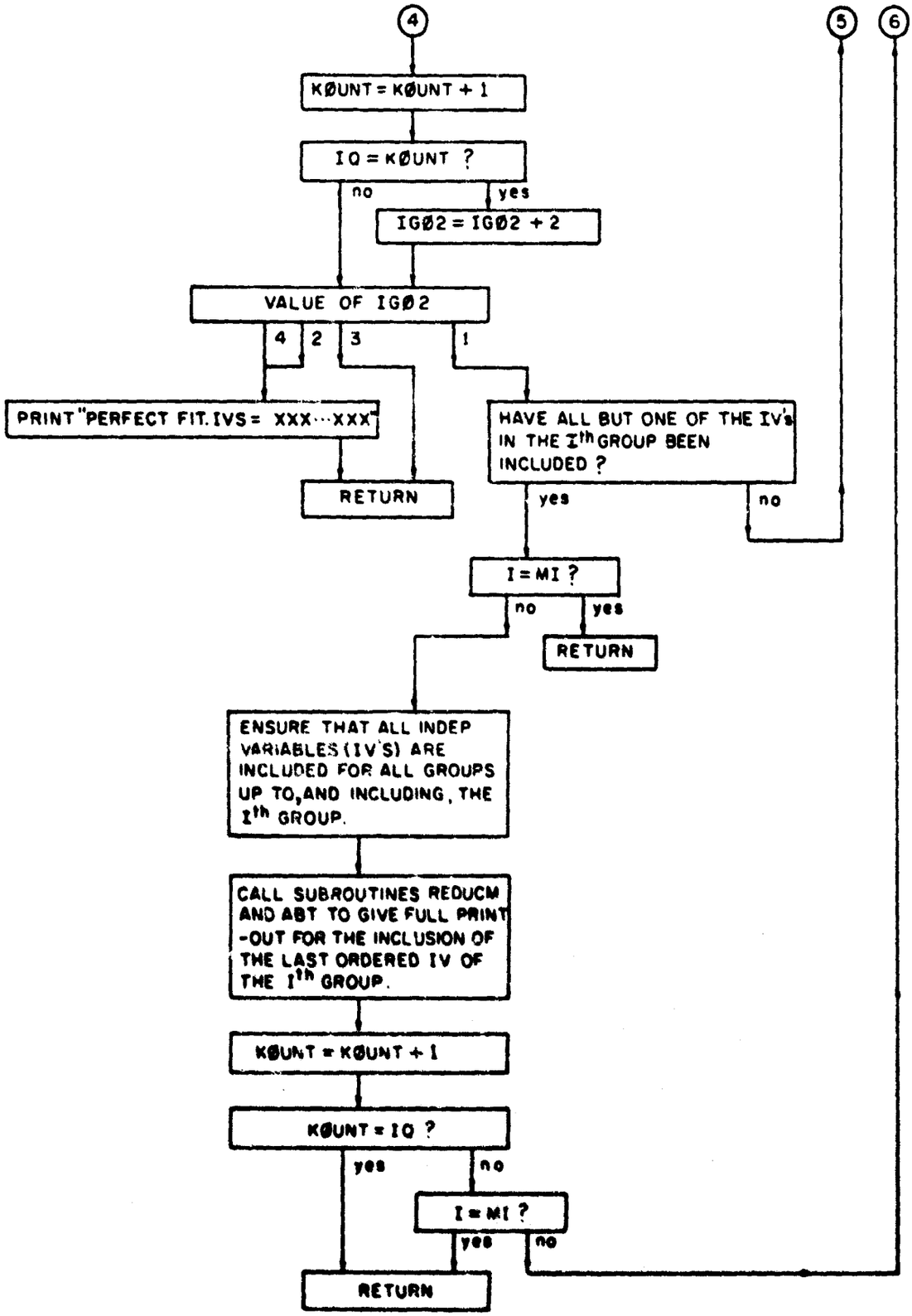

3
IS THIS THE MAIN RUN ?
no
yes
CALL SUBROUTINE FIX TO OBTAIN A BCD REPRESENTATION OF THE LOT ARRAY WHICH REPRESENTATION HAS BLANKS FOR THOSE ELEMENTS OF LOT WHICH DO NOT CORRESPOND TO INDEPENDENT VARIABLES IN THIS PROBLEM, i.e. COMPUTE LIT.
SAVE THE MAIN RUN VALUES OF ASSR AND SSE INTO RSSMO AND ESSO RESPECTIVELY.
COMPUTE THE DEGREES OF FREEDOM OF ERROR VARIANCE (NR1) AND SAVE THE NUMBER OF INDEPENDENT VARIABLES IN THE MAIN RUN INTO NRØ.
RSSMO NEGATIVE ?
yes
no
STORE PAGE HEADING, DEGREES OF FREEDOM OF ERROR VARIANCE AND TABLE HEADING FOR THE FINAL COMPREHENSIVE ON BCD TAPE 13.
IS RSSMO+1.0 NEGATIVE ?
yes
no
COMPUTE THE NUMBER OF DELETED VARIABLES (NR2) AND THE F RATIO FOR REGRESSION ON DELETED VARIABLES (FØUT).
PRINT "NO FINAL COMPREHENSIVE PRINTOUT SINCE MATRIX FOR MAIN RUN COULD NOT BE INVERTED".
STORE THE SQUARE OF THE CORRELATION COEFFICIENT, NR2, FØUT, AND LIT ON BCD TAPE 13.
STORE THE MAIN RUN VALUE OF THE SQUARE OF THE CORRELATION COEFFICIENT (CORSQ) ON BCD TAPE 13 WITH THE IDENTIFICATION "MAIN RUN".
RSSMO = -1.0
NPED = 0 ?
no
yes
REWIND DISK LOGICAL UNIT 10 CONTAINING THE INPUT DATA POINTS INCLUDING PRODUCT TERMS
READ THE DATA POINTS FROM DISK LOGICAL UNIT 10 AND COMPUTE THE PREDICTED VALUES, PREDICTION ERRORS, AND THE SUM OF THE SQUARES OF THE PREDICTION ERRORS.
CALL SUBROUTINE MAXMIN TO COMPUTE THE MAXIMUM AND MINIMUM OF THE PREDICTION ERRORS.
COMPUTE THE RANGE OF THE PREDICTION ERRORS (ERANGE).
COMPUTE THE INTERVAL SIZE (EDELTA)
COMPUTE THE UPPER BOUNDS OF THE 30 INTERVALS AND ZERO THE IFREQ ARRAY.
4
5


4
5
COMPUTE THE FREQUENCY OF OCCURRENCE OF THE PREDICTION ERRORS IN THE VARIOUS INTERVALS FOR ALL DATA POINTS AND STORE THESE FREQUENCIES IN THE IFREQ ARRAY
CALL SUBROUTINE CHISQ TO ATTEMPT TO COMPUTE THE CHI-SQUARE STATISTIC
NDPØ = 1 ?
no
yes
PRINT ITEM NUMBERS, PREDICTED VALUES, AND PREDICTION ERRORS IN THE "F" FORMAT FOR THE "M" DATA POINTS.
PRINT PREDICTED VALUES, AND PREDICTION ERRORS IN THE "E" FORMAT FOR THE "M" DATA POINTS
PRINT CHECK ERROR SUM OF SQUARES AND HEADING FOR THE PREDICTION ERROR FREQUENCY DISTRIBUTION. PRINT ERANGE IN THE "E" FORMAT.
PRINT CHECK ERROR SUM OF SQUARES AND HEADING FOR THE PREDICTION ERROR FREQUENCY DISTRIBUTION. PRINT ERANGE IN THE "F" FORMAT
PRINT PREDICTION ERROR FREQUENCY DISTRIBUTION WITH FREQUENCIES, HISTOGRAM, ALSO CHI-SQUARE CONTRIBUTIONS, AND OBSERVED AND EXPECTED FREQUENCIES IN THOSE INTERVALS CONTRIBUTING TO THE CHI-SQUARE STATISTIC
ERROR = 0.0
RETURN TO CALLING PROGRAM

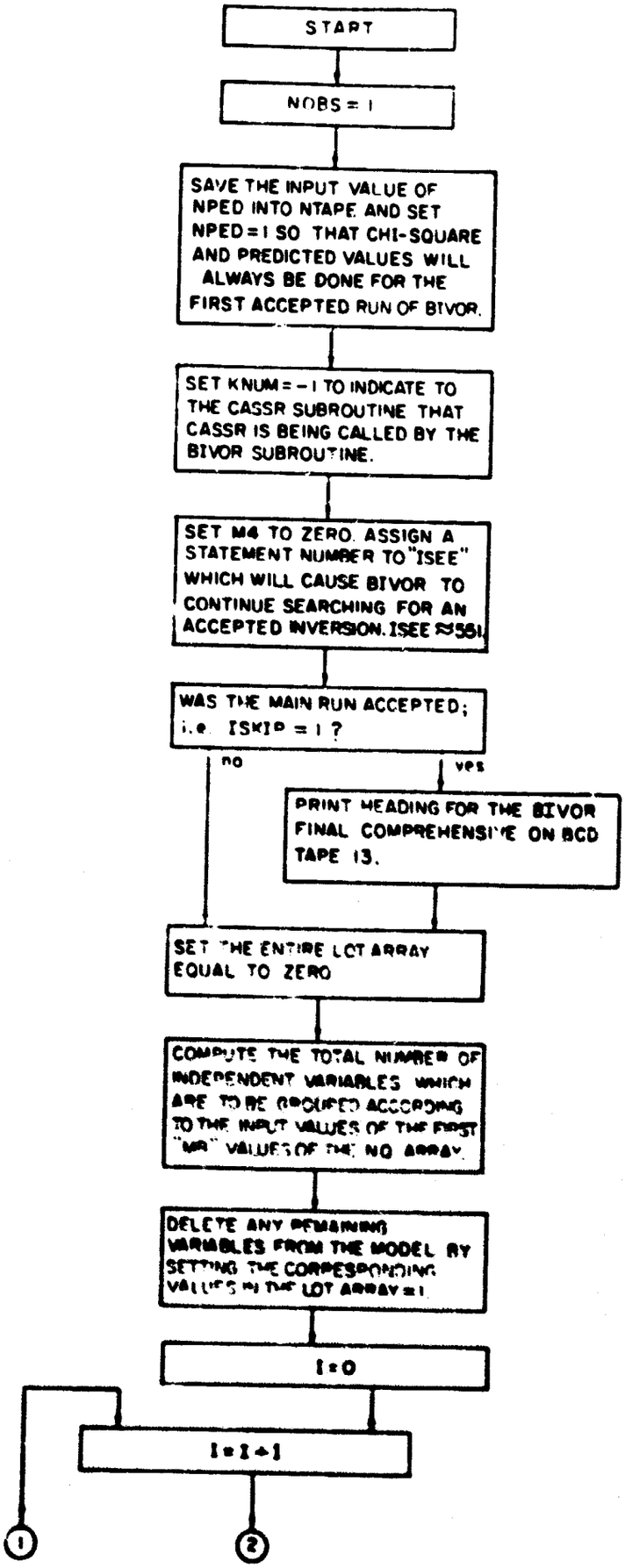
d. SUBROUTINE IVOR

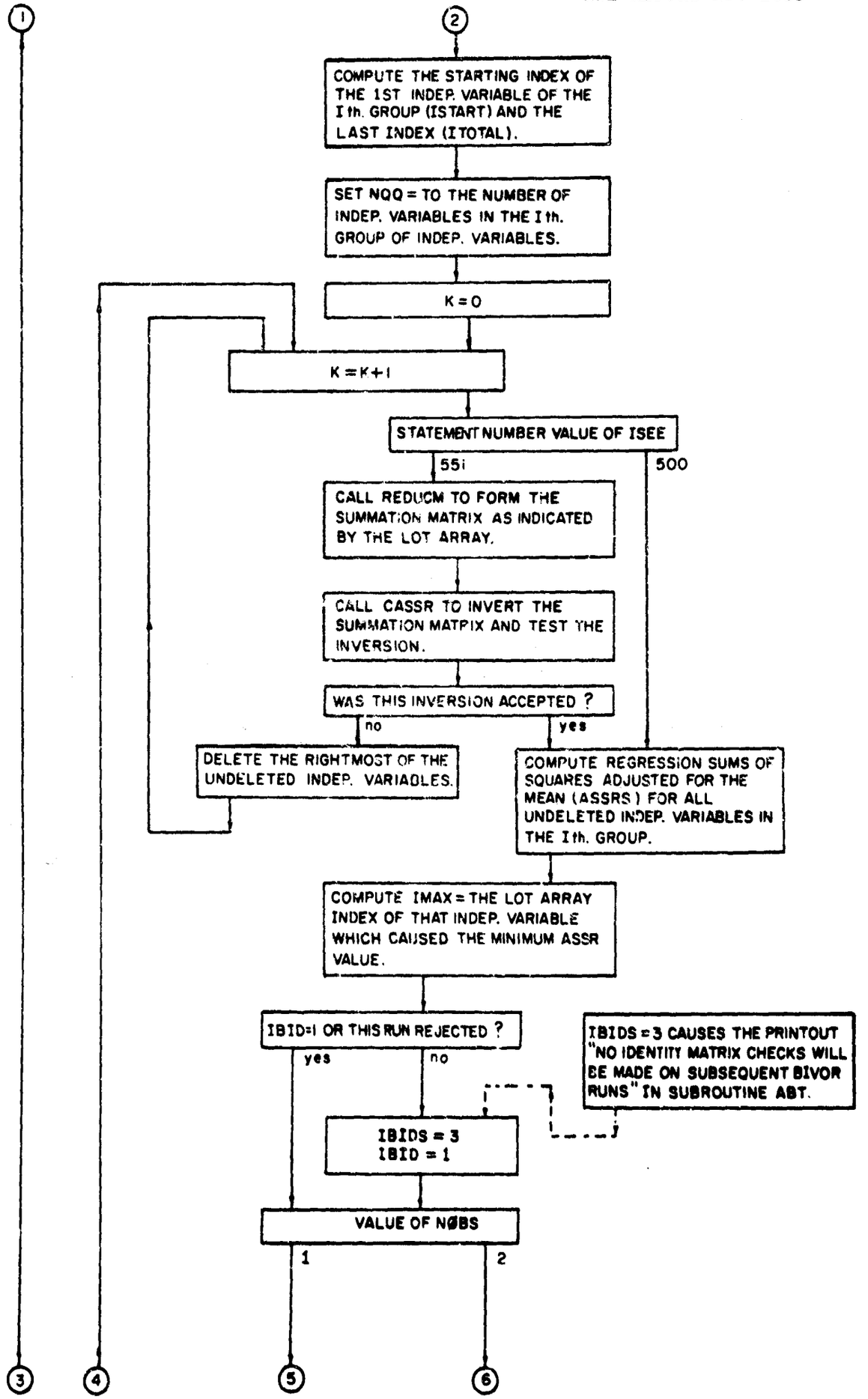


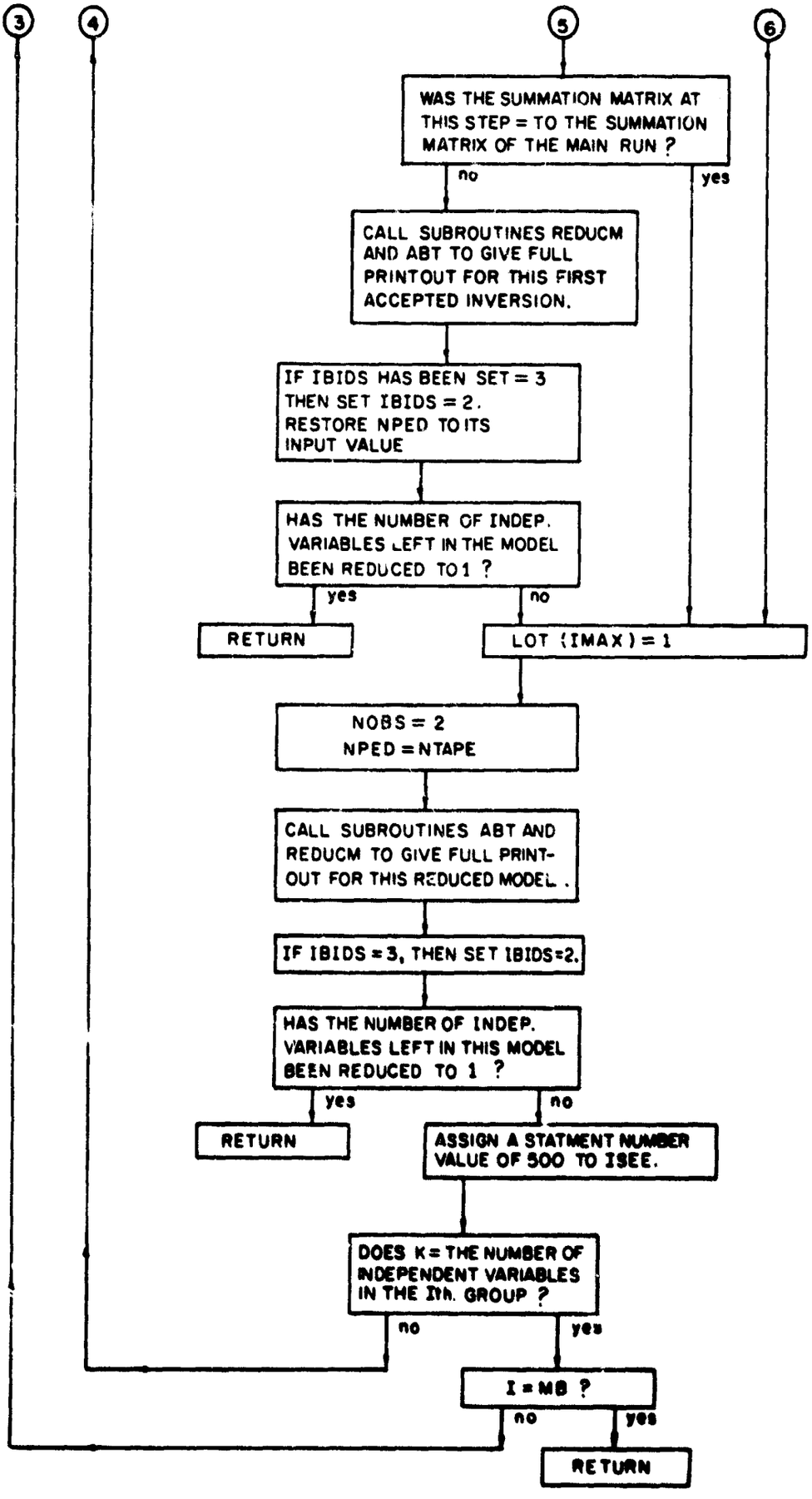

2
J = ISTART
HAS THE INDEP. VARIABLE IMPLIED BY "J" BEEN INCLUDED IN THE MODEL ?
yes
no
NUM = NUM + 1
INCLUDE THE INDEP. VARIABLE IN THE MODEL BY SETTING LØT (J) = 0.
J = ITOTAL ?
no
yes
J = J + 1
CALL SUBROUTINE REDUCM TO FORM THE SUMMATION MATRIX AS INDICATED BY LØT ARRAY.
WERE VALID ASSRS COMPUTED ?
no
yes
CALL SUBROUTINE CASSR TO INVERT THE SUMMATION MATRIX, TO TEST THE INVERSION AND, FOR ACCEPTABLE INVERSIONS, TO COMPUTE THE ASSR VALUE FOR THE (J-1)th INDEP. VARIABLE AND TO INCREASE KASSR BY 1.
PRINT "NO VALID ASSRS WERE COMPUTED".
RETURN
MORE THEN ONE VALID ASSR ?
yes
no
WAS THIS INVERSION ACCEPTABLE ?
no
yes
LAT(KASSR) = J
LØT(J) = 1
IXMAX = 1
CHECK THE ASSR VALUES TO DETERMINE IF THEY ARE EQUAL WITHIN A SPECIFIED TOLERANCE.
ALL ASSR VALUES EQUAL ?
no
yes
IGO2 = 2
IXMAX = 1
AT THIS POINT, INDEPENDENT VARIABLE (IMAX-1) HAS BEEN ORDERED.
COMPUTE THE LAT ARRAY INDEX OF THE MAXIMUM ASSR; i.e. COMPUTE IXMAX.
IMAX = LAT(IXMAX).
LØT (IMAX) = 0
CALL SUBROUTINES REDUCM AND ABT TO GIVE FULL PRINTOUT FOR THIS INDEPENDENT VARIABLE SELECTION.
4
1
3
5
6


4
5
6
KØUNT = KØUNT + 1
IQ = KØUNT ?
no
yes
IGØ2 = IGØ2 + 2
VALUE OF IGØ2
4
2
3
1
PRINT "PERFECT FIT. IVS = XXX···XXX"
RETURN
HAVE ALL BUT ONE OF THE IV's IN THE I^th GROUP BEEN INCLUDED ?
yes
no
I = MI ?
no
yes
RETURN
ENSURE THAT ALL INDEP VARIABLES (IV'S) ARE INCLUDED FOR ALL GROUPS UP TO, AND INCLUDING, THE I^th GROUP.
CALL SUBROUTINES REDUCM AND ABT TO GIVE FULL PRINT-OUT FOR THE INCLUSION OF THE LAST ORDERED IV OF THE I^th GROUP.
KØUNT = KØUNT + 1
KØUNT = IQ ?
yes
no
I = MI ?
yes
no
RETURN

e. SUBROUTINE BIVØR




1
2
COMPUTE THE STARTING INDEX OF THE 1ST INDEP. VARIABLE OF THE I th. GROUP (ISTART) AND THE LAST INDEX (ITOTAL).
SET NQQ = TO THE NUMBER OF INDEP. VARIABLES IN THE I th. GROUP OF INDEP. VARIABLES.
K = 0
K = K+1
STATEMENT NUMBER VALUE OF ISEE
55i
500
CALL REDUCM TO FORM THE SUMMATION MATRIX AS INDICATED BY THE LOT ARRAY.
CALL CASSR TO INVERT THE SUMMATION MATRIX AND TEST THE INVERSION.
WAS THIS INVERSION ACCEPTED ?
no
yes
DELETE THE RIGHTMOST OF THE UNDELETED INDEP. VARIABLES.
COMPUTE REGRESSION SUMS OF SQUARES ADJUSTED FOR THE MEAN (ASSRS) FOR ALL UNDELETED INDEP. VARIABLES IN THE I th. GROUP.
COMPUTE IMAX = THE LOT ARRAY INDEX OF THAT INDEP. VARIABLE WHICH CAUSED THE MINIMUM ASSR VALUE.
IBID=1 OR THIS RUN REJECTED ?
yes
no
IBIDS = 3 CAUSES THE PRINTOUT "NO IDENTITY MATRIX CHECKS WILL BE MADE ON SUBSEQUENT BIVOR RUNS" IN SUBROUTINE ABT.
IBIDS = 3
IBID = 1
VALUE OF NØBS
1
2
3
4
5
6


3
4
5
6
WAS THE SUMMATION MATRIX AT THIS STEP = TO THE SUMMATION MATRIX OF THE MAIN RUN ?
no
yes
CALL SUBROUTINES REDUCM AND ABT TO GIVE FULL PRINTOUT FOR THIS FIRST ACCEPTED INVERSION.
IF IBIDS HAS BEEN SET = 3 THEN SET IBIDS = 2. RESTORE NPED TO ITS INPUT VALUE
HAS THE NUMBER OF INDEP. VARIABLES LEFT IN THE MODEL BEEN REDUCED TO 1 ?
yes
no
RETURN
LOT (IMAX) = 1
NOBS = 2
NPED = NTAPE
CALL SUBROUTINES ABT AND REDUCM TO GIVE FULL PRINT-OUT FOR THIS REDUCED MODEL .
IF IBIDS = 3, THEN SET IBIDS=2.
HAS THE NUMBER OF INDEP. VARIABLES LEFT IN THIS MODEL BEEN REDUCED TO 1 ?
yes
no
RETURN
ASSIGN A STATMENT NUMBER VALUE OF 500 TO ISEE.
DOES K = THE NUMBER OF INDEPENDENT VARIABLES IN THE Ith. GROUP ?
no
yes
I = MB ?
no
yes
RETURN

VIII.3 Programming and Conversion Notes

a. Language - DA-MRCA is coded for the IBM 7030 computer (STRETCH) entirely in FORTRAN IV. FORMAT and DATA statements assume eight characters per word.

b. INPUT-OUTPUT Requirements - Three BCD tapes are required in addition to the system printer output tape. These BCD tapes have logical unit numbers of 5, 9, and 13, where 5 is the number for the tape unit containing the coordinates of the OCIV's and of the dependent variable when this data is on a separate tape; 9 is the number for the tape unit containing the analysis of variance tables which are computed in the program; and 13 is the number for the tape unit which contains the final comprehensive analysis table.

Two disk (or binary tape) logical units are required. Disk logical unit 10 is used to store the coordinates of the data points, and disk logical unit 11 is used to store the coordinates of the OCIV's for the synthetic design points.

The input-output requirements are described for the STRETCH in the IØD subprogram. The program listing contains a listing of this subprogram.

c. Storage Requirements - CØMMØN storage requires 25461 locations. The subprograms, excluding library functions and subroutines, require 4511 locations on the STRETCH but may require more or less on other machines.

d. Library Subroutines and Built-in Functions -

ABS - the absolute value function.

EØF - returns a value of .TRUE. if an end of file has been read, .FALSE. otherwise.

FLØAT - converts an integer to a floating point number.

FREQ(T) - the normal distribution function which gives

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{T} \exp\left[-y^2/2\right] dy.$$

INTVL - measures the interval, in seconds, between the current entry into INTVL and the exit from the immediately preceding TIME, INTVL, or SETIT subroutine.

KLØK - the time in hours/minutes/seconds since the last CALL SETCLK.

MINØ - chooses the smallest of its fixed point arguments.

SETCLK - used at the beginning of a portion of a program to be timed by the KLØK subroutine.

SETEØF - this function is necessary in order to use EØF; it causes EØF to be set to .TRUE. and termination of execution of the READ statement when an end of file has been reached.

SETIT - see INTVL and TIME.

TIME - measures the usable elapsed time, in seconds, between the exit from SETIT and the current entry into TIME.

VIII.4 Program Listing

```
T         SUBTYPE,FIOD
B        2IOD,$READER
B        3IOD,$PRINTER
B    5D   IOD,TAPE,,,,EVEN,,$AVE
B         REEL,PUL
B    9D   IOD,TAPE,,,,ECC,,
B         REEL,NULS9
B       10IOD,DISK,,,704
B       11IOD,DISK,,,100
B   13D   IOD,TAPE,,,,ECC,,
B         REEL,NULS13
          END
T         SUBTYPE,FORTRAN,LMAP,PBIN                                          MRCA0000
C    DA-MRCA    MULTIPLE REGRESSION COMPREHENSIVE ANALYSIS                   MRCA0010
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)               MRCA0020
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)    MRCA0030
      COMMON NNL,DETERM,NOBS,TOL11,TOL12 ,ERROR,NPED,ITOTAL,N,NDPO,ICASE    MRCA0040
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ        MRCA0050
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGL3(10)            MRCA0060
      COMMON IN(49,10),IR,IS,M1,JLIM,NN,M, TAPE                             MRCA0070
      COMMON SELECT,IBID,IBIDS                                              MRCA0080
C                                                                           MRCA0090
      DIMENSION EYY(7000),LET(15),IKEEPR(999),FORM(5)                       MRCA0100
C                                                                           MRCA0110
      EQUIVALENCE(B(1602),IKEEPR(1)),(YSDEV,EYY),(ICASE,IRUN)               MRCA0120
      EQUIVALENCE(LOT(1),LET(1)),(FORM(1),FIRM(2))                          MRCA0130
      EQUIVALENCE(NNXA,NEN,LIM)                                             MRCA0140
C                                                                           MRCA0150
      INTEGER TAPE                                                          MRCA0160
C                                                                           MRCA0170
      LOGICAL EOF                                                           MRCA0180
C                                                                           MRCA0190
      DATA LIMOB(7000),FIRMF(8H(I2,     ),SEVEN(8H7F10.4) ),CPAREN(1H))     MRCA0200
      DATA HANDS(8H(HAND)   ),IVORS(8H(IVOR)   ),BIVORS(8H(BIVOR) )         MRCA0210
C                                                                           MRCA0220
      CALL SETEOF                                                           MRCA0230
      REWIND 5                                                              MRCA0240
      REWIND 9                                                              MRCA0250
      REWIND 13                                                             MRCA0260
      FIRM(1)=FIRMF                                                         MRCA0270
 5401 CALL SETCLK                                                           MRCA0280
      CALL SETIT                                                            MRCA0290
      SELECT=HANDS                                                          MRCA0300
      M4=0                                                                  MRCA0310
      PRINT 2064                                                            MRCA0320
 2064 FORMAT(49H2DA-MRCA ... OUTPUT FROM PROGRAM VERSION  2/ 1/66)          MRCA0330
      READ 595,PGLB                                                         MRCA0340
  595 FORMAT(10A8)                                                          MRCA0350
      PRINT 594,PGLB                                                        MRCA0360
      READ 579,IR,IS,NR,NVP,NDR,NVPL,NPE,NDPO,TAPE,IVORGO,NFD,IBID,TOL11    MRCA0370
     1,TOL12,FORM                                                           MRCA0380
  579 FORMAT(2I2,3I3,5I1,I2,I1,1X,2E9.3,5A8)                                MRCA0390
      PRINT 975                                                             MRCA0400
  975 FORMAT(118H0IR IS  NR NVP NDR NVPL NPE NDPO TAPE IVORGO NFD IBID      MRCA0410
     1  TOL11      TOL12    FORM - INPUT DATA DESCRIPTION -CARD TYPE 2)     MRCA0420
      PRINT 976,IR,IS,NR,NVP,NDR,NVPL,NPE,NDPO,TAPE,IVORGO,NFD,IBID,TOL1    MRCA0430
     11,TOL12,FORM                                                          MRCA0440
  976 FORMAT(1H0,I2,1X,I2,3(1X,I3),2(3X,I1),2(4X,I1),5X,I1,4X,I2,3X,I1,1    MRCA0450
     1X,2(1X,E9.3),2X,5A8)                                                  MRCA0460
      N=IR+IS                                                               MRCA0470
```

```
      NPED=1                                                                MRCA0480
      NNN=N+1                                                               MRCA0490
      ITOTAL=NNN                                                            MRCA0500
      IF((N-1)*(N-50))1107,1107,107                                         MRCA0510
 1107 NSAV=N                                                                MRCA0520
      IF(NFD)2999,2999,3000                                                 MRCA0530
 2999 FIRM(2)=SEVEN                                                         MRCA0540
      KNUM=7                                                                MRCA0550
      KMUM=6                                                                MRCA0560
      GO TO 3001                                                            MRCA0570
 3000 FIRM(7)=CPAREN                                                        MRCA0580
      KNUM=NFD                                                              MRCA0590
      KMUM=NFD-1                                                            MRCA0600
 3001 ICASE=0                                                               MRCA0610
      NR=NR+1                                                               MRCA0620
      INDX=IR+1                                                             MRCA0630
      IBIDS=1                                                               MRCA0640
      IBID=IBID+1                                                           MRCA0650
      IVORGO=IVORGO+1                                                       MRCA0660
      IF(TAPE)8,10,8                                                        MRCA0670
   10 IAPE=2                                                                MRCA0680
      GO TO 11                                                              MRCA0690
    8 IAPE=5                                                                MRCA0700
   11 ISKIP=1                                                               MRCA0710
      TAPE=2                                                                MRCA0720
      RSSMO=-5.0                                                            MRCA0730
      NNNSAV=0                                                              MRCA0740
      IF(IS)13,13,12                                                        MRCA0750
   13 JLIM=IR+1                                                             MRCA0760
      LIM=JLIM                                                              MRCA0770
      NN=LIM+1                                                              MRCA0780
      GO TO 14                                                              MRCA0790
   12 READ 92,(((IN(K,L),L=1,10),K=1,IS)                                    MRCA0800
   92 FORMAT(40I2)                                                          MRCA0810
      PRINT 29,(((IN(K,L),L=1,10),K=1,IS)                                   MRCA0820
   29 FORMAT(35HOPRODUCT TERM DESCRIPTIONS -CARD TYPE 3/1H /(1X,10I3,1H/MRCA0830
     1(10I3,1H/,10I3,1H/,10I3,1H/))                                         MRCA0840
      JLIM=IR+1                                                             MRCA0850
      LIM=JLIM+IS                                                           MRCA0860
      NN=LIM+1                                                              MRCA0870
      DO 20 K=1,IS                                                          MRCA0880
      DO 20 L=1,10                                                          MRCA0890
   20 IN(K,L)=IN(K,L)+1                                                     MRCA0900
   14 GO TO(21,22,23,22),IVORGO                                             MRCA0910
C READ IVOR GROUPING VALUES                                                 MRCA0920
   22 READ 100,IQ,MI,(NJ(I),I=1,MI)                                         MRCA0930
  100 FORMAT(12,26I3)                                                       MRCA0940
      PRINT 1,IQ,MI,(NJ(I),I=1,MI)                                          MRCA0950
    1 FORMAT(41HOIQ  MI NJ(I),I=1,2,....,MI   -CARD TYPE 4/1X,I2,2X,I2,2XMRCA0960
     1,25I3)                                                                MRCA0970
      GO TO(221,21,221,23),IVORGO                                           MRCA0980
C READ BIVOR GROUPING VALUES                                                MRCA0990
   23 READ 100,MB,(LOT(I),I=1,MB)                                           MRCA1000
      DO 99 I=1,MB                                                          MRCA1010
      MMM=MB-I+1                                                            MRCA1020
   99 NQ(I)=LOT(MMM)                                                        MRCA1030
      PRINT 101,MB,(LOT(I),I=1,MB)                                          MRCA1040
  101 FORMAT(41HOMB LOT(I),I=1,2,....,MB      -CARD TYPE 5/1X,I2,2X,25I3)MRCA1050
   21 IF(NDR)26,26,40                                                       MRCA1060
   40 READ 41,(IKEEPR(I),I=1,NDR)                                           MRCA1070
```

```
   41 FORMAT(2014)                                                    MRCA1080
      PRINT 411,(IKEEPR(I),I=1,NDR)                                   MRCA1090
  411 FORMAT(53HONUMBERS OF SELECTED INPUT DESIGN POINTS -CARD TYPE 6/(2MRCA1100
     1015))                                                           MRCA1110
   26 IF(MVP)25,25,466                                               MRCA1120
  466 REWIND 11                                                      MRCA1130
      DO 79 K=1,MVP                                                  MRCA1140
C READ IN POINTS FOR VARIANCE OF PREDICTION,COMPUTE PRODUCT TERMS.   MRCA1150
      CALL RDIT                                                      MRCA1160
   79 WRITE(11)(X(I),I=2, LIM)                                       MRCA1170
   25 PRINT 594,PGLB                                                 MRCA1180
  594 FORMAT(1H1,10A8)                                               MRCA1190
      NNSAV=NN                                                       MRCA1200
      NNL=NN                                                         MRCA1210
      DO 2 I=1,NN                                                    MRCA1220
      DO 2 J=I,NN                                                    MRCA1230
C INITIALIZE SUMS TO ZERO                                            MRCA1240
    2 S(I,J)=0.0                                                     MRCA1250
C READ INPUT DATA                                                    MRCA1260
      REWIND 10                                                      MRCA1270
      TAPE=IAPE                                                      MRCA1280
      M=0                                                            MRCA1290
    5 CALL RDIT                                                      MRCA1300
      IF(M1)31,55,31                                                 MRCA1310
   55 WRITE(10)(X(I),I=2,NN)                                         MRCA1320
      IF(NDPO)5008,5006,5008                                         MRCA1330
 5006 PRINT 5506,M,(X(I),I=2,NN)                                     MRCA1340
 5506 FORMAT(1H I4,2X,9F13.6/(7X,9F13.6))                            MRCA1350
      GO TO 7                                                        MRCA1360
 5008 IF(NDPO-1)7,5007,7                                             MRCA1370
 5007 PRINT 6,M,(X(I),I=2,NN)                                        MRCA1380
    6 FORMAT(1H ,I4,2X,7E17.8/(7X,7E17.8))                           MRCA1390
    7 DO 3 I=2,NN                                                    MRCA1400
C GENERATE TRIANGULAR SUMMATION MATRIX                               MRCA1410
      S(1,I)=S(1,I)+X(I)                                             MRCA1420
      DO 3 J=I,NN                                                    MRCA1430
    3 S(I,J)=S(I,J)+X(I)*X(J)                                        MRCA1440
      GO TO 5                                                        MRCA1450
   31 PRINT 594,PGLB                                                 MRCA1460
      IF((M-2)*(LIMOB-M))2097,2095,2095                              MRCA1470
 2097 PRINT 2096                                                     MRCA1480
 2096 FORMAT(34H1TOO FEW OR TOO MANY DATA POINTS  )                  MRCA1490
      GO TO 5400                                                     MRCA1500
 2095 PRINT 15                                                       MRCA1510
   15 FORMAT(17HOSUMMATION MATRIX)                                   MRCA1520
      S(1,1)=M                                                       MRCA1530
      RECM=1.0/S(1,1)                                                MRCA1540
C FORM SYMMETRICAL NORMAL MATRIX A                                   MRCA1550
      DO 9 I=1,NNN                                                   MRCA1560
      LOT(I)=0                                                       MRCA1570
      DO 9 J=I,NNN                                                   MRCA1580
      A(I,J)=S(I,J)                                                  MRCA1590
      A(J,I)=S(I,J)                                                  MRCA1600
C FORM MATRIX WHICH SAVES A                                          MRCA1610
      AKP(I,J)=S(I,J)                                                MRCA1620
      AKP(J,I)=S(I,J)                                                MRCA1630
    9 CONTINUE                                                       MRCA1640
      DO 33 I=1,NNN                                                  MRCA1650
   33 PRINT 16,(A(I,J),J=1,NNN),S(I,NN)                              MRCA1660
      PRINT 16,(S(J,NN),J=1,NN)                                      MRCA1670
```

```
   16 FORMAT(1H0,7E17.8/(1X,7E17.8))                                    MRCA1680
      DO 679 I=1,NNN                                                    MRCA1690
      B(I)=S(I,NN)                                                      MRCA1700
C SAVE CONSTANTS                                                        MRCA1710
  679 BB(I)=B(I)                                                        MRCA1720
C INVERT A AND SOLVE NORMAL EQUATIONS                                   MRCA1730
   96 NR=NR-1                                                           MRCA1740
      IF(NNNSAV)500,501,500                                             MRCA1750
  501 IF((IVORGO-3)*(IVORGO-4))500,502,500                              MRCA1760
  502 IF(IBID.EQ.2)IBIDS=3                                              MRCA1770
  500 KOUNT=0                                                           MRCA1780
      CALL ABT                                                          MRCA1790
      IF(IBIDS.EQ.3)IBIDS=1                                             MRCA1800
 2050 IF(ERROR) 221,5660,833                                            MRCA1810
  833 IF(NNNSAV)2008,2008,83                                            MRCA1820
 2008 ISKIP=2                                                           MRCA1830
      GO TO 660                                                         MRCA1840
C ISKIP=2 MEANS NO FINAL COMPREHENSIVE                                  MRCA1850
 5660 IF(NNNSAV)660,660,39                                              MRCA1860
  660 DO 60 I=2,NN                                                      MRCA1870
C COMPUTE THE AVERAGES OF EACH VARIABLE                                 MRCA1880
   60 AVV(I)=S(1,I)/S(1,1)                                              MRCA1890
      IF(NOBS.EQ.4)ISKIP=2                                              MRCA1900
      NPED=NPE                                                          MRCA1910
      PRINT 61,(I,AVV(I+1),I=1,NNN)                                     MRCA1920
   61 FORMAT(57HOAVERAGES OF INDEPENDENT VARIABLES AND DEPENDENT VARIABLMRCA1930
     1E/(6(I3,E17.8)))                                                  MRCA1940
      IF(ERROR)39,39,83                                                 MRCA1950
   39 IF(NDR+MVP)83,83,62                                               MRCA1960
   62 PRINT 594,PGLB                                                    MRCA1970
      IF(NNNSAV)611,611,2023                                            MRCA1980
 2023 PRINT 5760,SELECT,(LOT(I),I=1,NNNSAV)                             MRCA1990
 5760 FORMAT(32HOINDEPENDENT VARIABLE SELECTION ,A8,1X,5I11)            MRCA2000
  611 IF(NDR)63,63,44                                                   MRCA2010
   44 PRINT 64                                                          MRCA2020
   64 FORMAT(32HOSELECTED INPUT DESIGN POINTS...)                       MRCA2030
C RDISK READS DATA FROM DISK(OR BINARY TAPE)FOR USE AS SELECTED DATA    MRCA2040
C INPUT OBSERVATIONS AND CALLS PREVAR TO COMPUTE PREDICTED VALUES AND   MRCA2050
C PREDICTION STANDARD DEVIATIONS.                                       MRCA2060
      CALL RDISK(KOUNT,INDX)                                            MRCA2070
   63 IF(MVP)7202,7202,46                                               MRCA2080
   46 REWIND 11                                                         MRCA2090
   45 PRINT 43                                                          MRCA2100
   43 FORMAT(27HOSYNTHETIC DESIGN POINTS...)                            MRCA2110
      DO 80 K=1,MVP                                                     MRCA2120
C READ IN POINTS FOR STANDARD DEVIATION OF PREDICTION,COMPUTE AND WRITE.MRCA2130
      READ(11)(X(I),I=2,LIM)                                            MRCA2140
   80 CALL PREVAR(KOUNT,INDX)                                           MRCA2150
 7202 IF(KOUNT)83,83,7201                                               MRCA2160
 7201 IF(MVPL)3022,2022,3022                                            MRCA2170
 2022 PRINT 82,(K,YY(K),YSDEV(K),K=1,KOUNT)                             MRCA2180
   82 FORMAT(90HOITEM NUMBER,PREDICTED VALUE,AND PREDICTION STANDARD DEVMRCA2190
     1IATION FOR INDIVIDUAL OBSERVATIONS/(3(I5,2E17.8)))                MRCA2200
      GO TO 83                                                          MRCA2210
 3022 PRINT 86,(K,YY(K),YSDEV(K),K=1,KOUNT)                             MRCA2220
   86 FORMAT(86HOITEM NUMBER,PREDICTED VALUE,AND PREDICTION STANDARD DEVMRCA2230
     1IATION FOR THE PREDICTION LINE/(   3(I5,2E17.8)))                 MRCA2240
   83 IF(NR)107,103,84                                                  MRCA2250
C RESET MATRIX DIMENSIONS                                               MRCA2260
   84 CALL TIME(XYIT)                                                   MRCA2270
```

```
 4009 FORMAT(4H4RUN,I5,5H TOOK,F13.8,9H SECONDS.)                    MRCA2280
      PRINT 4009,IRUN,XYIT                                           MRCA2290
      CALL SETIT                                                     MRCA2300
      N=NSAV                                                         MRCA2310
      NN=NNSAV                                                       MRCA2320
      NNN=NEN                                                        MRCA2330
C FORM NEW MATRIX A WITH SMALLER DIMENSIONS                          MRCA2340
      DO 701 I=1,51                                                  MRCA2350
  701 LOT(I)=1                                                       MRCA2360
      READ 85,(LOT(L),L=1,NNN)                                       MRCA2370
   85 FORMAT(51I1)                                                   MRCA2380
      NNNSAV=NNN                                                     MRCA2390
      ERROR=0.0                                                      MRCA2400
      CALL REDUCM                                                    MRCA2410
      NN=N+2                                                         MRCA2420
      GO TO 96                                                       MRCA2430
  107 PRINT 507                                                      MRCA2440
  507 FORMAT(29H0CARD TYPE 2 IS INCORRECT.    )                      MRCA2450
      GO TO 5400                                                     MRCA2460
  103 CALL TIME(XYIT)                                                MRCA2470
      PRINT 4009,IRUN,XYIT                                           MRCA2480
      GO TO(126,127,127,127),IVORGO                                  MRCA2490
  127 NNNSAV=NEN                                                     MRCA2500
      GO TO(126,128,129,128),IVORGO                                  MRCA2510
  128 CALL SETIT                                                     MRCA2520
      PRINT 2066                                                     MRCA2530
 2066 FORMAT(1H2/65H B E G I N  I V O R  R E G R E S S I O N  C A L C U MRCA2540
     1L A T I O N S)                                                 MRCA2550
      SELECT=IVORS                                                   MRCA2560
      WRITE(9,2068)                                                  MRCA2570
 2068 FORMAT(1H2,119X/71H0B E G I N  I V O R  A N A L Y S I S  O F  V A MRCA2580
     1R I A N C E  T A B L E S,49X)                                  MRCA2590
      CALL IVOR                                                      MRCA2600
      CALL TIME(XIT)                                                 MRCA2610
      PRINT 2093,XIT                                                 MRCA2620
 2093 FORMAT(21HLIVOR EXECUTION TIME ,F11.5,9H SECONDS.)             MRCA2630
      GO TO(126,126,126,129),IVORGO                                  MRCA2640
  129 CALL SETIT                                                     MRCA2650
      PRINT 2067                                                     MRCA2660
 2067 FORMAT(1H2/67H B E G I N  B I V O R  R E G R E S S I O N  C A L C MRCA2670
     1U L A T I O N S)                                               MRCA2680
      SELECT=BIVORS                                                  MRCA2690
      WRITE(9,2069)                                                  MRCA2700
 2069 FORMAT(1H2,119X/73H0B E G I N  B I V O R  A N A L Y S I S  O F  V MRCA2710
     1A R I A N C E  T A B L E S,47X)                                MRCA2720
      CALL BIVOR                                                     MRCA2730
      CALL TIME(XIT)                                                 MRCA2740
      PRINT 2098,XIT                                                 MRCA2750
 2098 FORMAT(22HLBIVOR EXECUTION TIME ,F11.8,9H SECONDS.)            MRCA2760
  126 CALL INTVL(XIT)                                                MRCA2770
      END FILE 9                                                     MRCA2780
      REWIND 9                                                       MRCA2790
      END FILE 13                                                    MRCA2800
      REWIND 13                                                      MRCA2810
 2062 READ(9,2061)LET                                                MRCA2820
 2061 FORMAT(15A8)                                                   MRCA2830
      IF(EOF(XYIT))GO TO 2065                                        MRCA2840
 2063 PRINT 2061,LET                                                 MRCA2850
      GO TO 2062                                                     MRCA2860
 2065 REWIND 9                                                       MRCA2870
```

```
 2094 READ(13,2061)LET                                                  MRCA2880
      IF(EOF(XYIT))GO TO 2092                                           MRCA2890
 2091 PRINT 2061,LET                                                    MRCA2900
      GO TO 2094                                                        MRCA2910
 2092 REWIND 13                                                         MRCA2920
      CALL INTVL(XIT)                                                   MRCA2930
      PRINT 4008,XIT                                                    MRCA2940
 4008 FORMAT(24HLCOMPREHENSIVE PRINTOUTS,F13.9,9H SECONDS.)             MRCA2950
 2010 CALL KLOK(XIT)                                                    MRCA2960
      PRINT 2013,XIT                                                    MRCA2970
 2013 FORMAT(44H4TOTAL PROBLEM RUNNING TIME(HRS./MIN./SEC.)=,A8)        MRCA2980
      GO TO 5401                                                        MRCA2990
  221 STOP                                                              MRCA3000
 5400 RETURN                                                            MRCA3010
      END                                                               MRCA3020
T         SUBTYPE,FORTRAN,LMAP,PBIN                                    ABT00000
      SUBROUTINE ABT                                                    ABT 0010
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)           ABT 0020
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)ABT 0030
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASEABT 0040
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),I0    ABT 0050
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)        ABT 0060
      COMMON IN(49,10),IR,IS,MI,JLIM,NN,M,NTAPE                         ABT 0070
      COMMON SELECT,ISID,IBIDS                                          ABT 0080
      DIMENSION ESTEP(30),IFREQ(31),FGRAPH(65),IOBF(30),CMPFR(30)       ABT 0090
      DIMENSION CHI(30)                                                 ABT 0100
      DIMENSION EYY(7000)                                               ABT 0110
      DIMENSION LIT(52)                                                 ABT 0120
      EQUIVALENCE(IDGO,NOBS)                                            ABT 0130
      EQUIVALENCE(LIT(1),B(1396))                                       ABT 0140
      EQUIVALENCE(ESTEP(1),B(201)),(IFREQ(1),B(231))                    ABT 0150
      EQUIVALENCE(FGRAPH(1),B(262)),(IOBF(1),B(327))                    ABT 0160
      EQUIVALENCE(CMPFR(1),B(357)),(CHI(1),B(387))                      ABT 0170
      EQUIVALENCE(EYY,YSDEV)                                            ABT 0180
      DATA BLANK(6H      )                                              ABT 0190
      DATA XXX(8HX       )                                              ABT 0200
      DATA ZZZ(8H0       )                                              ABT 0210
  731 CALL INTVL(XIT)                                                   ABT 0220
      CALL GAUSS                                                        ABT 0230
      CALL INTVL(XIT)                                                   ABT 0240
      PRINT 987,ICASE,XIT                                               ABT 0250
  987 FORMAT(19H0MATRIX INVERSION  ,I4,21H ...EVALUATION TIME =,F13.8,9HABT 0260
     1 SECONDS.)                                                        ABT 0270
  745 IF(ERROR)106,998,106                                              ABT 0280
  998 IF(DETERM)106,106,999                                             ABT 0290
  999 PRINT 35,DETERM                                                   ABT 0300
   35 FORMAT(13H0DETERMINANT=,G18)                                      ABT 0310
C     PRINT A INVERSE AND SOLUTION TO SIMULTANEOUS EQUATIONS            ABT 0320
      PRINT 17                                                          ABT 0330
   17 FORMAT(59H0INVERSE OF MATRIX A AND SOLUTION TO SIMULTANEOUS EQUATIABT 0340
     1ONS)                                                              ABT 0350
        DO 57 I = 1,NNN                                                 ABT 0360
   57 PRINT 16,(A(I,J),J=1,NNN),B(I)                                    ABT 0370
   16 FORMAT(7E17.8)                                                    ABT 0380
      SSR=0.                                                            ABT 0390
      DO 20 I=1,NNN                                                     ABT 0400
   20 SSR=SSR+B(I)*BB(I)                                                ABT 0410
      SSE=S(NNL,NNL)-SSR                                                ABT 0420
      ATSS=S(NNL,NNL)-S(1,NNL)*(S(1,NNL)/S(1,1))                        ABT 0430
      ASSR=ATSS-SSE                                                     ABT 0440
```

```
      CORSQ=ASSR/ATSS                                                    ABT 0450
      IF(CORSQ)109,23,23                                                 ABT 0460
   23 COR=SQRT(CORSQ)                                                    ABT 0470
      FN=NNN                                                             ABT 0480
      IF(S(1,1),EQ,FN)GO TO 31                                           ABT 0490
      SDEVSQ=SSE/(S(1,1)-FN)                                             ABT 0500
      IF(SDEVSQ)108,24,24                                                ABT 0510
   31 SDEVSQ=0.0                                                         ABT 0520
   24 SDEV=SQRT(SDEVSQ)                                                  ABT 0530
      DO 21 I=1,NNN                                                      ABT 0540
      IF(A(I,I))996,997,997                                              ABT 0550
  997 BSDEV(I)=SDEV*SQRT(A(I,I))                                         ABT 0560
   21 CONTINUE                                                           ABT 0570
      GO TO(150,151,150),IBIDS                                           ABT 0580
  150 CALL IDENTM                                                        ABT 0590
      CALL PRINTM                                                        ABT 0600
      IF(IDGO.GT.1)GO TO 151                                             ABT 0610
      GO TO(151,151,152),IBIDS                                           ABT 0620
  152 PRINT 153                                                          ABT 0630
  153 FORMAT(65HONO IDENTITY MATRIX CHECKS WILL BE MADE ON SUBSEQUENT BI ABT 0640
     1VOR RUNS.)                                                         ABT 0650
  151 PRINT 58,(I,BSDEV(I),I=1,NNN)                                      ABT 0660
   58 FORMAT((35HOSTANDARD DEVIATION OF COEFFICIENTS   )/(6(I3,E17.8))) ABT 0670
C THE G FORMAT IS USED TO PRINT THE MAXIMUM NUMBER OF SIGNIFICANT DIGITSABT 0680
C IN THE GIVEN NUMBER OF COLUMNS.                                        ABT 0690
 1074 PRINT 574,SSE                                                      ABT 0700
  574 FORMAT (1HO,  G18.35H  RESIDUAL OR ERROR SUM OF SQUARES.)          ABT 0710
 1075 PRINT 575,ATSS                                                     ABT 0720
  575 FORMAT (1H ,  G18.45H  TOTAL SUM OF SQUARES ADJUSTED FOR THE MEAN.ABT 0730
     1)                                                                  ABT 0740
 1076 PRINT 576,ASSR                                                     ABT 0750
  576 FORMAT (1H ,  G18.50H  REGRESSION SUM OF SQUARES ADJUSTED FOR THE  ABT 0760
     1MEAN.)                                                             ABT 0770
 1077 PRINT 577,COR                                                      ABT 0780
  577 FORMAT (1H ,  G18.30H  CORRELATION COEFFICIENT (R).)               ABT 0790
 1078 PRINT 578,SDEV                                                     ABT 0800
  578 FORMAT (1H ,  G18.35H  SQUARE ROOT OF RESIDUAL VARIANCE.)          ABT 0810
      CALL CMPR(ASSR,SSE,N,M,COR,B,PGLB,LGT,NNNSAV,M4)                   ABT 0820
      IF(NNNSAV)2083,2084,2083                                           ABT 0830
 2084 RSSMO=ASSR                                                         ABT 0840
      ESSO= SSE                                                          ABT 0850
      NR1=N-NNN                                                          ABT 0860
      NRO = N                                                            ABT 0870
      WRITE(13,2093)PGLB                                                 ABT 0880
 2093 FORMAT(1H1,10A8,39X)                                               ABT 0890
      WRITE (13,2085)NR1                                                 ABT 0900
 2085 FORMAT(1HO,39HDEGREES OF FREEDOM OF ERROR VARIANCE = ,I4,76X)      ABT 0910
      WRITE (13,2088)                                                    ABT 0920
 2088 FORMAT(1HO,7HCOEFFICIENT OF     NO (DF) OF              F FOR REGABT 0930
     1RESSION ON     INDEPENDENT,41X/1H ,8HDETERMINATION     DELETED VABT 0940
     2ARIABLES      DELETED VARIABLES      VARIABLE SELECTION,35X/120X) ABT 0950
      WRITE(13,2090)CORSQ                                                ABT 0960
 2090 FORMAT(1HO,F9.7,11H - MAIN RUN,99X)                                ABT 0970
      GO TO 59                                                           ABT 0980
 2083 CALL FIX                                                           ABT 0990
      IF(RSSMO)  94,2086,2086                                            ABT 1000
 2086 NR2=NRO-N                                                          ABT 1010
      FOUT=(RSSMO-ASSR)*(FLOAT(NR1))/(ESSO*FLOAT(NR2))                   ABT 1020
      IF(NR1.EQ.0)FOUT=999999999.999                                     ABT 1021
      WRITE(13,2089)CORSQ,NR2,FOUT,LIT                                   ABT 1030
```

```
 2089 FORMAT(1H ,F9.7,13X,I2,22X,F14.3,7X,52A1)                          ABT 1040
      GO TO 59                                                           ABT 1050
   54 IF(RSSMO+1.0)55,59,59                                              ABT 1060
   55 WRITE(13,2093)PGLB                                                 ABT 1070
      WRITE (13,56)                                                      ABT 1080
   56 FORMAT(1HO,80HNO FINAL COMPREHENSIVE PRINTOUT SINCE MATRIX FOR MA  ABT 1090
     1N RUN COULD NOT BE INVERTED ,39X)                                  ABT 1100
      RSSMO=-1.0                                                         ABT 1110
   57 IF(NPED)99,5660,99                                                 ABT 1120
C  COMPUTE PREDICTION ERROR                                              ABT 1130
   99 REWIND 10                                                          ABT 1140
      ES2 = 0.0                                                          ABT 1150
   25 DO 26 K=1,M                                                        ABT 1160
   28 READ (10)(X(I),I=2,NNSAV)                                          ABT 1170
      IF(NNNSAV)29,29,7280                                               ABT 1180
 7280 NNW=2                                                              ABT 1190
      DO 7299 I=2,ITOTAL                                                 ABT 1200
      IF(LOT(I))104,727,7299                                             ABT 1210
  727 X(NNW)=X(I)                                                        ABT 1220
  728 NNW=NNW+1                                                          ABT 1230
 7299 CONTINUE                                                           ABT 1240
   29 YY(K)=B(1)                                                         ABT 1250
      DO 30 I=2,NNN                                                      ABT 1260
   30 YY(K)=YY(K)+X(I)*B(I)                                              ABT 1270
      EYY(K)=X(NNSAV)-YY(K)                                              ABT 1280
      ES2=  ES2 + EYY(K)*EYY(K)                                          ABT 1290
   26 CONTINUE                                                           ABT 1300
      REWIND 10                                                          ABT 1310
C  COMPUTE RANGE OF ERRORS                                               ABT 1320
      CALL MAXMIN(M,EYY,EYYU,EYYL,IXMAX,IXMIN)                           ABT 1330
CDETERMINE AND PLOT  DISTRIBUTION OF ERRORS.  PERFORM CHI SQUARE TEST    ABT 1340
C IF POSSIBLE                                                            ABT 1350
   22 ERANGE = EYYU-EYYL                                                 ABT 1360
      EDELTA = ERANGE/30.0                                               ABT 1370
      ESTEP(1) = EYYL + EDELTA                                           ABT 1380
      IFREQ(1) = 0.0                                                     ABT 1390
      DO 2003 II=2,30                                                    ABT 1400
      IFREQ(II)= 0                                                       ABT 1410
 2003 ESTEP(II) = ESTEP(II-1)+EDELTA                                     ABT 1420
      ESTEP(30)=  EYYU                                                   ABT 1430
      IFREQ(31)= 0                                                       ABT 1440
      DO 2004 II=1,M                                                     ABT 1450
      JJ= (EYY(II)-EYYL)/EDELTA                                          ABT 1460
 2004 IFREQ(JJ+1)=IFREQ(JJ+1)+1                                          ABT 1470
      IFREQ(30)=IFREQ(30)+IFREQ(31)                                      ABT 1480
      CALL CHISQ(N,ESTEP,IFREQ,S(1,1),SDEV,IDF,CHI,CHISUM,IOBF,CHIFR)    ABT 1490
 2002 IF(INDPO-1)5026,5027,5026                                          ABT 1500
 5026 PRINT 594,PGLB                                                     ABT 1510
  594 FORMAT(1H1,10A8)                                                   ABT 1520
      IF(NNNSAV)2014,2013,2014                                           ABT 1530
 2014 PRINT 5760,SELECT,(LOT(LIO),LIO=1,NNNSAV)                          ABT 1540
 5760 FORMAT(32HO1NDEPENDENT VARIABLE SELECTION ,A8,1X,9I11)             ABT 1550
 2013 PRINT 5526,(K,YY(K),EYY(K),K=1,M)                                  ABT 1560
 5526 FORMAT((49HO)ITEM NUMBER PREDICTED VALUE AND PREDICTION ERROR )/   ABT 1570
     1(3(I8,2F15.6)))                                                    ABT 1580
      PRINT 5528,ES2                                                     ABT 1590
      PRINT 594,PGLB                                                     ABT 1600
      IF(NNNSAV)2017,2016,2017                                           ABT 1610
 2017 PRINT 5760,SELECT,(LOT(LIO),LIO=1,NNNSAV)                          ABT 1620
 2016 PRINT 2005,ERANGE                                                  ABT 1630
```

```
2005 FORMAT(        40HOPREDICTION ERROR FREQUENCY DISTRIBUTION  /      ABT 1640
     1 9H RANGE = F15.4, /                                             ABT 1650
     228H       UPPER BOUND  FREQUENCY  2X,                            ABT 1660
     3 9HBAR CHART,61X,3HCHI,3X,6HOBS FR,3X,7HEXPD FR)                 ABT 1670
2038 DO 2032 II = 1,30                                                 ABT 1680
      IFGRPH=IFREQ(II)                                                 ABT 1690
      IFGRPH =MINO(65,IFGRPH)                                          ABT 1700
      IF(IFGRPH)2024,2026,2024                                         ABT 1710
2024 DO 2025 IFG= 1,IFGRPH                                             ABT 1720
2025 FGRAPH(IFG)=XXX                                                   ABT 1730
2026 IFGRPH= IFGRPH +1                                                 ABT 1740
      IF(IFGRPH-66)2033,2034,2034                                      ABT 1750
2034 FGRAPH(65)=ZZZ                                                    ABT 1760
      GO TO 2035                                                       ABT 1770
2033, DO 2027 IFG =IFGRPH,65                                           ABT 1780
2027 FGRAPH(IFG)=BLANK                                                 ABT 1790
2035 IF(NDPO-1)2028,2030,2028                                          ABT 1800
2028 IF(CHI(II))2040,2041,2041                                         ABT 1810
2040 PRINT 2029,ESTEP(II),IFREQ(II),FGRAPH                             ABT 1820
      GO TO 2032                                                       ABT 1830
2041 PRINT 2029,ESTEP(II),IFREQ(II),FGRAPH,CHI(II),IOBF(II),CMPFR(II)  ABT 1840
2029 FORMAT(2X,F15.4,2X,I5,6X,1HI,65A1,1X,F8.3,1X,I5,2X,F9.3)          ABT 1850
      GOTO 2032                                                        ABT 1860
2030 IF(CHI(II))2042,2043,2043                                         ABT 1870
2042 PRINT 2031,ESTEP(II),IFREQ(II),FGRAPH                             ABT 1880
      GO TO 2032                                                       ABT 1890
2043 PRINT 2031,ESTEP(II),IFREQ(II),FGRAPH,CHI(II),IOBF(II),CMPFR(II)  ABT 1900
2031 FORMAT(2X,E15.8,2X,I5,6X,1HI,65A1,1X,F8.3,1X,I5,2X,F9.3)          ABT 1910
2032 CONTINUE                                                          ABT 1920
      IF(IDF)2048,2048,2049                                            ABT 1930
2048 PRINT 2050                                                        ABT 1940
2050 FORMAT(1X,31HCHISQUARE COULD NOT BE COMPUTED)                     ABT 1950
      GO TO 5660                                                       ABT 1960
2049 PRINT 2039,CHISUM,IDF                                             ABT 1970
2039 FORMAT(12H CHISQUARE = F15.3,22H  DEGREES OF FREEDOM = I5 )       ABT 1980
      GO TO 5660                                                       ABT 1990
5027 PRINT 594,PGLB                                                    ABT 2000
      IF(NNNSAV)2019,2018,2019                                         ABT 2010
2019 PRINT 5760,SELECT,(LOT(LIQ),LIQ=1,NNNSAV)                         ABT 2020
2018 PRINT 75,(K,YY(K),EYY(K),K=1,M)                                   ABT 2030
  75 FORMAT((49H0ITEM NUMBER PREDICTED VALUE AND PREDICTION ERROR )/   ABT 2040
     1(3(I5,2E15.6)))                                                  ABT 2050
      PRINT 5528,ES2                                                   ABT 2060
5528 FORMAT(27H0CHECK ERROR SUM OF SQUARES /1H ,G18)                   ABT 2070
      PRINT 594,PGLB                                                   ABT 2080
      IF(NNNSAV)2021,2020,2021                                         ABT 2090
2021 PRINT 5760,SELECT,(LOT(LIQ),LIQ=1,NNNSAV)                         ABT 2100
2020 PRINT 2006,ERANGE                                                 ABT 2110
2006 FORMAT(        40HOPREDICTION ERROR FREQUENCY DISTRIBUTION  /     ABT 2120
     1 9H RANGE = E15.6, /                                             ABT 2130
     228H       UPPER BOUND  FREQUENCY  2X,                            ABT 2140
     3 9HBAR CHART,61X,3HCHI,3X,6HOBS FR,3X,7HEXPD FR)                 ABT 2150
      GO TO 2038                                                       ABT 2160
  104 PRINT 504                                                        ABT 2170
  504 FORMAT(36H0A RERUN CARD IS MADE UP INCORRECTLY)                  ABT 2180
      ERROR=-1.0                                                       ABT 2190
      GO TO 5661                                                       ABT 2200
  106 PRINT 505                                                        ABT 2210
505   FORMAT (25H0MATRIX FAILED TO INVERT.)                            ABT 2220
      GO TO 83                                                         ABT 2230
```

```
  108 PRINT 506                                                          ABT 2240
  506 FORMAT(22H0VARIANCE IS NEGATIVE.)                                  ABT 2250
      GO TO 83                                                           ABT 2260
  109 PRINT 503                                                          ABT 2270
  503 FORMAT(55H0THE SQUARE OF THE CORRELATION COEFFICIENT IS NEGATIVE.)ABT 2280
      GO TO 83                                                           ABT 2290
  996 PRINT 995                                                          ABT 2300
  995 FORMAT(67H0AN ELEMENT OF THE MAIN DIAGONAL OF THE INVERSE MATRIX IABT 2310
     1S NEGATIVE.)                                                       ABT 2320
   83 ERROR=1.0                                                          ABT 2330
      GO TO 5661                                                         ABT 2340
 5660 ERROR=0.0                                                         ABT 2350
 5661 RETURN                                                            ABT 2360
      END                                                               ABT 2370
T        SUBTYPE,FORTRAN,LMAP,PBIN                                      BIVOR000
      SUBROUTINE BIVOR                                                  BIVOR010
C BIVOR-BACKWARD IVOR-INDEPENDENT VARIABLE SELECTION SUBROUTINE FOR THE BIVOR020
C ORDERING OF INDEPENDENT VARIABLES ACCORDING TO MAGNITUDES OF          BIVOR030
C REGRESSION SUMS OF SQUARES.                                           BIVOR040
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)           BIVOR050
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)BIVOR060
      COMMON NNL,DETERM,NIBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASEBIVOR070
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ    BIVOR080
      COMMON NNXA,NNSAV,SDEV,AKP(51,31),BB(52),S(52,52),PGLB(10)        BIVOR090
      COMMON IN(49,10),IR,IS,MI,JLIM,NN,M,NTAPE                         BIVOR100
      COMMON SELECT,IBID,IBIDS                                          BIVOR110
      DIMENSION LAT(51)                                                 BIVOR120
      EQUIVALENCE (LAT,XD)                                              BIVOR130
      EQUIVALENCE(NIBS,IDGO)                                            BIVOR140
C  M4 COUPLED WITH THE VARIABLE LL(IN CMPR) CONTROLS THE PRINTING OF    BIVOR150
C  ANALYSIS OF VARIANCE TABLES.                                         BIVOR160
      NOBS=1                                                            BIVOR170
C  SAVE NPED                                                            BIVOR180
      NTAPE=NPED                                                        BIVOR190
      KNUM=-1                                                           BIVOR200
C KNUM=-1 LETS CASSR KNOW THAT BIVOR(INSTEAD OF IVOR)IS BEING USED.     BIVOR210
      NPED=1                                                            BIVOR220
      ASSIGN 551 TO ISEE                                                BIVOR230
      M4=0                                                              BIVOR240
      GO TO(1,22),ISKIP                                                 BIVOR250
    1 WRITE(13,103)                                                     BIVOR260
  103 FORMAT(56H0*** B I V O R  F I N A L  C O M P R E H E N S I V E ***BIVOR270
     1,64X/120X)                                                        BIVOR280
   22 DO 101 I=1,51                                                     BIVOR290
  101 LOT(I)=0                                                          BIVOR300
      ITOTAL=1                                                          BIVOR310
      DO 102 I=1,MB                                                     BIVOR320
  102 ITOTAL=ITOTAL+NQ(I)                                               BIVOR330
      ISTART=ITOTAL+1                                                   BIVOR340
      IF(ISTART-51)106,107,107                                          BIVOR350
  106 DO 105 I=ISTART,51                                                BIVOR360
  105 LOT(I)=1                                                          BIVOR370
  107 DO 200 I=1,MB                                                     BIVOR380
      IDUM=0                                                            BIVOR390
      ITOTAL=ISTART-1                                                   BIVOR400
      ISTART=ISTART-NQ(I)                                               BIVOR410
      NQQ=NQ(I)                                                         BIVOR420
      JSAVE=ITOTAL                                                      BIVOR430
      JLOT=ISTART-1                                                     BIVOR440
      DO 600 K=1,NQQ                                                    BIVOR450
```

```
      GO TO ISEE                                                        BIVOR460
  551 ERROR=1.0                                                         BIVOR470
      CALL REDUCM                                                       BIVOR480
      CALL CASSR(IDUM,KGO)                                              BIVOR490
      GO TO(500,501),KGO                                                BIVOR500
  501 LOT(JSAVE)=1                                                      BIVOR510
      JSAVE=JSAVE-1                                                     BIVOR520
      GO TO 600                                                         BIVOR530
  500 JLOT=ISTART-1                                                     BIVOR540
      KASSR=0                                                           BIVOR550
      DO 300 J=ISTART,ITOTAL                                            BIVOR560
      IF(LOT(J))301,301,300                                             BIVOR570
  301 JLOT=JLOT+1                                                       BIVOR580
      KASSR=KASSR+1                                                     BIVOR590
      AW(KASSR)=B(JLOT)*(B(JLOT)/A(JLOT,JLOT))                          BIVOR600
      LAT(KASSR)=J                                                      BIVOR610
  300 CONTINUE                                                          BIVOR620
  204 IF(KASSR-1)221,400,404                                            BIVOR630
  400 IXMIN=1                                                           BIVOR640
      GO TO 402                                                         BIVOR650
  404 CALL MAXMIN(KASSR,AW,AMAX,AMIN,IXMAX,IXMIN)                       BIVOR660
  402 IMAX=LAT(IXMIN)                                                   BIVOR670
      IF(IBID.NE.2.OR.IDGO.NE.1)GO TO 100                               BIVOR680
      IBIDS=3                                                           BIVOR690
      IBID=1                                                            BIVOR700
  100 GO TO(524,526),NOBS                                               BIVOR710
  524 IF(NNN-NNNSAV)525,526,525                                         BIVOR720
  525 ERROR=0.0                                                         BIVOR730
   09 CALL REDUCM                                                       BIVOR740
      CALL ABT                                                          BIVOR750
      IF(IBIDS.EQ.3)IBIDS=2                                             BIVOR760
      NPED=NTAPE                                                        BIVOR770
      IF(NNN-2)220,220,526                                              BIVOR780
  526 LOT(IMAX)=1                                                       BIVOR790
      NOBS=2                                                            BIVOR800
      NPED=NTAPE                                                        BIVOR810
      ERROR=0.0                                                         BIVOR820
      CALL REDUCM                                                       BIVOR830
      CALL ABT                                                          BIVOR840
      IF(IBIDS.EQ.3)IBIDS=2                                             BIVOR850
  202 IF(NNN-2)220,220,599                                              BIVOR860
  599 ASSIGN 500 TO ISEE                                                BIVOR870
  600 CONTINUE                                                          BIVOR880
  200 CONTINUE                                                          BIVOR890
  220 RETURN                                                            BIVOR900
  221 STOP                                                              BIVOR910
      END                                                               BIVOR920
         SUBTYPE,FORTRAN,LMAP,PBIN                                      CASSR000
      SUBROUTINE CASSR(KASSR,KGO)                                       CASSR010
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)           CASSR020
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)CASSR030
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASECASSR040
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ    CASSR050
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)        CASSR060
      COMMON IN(49,10),IR,IS,MI,JLIM,NN,M,NTAPE                         CASSR070
      COMMON SELECT,IBID,IBIDS                                          CASSR080
      DIMENSION EYY(7000)                                               CASSR090
      EQUIVALENCE(EYY,YSDEV)                                            CASSR100
      EQUIVALENCE(IDGO,NOBS)                                            CASSR110
      EQUIVALENCE(TOL12,TOLCES)                                         CASSR120
```

```
      FNNN=NNN                                                          CASSR130
      CALL GAUSS                                                        CASSR140
  745 IF(FRROR)106,998,106                                              CASSR150
  998 IF(DETERM)106,106,57                                              CASSR160
   57 SSR=0.0                                                           CASSR170
      DO 20 I=1,NNN                                                     CASSR180
   20 SSR=SSR+B(I)*BB(I)                                                CASSR190
      SSE=S(NNL,NNL)-SSR                                                CASSR200
      ATSS=S(NNL,NNL)-((S(1,NNL)**2)/S(1,1))                            CASSR210
      ASSR=ATSS-SSE                                                     CASSR220
C ISKIP IS SET IN THE MAIN PROGRAM.ISKIP=1 IF MAIN RUN IS SUCCESSFUL,=2 CASSR230
C MAIN RUN IS UNSUCCESSFUL.                                             CASSR240
      GO TO(579,580),ISKIP                                              CASSR250
  580 CORSQ=ASSR/ATSS                                                   CASSR260
      IF(CORSQ)109,23,23                                                CASSR270
   23 SDEVSQ=SSE/(S(1,1)-FNNN)                                          CASSR280
      IF(SDEVSQ)108,24,24                                               CASSR290
   24 DO 21 I=1,NNN                                                     CASSR300
      IF(A(I,I))996,21,21                                               CASSR310
   21 CONTINUE                                                          CASSR320
  999 CALL IDENTM                                                       CASSR330
      GO TO(579,579,579,17),IDGO                                        CASSR340
  579 KASSR=KASSR+1                                                     CASSR350
      AW(KASSR)=ASSR                                                    CASSR360
C ASSR - REGRESSION SUM OF SQUARES ADJUSTED FOR THE MEAN                CASSR370
C KGO=1 MEANS VALID ASSR WAS COMPUTED    =2 INVALID ASSR                CASSR380
      KGO=1                                                             CASSR390
      GO TO 221                                                         CASSR400
   17 IF(KNUM+1)19,18,19                                                CASSR410
   18 IF(NNN.EQ.NNNSAV)GO TO 19                                         CASSR420
      CALL REDUCM                                                       CASSR430
      CALL ABT                                                          CASSR440
      KGO=2                                                             CASSR450
      GO TO 221                                                         CASSR460
   19 PRINT 2009,TOL12                                                  CASSR470
 2009 FORMAT(79H0DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MACASSR480
     1TRIX LARGER THAN 1(2)= ,G9.15H ,RUN REJECTED.)                    CASSR490
      GO TO 83                                                          CASSR500
  104 PRINT 110                                                         CASSR510
  110 FORMAT(32H IVS CONTAINED NEGATIVE ELEMENT.)                       CASSR520
      GO TO 83                                                          CASSR530
  995 PRINT 995                                                         CASSR540
  995 FORMAT(67H0AN ELEMENT OF THE MAIN DIAGONAL OF THE INVERSE MATRIX ICASSR550
     1S NEGATIVE.)                                                      CASSR560
      GO TO 83                                                          CASSR570
  106 PRINT 505                                                         CASSR580
505   FORMAT (25H0MATRIX FAILED TO INVERT.)                             CASSR590
      GO TO 83                                                          CASSR600
  108 PRINT 506                                                         CASSR610
  506 FORMAT(22H0VARIANCE IS NEGATIVE.)                                 CASSR620
      GO TO 83                                                          CASSR630
  109 PRINT 503                                                         CASSR640
  503 FORMAT(55H0THE SQUARE OF THE CORRELATION COEFFICIENT IS NEGATIVE.)CASSR650
   83 PRINT 2089,(LOT(I),I=1,NNNSAV)                                    CASSR660
 2089 FORMAT(6H IVS= ,5111)                                             CASSR670
      KGO=2                                                             CASSR680
  221 RETURN                                                            CASSR690
      END                                                               CASSR700
T        SUBTYPE,FORTRAN,LMAP,PBIN                                      CHISQ000
       SUBROUTINE CHISQ(N,ESTEP,IFREQ,OB    ,SDEV,IDF,CHI,CHISUM,IOBFR,  CHISQ010
```

```
     1CMPFR)                                                          CHISQ020
C THIS SUBROUTINE FITS A NORMAL CURVE WITH MEAN 0 AND STANDARD DEVIATIONCHISQ030
C SDEV TO THE DATA IN IFREQ WHERE THE UPPER BOUND OF EACH INTERVAL IS INCHISQ040
C THE CORRESPONDING ENTRY IN ESTEP.  N IS THE NUMBER OF INDEPENDENT     CHISQ050
C VARIABLES. OB    IS THE NUMBER OF OBSERVATIONS.                       CHISQ060
C THE ROUTINE GROUPS THE DATA SO THAT THERE ARE AT LEAST 5 COMPUTED     CHISQ070
C VALUES IN EACH INTERVAL AND THEN COMPUTES THE CHISQUARE STATISTIC TO  CHISQ080
C GIVE AN EST'MATION OF THE GOODNESS OF FIT.  ON EXIT FROM THE ROUTINE  CHISQ090
C IDF CONTAINS THE NUMBER OF DEGREES OF FREEDOM. CHISUM THE CHISQUARE   CHISQ100
C VALUE. AND CHI(J) CONTAINS A -1 IF THE JTH INTERVAL WAS NOT THE LAST  CHISQ110
C OF A GROUP OTHERWISE IT CONTAINS (OBSERVED FREQUENCY-THEORETICAL      CHISQ120
C FREQUENCY)**2/THEORETICAL FREQUENCY.  IF THERE IS AN INSUFFICIENT     CHISQ130
C NUMBER OF OBSERVATIONS THE FIT IS NOT ATTEMPTED AND ALL OUTPUT VALUES CHISQ140
C ARE SET TO -1 EXCEPT FOR IDF WHICH WILL BE -(N+3).                    CHISQ150
C IOBF(J) CONTAINS ON EXIT THE OBSERVED FREQUENCY IF THE JTH INTERVAL   CHISQ160
C WAS THE LAST OF A GROUP. OTHERWISE ITS CONTENTS ARE MEANINGLESS.      CHISQ170
C LIKEWISE CMPFR(J) CONTAINS THE THEORETICAL FREQUENCY.                 CHISQ180
      ODIMENSION ESTEP(30),IFREQ(30),CHI(30)                           CHISQ190
      ODIMENSION IOBFR(30),CMPFR(30)                                   CHISQ200
        FOT=0.0                                                        CHISQ210
       KOUNT=0                                                         CHISQ220
       CHISUM=0.0                                                      CHISQ230
       PROBO=0.0                                                       CHISQ240
       FO=0.0                                                          CHISQ250
       IF(OB/5.0-FLOAT(N)-3.0)1,1,3                                    CHISQ260
     1 JJ=1                                                            CHISQ270
       CHISUM=-1.0                                                     CHISQ280
       CHI(30)=-1.0                                                    CHISQ290
       GO TO 14                                                        CHISQ300
     3 DO 10 J=1,30                                                    CHISQ310
       IF(J.NE.30)GO TO 11                                             CHISQ320
       PROBN=1.0                                                       CHISQ330
       GO TO 2                                                         CHISQ340
    11 PROBN=FREQ(ESTEP(J)/SDEV)                                       CHISQ350
     2 FOC=OB*(PROBN-PROBO)                                            CHISQ360
       FO=FLOAT(IFREQ(J))+FO                                           CHISQ370
        IF(FOC-5.0)4,4,5                                               CHISQ380
     4 CHI(J)=-1.0                                                     CHISQ390
       GO TO 10                                                        CHISQ400
     5 FOT=FOT+FO                                                      CHISQ410
       REMAIN=OB*(1.0-PROBO)                                           CHISQ420
       IF(OB*(1.0-PROBN)-5.0)6,6,7                                     CHISQ430
     6  FO=FO+(OB-FOT)                                                CHISQ440
       FOMRE=FO-REMAIN                                                 CHISQ450
       CHI(30)=FOMRE*FOMRE/REMAIN                                      CHISQ460
       IOBFR(30)=FO                                                    CHISQ470
       CMPFR(30)=REMAIN                                                CHISQ480
       CHISUM=CHISUM+CHI(30)                                           CHISQ490
       KOUNT = KOUNT +1                                                CHISQ500
       JJ= J                                                           CHISQ510
       GO TO 12                                                        CHISQ520
     7  IF(J-30)9,8,9                                                 CHISQ530
     8 FOC=REMAIN                                                      CHISQ540
     9  CHI(J)=(FO-FOC)**2/FOC                                        CHISQ550
       CHISUM= CHISUM + CHI(J)                                         CHISQ560
       IOBFR(J)=FO                                                     CHISQ570
        CMPFR(J)=FOC                                                   CHISQ580
       KOUNT= KOUNT + 1                                                CHISQ590
       FO = 0.0                                                        CHISQ600
       PROBO = PROBN                                                   CHISQ611
```

```
   10 CONTINUE                                                          CHISQ620
      GO TO 17                                                          CHISQ630
   12 IF(JJ-29)14,14,17                                                 CHISQ640
   14 DO 16 J=JJ,29                                                     CHISQ650
   16 CHI(J) = -1.0                                                     CHISQ660
   17 IDF = KOUNT - N - 3                                               CHISQ670
      RETURN                                                            CHISQ680
      END                                                               CHISQ690
T        SUBTYPE,FORTRAN,LMAP,PBIN                                      CMPR0000
      SUBROUTINE CMPR(RSSM,ESS,IR,N,CORR,B,PGLB,LOT,NNNSAV,44)          CMPR0010
      DIMENSION B(51),J(52),PGLB(10),LOT(51)                            CMPR0020
      COMMON DUM(25458),SELECT                                          CMPR0030
      DIMENSION LIT(52)                                                 CMPR0040
      DIMENSION FORM (9),BCDA(4),BCDB(4),BCDC(4),FORM2(8)               CMPR0050
      EQUIVALENCE(DUM(4048),LIT(1)),(DUM(4100),J(1))                    CMPR0060
      DATA(FORM(I),I=1,9)(8H(5HOY = ,8HE20.14, ,8H         ,8H(3H + , ,8HCMPR0070
     1E20.14,3,8HH X(,I2,,8H1H)),    ,8H         ,8HX)       )          CMPR0080
      DATA(BCDA(I),I=1,4)(1H1,1H2,1H3,1H4)                              CMPR0090
      DATA(BCDB(I),I=1,4)(1H0,2H66,2H37,2H 8)                           CMPR0100
      DATA(BCDC(I),I=1,4)(2H88,2H58,2H29,1H1)                           CMPR0110
      DATA(FORM2(I),I=1, 8)(8H(4X,     ,8H         ,8H(2H+ ,E2,8H0.14,3H ,CMPR0120
     18HX(,I2,1H,8H),1X) , ,8H         ,8HX)       )                   CMPR0130
      DATA BLANK(6H       )                                             CMPR0140
      DATA L(9)                                                         CMPR0150
      IF(M4)88,87,88                                                    CMPR0160
   87 LL=1                                                              CMPR0170
      M4=1                                                              CMPR0180
   88 R=IR                                                              CMPR0190
      AN=N                                                              CMPR0200
      RSQUOT=RSSM/R                                                     CMPR0210
      OMR=AN-R-1.                                                       CMPR0220
      NOMR=OMR                                                          CMPR0230
      IF(OMR.EQ.0.0)GO TO 200                                           CMPR0231
      ESQUOT=ESS/OMR                                                    CMPR0240
      FQUOT=RSQUOT/ESQUOT                                               CMPR0250
      GO TO 201                                                         CMPR0251
  200 ESQUOT=0.0                                                        CMPR0252
      FQUOT=9999999999.9999                                             CMPR0253
  201 IRCT=IR+1                                                         CMPR0254
   62 IF(NNNSAV)38,63,64                                                CMPR0270
   63 LL=2                                                              CMPR0280
      K=IRCT+1                                                          CMPR0290
      DO 56 I=2,K                                                       CMPR0300
   56 J(I)=I-1                                                          CMPR0310
   70 WRITE (L,1)PGLB,BLANK                                             CMPR0320
   21 WRITE (L,3)                                                       CMPR0330
   85 WRITE (L,86)                                                      CMPR0340
   22 WRITE (L,4)IR,RSSM,RSQUOT,FQUOT,BLANK                             CMPR0350
      DO 23 I=1,2                                                       CMPR0360
   23 WRITE (L,5)                                                       CMPR0370
      WRITE (L,6)NOMR,ESS,ESQUOT,BLANK                                  CMPR0380
      WRITE (L,7)CORR,BLANK                                             CMPR0390
   84 DO 27 I=1,IRCT,4                                                  CMPR0400
      LAST=I+3                                                          CMPR0410
      IF(LAST-IRCT)51,51,52                                             CMPR0420
   52 LAST=IRCT                                                         CMPR0430
   51 IW=LAST-I+1                                                       CMPR0440
      IF(I-1)53,53,54                                                   CMPR0450
   53 FORM (3)=BCDA(IW-1)                                               CMPR0460
      FORM (8)=BCDB(IW)                                                 CMPR0470
```

```
      WRITE(L,FORM )B(1),(B(K),J(K   ),K=2,LAST)                        CMPR0480
      GO TO 27                                                          CMPR0490
   54 FORM2(2)=BCDA(IW)                                                 CMPR0500
      FORM2(7)=BCDC(IW)                                                 CMPR0510
      WRITE(L,FORM2)(B(K),J(K   ),K=1,LAST)                             CMPR0520
   27 CONTINUE                                                          CMPR0530
   38 RETURN                                                            CMPR0540
   64 I=0                                                               CMPR0550
      CALL FIX                                                          CMPR0560
      DO 101 JK=1,NNNSAV                                                CMPR0570
      IF(LOT(JK))101,100,101                                            CMPR0580
  100 I=I+1                                                             CMPR0590
      J(I)=JK-1                                                         CMPR0600
  101 CONTINUE                                                          CMPR0610
      GO TO (65,66),LL                                                  CMPR0620
   65 WRITE (L,1)PGLB,BLANK                                             CMPR0630
      GO TO 80                                                          CMPR0640
   66 DO 67 I=1,6                                                       CMPR0650
   67 WRITE (L,2)                                                       CMPR0660
   80 WRITE(L,8)SELECT,LIT                                              CMPR0670
      WRITE (L,83)                                                      CMPR0680
      LL=3-LL                                                           CMPR0690
      GO TO 85                                                          CMPR0700
    1 FORMAT(1H1,10A8,39X/119X,A1)                                      CMPR0710
    2 FORMAT(1H ,119X)                                                  CMPR0720
    3 FORMAT(9HOMAIN RUN,111X)                                          CMPR0730
    4 FORMAT(11H REGRESSION,20X,I5,1X,F20.09,1X,F20.09,1X,F20.09,20X,A1)CMPR0740
    5 FORMAT(11HOREGRESSION,109X)                                       CMPR0750
    6 FORMAT(6HOERROR,25X,I5,1X,F20.09,1X,F20.09,41X,A1)                CMPR0760
    7 FORMAT(12HOCORRELATION,4X,F10.9,93X,A1)                           CMPR0770
    8 FORMAT(32HOINDEPENDENT VARIABLE SELECTION ,A8,1X,52A1,27X)        CMPR0780
   83 FORMAT(100X,5H      ,15X)                                         CMPR0790
   86 FORMAT(1HO,33X,2HDF,13X,2HSS,19X,2HMS,19X,1HF,30X)                CMPR0800
      END                                                               CMPR0810
T        SUBTYPE,FORTRAN,LMAP,PBIN                                      FIX  000
      SUBROUTINE FIX                                                    FIX  010
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)           FIX  020
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)FIX  030
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASEFIX  040
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ    FIX  050
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)        FIX  060
      DIMENSION LIT(52)                                                 FIX  070
      EQUIVALENCE(LIT(1),S(1396))                                       FIX  080
      DATA KZERO(1HO),KONE(1H1),KBLANK(1H )                             FIX  090
      DO 3 I=1,NNNSAV                                                   FIX  100
      IF(LOT(I))2,2,1                                                   FIX  110
    1 LIT(I)=KONE                                                       FIX  120
      GO TO 3                                                           FIX  130
    2 LIT(I)=KZERO                                                      FIX  140
    3 CONTINUE                                                          FIX  150
      DO 4 I=NNSAV,52                                                   FIX  160
    4 LIT(I)=KBLANK                                                     FIX  170
      RETURN                                                            FIX  180
      END                                                               FIX  190
T        SUBTYPE,FORTRAN,LMAP,PBIN                                      GAUSS000
      SUBROUTINE GAUSS                                                  GAUSS010
      COMMON A(51,51),IPIVOT(51),B(51,51),YY(7000),X(52),XD(51)         GAUSS020
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)GAUSS030
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,V,NDPO,ICASEGAUSS040
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ    GAUSS050
```

```
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)          GAUSS060
      COMMON IN(49,10),IR,IS,M1,JLIM,NN,Z,NTAPE                          GAUSS070
      COMMON SELECT,IBID,IBIDS                                           GAUSS080
      DIMENSION INDEX(51,2)                                              GAUSS090
C                                                                        GAUSS100
      EQUIVALENCE (N,NNN)                                                GAUSS110
      EQUIVALENCE (YY(1),INDEX(1))                                       GAUSS120
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)          GAUSS130
C                                                                        GAUSS140
C     INITIALIZATION                                                     GAUSS150
C                                                                        GAUSS160
      ERROR=0.0                                                          GAUSS170
      M=1                                                                GAUSS180
   10 DETERM=1.0                                                         GAUSS190
   15 DO 20 J=1,N                                                        GAUSS200
   20 IPIVOT(J)=0                                                        GAUSS210
   30 DO 550 I=1,N                                                       GAUSS220
      IGO=1                                                              GAUSS230
C                                                                        GAUSS240
C     SEARCH FOR PIVOT ELEMENT                                           GAUSS250
C                                                                        GAUSS260
   40 AMAX=0.0                                                           GAUSS270
   45 DO 105 J=1,N                                                       GAUSS280
   50 IF (IPIVOT(J)-1) 60, 105, 60                                       GAUSS290
   60 DO 100 K=1,N                                                       GAUSS300
70    IF (IPIVOT(K)-1) 80,100,899                                        GAUSS310
   80 IF (ABS(AMAX)-ABS(A(J,K))) 85, 100, 100                            GAUSS320
   85 IROW=J                                                             GAUSS330
   90 ICOLUM=K                                                           GAUSS340
   95 AMAX=A(J,K)                                                        GAUSS350
      IGO=2                                                              GAUSS360
  100 CONTINUE                                                           GAUSS370
  105 CONTINUE                                                           GAUSS380
      GO TO(106,110),IGO                                                 GAUSS390
  106 DETERM=0.0                                                         GAUSS400
      GO TO 740                                                          GAUSS410
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1                                    GAUSS420
      IF (A(ICOLUM,ICOLUM))130,899,130                                   GAUSS430
C                                                                        GAUSS440
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL                  GAUSS450
C                                                                        GAUSS460
  130 IF (IROW-ICOLUM) 140, 260, 140                                     GAUSS470
  140 DETERM=-DETERM                                                     GAUSS480
  150 DO 200 L=1,N                                                       GAUSS490
  160 SWAP=A(IROW,L)                                                     GAUSS500
  170 A(IROW,L)=A(ICOLUM,L)                                              GAUSS510
  200 A(ICOLUM,L)=SWAP                                                   GAUSS520
  210 DO 250 L=1, M                                                      GAUSS530
  220 SWAP=B(IROW,L)                                                     GAUSS540
  230 B(IROW,L)=B(ICOLUM,L)                                              GAUSS550
  250 B(ICOLUM,L)=SWAP                                                   GAUSS560
  260 INDEX(I,1)=IROW                                                    GAUSS570
  270 INDEX(I,2)=ICOLUM                                                  GAUSS580
  310 PIVOT    =A(ICOLUM,ICOLUM)                                         GAUSS590
  320 DETERM=DETERM*PIVOT                                                GAUSS600
C                                                                        GAUSS610
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                                  GAUSS620
C                                                                        GAUSS630
  330 A(ICOLUM,ICOLUM)=1.0                                               GAUSS640
  340 DO 350 L=1,N                                                       GAUSS650
```

```
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT                                    GAUSS660
  360 DO 370 L=1,M                                                     GAUSS670
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT                                    GAUSS680
C                                                                      GAUSS690
C     REDUCE NON-PIVOT ROWS                                            GAUSS700
C                                                                      GAUSS710
  380 DO 550 L1=1,N                                                    GAUSS720
  390 IF(L1-ICOLUM) 400, 550, 400                                      GAUSS730
  400 T=A(L1,ICOLUM)                                                   GAUSS740
  420 A(L1,ICOLUM)=0.0                                                 GAUSS750
  430 DO 450 L=1,N                                                     GAUSS760
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T                                    GAUSS770
  460 DO 500 L=1,M                                                     GAUSS780
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T                                    GAUSS790
  550 CONTINUE                                                         GAUSS800
C                                                                      GAUSS810
C     INTERCHANGE COLUMNS                                              GAUSS820
C                                                                      GAUSS830
  600 DO 710 I=1,N                                                     GAUSS840
  610 L=N+1-I                                                          GAUSS850
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                         GAUSS860
  630 JROW=INDEX(L,1)                                                  GAUSS870
  640 JCOLUM=INDEX(L,2)                                                GAUSS880
  650 DO 705 K=1,N                                                     GAUSS890
  660 SWAP=A(K,JROW)                                                   GAUSS900
  670 A(K,JROW)=A(K,JCOLUM)                                            GAUSS910
  700 A(K,JCOLUM)=SWAP                                                 GAUSS920
  705 CONTINUE                                                         GAUSS930
  710 CONTINUE                                                         GAUSS940
  740 RETURN                                                           GAUSS950
899   ERROR=1.0                                                        GAUSS960
      RETURN                                                           GAUSS970
      END                                                              GAUSS980
T        SUBTYPE,FORTRAN,LMAP,PBIN                                     IDENTM00
      SUBROUTINE IDENTM                                                IDENTM01
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)          IDENTM02
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)IDENTM03
      COMMON NNL,DETERM,NOBS,TOL11,TOL12 ,ERROR,NPED,ITOTAL,N,NDPO,ICASEIDENTM04
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ   IDENTM05
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)       IDENTM06
      DIMENSION AIDENT(51,51)                                          IDENTM07
      EQUIVALENCE(IDGO,NOBS)                                           IDENTM08
      EQUIVALENCE(AIDENT(1),YY(1))                                     IDENTM09
    1 DO 3 I=1,NNN                                                     IDENTM10
      DO 4 K=1,NNN                                                     IDENTM11
      SUM=0.0                                                          IDENTM12
      DO 5 J=1,NNN                                                     IDENTM13
    5 SUM=SUM+A(I,J)*AKP(J,K)                                          IDENTM14
      AIDENT(I,K)=SUM                                                  IDENTM15
    4 CONTINUE                                                         IDENTM16
    3 CONTINUE                                                         IDENTM17
      IDGO=1                                                           IDENTM18
      DO 7 I=1,NNN                                                     IDENTM19
      GO TO(16,17),IDGO                                                IDENTM20
   16 IF(ABS(AIDENT(I,1)-1.0)-TOL11)7,8,8                              IDENTM21
    8 IDGO=2                                                           IDENTM22
   17 IF(ABS(AIDENT(I,1)-1.0)-TOL12)7,10,10                            IDENTM23
    7 CONTINUE                                                         IDENTM24
      GO TO(20,220),IDGO                                               IDENTM25
   20 DO 13 I=1,N                                                      IDENTM26
```

```
      K=I+1                                                              IDENTM27
      DO 13 J=K,NNN                                                      IDENTM28
      IF(ABS(AIDENT(I,J))-TOLI)14,13,15                                  IDENTM29
   14 IF(ABS(AIDENT(J,I))-TOLI)13,15,15                                  IDENTM30
   13 CONTINUE                                                           IDENTM31
      GO TO 220                                                          IDENTM32
   10 IDGO=4                                                             IDENTM33
      GO TO 220                                                          IDENTM34
   15 GO TO(18,220),IDGO                                                 IDENTM35
   18 IDGO=3                                                             IDENTM36
  220 RETURN                                                             IDENTM37
      END                                                                IDENTM38
T        SUBTYPE,FORTRAN,LMAP,PBIN                                       IVOR 000
      SUBROUTINE IVOR                                                    IVOR 010
C IVOR - INDEPENDENT VARIABLE SELECTION SUBROUTINE FOR THE ORDERING OF   IVOR 020
C        INDEPENDENT VARIABLES ACCORDING TO MAGNITUDES OF REGRESSION     IVOR 030
C        SUMS OF SQUARES.                                                IVOR 040
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)            IVOR 050
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)IVOR 060
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASEIVOR 070
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ     IVOR 080
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGL3(10)         IVOR 090
      COMMON IN(49,10),IR,IS,M1,JLIM,NN,M,NTAPE                          IVOR 100
      COMMON SELECT,IBID,IBIDS                                           IVOR 110
      DIMENSION LAT(51)                                                  IVOR 120
      EQUIVALENCE (LAT,XD)                                               IVOR 130
      DATA TOLSS(.5E-8)                                                  IVOR 140
      IF(IQ)500,500,501                                                  IVOR 150
  500 IQ=NNSAV-3                                                         IVOR 160
  501 KOUNT=0                                                            IVOR 170
      IGO2=1                                                             IVOR 180
C  SEE NOTE IN BIVOR ON THE USE OF M4.                                   IVOR 190
      M4=0                                                               IVOR 200
      GO TO(1,2),ISKIP                                                   IVOR 210
    1 WRITE(13,103)                                                      IVOR 220
  103 FORMAT(56H0***   I V O R  F I N A L  C O M P R E H E N S I V E ***IVOR 230
     1,64X/120X)                                                         IVOR 240
    2 DO 101 I=2,51                                                      IVOR 250
  101 LOT(I)=1                                                           IVOR 260
      ITOTAL=1                                                           IVOR 270
      LOT(1)=0                                                           IVOR 280
      DO 200 I=1,MI                                                      IVOR 290
      ISTART=ITOTAL+1                                                    IVOR 300
      ITOTAL=ITOTAL+ NJ(I)                                               IVOR 310
  201 NUM=2                                                              IVOR 320
      KASSR=0                                                            IVOR 330
C KASSR COUNTS THE NUMBER OF ASSR-S COMPUTED                             IVOR 340
      DO 300 J=ISTART,ITOTAL                                             IVOR 350
      IF(LOT(J))301,307,301                                              IVOR 360
  301 LOT(J)=0                                                           IVOR 370
C  IN IVOR,ERROR=1.0 MEANS THAT REDUCM WILL NOT PRINT IDENTIFICATION     IVOR 380
C  AND IVS,ERROR=0.0 MEANS PRINT.                                        IVOR 390
      ERROR=1.0                                                          IVOR 400
  302 CALL REDUCM                                                        IVOR 410
    3 CALL CASSR(KASSR,KGO)                                              IVOR 420
  308 KASSR=KASSR                                                        IVOR 430
      GO TO(303,304),KGO                                                 IVOR 440
  303 LAT(KASSR)=J                                                       IVOR 450
  304 LOT(J)=1                                                           IVOR 460
      GO TO 300                                                          IVOR 470
```

```
  307 NUM=NUM+1                                                          IVOR 480
  300 CONTINUE                                                           IVOR 490
      IF(KASSR)221,227,204                                               IVOR 500
  202 IF(NJ(I)-NUM)221,203,201                                           IVOR 510
  203 IF(I-MI)210,220,221                                                IVOR 520
  210 DO 211 L=2,ITOTAL                                                  IVOR 530
  211 LOT(L)=0                                                           IVOR 540
      ERROR=0.0                                                          IVOR 550
      CALL REDUCM                                                        IVOR 560
  209 CALL ABT                                                           IVOR 570
      KOUNT=KOUNT+1                                                      IVOR 580
      IF(IQ-KOUNT)221,220,200                                            IVOR 590
  204 IF(KASSR-1)221,400,401                                             IVOR 600
  400 IXMAX=1                                                            IVOR 610
      GO TO 402                                                          IVOR 620
  401 DO 229 J=2,KASSR                                                   IVOR 630
      IF(ABS((AW(1)-AW(J  ))/AW(1))-TOLSS)229,229,404                    IVOR 640
  229 CONTINUE                                                           IVOR 650
  405 IGO2=2                                                             IVOR 660
      IXMAX=1                                                            IVOR 670
      GO TO 402                                                          IVOR 680
  404 CALL MAXMIN(KASSR,AW,AMAX,AMIN,IXMAX,IXMIN)                        IVOR 690
  402 IMAX=LAT(IXMAX)                                                    IVOR 700
      LOT(IMAX)=0                                                        IVOR 710
      ERROR=0.0                                                          IVOR 720
      CALL REDUCM                                                        IVOR 730
      CALL ABT                                                           IVOR 740
      KOUNT=KOUNT+1                                                      IVOR 750
      IF(IQ-KOUNT)221,302,205                                            IVOR 760
  302 IGO2=IGO2+2                                                        IVOR 770
  205 GO TO(202,408,220,408),IGO2                                        IVOR 780
  200 CONTINUE                                                           IVOR 790
  220 RETURN                                                             IVOR 800
  227 PRINT 228                                                          IVOR 810
  228 FORMAT(30H4NO VALID ASSRS WERE COMPUTED.)                          IVOR 820
      GO TO 220                                                          IVOR 830
  408 PRINT 411,(LOT(I),I=1,NNNSAV)                                      IVOR 840
  411 FORMAT(18H4PERFECT FIT.IVS= ,9I11)                                 IVOR 850
      GO TO 220                                                          IVOR 860
  221 STOP                                                               IVOR 870
      END                                                                IVOR 880
T        SUBTYPE,FORTRAN,LMAP,PBIN                                       MAXMIN00
      SUBROUTINE MAXMIN(N,A,AMAX,AMIN,IXMAX,IXMIN)                       MAXMIN01
      DIMENSION A(N)                                                     MAXMIN02
      AMAX=A(1)                                                          MAXMIN03
      AMIN=AMAX                                                          MAXMIN04
      IXMAX=1                                                            MAXMIN05
      IXMIN=1                                                            MAXMIN06
      IF(N.EQ.1)GO TO 220                                                MAXMIN07
      DO 1 I=2,N                                                         MAXMIN08
      IF(A(I).GE.AMAX)GO TO 2                                            MAXMIN09
      IF(A(I).GT.AMIN)GO TO 1                                            MAXMIN10
      IXMIN=I                                                            MAXMIN11
      AMIN=A(I)                                                          MAXMIN12
      GO TO 1                                                            MAXMIN13
    2 AMAX=A(I)                                                          MAXMIN14
      IXMAX=I                                                            MAXMIN15
    1 CONTINUE                                                           MAXMIN16
  220 RETURN                                                             MAXMIN17
      END                                                                MAXMIN18
```

```
T         SUBTYPE,FORTRAN,LMAP,PBIN                                          PREVAR00
        SUBROUTINE PREVAR(KOUNT,INDX)                                        PREVAR01
        COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)              PREVAR02
        COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)PREVAR03
        COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NOPO,ICASEPREVAR04
        COMMON RSSMC,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IG       PREVAR05
        COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGL3(10)           PREVAR06
        COMMON IN(49,10),IR,IS,MI,JLIM,NN,M,NTAPE                            PREVAR07
        COMMON SELECT,IBID,IBIDS                                             PREVAR08
        DIMENSION XX(51)                                                     PREVAR09
        EQUIVALENCE(XX(1),B(1549))                                           PREVAR10
        KOUNT=KOUNT+1                                                        PREVAR11
        IF(NNNSAV)650,66,650                                                 PREVAR12
  650   JJJ=1                                                                PREVAR13
        DO 652 JJ=2,NNXA                                                     PREVAR14
        IF(LOT(JJ))104,651,652                                               PREVAR15
  651   JJJ=JJJ+1                                                            PREVAR16
        X(JJJ)=X(JJ)                                                         PREVAR17
        AW(JJJ)=AVV(JJ)                                                      PREVAR18
        IF(JJ-INDX)654,654,652                                               PREVAR19
  654   J=JJJ                                                                PREVAR20
  653   XX(JJJ)=X(JJ)                                                        PREVAR21
  652   CONTINUE                                                             PREVAR22
        PRINT 70,KOUNT,(XX(I),I=2,J)                                         PREVAR23
   70   FORMAT(2H (,I3,1H), 9(1X,E12.6)/(5X,9(1X,E12.6)))                    PREVAR24
        DO 468 I=2,NNN                                                       PREVAR25
  468   XD(I)=X(I)-AW(I)                                                     PREVAR26
        GO TO 1066                                                           PREVAR27
   66   DO 68 I=2,NNN                                                        PREVAR28
   68   XD(I) = X(I)-AVV(I)                                                  PREVAR29
        PRINT 70,KOUNT,(X(I),I=2,INDX)                                       PREVAR30
 1066   YY(KOUNT)=B(1)                                                       PREVAR31
        DO 67 I=2, NNN                                                       PREVAR32
   67   YY(KOUNT)=YY(KOUNT)+X(I)*B(I)                                        PREVAR33
        TEM XX =0.0                                                          PREVAR34
        DO 81 I=2,NNN                                                        PREVAR35
        DO 81 J=2,NNN                                                        PREVAR36
   81   TEMXX=TEMXX+ A(I,J)*XD(I)*XD(J)                                      PREVAR37
        IF(MVPL)812,811,812                                                  PREVAR38
  811   YSDEV(KOUNT)=SDEV*SQRT(1.0+RECM*TEMXX)                               PREVAR39
        GO TO 80                                                             PREVAR40
  812   YSDEV(KOUNT)=SDEV*SQRT(RECM*TEMXX)                                   PREVAR41
   80   RETURN                                                               PREVAR42
  104   STOP                                                                 PREVAR43
        END                                                                  PREVAR44
T         SUBTYPE,FORTRAN,LMAP,PBIN                                          PRINTM00
        SUBROUTINE PRINTM                                                    PRINTM01
        COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)              PRINTM02
        COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)PRINTM03
        COMMON NNL,DETERM,NOBS,TOL11,TOL12,ERROR,NPED,ITOTAL,N,NOPO,ICASEPRINTM04
        COMMON RSSMC,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IG       PRINTM05
        COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGL3(10)           PRINTM06
        DIMENSION AIDENT(51,51)                                              PRINTM07
        EQUIVALENCE(IDGO,NOBS)                                               PRINTM08
        EQUIVALENCE(AIDENT(1),YY(1))                                         PRINTM09
        GO TO(1,2,2,2),IDGO                                                  PRINTM10
    2   PRINT 3                                                              PRINTM11
    3   FORMAT(16H0IDENTITY MATRIX)                                          PRINTM12
        DO 4 I=1,NNN                                                         PRINTM13
    4   PRINT 5,(AIDENT(I,K),K=1,NNN)                                        PRINTM14
```

```
    5 FORMAT(1H0,7E17.8/(1X,7E17.8))                                    PRINTM15
      GO TO(1,7,8,9),IDGO                                               PRINTM16
    1 PRINT 6,TOLI1                                                     PRINTM17
    6 FORMAT(70H0DEVIATIONS OF ALL ELEMENTS OF THE IDENTITY MATRIX SMALLPRINTM18
     1ER THAN I(1)= ,G9.15H ,RUN ACCEPTED.)                             PRINTM19
      GO TO 220                                                         PRINTM20
    7 PRINT 10,TOLI1,TOLI2                                              PRINTM21
   10 FORMAT(78H0DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MAPRINTM22
     1TRIX LARGER THAN I(1)= ,G9.   21H BUT LESS THAN I(2)= ,G9.15H ,RUNPRINTM23
     2 ACCEPTED.)                                                       PRINTM24
      GO TO 220                                                         PRINTM25
    8 PRINT 11,TOLI1                                                    PRINTM26
   11 FORMAT(84H0DEVIATIONS OF ALL MAIN DIAGONAL ELEMENTS IN THE IDENTITPRINTM27
     1Y MATRIX SMALLER THAN I(1)= ,G9/68H DEVIATION OF AN OFF-DIAGONAL EPRINTM28
     2LEMENT LARGER THAN I(1),RUN ACCEPTED.)                            PRINTM29
      GO TO 220                                                         PRINTM30
    9 PRINT 12,TOLI2                                                    PRINTM31
   12 FORMAT(79H0DEVIATION OF A MAIN DIAGONAL ELEMENT IN THE IDENTITY MAPRINTM32
     1TRIX LARGER THAN I(2)= ,G9.15H ,RUN REJECTED.)                    PRINTM33
  220 RETURN                                                            PRINTM34
      END                                                               PRINTM35
T        SUBTYPE,FORTRAN,LMAP,PBIN                                      RDISK 00
      SUBROUTINE RDISK(KOUNT,INDX)                                      RDISK 01
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)           RDISK 02
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)RDISK 03
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASERDISK 04
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ    RDISK 05
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)        RDISK 06
      COMMON IN(49,10),IR,IS,M1,JLIM,NN,M,NTAPE                         RDISK 07
      COMMON SELECT,IBID,IBIDS                                          RDISK 08
      DIMENSION IKEEPR(999)                                             RDISK 09
      EQUIVALENCE (B(1502),IKEEPR(1))                                   RDISK 10
      REWIND 10                                                         RDISK 11
      ISTART=1                                                          RDISK 12
      DO 1 I=1,NDR                                                      RDISK 13
      IWHICH=IKEEPR(I)                                                  RDISK 14
      NUMBER=IWHICH-ISTART                                              RDISK 15
      IF(NUMBER)2,3,4                                                   RDISK 16
    4 DO 11 J=1,NUMBER                                                  RDISK 17
   11 READ (10) SKIP                                                    RDISK 18
    3 READ(10)(X(K),K=2,NNSAV)                                          RDISK 19
      CALL PREVAR(KOUNT,INDX)                                           RDISK 20
      ISTART=IWHICH+1                                                   RDISK 21
    1 CONTINUE                                                          RDISK 22
      GO TO 5                                                           RDISK 23
    2 STOP                                                              RDISK 24
    5 RETURN                                                            RDISK 25
      END                                                               RDISK 26
T        SUBTYPE,FORTRAN,LMAP,PBIN                                      RDIT  00
      SUBROUTINE RDIT                                                   RDIT  01
C  RDIT-A PROGRAM TO READ TAPE OR CARDS AND COMPUTE HIGHER ORDER        RDIT  02
C PRODUCT TERMS OF THE DATA.                                            RDIT  03
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)           RDIT  04
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)RDIT  05
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASERDIT  06
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,NQ(25),IQ    RDIT  07
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)        RDIT  08
      COMMON IN(49,10),IR,IS,M1,JLIM,NN,M, TAPE                         RDIT  09
      COMMON SELECT,IBID,IBIDS                                          RDIT  10
      DIMENSION Y(52)
```

```
      EQUIVALENCE(Y(1),B(53))
      EQUIVALENCE(LIM,NNXA)                                              RDIT  11
      EQUIVALENCE(KNUM,NUM),(KMUM,MUM)                                   RDIT  12
      INTEGER TAPE                                                       RDIT  13
    3 DO 33 J=1,JLIM,NUM                                                 RDIT  14
      J1=J+MUM                                                           RDIT  15
      IF(JLIM-J1)11,10,10                                                RDIT  16
   11 J1=JLIM                                                            RDIT  17
   10 IF (J-1)8,8,9                                                      RDIT  18
    8 READ(TAPE,FIRM)M1,(Y(J2),J2=J,J1)                                  RDIT  19
      IF(M1)2,33,2                                                       RDIT  20
    9 READ(TAPE,FIRM)MZ,(Y(J2),J2=J,J1)                                  RDIT  21
   33 CONTINUE                                                           RDIT  22
      X(NN)=Y(1)                                                         RDIT  23
      Y(1)=1.                                                            RDIT  24
    4 M=M+1                                                              RDIT  25
      IF(IS)200,200,100                                                  RDIT  26
  100 DO 5 K=1,IS                                                        RDIT  27
      KK=IR+K+1                                                          RDIT  28
      Y(KK)=1.                                                           RDIT  29
      DO 5 L=1,10                                                        RDIT  30
      INDEX=IN(K,L)                                                      RDIT  31
    5 Y(KK)=Y(KK)*Y(INDEX)                                               RDIT  32
  200 DO 6 J=2,LIM                                                       RDIT  33
    6 X(J)=Y(J)                                                          RDIT  34
    2 RETURN                                                             RDIT  35
      END                                                                RDIT  36
T        SUBTYPE,FORTRAN,LMAP,PBIN                                       REDUCM00
      SUBROUTINE REDUCM                                                  REDUCM01
      COMMON A(51,51),BSDEV(51),B(2601),YY(7000),X(52),XD(51)            REDUCM02
      COMMON AVV(52),YSDEV(7000),AW(51),RECM,NDR,MVPL,NNNSAV,NNN,LOT(51)REDUCM03
      COMMON NNL,DETERM,NOBS,TOLRS,TOLCES,ERROR,NPED,ITOTAL,N,NDPO,ICASEREDUCM04
      COMMON RSSMO,ISKIP,NJ(25),M4,FIRM(7),KNUM,KMUM,MB,MI,VQ(25),IQ     REDUCM05
      COMMON NNXA,NNSAV,SDEV,AKP(51,51),BB(52),S(52,52),PGLB(10)         REDUCM06
      COMMON IN(49,10),IR,IS,M1,JLIM,NN,M,NTAPE                          REDUCM07
      COMMON SELECT,IBID,IBIDS                                           REDUCM08
      EQUIVALENCE(LI,NNN)                                                REDUCM09
      LI=0                                                               REDUCM10
      DO 95 I=1,ITOTAL                                                   REDUCM11
      IF(LOT(I))95,91,95                                                 REDUCM12
   91 LI=LI+1                                                            REDUCM13
      B(LI)=S(I,NNL)                                                     REDUCM14
      BB(LI)=S(I,NNL)                                                    REDUCM15
      J=LI-1                                                             REDUCM16
      DO 200 L=I,ITOTAL                                                  REDUCM17
      IF(LOT(L))200,203,200                                              REDUCM18
  203 J=J+1                                                              REDUCM19
      AKP(J,LI)=S(I,L)                                                   REDUCM20
      AKP(LI,J)=S(I,L)                                                   REDUCM21
      A(J,LI)=S(I,L)                                                     REDUCM22
      A(LI,J)=S(I,L)                                                     REDUCM23
  200 CONTINUE                                                           REDUCM24
   95 CONTINUE                                                           REDUCM25
      N=LI-1                                                             REDUCM26
      IF(ERROR)219,219,220                                               REDUCM27
  219 PRINT 594,PGLB                                                     REDUCM28
  594 FORMAT(1H1,10A8)                                                   REDUCM29
      PRINT 5760,SELECT,(LOT(I),I=1,NNNSAV)                              REDUCM30
 5760 FORMAT(32H0INDEPENDENT VARIABLE SELECTION ,A8,1X,51I1)             REDUCM31
      ICASE=ICASE+1                                                      REDUCM32
```

```
220 RETURN                                                    REDUCM33
    END                                                       REDUCM34
```

IX. REFERENCES

1. Abt, K. [1965], Principles for Ranking Independent Variables by Order of Importance in the General Linear Hypothesis Model, Paper presented at the Meeting of the Institute of Mathematical Statistics, April 29-May 1, 1965, Tallahassee, Florida.

2. Anderson, R. L. and Bancroft, T. A. [1952], Statistical Theory in Research, McGraw-Hill Book Co., Inc., New York.

3. Brownlee, K. A. [1960], Statistical Theory and Methodology in Science and Engineering, John Wiley and Sons, Inc., New York.

4. Cohen, C. J. [1959], Inverting Symmetric Definite Matrices by Gauss' Method with the Largest Element for Pivot Element, Unpublished Note, U. S. Naval Weapons Laboratory, Dahlgren, Virginia.

5. Duncan, A. J. [1959], Quality Control and Industrial Statistics, Richard D. Irwin, Inc., Homewood, Illinois.

6. Efroymson, M. A. [1960], Multiple Regression Analysis. In: Mathematical Methods for Digital Computers, (Ralston, A., and Wilf, H. S., Ed.) John Wiley and Sons, Inc., New York.

7. Gorman, J. W. and Toman, R. J. [1966], Selection of Variables for Fitting Equations to Data, Technometrics, 8, 27-51.

8. Graybill, F. A. [1961], An Introduction to Linear Statistical Models, Volume I, McGraw-Hill Book Co., Inc., New York.

9. Hader, R. J. and Grandage, A. H. E. [1958], Simple and Multiple Regression Analyses. In: Experimental Designs in Industry, (Chew, V., Ed.) John Wiley and Sons, Inc., New York.

10. Hamaker, H. C. [1962], On Multiple Regression Analysis, Statistica Neerlandica, 16, 31-56.

11. Herring, T. L. [1966], MTRAN - A Stretch Computer Program to Preprocess Data for the DA-MRCA Multiple Regression Program, Technical Memorandum No. K-40/66. U. S. Naval Weapons Laboratory Dahlgren, Virginia.

12. Scheffé, H. [1959], The Analysis of Variance, John Wiley and Sons, Inc., New York.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Weapons Laboratory	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b GROUP

3 REPORT TITLE
DA-MRCA: A FORTRAN IV PROGRAM FOR MULTIPLE LINEAR REGRESSION

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (Last name, first name, initial)
Abt, K., Gemmill, G., Herring, T., and Shade, R.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1966	207	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	TR-2035
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report contains the documentation of a multiple linear regression program for up to 50 independent variables, written in FORTRAN IV for the IBM 7030 (STRETCH) computer. The program incorporates part of the results obtained from an effort to explore the present limitations of high speed computation in the area of linear statistical models. DA-MRCA includes options for both forward and backward automatic ranking of the independent variables by order of prediction power for the dependent variable. The report contains the description of these options, along with an outline of the applicability of the program which includes, in a convenient form, non-orthogonal analysis of variance. Justifications are given for extensive checks made on the accuracy of the matrix inversions. The resulting internal decisions and their effects on the computational flow are described in detail. Also, a failure analysis is given in which causes for failures to obtain acceptable inverses and possible consequences of corrective measures are discussed.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification