Technical Report No. 34

CURRENT GENERALIZED COMPUTER PROGRAMS USED IN GRASSLAND BIOME ANALYSES

Coordinated by D. M. Swift

Contributors:

C. V. Baker
L. J. Bledsoe
J. D. Gustafson
R. D. Robinson

GRASSLANDS BIOME

U. S. International Biological Program

January 1970

INTRODUCTION

This report deals with the functioning programs which are available at the Natural Resource Ecology Laboratory at CSU and have been useful in the IBP Grasslands Biome activities. These include generalized statistical programs as well as generalized mathematical programs such as routines for solving differential equations.

All programs are described in general terms so that investigators not familiar with programming may evaluate their utility to various types of data. Those programs which are mot frequently used or which are deemed most useful to the field investigator or the modeler are discussed in some detail. The reader will note that most of the programs can be set up and run very simply by merely adding a few control cards and a data deck to the source deck. Others, however, require a user-supplied subroutine, which presupposes that the potential user have a knowledge of FORTRAN programming. FORTRAN listings are included for these programs unless such listings are readily obtainable from some other source. Programs less frequently used or of less general interest are discussed briefly in Appendix 1.

REGRES1

INTRODUCTION

REGRES1 is a general multiple regression and correlation program which can handle up to 60 variables with 9999 observations per variable. It is written in FORTRAN extended and operates on both the SCOPE and NCAR compilers. It provides summary statistics on the input variables as well as correlation and regression analyses. Deviations from the model are calculated. Options within the program allow reading data from cards or tape, variable format or standard format, names for variables generated or names read in, a variety of transgenerations, regression through the origin or not through the origin, and weighted or nonweighted regression.

This is a fairly quick program which can be used for multiple correlation and regressions in cases where screening of variables is not required.

Information in this report was drawn liberally from ORNL-TM-1288, REGRESS - *A multiple regression and correlation program with graphic output for model evaluation*, George M. Van Dyne, published by Oak Ridge National Laboratory.

HOW TO USE THE PROGRAM

The sequence of input is given below. Systems cards vary from computer center to computer center and are not described herein, except that the program as compiled here requires a standard input tape, a standard output tape for printing, two scratch tapes for intermediate calculations and storage, and an output tape for the Calcomp plotter. Three tapes, logical units 2, 3, and 4, ne ' to be specified on the job card. The "Glorius" dump system is utilized, thus requiring one of the standard scratch tapes.

Sequence of Input to REGRESS1

	REGRESS1
Program	System cards are required before and after the program deck
Control card	Always required
Name card(s) Variable format card(s) Transgeneration card(s) Constant card(s)	Each of these may or may not be required as specified by the control card
Data deck	Data may be read from cards or from tape 4 as specified by the control card
Plot dimension card(s)	Required only if type 9 plots are specified on the control card
Control card . . .	Repeat as above following control card for as many sets of data as desired
Blank card	After last data deck

The control card is described below. The control card always must be read by REGRESS1 but the actual information required may be minimal. For example, if a standard input format is used, if no transgenerations are required, if no plots are to be made, and if the data is to be unweighted and not through the origin, then all that is required is the number of observations (NOBS) and the number of variables before transgeneration (N). If the number of variables after transgeneration is left blank, it is assumed that the number of variables after transgeneration is the same as the number before transgeneration (NA = N).

A variable can be an independent variable, a dependent variable, or a weighting factor. If no transgenerations are to be made the dependent variable must be the right-most one which is read.

The control card, the input format, the transgenerations, and the constants are read and are printed on the output for purposes of checking the results.

If variable names are read in, they should be read in order of the variables after transgeneration rather than before transgeneration. These variable names are listed in the printed and graphic output and may be checked readily. If no variable names are called for, then numbers 1 through 60 are generated for variable names.

Variable formats are allowed for the input by making NCARDS $\geq$ 1 thereby specifying the number of cards to be read containing the variable format. For example, if NCARDS = 1 and the following format is used

(3F5.1,21X,F7.2)

then four variables would be read per input card. The left parenthesis should be in column 1 of the format card.

Description of the control card for REGRESS1

Columns in cards	Designation on printed output	Designation in program	Meaning and coded values
1-2	Printed statement	NTAPE	Controls the source of the data: < 0 means read data from tape 4; 0 means read the data from cards; and > 0 means read the data from cards and write it on tape 4. If NTAPE > 0 then NTAPE should be the number of cards per data point (see Appendix example).
3-6	OBS	NOBS	Number of data points (2 $\leq$ NOBS $\leq$ 9999).
7, 8	NBG	N	Number of variables before transgenerati (2 $\leq$ N $\leq$ 61).
9, 10	NAF	NA	Number of variables after transgeneratio (2 $\leq$ NA $\leq$ 60).
12	WGT	IWT	< 1 do not weight data; $\geq$ 1 weight data.

14	PLT	IPLOT	$\leq$ 1 do not plot = 2 plot deviations against variables = 3 plot independent variables against each other = 4 plot dependent variables against independent variables = 5 do as 2 + 3 = 6 do as 2 + 4 = 7 do as 3 + 4 = 8 do as 2 + 3 + 4 $\geq$ 9 plot each independent variable with the dependent variable which is adjusted for the variation of all other independent variables from their means (see text for description of plot dimension card).
16	ORG	IORIGIN	$\leq$ 1 not through the origin $\geq$ 2 through the origin
18	NAM	INAME	= 1 generate numbers for names = 2 read names of variables
20	FMT	NCARDS	= 0 read data according to standard format = 1, 2, ..., 9 number of variable format card to read
21, 22	TRN	NTRAN	= 0 no transgenerations = 1, 2, ..., 99 number of transgeneration codes to read (up to 10 codes per card)
23,24	CON	NCON	= 0 no constants = 1, 2, ..., 99 number of constants to be read (up to 8 constants per card)
25-80	IDENTIFICATION	ID	Alphanumeric identification of the problem

If NCARDS $\leq$ 0 then the data are read according to a standard format of

10X,10F7.0

"Transgenerations," meaning transformations of variables and generation of new variables, are described below. Each transgeneration is defined by an 8 digit code and up to 10 transgeneration codes are read in on a card. The first 2 digits of the transgeneration code define the type of transgeneration which is to be made, the second two digits define the index of the variable after transgeneration, the next two digits define the index of the variable before transgeneration, and the last two digits, if required, define the index of a second variable involved in the transgeneration or the index of the constant involved.

An example transgeneration is as follows:

05030102

This means to use transgeneration 5 and make variable 3 from variable 1 times variable 2. Caution must be taken not to eliminate a variable while in the process of transgenerations.

Constants are read with an F10.0 format, thus allowing up to 8 constants per card.

If weighted regression is to be run it is the user's responsibility to transfer the weight to variable 61. For example, if the weight was the 6th of 6 variables read in, the transgeneration would be as follows:

01610600

If weighted regression is not run a 1 is automatically placed in variable 61 which is used as the weighting factor. If the regression is not weighted the output will show that the number of data equals the sum of weights. The number of variables printed on the output should be equal to the number of variables after transgenerations.

Transgeneration codes (n, i, j, k)*

Code	Meaning
01	$X_i = X_j$
02	$= X_j \times X_k$
03	$= X_j \div X_k$
04	$= X_j + X_k$
05	$= X_j - X_k$
06	$= \frac{1}{X_j}$
07	$= X_j + C_k$
08	$= X_j \div C_k$
09	$= C_k \div X_j$
10	$= X_j \times C_k$
11	$= X_j^{C_k}$
12	$= \mathrm{Log}_e X_j$
13	$= \mathrm{Log}_{10} X_j$
14	$= e^{X_j}$
15	$= \sin X_j$
16	$= \arctan X_j$
17	$= \sin (X_j - C_k) + 1$
18	$X_i = \max(X_j, X_k)$

* where n = transformation code

i = index of the variable after transgeneration

j = index of the variable before transgeneration

k = index of a second variable or constant used in the transgeneration

OUTPUT

The sequence of output is constant and generally is self explanatory (see Appendices II-IV). The upper triangular portions of the sums of squares and cross-product matrix and the correlation matrices are given in the output. To be comparable to the standard notation of writing an upper triangular matrix, each row of the output should be adjusted to the right as shown in the following hypothetical example:

Program Output Form			
1.00	.26	.35	.78
1.00	.75	.61	
1.00	.41		
1.00			

Conventional Form			
1.00	.26	.35	.78
	1.00	.75	.61
		1.00	.41
			1.00

PROGRAM OPERATION

Most of the routines used herein are found in regression programs and in statistical textbooks. A complete 'well commented" listing of REGRESS1 is given in Appendix I, therefore, only special features of the program will be discussed.

The structure of the program and the subroutines and sequence of use of subroutines is as follows:

```
                    REGRESS

          INVERT                 TRANSFRM
```

TRANSFRM is a subroutine which will make the required transgenerations. INVERT is a subroutine which is called to invert correlation matrices.

Some important steps in the input sequence are as follows:

1. After clearing certain storage areas and setting certain counters, the control card is read and the number of variables before transgeneration

(N) is evaluated. If $N \leq 1$, control is transferred to the exit to return to the monitor. A message is printed indicating the END OF RUN. If $N \geq 2$ the control card values are printed.

2. If name cards are not to be read then NAME (I) is set equal to I for all I. Otherwise names are read in a 10A8 format.

3. If a variable format card(s) is (are) not needed the data are read according to a standard format of 10X,10F7.0. Otherwise the variable data input format, IFMT, is read.

4. If transgenerations are to be made the necessary transgeneration codes are read and printed. Then if constants are to be read, they are read and printed.

5. The data are read, according to the format (IFMT) defined above, in a list of X(I) with I = 1,...N.

6. Data normally are read from cards (NTAPE = 0), but they may be read from tape 4 (NTAPE $\geq$ 1) or they may be read from cards and written onto tape 4 (NTAPE $\leq$ -1). If data are read from cards and written onto tape 4 or if they are read from tape 4, then NTAPE should be the number of card images (80 column units) to be read per observation. If transgenerations are to be made, the transgeneration subroutine (TRANSFRM) is called to make them. Calling the transgeneration subroutine involves giving a list of X's, the transgeneration codes, and constants (if required). If the data are not to be weighted then X(61) is automatically set equal 1. Otherwise, a transgeneration must be made to set some given X into X(61).

7. After the transgenerations have been made the list of X's is searched and compared to he values stored as the maximum and minimum values for each X(I). If a given X(I) is larger or smaller than the one previously stored that value is replaced.

8. The count of data points is then incremented by 1 and the string of X's is written on scratch tape (tape 2), and the weighting factor is added to the sum of weights. The weighted sums, sums of squares, and sums of cross-products are accumulated while reading the data in.

The above procedures are repeated until all the observations, NOBS, have been read.

The ranges and corrected sums of squares are then calculated. The corrected sums of squares defined as XX(I) and XY(I), and the total sum of squares, TOTSS, are set aside for later use. The coefficients of variation, CV(J), standard errors, SE(J), and standard deviations, X(J), of the variables are then calculated. These statistics are printed in table form and are identified by either the generated name-numbers of the alphabetic names which were read in (see Appendices II-IV).

Calculations proceed with formation of a matrix of simple correlation coefficients which is then printed. This simple correlation matrix (composed of all independent variables plus the dependent variable) is inverted by a Gauss-Jordan elimination method (subroutine INVERT) and the inverse of the matrix is printed.

A comparison of the computational scheme used in this program with other commonly used schemes is given in Table 1. An essential point is that this program inverts the complete or augumented matrix of correlation coefficients rather than the predictor matrix of correlation coefficients. The highest order partial correlation coefficients among the variables are calculated according to the method in Table and then are printed. A check is made to see if the regression is to be through the origin. If not, (IORIGIN $\leq$ 0), the regression coefficients are calculated by the method described in

Table 1. If the regression is through the origin (IORIGIN $\geq$ 1) then a matrix of "pseudo" correlation coefficients is formed and inverted before the regression coefficients are calculated.

The standard error of estimate, SY, the square of the multiple correlation coefficient, RSQ, the standard partial regression coefficients, BPRIME(J), the partial regression coefficients, B(J), the standard errors of the partial regression coefficients, S(J), and "t" tests of the regression coefficient, T(J), are calculated next.

If the regression is not through the origin the constant term (BO) is set equal to XBAR(N) and the regression coefficients are multiplied by the mean values for each independent variable and summed for all independent variables. This value is subtracted from BO to give the intercept value. The "t" test of the intercept is calculated as the value of the intercept divided by the standard error of estimate. If the regression is to be through the origin, the intercept value is set equal to 0 and the "t" test for the intercept is also set equal to 0.

Values are predicted for each data point (using the regression equation) from the independent variables which are read in from a scratch tape. This predicted value is compared to the observed value for the independent variable to obtain the deviation. The deviation is checked against the maximum and minimum deviations that have been obtained up to that point and these are replaced if necessary. The data are again written onto a scratch tape (to save the input variables as remaining after transgeneration), along with the predicted value for the independent variable, and the deviation.

Regression coefficients are printed with their standard errors, "t" values, and simple and partial correlations. A test statistic for "outliers" is calculated. This is the absolute difference of the maximum and minimum deviations from the model divided by the standard error of the estimate.

The analysis of variance is printed and in the process the analysis of variance for each variable (as if it were a simple linear regression) is calculated.

Sample Run of Program

The sample is a regression of streamflow in the Cache La Poudre River on various snow course measurements for 15 years of record. As they appear on the data cards; item 1 is water equivalent, Cameron Pass snow course, April, in inches (name-CPA); item 2 is water equivalent, Cameron Pass snow course, May, in inches (name-CPM); item 3 is water equivalent, Big South snow course, April, in inches (name-BSA); and item 4 is stream flow, April through September, in thousands of acre feet (name-POUDRE). Before the regression was performed, item 1 (CPA) and item 2 (CPM) were transgenerated into a single variable (CPAM) such that CPAM = CPA + 2 (CPM). The regression then had two independent variables (CPAM and BSA) and the dependent (POUDRE). A listing of the program control cards and the data deck from this run is attached, along with the resultant output.

Table 1. A comparison of procedures used in multiple regression analysis[1,2]

	Symbol	This Program	Alternatives		
Special definitions	-	-	-	$c_{ij} = \frac{p_{ij}^{-1}\cdot r_{ij}}{\Sigma x_i x_j}$	
Matrix to be inverted		$\underset{\sim}{R}$	$\underset{\sim}{A}$	$\underset{\sim}{P}$	$\underset{\sim}{S}$
Standard partial regression coefficients	b_j^*	$\frac{-r_{jy}^{-1}}{r_{yy}^{-1}}$	$b_j\sqrt{\frac{\Sigma x_j^2}{\Sigma y^2}}$	$\sqrt{b_j \frac{\Sigma x_j^2}{\Sigma y^2}}$	$\frac{-s_{jy}^{-1}}{s_{yy}^{-1}}$
Partial regression coefficients	b_j	$b_j^*\sqrt{\frac{\Sigma y^2}{\Sigma x_j^2}}$	$\sum_i (a_{ji}^{-1}\cdot\Sigma x_i y)$	$\sum_i (c_{ij}\Sigma x_i y)$	$b_j^*\sqrt{\frac{\Sigma y^2}{\Sigma x_j^2}}$
Standard errors of regression coefficients	s_{b_i}	$S_y\sqrt{\frac{r_{jy}^{-1}\cdot r_{yy}^{-1}-(r_{jy}^{-1})^2}{\Sigma x_j^2\cdot r_{yy}^{-1}}}$	$S_y\cdot\sqrt{a_{jj}^{-1}}$	$S_y\sqrt{c_{ii}}$	$\sqrt{\frac{s_{yy}^{-1}\cdot s_{jj}^{-1}-(s_{yj}^{-1})^2}{(N-k)\cdot s_{yy}^{-1}}}$
Standard error of estimate	S_y	$\sqrt{\frac{\Sigma y^2}{r_{yy}^{-1}\cdot(N-k-1)}}$	$\sqrt{\frac{\Sigma y_i^2-\sum_j b_j\cdot(\Sigma x_j y)}{N-k-1}}$	$\sqrt{\frac{\Sigma y_i^2-R_m^2\cdot\Sigma y_i^2}{N-k-1}}$	$\sqrt{\frac{1}{N\cdot(N-k)\cdot s_{yy}^{-1}}}$
Multiple correlation coefficient	R_m	$\sqrt{1-\frac{1}{r_{yy}^{-1}}}$		$\sqrt{\frac{\sum_j(b_j\cdot\Sigma x_j y)}{\Sigma y^2}}$	$\sqrt{1-\frac{1}{s_{yy}^{-1}\cdot\Sigma y_i^2}}$
Partial correlation coefficients	r_{ij}^*	$\frac{-r_{ij}^{-1}}{\sqrt{r_{ii}^{-1}\cdot r_{jj}^{-1}}}$			

[1] Derived from Goulden (1952), Kelley (1947), Efromyson (1960, Steel and Torrie (1960), Ostle (1963), and Friedman and Foote (1955).

[2] Let: Σx_i^2 be the corrected sum of squares for variable X_i; $\Sigma x_i y$ be the corrected sum of cross-products for $X_i Y$; $\underset{\sim}{R}$ be a matrix of simple linear correlation coefficients r_{ij} composed of independent variables and the dependent variable (augumented) where r_{iy} is a correlation of an X_i with Y; inverse elements be denoted by r_{ij}^{-1}; $\underset{\sim}{P}$ be a matrix of simple linear correlation coefficients among only the independent variables (predictors); $\underset{\sim}{S}$ be a complete moment matrix of sums of squares and cross products composed of elements Σx_i^2, $\Sigma x_i x_j$, $\Sigma x_i y$, Σy^2, and $\underset{\sim}{A}$ be a matrix of sums of squares and cross products only of Σx_i^2 and $\Sigma x_i x_j$ with elements $i=1,\ldots k$ independent variables; and N be the number of observations or data points.

```
      PROGRAM REGRES1
     1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2=2028,TAPE3=2028,TAPE4
     2=2028)
C         TAPE 57 IS USED FOR THE  GLORIUS  DUMP PROCEDURE
C      NO GDUMP UNTIL ROUTINE IS SUPPLIED

C=========================================================================

      DIMENSION X(61), SUMX(60), XBAR(60), DMAX(60), DMIN(60), RNG(60)

C  SEPTEMBER 1965 VERSION  --  GEORGE M. VAN DYNE
C      REVISED 4/6/67 FOR CDC 6400, L J BLEDSOE
C      REVISED 9/29/69 FOR SCOPE, DAVID M. SWIFT
C         PLOT ON TAPE 9
C      NO PLOTS UNTIL SUBROUTINES ARE REVISED
C         SCRATCH TAPES 3 AND 2 (2=56)
C         WRITE DATA ON OR READ DATA FROM TAPE 4
      DIMENSION SIGMAX(60), SE(60), CV(60), SAVER(60), BPRIME(60), B(60)
      DIMENSION S(60), T(60), XX(60), XY(60), FSS(60)
      DIMENSION IFMT(90), NAME(60), ID(7), CARD(100), XLABEL(4)
      DIMENSION NT(99), NI(99), NJ(99), NK(99), C(99), ZZ(14)
      DIMENSION A(60,60), R(60,60), RPRIME(60,60), UNCOR(60,60)
      DIMENSION V1(3600), V2(3600), V3(3600)
      COMMON V1,V2,V3

      EQUIVALENCE (UNCOR,V1),(A,V2),(R,RPRIME,V3)

C=========================================================================
C  BEGIN PROGRAM BY SETTING UP NECESSARY COUNTERS AND CLEARING STORAGE AREAS
    1 REWIND 3 $ REWIND 4
      DO 4444 I=1,4
 4444 XLABEL(I)=0.
      DO 4445 I=1,14
 4445 ZZ(I)=0.
      WT=NTOTAL=DEVMIN=DEVMAX=SMALL=I=J=NTAPE=NOBS=N=NA=IWT=IPLOT=0
      IORIGN=INAME=NCARDS=NTRAN=NCON=JVAR=MTAPE=KKK=W=NN=TOTSS=PHI=DIV=0
      XN=SY=RSQ=RMULT=RDETR=L=B0=DEVSQ=II=YPRED=JJ=DEV=SSREG=SMREG=0
      SMDEV=DFTOT=F=ICOUNT=NPLOTS=LAST=0
      DO 222 I=1,60   $   DMAX(I)=-99.E99   $   DMIN(I)=+99.E99
      SIGMAX(I)=SE(I)=XBAR(I)=X(I)=S(I)=CV(I)=T(I)=SUMX(I)=B(I)=0.0
      BPRIME(I)=XX(I)=FSS(I)=XY(I)=SAVER(I)=0.0
      RNG(I)=FSS(I)=0.0
      DO 222 J=1,60
  222 A(I,J)=R(I,J)=UNCOR(I,J)=0.
      DEVMAX=-99.E99   $   DEVMIN=+99.E99   $   SMALL=+99.E99

C=========================================================================
C READ CONTROL DATA
      KKZ=1
      READ 99,NTAPE,NOBS,N,NA,IWT,IPLOT,IORIGN,INAME,NCARDS,NTRAN,
     1NCON,ID
      IF (NOBS) 5001,5002,5002
 5001 KKZ=2 $ NOBS=-NOBS
 5002 IF (N) 60,60,61
61    WRITE(6,66665)
      WRITE(6,999)NOBS,N,NA,IWT,IPLOT,IORIGN,INAME,NCARDS,NTRAN,NCON,ID

C=========================================================================
C  SEE IF NAMES ARE GIVEN, IF NOT THEN GENERATE NUMBERS FOR NAMES
      GO TO (515,514) INAME
```

```
515    DO 500I=1,60
500    NAME(I)=I  $  GO TO 516
514    READ1001,(NAME(I),I=1,NA)

C=====================================================================================
C  CHECK TO SEE IF VARIABLE FORMAT CARDS ARE NEEDED
516    IF (NCARDS) 3001, 3001, 3000
3001   IFMT(1)=8H(10X,10F  $  IFMT(2)=8H7.0)
       IFMT(3)=IFMT(4)=IFMT(5)=IFMT(6)=IFMT(7)=IFMT(8)=IFMT(9)=8H
       GO TO 3002
 3000 JVAR=8*NCARDS
       READ1000,(IFMT(I),I=1,JVAR)
3002   WRITE(6,66667)(IFMT(I),I=1,JVAR)

       IF(NTAPE)3005,3004,3003
3003   WRITE(6,66662)  $  GO TO 3006
3004   WRITE(6,66663)  $  GO TO 3006
3005   WRITE(6,66664)  $  GO TO 3006
3006   CONTINUE

C=====================================================================================
C  CHECK TO SEE IF TRANSGENERATIONS AND CONSTANTS ARE TO BE READ.  IF SO, THEN
C  READ AND PRINT THEM.
       IF(NTRAN.LE.0)8022,8020
 8020 READ8100,(NT(I),NI(I),NJ(I),NK(I),I=1,NTRAN)
       WRITE(6,8101)(NT(I),NI(I),NJ(I),NK(I),I=1,NTRAN)
       IF(NCON.LE.0)8022,8021
8021   READ 8103, (C(I),I=1,NCON)  $  WRITE(6,8104)(C(I),I=1,NCON)
8022   WRITE(6,996)

C=====================================================================================
C  READ DATA, WRITE ON TAPE 4 IF REQUIRED. TRANSGENERATE IF REQUIRED
C  WEIGHT IF REQUIRED, SEARCH FOR EXTREMES
C  LET NTAPE = THE NUMBER OF CARDS TO BE READ PER DATA POINT, CHANGE TO BE ABLE
C  TO READ WITH AN 8A10 FORMAT AND TO WRITE ON TAPE 4 IN CARD IMAGE
       NTAPE = NTAPE*8
C  SUM COUNTS, WRITE ON TAPE, AND GET SUMS AND SUMS OF SQUARES
       IF(NA)8066,8066,8067
 8066 NA=N

 8067 KKY=NOBS
       DO 8060 KKK=1,KKY
       IF (NTAPE) 9002,9001,9003
9003   READ 9100, (CARD(I), I=1,NTAPE)
       WRITE(4,9100) (CARD(I), I=1,NTAPE)
       DECODE(80,IFMT,CARD) (X(I),I=1,N) $ GO TO 9004
 9001 READ IFMT,(X(I),I=1,N)  $  GO TO 9004
9002   READ (4,IFMT) (X(I),I=1,N)
 9004 GO TO (5006,5003) KKZ
 5003 DO 5004 I=1,N
       IF (X(I)) 5004,5005,5004
 5004 CONTINUE
       GO TO 5006
 5005 NOBS=NOBS-1 $ GO TO 8060
 5006 IF (NTRAN)  8061,8061,8062
8062   DO 8063 I=1,NTRAN
 8063 CALL TRNSFRM(NT,NI,NJ,NK,C,X,I)
 8061 IF(IWT)8064,8064,8065
8064   X(61)=1.0
8065   DO 834 I=1,NA  $  IF(X(I)-DMAX(I))831,831,832
```

```
832     DMAX(I)=X(I)
831     IF(X(I)-DMIN(I))833,834,834
833     DMIN(I)=X(I)
834     CONTINUE
        W=X(61)   $  NTOTAL=NTOTAL+1   $   WT=WT+W
        WRITE(3) (X(I),I=1,NA)
        DO 11 I=1,NA   $   SUMX(I)=SUMX(I)+W*X(I)
        DO 11 J=1,NA
11      A(I,J)=A(I,J)+W*X(I)*X(J)
 8060   CONTINUE

        N=NA
        IF(NTAPE)9006,9006,9007
9007    END FILE 4
9006    CONTINUE

C  IF REGRESSION IS TO BE THROUGH THE ORIGIN THEN SAVE THE UNCORRECTED SUM OF
C  SQUARES AND CROSS-PRODUCTS
        IF (IORIGN .LE. 1) GO TO 12
874     DO 875 I=1,N   $   XX(I)=A(I,I)   $   XY(I)=A(I,N)   $   DO 875 J=1,N
875     UNCOR(I,J)=A(I,J)

C================================================================================
C  PRINT COUNTS THEN GET MEANS, RANGES, AND CORRECTED SUMS OF SQUARES AND PRINT
12      WRITE(6,202)NTOTAL,N,WT  $  WRITE(6,203)(SUMX(I),I=1,N)  $  WRITE
       1(6,204)
        DO 140 I=1,N
140     WRITE(6,225)(A(I,J),J=I,N)
        REWIND 3   $   WRITE(6,206)
        DO 15 I=1,N   $   XBAR(I)=SUMX(I)/WT
        DO 15 J=1,N
 15     A(I,J)=A(I,J)-SUMX(I)*SUMX(J)/WT
        DO735I=1,N
735     RNG(I)=DMAX(I)-DMIN(I)
        DO 141 I=1,N
141     WRITE(6,225)(A(I,J),J=I,N)

C================================================================================
C SET ASIDE CORRECTED SUMS OF SQUARES
        NN=N-1   $   TOTSS=A(N,N)
        IF (IORIGN .GE. 2) GO TO 7131
7130    DO 713I=1,N   $   XX(I)=A(I,I)
713     XY(I)=A(I,N)
7131    CONTINUE

C================================================================================
C CALCULATE STD. DEVIATIONS, STD. ERRORS, AND COEFFICIENTS OF VARIATION
        PHI = NOBS - 1
        DO 20 I=1,N
        SIGMAX(I)=SQRT (A(I,I)) $X(I)=SIGMAX(I)/SQRT (WT-1.)
        SE(I)=X(I)/SQRT (PHI+1.0)
20      CV(I)=100.*X(I)/XBAR(I)

C================================================================================
C  PRINT MEANS, STD. DEV., STD. ERRORS, COEFF. VAR., MAXIMUMS, MINIMUMS, RANGES
        WRITE(6,996)   $  WRITE(6,736)   $  WRITE(6,737)   $  WRITE(6,997)
        DO 739 I=1,N
739     WRITE(6,738)NAME(I),XBAR(I),X(I),SE(I),CV(I),DMAX(I),DMIN(I),RNG(I
       1)
```

```
C=========================================================================================
C  CALCULATE MATRIX OF SIMPLE CORRELATION COEFFICIENTS AND PRINT
      WRITE(6,996)  $  WRITE(6,207)
      DO 21 I=1,N  $  DO 21 J=1,N
21    R(I,J) = A(I,J)/(SIGMAX(I)*SIGMAX(J))
      DO 2207 I=1,N  $  SAVER(I)=R(I,N)
2207  WRITE(6,300)NAME(I), (R(I,J),J=I,N)

C=========================================================================================
C  SET MATRIX A = CORRELATION MATRIX THEN INVERT A AND PRINT INVERSE
      DO 2297 I=1,N  $  DO 2297 J=1,N
2297  A(I,J)=R(I,J)
      CALL INVERT(A,N)
      WRITE(6,298)
      DO 2298 I=1,N
2298  WRITE(6,225)(A(I,J),J=I,N)

C=========================================================================================
C  CALCULATE AND PRINT HIGHEST ORDER PARTIAL CORRELATION COEFFICIENTS
      WRITE(6,707)
      DO 706 I=1,N  $  DO 706 J=1,N
706   RPRIME(I,J)=-A(I,J)/SQRT (A(I,I)*A(J,J))
      DO 708 I=1,N  $  RPRIME(I,I) = 1.0
708   WRITE(6,300)NAME(I), (RPRIME(I,J),J=I,N)

C=========================================================================================
C  CHECK TO SEE IF THE REGRESSION IS TO BE THROUGH THE ORIGIN.  IF SO, THEN
C   FORM A MATRIX OF PSUEDO CORRELATION COEFFICIENTS AND INVERT THAT MATRIX
C  THEN FORM PSUEDO STANDARD DEVIATIONS
      XN = NN
      IF (IORIGN .LE. 1) GO TO 8711
870   PHI=PHI-XN+1
      DO 876 I=1,N  $  DO 876 J=1,N
876   A(I,J)=UNCOR(I,J)/SQRT (UNCOR(I,I)*UNCOR(J,J))
      CALL INVERT(A,N)
      DO 8761 I=1,N
8761  SIGMAX(I) = SQRT (UNCOR(I,I))
      GO TO 8712

C=========================================================================================
C  CALCULATE REGRESSION COEFFICIENTS, STD. ERRORS, T VALUES, AND CONSTANT TERM
8711  PHI=PHI-XN
8712  SY=SIGMAX(N)*SQRT (1./(A(N,N)*PHI))
      RSQ = 1.-1./A(N,N)  $  RMULT=SQRT (RSQ)  $  RDETR=RSQ*100.
      DO 7700 L=1,NN
      BPRIME(L)=-A(L,N)/A(N,N)  $  B(L)=BPRIME(L)*SIGMAX(N)/SIGMAX(L)
      S(L)=SQRT ((A(L,L)*A(N,N)-A(L,N)*A(L,N))/A(N,N))*SY/SIGMAX(L)
7700  T(L)=B(L)/S(L)
      IF (IORIGN .GE. 2) GO TO 504
505   BO=XBAR(N)
      DO506L=1,NN
506   BO=BO-B(L)*XBAR(L)  $  T(N)=BO/SY  $  GO TO 507
504   BO=0.  $  T(N)=0.
507   CONTINUE

C=========================================================================================
C  CALCULATE THE DEVIATIONS FROM THE MODEL AND THEIR RANGE AND WRITE ON TAPE
      DEVSQ=0. $ REWIND 2
      DO 2201II=1,NTOTAL  $  READ (3) (X(I),I=1,N) $ YPRED=BO
      DO 2202 JJ=1,NN
```

```
2202   YPRED=YPRED+B(JJ)*X(JJ)
       DEV=X(N)-YPRED
       IF(DEV-DEVMAX)748,748,749
749    DEVMAX=DEV
748    IF(DEVMIN-DEV)750,751,751
751    DEVMIN=DEV
750    CONTINUE
       IF(ABS (DEV)-ABS (SMALL))508,509,509
508    SMALL=DEV
509    DEVSQ=DEVSQ+DEV*DEV
 2201  WRITE(2) II,(X(I),I=1,N),YPRED,DEV

C=====================================================================================
C CALCULATE ANALYSIS OF VARIANCE FOR REGRESSION
       IF(IORIGN.GE.2) 22011,22022
22011  TOTSS=XX(N)
22022  SSREG=TOTSS-DEVSQ  $  SMREG=SSREG/XN  $  SMDEV=DEVSQ/PHI
       DFTOT=XN+PHI  $  F=SMREG/SMDEV

C=====================================================================================
C  PRINT REGRESSION CHARACTERISTICS
       WRITE(6,996)  $  WRITE(6,743)  $  WRITE(6,753)  $  WRITE(6,754)  $
      1  WRITE(6,997)
       WRITE(6,755)RDETR,RMULT,SY,DEVMAX,DEVMIN,BO,T(N)
C      THE FOLLOWING WAS ADJUSTED BY B A PETTY, 6/8/67.
       SMALL=(DEVMAX-DEVMIN)/SY
512    WRITE(6,513)SMALL

C=====================================================================================
C PRINT REGRESSION COEFFICIENTS, STD. ERRORS, T VALUES , AND CORRELATIONS WITH
       WRITE(6,745)  $  WRITE(6,997)
       IF(N-2)757,756,757
756    RPRIME(1,N) = SAVER(1)
757    DO746I=1,NN
746    WRITE(6,744)NAME(I),B(I),S(I),T(I),BPRIME(I),SAVER(I),RPRIME(I,N)

C=====================================================================================
C  PRINT ANALYSIS OF VARIANCE OF REGRESSION
       WRITE(6,709)  $  WRITE(6,997)
       WRITE(6,710)SSREG,XN,SMREG,F  $  WRITE(6,711)DEVSQ,PHI,SMDEV
       IF(IORIGN.GE.2.OR.IWT.GE.2) 7161, 7160
7160   DO716 I=1,NN
       XX(I)=XY(I)*XY(I)/XX(I)  $  FSS(I)=XX(I)*(DFTOT-1.)/(TOTSS-XX(I))
716    WRITE(6,994)NAME(I), XX(I),  XX(I), FSS(I)
7161   WRITE(6,712)TOTSS,DFTOT

C=====================================================================================
C  READ DATA POINTS AND DEVIATIONS FROM TAPE, CHECK FOR OUTLIERS, PRINT
       REWIND 2  $  WRITE(6,996)  $  WRITE(6,995)  $  WRITE(6,997)
       DO 747 II=1,NTOTAL
       READ(2) ICOUNT,(X(I),I=1,N),YPRED,DEV
       IF(ABS (DEV)-2.*SY)7477,7477,7499
7499   IF(ABS (DEV)-3.0*SY)7500,7500,7488
7500   WRITE(6,992)  $  GO TO 7477
7488   WRITE(6,993)
7477   WRITE(6,200)X(N), YPRED, DEV
747    WRITE(6,201) II, (X(I),I=1,NN)

C=====================================================================================
C  SEE IF PLOTS ARE CALLED FOR
```

```
      IF(IPLOT.GE.1.AND.IPLOT.LE.8)8004,8005
8004  CALL DRAW(INAME,NAME,DMIN,DMAX,RNG,N,NPLOTS,SY,ID,LAST,NTOTAL,X,
     1 IPLOT, V1, V2, V3)

C===========================================================================
C  PLOT Y AGAINST EACH X WHEN Y IS CORRECTED FOR DEVIATIONS OF ALL OTHER
C  X S FROM THEIR RESPECTIVE MEANS
8005  IF(IPLOT.EQ.9)5,1
5     CALL CORRECT(B,BO,NN,XBAR,N,NAME,ID,NTOTAL,V1,V2,V3,INAME,X)

C===========================================================================
C RETURN TO READ NEXT SET OF DATA
      GO TO 1

C===========================================================================
C  END OF RUN SO QUIT
60    WRITE(6,66666)  $  WRITE(6,998)  $  CALL EXIT
C===========================================================================
C  INPUT FORMATS -- IF VARIABLE FORMAT ISN T SPECIFIED. THE STANDARD INPUT IS
C  (10X,10F7.0).  CONTROL CARD AND NAME INPUT FORMATS ARE AS FOLLOWS
   99 FORMAT(I2,I4,9I2,7A8)
 1000 FORMAT(8A10)
 1001 FORMAT(10A8)

C  OUTPUT FORMATS
200   FORMAT(1H , 74X, 3F14.5)
201   FORMAT(1H ,I4,5F14.5,/(5X,5F14.5))
202   FORMAT(14H0NO. OF DATA =,I5,5X,19H NO. OF VARIABLES =,
     1I4,5X,16HSUM OF WEIGHTS =,F19.5)
203   FORMAT(18H0SUMS OF VARIABLES,/(1H ,6F19.5))
206   FORMAT(51H0RESIDUAL SUMS OF SQUARES AND CROSS-PRODUCTS MATRIX)
207   FORMAT(33H0SIMPLE LINEAR CORRELATION MATRIX)
225   FORMAT(1H , 6F19.5)
251   FORMAT(1H0,I5,5X,3(6X,F14.5))
204   FORMAT(46H0RAW SUMS OF SQUARES AND CROSS-PRODUCTS MATRIX)
298   FORMAT(30H0INVERSE OF CORRELATION MATRIX)
300   FORMAT(1H , A8, 12F9.4/(8X, 12F9.4))
513   FORMAT(38H0TEST STATISTIC FOR OUTLIERS (IF MORE ,
     122HTHAN 30 DATA POINTS) =,F14.5)
707   FORMAT(35H0HIGHEST ORDER PARTIAL CORRELATIONS)
709   FORMAT(7H0SOURCE25X14HSUM OF SQUARES5X2HDF6X12HMEAN SQUARES9X1HF)
710   FORMAT(27H0DUE TO REGRESSION OF ALL X,F19.5,F6.0,F19.5,F13.3)
711   FORMAT(27H0DEVIATIONS FROM REGRESSION,F19.5,F6.0,F19.5/)
712   FORMAT(6H0TOTAL,21X,F19.5,F6.0)
736   FORMAT(30H0CHARACTERISTICS OF INPUT DATA)
737   FORMAT(117H0VARIABLE              MEAN                SD              SE
     1              CV                MAX                MIN              RNG)
738   FORMAT(1H , A8, 6F16.5, F15.5)
743   FORMAT(39H0CHARACTERISTICS OF REGRESSION ANALYSIS)
744   FORMAT(1H , A8, 2F20.6, 2F20.5, 2F15.4)
745   FORMAT(120H0VARIABLE             COEFFICIENT         STD. ERROR OF B
     1    T TEST OF B    STD. COEFFICIENT         SIMPLE R       PARTIAL R )
753   FORMAT(111H0  COEF.  MULT.      STANDARD ERROR    MAXIMUM DEVIATION
     1S FROM THE MODEL            Y INTERCEPT        T OF CONSTANT)
754   FORMAT(69H   DETR.  CORR.          OF ESTIMATE           POSITIVE
     1      NEGATIVE)
755   FORMAT(1H , F7.2, F7.4, 5F19.5)
992   FORMAT(86H CHECK THE NEXT DATA POINT AS AN OUTLIER, IT DEVIATES FR
     1OM THE MODEL MORE THAN 2 SIGMA)
```

```
993    FORMAT(86H CHECK THE NEXT DATA POINT AS AN OUTLIER, IT DEVIATES FR
      1OM THE MODEL MORE THAN 3 SIGMA)
994    FORMAT(1H ,4X,A8,14H INDEPENDENTLY,F19.5,5X,1H1,F19.5,F13.3)
995    FORMAT(1H0, 4H NO., 24X, 21HINDEPENDENT VARIABLES, 33X, 1HY, 10X,
      19HPREDICTED, 5X, 9HDEVIATION)
996    FORMAT(1H0,119(1H*))
997    FORMAT(1H ,119(1H-))
998    FORMAT(118H VAN DYNE, G. M. 1965 REGRESS - A MULTIPLE REGRESSION A
      1ND CORRELATION PROGRAM WITH GRAPHIC OUTPUT FOR MODEL EVALUATION/,*
      2 ORNL TM 1288*)
999    FORMAT(15H CONTROL DATA =, I6, 9I4, 7H......., 7A8)
 8100 FORMAT(40I2)
 8101 FORMAT(20H TRANSGENERATIONS = ,6(I2,1H-,I2,1H-,I2,1H-,I2,5X)/
      1(20X,6(I2,1H-,I2,1H-,I2,1H-,I2,5X)))
 8103 FORMAT(8F10.0)
8104   FORMAT(13H0CONSTANTS = , 5(F17.5, 3X)/(13X, 5(F17.5, 3X)))
 9100 FORMAT(8A10)
66662 FORMAT(1H ,*DATA WERE READ FROM CARDS AND WERE WRITTEN ON TAPE 4*)
66663 FORMAT(1H , *DATA WERE READ FROM CARDS*)
66664 FORMAT(1H , *DATA WERE READ FROM TAPE 4*)
66665 FORMAT(1H1,18X,59HOBS NBE NAF WGT PLT ORG NAM FMT TRN CON       IDE
      1NTIFICATION)
66666 FORMAT(11H0END OF RUN)
66667 FORMAT(19H INPUT FORMAT WAS =, 8A10)

      END
```

```
  INVERT   FORTRAN EXTENDED VERSION 2.0               03/14/70             *00.39.34.
      SUBROUTINE INVERT(A,N)
C     MATRIX INVERSION BY GAUSS-JORDAN ELIMINATION
C  THIS SUBROUTINE TAKES MATRIX A OF N BY N DIMENSION, INVERTS IT, STORES IN A
      DIMENSION A(60,60),B(60),C(60),LZ(60)
      DO 10 J=1,N
   10 LZ(J)=J
      DO 20 I=1,N
      K=I
      Y=A(I,I)
      L=I-1
      LP=I+1
      IF(N-LP)14,11,11
   11 DO 13 J=LP,N
      W=A(I,J)
      IF(ABS (W)-ABS (Y))13,13,12
   12 K=J
      Y=W
   13 CONTINUE
   14 DO 15 J=1,N
      C(J)=A(J,K)
      A(J,K)=A(J,I)
      A(J,I)=-C(J)/Y
      A(I,J)=A(I,J)/Y
   15 B(J)=A(I,J)
      A(I,I)=1.0/Y
      J=LZ(I)
      LZ(I)=LZ(K)
      LZ(K)=J
      DO 19 K=1,N
      IF(I-K)16,19,16
   16 DO 18 J=1,N
      IF(I-J)17,18,17
   17 A(K,J)=A(K,J)-B(J)*C(K)
   18 CONTINUE
   19 CONTINUE
   20 CONTINUE
      DO 200 I=1,N
      IF(I-LZ(I))100,200,100
  100 K=I+1
      DO 500 J=K,N
      IF(I-LZ(J))500,600,500
  600 M=LZ(I)
      LZ(I)=LZ(J)
      LZ(J)=M
      DO 700 L=1,N
      C(L)=A(I,L)
      A(I,L)=A(J,L)
  700 A(J,L)=C(L)
  500 CONTINUE
  200 CONTINUE
      RETURN
      END
      SUBROUTINE TRNSFRM(NT,NI,NJ,NK,C,X,I)
C   GIVEN A ONE-DIMENSIONAL ARRAY X AND ARRAYS OF TRANSGENERATION CODES (NT, NI
C   NJ, AND NK) AND A LIST OF CONSTANTS C AND AN INDEX I THIS SUBROUTINE WILL
C   PERFORM A VARIETY OF TRANSGENERATIONS (TRANSFORMATIONS + GENERATION)
C   N1 = THE TYPE OF TRANSGENERATION
C   N2 = THE INDEX (SUBSCRIPT) OF THE VARIABLE IN ARRAY X AFTER TRANSGENERATION
C   N3 = INDEX OF THE VARIABLE BEFORE TRANSGENERATION
C   N4 = THE INDEX OF A SECOND VARIABLE OR OF A CONSTANT USED IN TRANSGENERATIO
      DIMENSIONNT(99),NI(99),NJ(99),NK(99),X(61),C(99)
      N1=NT(I)
      N2=NI(I)
      N3=NJ(I)
      N4=NK(I)
      GOTO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19)N1
    1 X(N2)=X(N3)   $ GO TO 99
    2 X(N2)=X(N3)*X(N4)$GOTO99
    3 X(N2)=X(N3)/X(N4)$GOTO99
    4 X(N2)=X(N3)+X(N4)$GOTO99
    5 X(N2)=X(N3)-X(N4)$GOTO99
    6 X(N2)=1.0/X(N3)$GOTO99
    7 X(N2)=X(N3)+C(N4)$GOTO99
    8 X(N2)=X(N3)/C(N4)$GOTO99
    9 X(N2)=C(N4)/X(N3)$GOTO99
   10 X(N2)=X(N3)*C(N4)$GOTO99
   11 X(N2)=X(N3)**C(N4)$GOTO99
   12 IF(X(N3).EQ.0)18,40
   18 X(N2)=0.0 $ GOTO99
   40 X(N2)=ALOG(X(N3))   $   GOTO99
   41 X(N2)=ALOG(X(N3))*.4342917   $   GOTO99
   13 IF(X(N3).EQ.0)18,41
   14 X(N2)=EXP (X(N3))   $   GOTO99
   15 TEMP=X(N3)*.0174533
      X(N2)=SIN (TEMP)   $   GO TO 99
   16 X(N2)=ATAN (X(N3))   $   GO TO 99
   17 IF(X(N3).LT.C(N4))20,21
   20 X(N3)=X(N3)+360
   21 TEMP=X(N3)-C(N4)
      TEMP=TEMP/57.29578
      X(N2)=SIN (TEMP)+1.0
      GO TO 99
   19 IF(X(N3).GE.X(N4))22,23
   22 X(N2)=X(N3)
      GO TO 99
   23 X(N2)=X(N4)
   99 RETURN
      END
```

```
  SORT     FORTRAN EXTENDED VERSION 2.0           03/14/70            *00.39.34.

      SUBROUTINE SORT (X,KEY,NO)
C  ACCEPT AN ARRAY OF N (NO) VALUES (X) AND SORT THEM INTO ASCENDING ORDER.
C  STORE THEIR ORIGINAL SUBSCRIPTS IN ARRAY KEY.
      DIMENSION KEY(61),X(61)
      DO 1 I=1,NO
 1            KEY (I)=I
      MO=NO
 2    IF(MO-15) 21,21,23
 21   IF(MO-1) 9,9,22
 22   MO=2*(MO/4)+1
      GO TO 24
 23   MO=2*(MO/8)+1
 24   KO=NO-MO
      JO=1
 25   I=JO
 26   IF(X(I)-X(I+MO)) 28,28,27
 27   TEMP=X(I)
 2701 X(I)=X(I+MO)
 2702 X(I+MO)= TEMP
 2703 KEMP=KEY(I)
 2704 KEY(I)=KEY(I+MO)
 2705 KEY(I+MO)=KEMP
      I=I-MO
      IF(I-1) 28,26,26
 28   JO=JO+1
      IF(JO-KO) 25,25,2
 9    RETURN
      END
      SUBROUTINE DRAW(X)
      DIMENSION X(16)
C     DUMMY
      WRITE(6,100)
  100 FORMAT(*0 NO PLOTS UNTIL SUBROUTINES ARE SUPPLIED*)
      RETURN
      END
      SUBROUTINE CORRECT(X)
      DIMENSION X(13)
C     DUMMY
      RETURN
      END
```

INPUT DECK FOR SAMPLE RUN OF REGRES1

```
 0  15 4 3 0 0 0 2 1 4 1SAMPLE OUTPUT OF REGRES1 - SNOW COURSES VS. STREAMF
CPAM    BSA     POUDRE
(F4.1,2(F5.1),1X,F3.0)
100202010401010201020300001030400
2.0
20.8 22.1  0.4 147       CPA,CPM,BSA,POUDRE 53
25.1 17.6  1.6 106       CPA,CPM,BSA,POUDRE 54
19.3 17.3  0.2 133       CPA,CPM,BSA,POUDRE 55
30.5 38.9  2.3 202       CPA,CPM,BSA,POUDRE 56
29.7 36.1  5.0 378       CPA,CPM,BSA,POUDRE 57
29.4 32.9  4.0 256       CPA,CPM,BSA,POUDRE 58
30.9 29.7  5.0 228       CPA,CPM,BSA,POUDRE 59
29.2 26.0  2.8 199       CPA,CPM,BSA,POUDRE 60
22.3 27.5  2.8 317       CPA,CPM,BSA,POUDRE 61
38.9 36.5  2.1 274       CPA,CPM,BSA,POUDRE 62
20.1 19.8  2.4 146       CPA,CPM,BSA,POUDRE 63
35.4 32.2  1.9 176       CPA,CPM,BSA,POUDRE 64
27.7 34.3  4.1 325       CPA,CPM,BSA,POUDRE 65
21.1 21.9  0.0 110       CPA,CPM,BSA,POUDRE 66
27.6 33.9  0.5 242       CPA,CPM,BSA,POUDRE 67
```

```
                  OBS NBE NAF WGT PLT ORG NAM FMT TRN CON       IDENTIFICATION
CONTROL DATA =     15   4   3   0   0   0   2   1   4   1.......SAMPLE OUTPUT OF REGRES1 - SNOW COURSES VS. STREAMFLOW
INPUT FORMAT WAS =(F4.1,2(F5.1),1X,F3.0)
DATA WERE READ FROM CARDS
TRANSGENERATIONS = 10- 2- 2- 1      4- 1- 1- 2      1- 2- 3- 0      1- 3- 4- 0

CONSTANTS =            2.00000

==========================================================================================================

NO. OF DATA =   15      NO. OF VARIABLES =   3     SUM OF WEIGHTS =          15.00000
SUMS OF VARIABLES
          1261.40000          35.10000          3239.00000

RAW SUMS OF SQUARES AND CROSS-PRODUCTS MATRIX
        111297.26000        3178.37000       287597.70000
           120.57000        8932.10000
        795489.00000

RESIDUAL SUMS OF SQUARES AND CROSS-PRODUCTS MATRIX
          5221.92933         225.69400        15219.39333
            38.43600        1352.64000
         96080.93333

==========================================================================================================

CHARACTERISTICS OF INPUT DATA

VARIABLE        MEAN          SD           SE           CV           MAX          MIN          RNG
----------------------------------------------------------------------------------------------------------
CPAM          84.09333     19.31306      4.98662     22.96624    111.00000     53.00000     58.00000
BSA            2.34000      1.65693       .42782     70.80912      5.00000      0.00000      5.00000
POUDRE       215.93333     82.84277     21.38991     38.36497    378.00000    106.00000    272.00000

==========================================================================================================

SIMPLE LINEAR CORRELATION MATRIX
CPAM        1.0000     .5066     .6795
BSA         1.0000     .7040
POUDRE      1.0000

INVERSE OF CORRELATION MATRIX
             1.86283           -.10223           -1.19375
             1.98810          -1.33012
             2.74748

HIGHEST ORDER PARTIAL CORRELATIONS
CPAM        1.0000     .0631     .5277
BSA         1.0000     .5691
POUDRE      1.0000

==========================================================================================================

CHARACTERISTICS OF REGRESSION ANALYSIS

 COEF.  MULT.      STANDARD ERROR    MAXIMUM DEVIATIONS FROM THE MODEL          Y INTERCEPT       T OF CONSTANT
 DETR.  CORR.        OF ESTIMATE        POSITIVE           NEGATIVE
----------------------------------------------------------------------------------------------------------
 63.60  .7975         53.98344         162.59330          -63.88621              2.56640             .04754

TEST STATISTIC FOR OUTLIERS (IF MORE THAN 30 DATA POINTS) =        3.08390
VARIABLE          COEFFICIENT        STD. ERROR OF B      T TEST OF B   STD. COEFFICIENT      SIMPLE R      PARTIAL R
----------------------------------------------------------------------------------------------------------
CPAM               1.863729            .866106             2.15165          .43449           .6795          .5277
BSA               24.2[illegible]      10.098261           2.39766          .48412           .7040          .5691
```

```
SOURCE                              SUM OF SQUARES    DF      MEAN SQUARES        F
-------------------------------------------------------------------------------------------------------------

DUE TO REGRESSION OF ALL X           61110.33484      2.       30555.16742      10.485

DEVIATIONS FROM REGRESSION           34970.59849     12.        2914.21654

   CPAM      INDEPENDENTLY           44357.15588      1        44357.15588      11.149
   BSA       INDEPENDENTLY           47616.19486      1        47616.19486      12.772
TOTAL                                96080.93333     14.

=============================================================================================================

 NO.                           INDEPENDENT VARIABLES                      Y          PREDICTED     DEVIATION
-------------------------------------------------------------------------------------------------------------
  1       65.00000        .40000                                    147.00000    133.39081      13.60919
  2       60.30000       1.60000                                    106.00000    153.67729     -47.67729
  3       53.90000        .20000                                    133.00000    107.86241      25.13759
  4      108.30000       2.30000                                    202.00000    260.07981     -58.07981
  5      101.90000       5.00000                                    378.00000    313.50547      64.49453
  6       95.20000       4.00000                                    256.00000    276.81347     -20.81347
  7       90.30000       5.00000                                    228.00000    291.88621     -63.88621
  8       81.20000       2.80000                                    199.00000    221.67525     -22.67525
  9       77.30000       2.80000                                    317.00000    214.40670     102.59330
 10      111.90000       2.10000                                    274.00000    261.94823      12.05177
 11       59.70000       2.40000                                    146.00000    171.92306     -25.92306
 12       99.80000       1.90000                                    176.00000    234.55610     -58.55610
 13       96.30000       4.10000                                    325.00000    281.28407      43.71593
 14       64.90000       0.00000                                    110.00000    123.52243     -13.52243
 15       95.40000        .50000                                    242.00000    192.46868      49.53132
END OF RUN
VAN DYNE, G. M. 1965 REGRESS - A MULTIPLE REGRESSION AND CORRELATION PROGRAM WITH GRAPHIC OUTPUT FOR MODEL EVALUATION
ORNL TM 1288
```

AUTOREG

INTRODUCTION

Many economic and biological phenomena involve a process of gradual adjustment to change. Because of this process, the current value of one variable depends on present and past values of other variables. Models which describe this process are known as distributed-lag functions. It is also possible that, if the observations of the dependent variable are taken in a time sequence, consistent measurement errors will be made. If either of these conditions exist, the application of a least squares method to the data will result in a lack of fit caused from autocorrelation in the error term.

AUTOREG is a flexible regression system which allows for distributed lag functions and autoregressive errors. It is capable of estimating the parameters of models which are either linear or nonlinear in the parameter space under the statistical assumption of either independent errors or first order autoregressive errors. Conditional regressions may also be run using any of seven combinations of models and error assumptions.

AUTOREG was written in 1965 by Jack E. Martin for use on the IBM 7040 and 7090194 computer systems. It has been converted for use on the NCAR and SCOPE systems of the CDC 6400 computer.

The most general form of the equations analyzed is as follows:[1/]

[1/] The derivation of equation 1 is presented in *The use of distributed lag models containing two lag parameters in the estimation of elasticities of demand*, J. Farm Econ. 45:1474-1481, December 1963.

SECTION A

1. $$Y_t = a_o(1 - \lambda)(1 - \mu)(1 - \beta) + \sum_{i=1}^{A} a_i X_{it} - (\mu + \beta) \sum_{i=1}^{A} a_i X_{it-1}$$

$$+ \mu\beta \sum_{i=1}^{A} a_i X_{it-2} + \sum_{j=1}^{B} b_j Z_{jt} - (\lambda + \beta) \sum_{j=1}^{B} b_j Z_{jt-1}$$

$$+ \lambda\beta \sum_{j=1}^{B} b_j Z_{jt-2} + (\lambda + \mu + \beta)\, Y_{t-1}$$

$$- [(\lambda + \mu)\, \beta + \lambda\mu]\, Y_{t-2} + \lambda\mu\beta Y_{t-3}$$

$$+ \sum_{k=1}^{C} d_k D_{kt} + e_t$$

where

Y_{t-m} = the current and lagged values of the dependent variable ($m = 0,1,2,3$)

X_{it-m} = the current and lagged values of the exogenous variables associated with the lag parameter λ ($m = 0,1,2$)

Z_{jt-m} = the current and lagged values of the exogenous variables associated with the lag parameter $\underset{\mu}{\lambda}$ ($m = 0,1,2$)

D_{kt} = the current exogenous and/or dummy variables which are not associated with a lag

e_t = the errors in the equation

$a_o(1-\lambda)(1-\mu)(1-\beta)$ = the pure constant term

a_i = the parameters of the set of exogenous variables, X_{it}, ($i = 1,\ldots,A$)

2. b_j = the parameters of the set of exogenous variables, Z_{jt}, $(j = 1,\ldots,B)$

λ = the lag parameter associated with the set of exogenous variables, X_{it}

μ = the lag parameter associated with the set of exogenous variables, Z_{jt}

β = the first order autocorrelation coefficient

d_k = the parameters associated with the set of exogenous variables, D_{kt}

The estimation procedure used is presented in more detail by Fuller and Martin[2/] and includes a modification to insure convergence.[3/]

This is known as a two lag model with autoregressive errors (model 7). The other six models may be considered by restricting the values of the parameters as follows:

1. A two lag model with independent errors is obtained by setting the parameter β equal to zero.
2. A Nerlove model with autoregressive errors may be derived by either setting y equal to zero and omitting the Z terms or λ equal to zero and omitting the X terms.
3. A static model with autoregressive errors results when $\lambda = y = 0$ and either the X terms or the Z terms are omitted from the general equation.

[2/] Wayne A. Fuller and James E. Martin, *The effects of autocorrelated errors on the statistical estimation of distributed lag models*, J. Farm Econ. 43: 71-82, February 1961.

[3/] Wayne A. Fuller and James E. Martin, A Note on *The effects of autocorrelated errors on the statistical estimation of distributed lag models*, J. Farm Econ. 44:407-410, May 1962.

4. A Nerlove model with independent errors by restricting $\beta = 0$ in addition to the restrictions in model 2.

5. A first difference model with independent errors results from setting $\beta = 1$ in addition to the restrictions specified by model 3.

6. A static model with independent errors has no X or Z terms and $\lambda = y = \beta = 0$.

METHOD OF OPERATION

The method used is described in detail in the program document by James E. Martin[4/] and will be omitted here.

INPUT REQUIREMENTS

The concept used in this program is that of a "Problem" which may consist of one or more "Jobs" or regressions. Thus, a "Problem Card" causes the program to read a set of data and indicates the total number of regressions or Jobs which are to be run using these data. Two "Control Cards" and an appropriate number of "Start Vector Cards" are associated with each Job.

A given Problem must consist of the following cards in the order specified:

a. Problem Card (one card)

b. Data Deck (the number of cards may vary)

c. Control Card 1 (one card)

d. Control Card 2 (one card)

e. Start Vector Card(s) (the number of cards depends on the number of parameters in the Job)

Additional Jobs may be run using *the same Data Deck* by repeating items c, d, and e above for each additional Job.

[4/] James E. Martin, *Computer algorithms for estimating the parameters of selected classes of nonlinear, single equation models*. May 1968.

INPUT FORMATS

Problem Card

This card controls the parameters which specify the number of variables, the number of observations per variable and the number of Jobs which are associated with a given Data Deck. The format of the Problem Card is as follows:

Cols.	1- 5	...	the Problem Number
Cols.	6-10	...	the Number of variables in the Data Deck
Cols.	11-~~20~~ 15	...	the Number of observations per variable in the Data Deck
Cols.	16-20	....	the Number of Jobs to be run using this Data Deck
Cols.	21-80	...	Blank or any identifying information the user wishes to use.

All values which are punched into columns 1-20 of the Problem Card must be right justified integer values. Columns 21-80 may contain any alphanumeric information.

Data Deck

The Data Deck is punched in *row order* starting with the first observation on each variable. Seven observations are punched per card using FORTRAN FORMAT (7F10.4).

In order for the Data Deck to conform with the MRHS-1 and MRHS-2 programs, the variables in the Data Deck must appear as column vectors in the following order:

3. $[X_{11},X_{21},\ldots,X_{A1},Z_{11},Z_{21},\ldots,Z_{B1},D_{11},D_{21},\ldots,D_{C1},Y_{11},Y_{21},\ldots,Y_{y1}]$

where the subscripts A and B denote the total number of *current* exogenous variables associated with lag parameters λ and μ, respectively. The variables $D_{it}(i = 1,\ldots,C)$ represent the exogenous variables which are not associated with the lag parameters λ and μ. The subscript C is the total number of D_{it} variables. *Note*: The y dependent variables, $Y_{it}(i = 1,\ldots,y)$ must *always* appear on the right hand side of the exogenous variables in the Data Deck.

Control Card 1

This card defines most of the major parameters which control the type of estimation which is to be employed for the Job, the size of the Job, etc. The card format is as follows:

Cols.	1- 5 ...	Job Number
Cols.	6- 7 ...	The total number of exogenous variables associated with lag parameter λ ($A \geq 0$)
Cols.	8- 9 ...	The total number of exogenous variables associated with lag parameter μ ($B \geq 0$)
Cols.	10-11 ...	The total number of exogenous variables and/or dummy variables which are not associated with a lag parameter ($C \geq 0$)
Cols.	12-13 ...	The column number of the variable in the Data Deck which is to be used as the dependent variable for this Job.
Cols.	14-15 ...	The total number of parameters or rows to be controlled or skipped, NSKIP ($NSKIP \geq 0$)
Cols.	16-17 ...	"01" for least squares estimation, otherwise, the maximum number of iterations to be performed in a nonlinear estimation Job ("25" is suggested for nonlinear Jobs)
Col.	18 ...	"1" if the calculation of the pure constant term is desired, otherwise, "0" when regressions through the origin are desired
Col.	19 ...	"1" if the calculation of the predicted values, residuals and Durbin-Watson "d" statistic is desired, otherwise, "0"

Col.	20	...	"1" if a listing of the correlation matrix is desired, otherwise, "0". *Note*: This option is available only when Col. 31 of Control Card 1 contains "0"
Cols.	21-30	...	The "test criterion" to be used to halt the iterative process in nonlinear Jobs. The value of the "test criterion" is punched using FORTRAN FORMAT (F10.4). ("0.001" is recommended for most nonlinear Jobs
Col.	31	...	"0" if the Job to be run is: (1) the first Job in a Problem (2) involves a different dependent variable from the preceding Job *or* (3) involves a different computation on the pure constant term from the preceding Job, otherwise, "1"
Cols.	32-80	...	Blank or any identifying information the user wishes to punch into this card.

Control Card 2

This card specifies the actual row numbers (or parameter numbers) of those parameters which are to be controlled or skipped. The order or numbering of the parameters in any given Job is always assumed to be: $a_i (i = 1,\dots,A)$ provided $A > 0$; $b_j (j = 1,\dots,B)$ provided $B > 0$; λ; μ; β; and $d_k (k = 1,\dots,C)$ provided $C > 0$. Thus, the programs are written such that the first r parameters, $r = A + B \geq 0$, are the total number of rows for the parameters of the current exogenous variables, X_{it} and Z_{jt}, row $r + 1$ is assumed to be the parameter λ, row $r + 2$ is assumed to be the parameter μ, row $r + 3$ is assumed to be the parameter β, and row $r + 3 + k$, $(k \geq 0)$ is assumed to be the parameter of the *k*th current exogenous or dummy variable, D_{kt}, if such a variable is included in the Job being run.

The format for Control Card 2 is FORTRAN FORMAT (15(I2,2X)). Thus, the card is punched in the following manner:

Cols.	1- 2	... The actual row number of the first parameter to be controlled or skipped.
Cols.	3- 4	... Blank
Cols.	5- 6	... The actual row number of the second parameter to be controlled or skipped
Cols.	7- 8	... Blank
.		
.		
etc.		
.		
.		
Cols.	61-80	... Blank

Note: The number of parameters which are punched into Control Card 2 and which are to be controlled or skipped *must* agree with the number punched into columns 14-15 of Control Card 1.

Start Vector Card(s)

The total number of Start Vector Card(s), NSVC, required for a given Job depends upon the number of parameters involved and may differ from Job to Job. The number of Start Vector Card(s) required for each Job is determined by rounding NSVC to the largest integer value where:

$$4. \quad \text{NSVC} = \frac{A + B + C + 3}{8}$$

The Start Vector Card(s) contain(s) all initial parameter estimates. Therefore, the order of the initial parameter estimates *must be identical* to the actual row numbers described under Control Card 2 for each parameter. The format used for punching the initial parameter estimates in the Start Vector Card(s) is FORTRAN FORMAT (8F10.5). All columns of the last Start Vector Card which are not required for initial parameter estimates may be left blank. *Note*: In all least squares estimation Jobs, the appropriate number of blank card(s) may be used as Start Vector Card(s). In all non-linear estimation Jobs, the user's best estimate of each parameter value should be punched into the appropriate locations of the Start Vector Card(s). The nearer the user's initial estimates are to the final parameter estimates, the shorter will be the computing time for the Job.

OUTPUT

At the conclusion of the run, the printed output will consist of:

1. For each Problem: Problem Card
2. A listing of the input data
3. For each Job: Control Card 1
4. The Correlation Matrix if "1" appears in column 20 of Control Card 1 printed in lower triangular form
5. For each trial: k, IDF, TSS, SSE, MSE, SSR, R Square
6. For each trial: values of parameter estimates
7. For the last iteration: i, θ_i, $\Delta\theta_i$, Test_i, Variance_i, Stand Er_i, $\text{Student } t_i$
8. For each Job:
    a. The dimensions of the inverse matrix
    b. The inverse matrix at the last iteration
    c. The pure constant term *if* "1" appears in column 18 of Control Card 1
    d. The a_o term
    e. The error routine and Durbin-Watson "d" if "1" appears in column 19 of Control Card 1.
    f. The means of the variables at times t, t-1, t-2, and t-3

where

k = the denominator in the geometric series 1, 1/2, 1/4,...

IDF = the degrees of freedom

TSS = the total sum of squares

SSE = the sum of squares for error

MSE = the mean square error

SSR = the sum of squares for regression

R square = R^2

i = the parameter row number

θ_i = the estimate of the ith parameter

$\Delta\theta_i$ = the change in the ith parameter

$\text{Test}_i = (\Delta\theta_i)^2/\text{Var}(\theta_i)$

Variance_i = the variance of the ith parameter

Stand Er_i = the standard error of the ith parameter estimate

Student t_i = the student "t" test of the ith parameter estimate against zero. All printed output will possess column headings above the numerical data and/or estimate.

ERROR CONDITIONS

The following list of error conditions are checked by the programs:

Error 107 = columns 6-7 of Control Card 1 contain a negative punch.

Error 135 = columns 8-9 of Control Card 1 contain a negative punch.

Error 191 = columns 10-11 of Control Card 1 contain a negative punch.

Error 453 = columns 14-15 of Control Card 1 contain a negative punch.

Error 524 = singular matrix, matrix cannot be inverted.

Error 727 = the test value in columns 21-30 of Control Card 1 is negative.

Should an error condition occur, all subsequent Jobs and Problems contained in the same run in which the error occurs *will be* terminated.

LISTING AND EXAMPLE PROBLEM

The example problem shown is a two-lag model with autoregressive errors -- equation 1. There are 27 observations for each of seven variables, which include two variables associated with the lag λ, two variables associated with the lag y, two non-lagged variables, and the dependent variable, Y.

```
      PROGRAM AUTOREG(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)            A
      DIMENSION KZ(15), T(35), VB(35), E(47), C(35,35), B(35,47), A(12), A
     1 X(47,47), XM(47), TSQR(35), FTSQR(35), STUDT(35), Y(375,8), XS(47 A
     2), IDENT(35), KDENT(4)                                            A
      DIMENSION VALUES(9,100)                                           A
      COMMON A,T,IN1,IN2,IN3                                            A
C-----                                                                  A
C-----SSE3=SSE1= A MAGIC NUMBER WHICH IS ATTEMPTED UPPER LIMIT OF SS ERR A
      DATA SSE3/37767777777777777777B/                                  A
C-----                                                                  A
C-----TEST1 IS AN ARBITRARILY SMALL NUMBER USED IN CHECKING MATRIX INVER A
1     TEST1=0.0000000001                                                A
      NN=0                                                              A
C-----READ AND PRINT DATA AND CONTROL CARDS.                             A
      READ (5,143) IPROB,NVAR,NOBS,NJOBS                                A
      IF (IPROB.EQ.-0) GO TO 138                                        A
      WRITE (6,144) IPROB,NVAR,NOBS,NJOBS                               A
      WRITE (6,174) NVAR,NOBS                                           A
C-----Y(I,J) IS THE ARRAY OF ORIGINAL VARIABLES.                         A
      DO 2 I=1,NOBS                                                     A
      READ (5,145) (Y(I,J),J=1,NVAR)                                    A
2     WRITE (6,173) (Y(I,J),J=1,NVAR)                                   A
3     SSE2=SSE3                                                         A
      READ (5,147) JOBNO,IA,IB,ID,IRHS,NSKIP,ISS1,ISS2,IERR,ICORR,TEST,N A
     1AME1,(KDENT(I),I=1,4)                                             A
      WRITE (6,146)                                                     A
      WRITE (6,148) JOBNO,IA,IB,ID,IRHS,NSKIP,ISS1,ISS2,IERR,ICORR,TEST, A
     1NAME1,(KDENT(I),I=1,4)                                            A
C-----                                                                  A
C-----SET UP INDICES                                                     A
C-----IA= NUMBER OF X VARIABLES ASSOCIATED WITH A LAG PARAMETER          A
C-----IB = NUMBER OF Z VARIABLES ASSOCIATED WITH A LAG PARAMETER         A
C-----ID = NUMBER OF UNLAGGED D VARIABLES                                A
C-----IL = NUMBER OF LAGGED VARIABLES                                    A
C-----IX = TOTAL NUMBER OF VARIABLES INCLUDING GERATED ONES              A
      IL=IA+IB                                                          A
      IX=3*IL+ID+4                                                      A
      INFM1=IL+ID+3                                                     A
      IN1=IL+1                                                          A
      IN2=IL+2                                                          A
      IN3=IL+3                                                          A
      INF=INFM1+1                                                       A
      IM1=3*IL+1                                                        A
      IM2=IM1+1                                                         A
      IM3=IM2+1                                                         A
      OBS=NOBS-3                                                        A
C-----                                                                  A
      IF (NAME1.EQ.1) GO TO 15                                          A
      DO 4 I=1,IX                                                       A
      DO 4 J=1,IX                                                       A
      X(I,J)=0.0                                                        A
      XS(I)=0.0                                                         A
4     XM(I)=0.0                                                         A
C-----                                                                  A
C-----GENERATE NEEDED LAGGED INDEPENDENT AND/OR DEPENDENT VARIABLES.     A
C-----STORE APPROPRIATE SQUARES AND CROSS PRODUCTS IN XS(IX)             A
```

```
C-----FINALLY PUT THE SUM OF SQUARES AND CROSS PRODUCTS INTO X(IX,IX).
      DO 10 I=4,NOBS
      IF (IL.EQ.0) GO TO 6
      DO 5 K=1,IL
      E(3*K-2)=Y(I,K)
      E(3*K-1)=Y(I-1,K)
      E(3*K)=Y(I-2,K)
5     CONTINUE
6     E(IM1)=Y(I-1,IRHS)
      E(IM2)=Y(I-2,IRHS)
      E(IM3)=Y(I-3,IRHS)
      IF (ID.EQ.0) GO TO 8
      DO 7 L=1,ID
      M=IL+L
      N=IM3+L
7     E(N)=Y(I,M)
8     E(IX)=Y(I,IRHS)
      DO 9 M=1,IX
      XS(M)=XS(M)+E(M)
      DO 9 N=1,IX
9     X(M,N)=X(M,N)+E(M)*E(N)
10    CONTINUE
C-----
      IF (ISS2.NE.1) GO TO 12
C-----
C-----GET X = SUM OF SQUARES + CROSS PRODUCTS CORRECTED
      DO 11 I=1,IX
      DO 11 J=1,IX
11    X(I,J)=X(I,J)-(XS(I)*XS(J))/OBS
C-----
C-----CALCULATE AND PRINT CORRELATION MATRIX IF REQUESTED.
C-----CORRELATION MATRIX IS C(IX,IX)
12    IF (ICORR.NE.1) GO TO 16
      DO 13 I=1,IX
      DO 13 J=1,IX
13    C(I,J)=X(I,J)/((X(I,I)*X(J,J))**0.5)
      WRITE (6,150)
      WRITE (6,149) (I,I=1,IX)
      DO 14 I=1,IX
14    WRITE (6,151) I,(C(I,J),J=1,I)
C-----
15    IDF=NOBS-IL-ID-6+NSKIP-ISS2
C-----
C-----START WITH ITERATION = 1
C-----READ CONTROL CARDS 3 AND 4, COEF. EST. = T(INFM1)
16    IY=1
      DF=IDF
      READ (5,152) (KZ(I),I=1,NSKIP)
      READ (5,153) (T(I),I=1,INFM1)
      DO 17 I=1,INFM1
17    VALUES(I,1)=T(I)
C-----
C-----CALCULATE MEANS OF ALL VARIABLES.
      DO 18 I=1,IX
18    XM(I)=XS(I)/OBS
      DO 19 I=1,IX
```

```
19     E(I)=0.0
C-----
       CALL ACOM
C-----
C-----WRITE HEADER FOR ITERATIVE PROCEDURES
       WRITE (6,176)
       WRITE (6,155)
       WRITE (6,154)
C-----
C-----B=AXX
20     DO 21 I=1,INF
       DO 21 J=1,IX
21     B(I,J)=0.
       IF (IA) 22,25,23
22     K=107
       GO TO 139
23     DO 24 I=1,IA
       DO 24 J=1,IX
       B(I,J)=B(I,J)+X(3*I-2,J)+A(2)*X(3*I,J)-A(1)*X(3*I-1,J)
       B(IN2,J)=B(IN2,J)+T(I)*(A(4)*X(3*I,J)-X(3*I-1,J))
       B(IN3,J)=B(IN3,J)+T(I)*(A(3)*X(3*I,J)-X(3*I-1,J))
24     B(INF,J)=B(INF,J)+T(I)*(A(1)*X(3*I-1,J)-X(3*I-2,J)-A(2)*X(3*I,J))
25     IF (IB) 26,29,27
26     K=135
       GO TO 139
27     DO 28 I=1,IB
       DO 28 J=1,IX
       M=(3*IA)+(3*I-2)
       K=IA+I
       B(K,J)=B(K,J)+X(M,J)+A(6)*X(M+2,J)-A(5)*X(M+1,J)
       B(IN1,J)=B(IN1,J)+T(K)*(A(4)*X(M+2,J)-X(M+1,J))
       B(IN3,J)=B(IN3,J)+T(K)*(A(7)*X(M+2,J)-X(M+1,J))
28     B(INF,J)=B(INF,J)+T(K)*(A(5)*X(M+1,J)-X(M,J)-A(6)*X(M+2,J))
29     DO 30 J=1,IX
       B(IN1,J)=B(IN1,J)+X(IM1,J)+A(2)*X(IM3,J)-A(1)*X(IM2,J)
       B(IN2,J)=B(IN2,J)+X(IM1,J)+A(6)*X(IM3,J)-A(5)*X(IM2,J)
       B(IN3,J)=B(IN3,J)+X(IM1,J)+A(10)*X(IM3,J)-A(9)*X(IM2,J)
30     B(INF,J)=B(INF,J)+A(11)*X(IM2,J)-A(8)*X(IM1,J)-A(12)*X(IM3,J)
       IF (ID) 31,34,32
31     K=191
       GO TO 139
32     DO 33 I=1,ID
       DO 33 J=1,IX
       M=IM3+I
       K=IN3+I
       B(INF,J)=B(INF,J)-T(K)*X(M,J)
33     B(K,J)=X(M,J)
34     DO 35 J=1,IX
35     B(INF,J)=B(INF,J)+X(IX,J)
C-----
C-----C=BA
       DO 36 I=1,INF
       DO 36 J=1,INF
36     C(I,J)=0.
       IF (IA) 37,40,38
37     K=303
```

```
      GO TO 139
38    DO 39 J=1,IA
      DO 39 I=1,INF
      C(I,J)=C(I,J)+B(I,3*J-2)+A(2)*B(I,3*J)-A(1)*B(I,3*J-1)
      C(I,IN2)=C(I,IN2)+T(J)*(A(4)*B(I,3*J)-B(I,3*J-1))
      C(I,IN3)=C(I,IN3)+T(J)*(A(3)*B(I,3*J)-B(I,3*J-1))
39    C(I,INF)=C(I,INF)+T(J)*(A(1)*B(I,3*J-1)-A(2)*B(I,3*J)-B(I,3*J-2))
40    IF (IB) 41,44,42
41    K=337
      GO TO 139
42    DO 43 J=1,IB
      DO 43 I=1,INF
      K=IA+J
      M=(3*IA)+(3*J-2)
      C(I,K)=C(I,K)+B(I,M)+A(6)*B(I,M+2)-A(5)*B(I,M+1)
      C(I,IN1)=C(I,IN1)+T(K)*(A(4)*B(I,M+2)-B(I,M+1))
      C(I,IN3)=C(I,IN3)+T(K)*(A(7)*B(I,M+2)-B(I,M+1))
43    C(I,INF)=C(I,INF)+T(K)*(A(5)*B(I,M+1)-B(I,M)-A(6)*B(I,M+2))
44    DO 45 I=1,INF
      C(I,IN1)=C(I,IN1)+B(I,IM1)+A(2)*B(I,IM3)-A(1)*B(I,IM2)
      C(I,IN2)=C(I,IN2)+B(I,IM1)+A(6)*B(I,IM3)-A(5)*B(I,IM2)
      C(I,IN3)=C(I,IN3)+B(I,IM1)+A(10)*B(I,IM3)-A(9)*B(I,IM2)
45    C(I,INF)=C(I,INF)+A(11)*B(I,IM2)-A(8)*B(I,IM1)-A(12)*B(I,IM3)
      IF (ID) 46,49,47
46    K=397
      GO TO 139
47    DO 48 I=1,INF
      DO 48 J=1,ID
      M=IM3+J
      N=IN3+J
      C(I,N)=B(I,M)
48    C(I,INF)=C(I,INF)-T(N)*B(I,M)
49    DO 50 I=1,INF
50    C(I,INF)=C(I,INF)+B(I,IX)
C-----
C-----SET TO SKIP OR FIX THETAS
      IF (NSKIP) 51,54,52
51    K=453
      GO TO 139
52    DO 53 I=1,NSKIP
      J=KZ(I)
      DO 53 K=1,INF
      C(J,K)=0.
53    C(K,J)=0.
C-----
C-----INVERT MATRIX
54    DO 62 J=1,INFM1
      DO 55 I=1,NSKIP
      IF (J-KZ(I)) 55,62,55
55    CONTINUE
      SAVE=C(J,J)
      IF (SAVE-TEST1) 56,58,58
56    IF (SAVE+TEST1) 58,58,57
57    K=524
      GO TO 139
58    PIVOT=1.0/SAVE
```

```
      C(J,J)=-1.0
      DO 59 I=1,INFM1
59    E(I)=-C(I,J)*PIVOT
      C(J,J)=SAVE
      DO 60 K=1,INF
      TEMP=C(J,K)
      C(J,K)=0.
      DO 60 I=1,INFM1
60    C(I,K)=C(I,K)+E(I)*TEMP
      DO 61 I=1,INFM1
61    C(I,J)=E(I)
62    CONTINUE
      IV=1
C-----
C-----ADJUST PARAMETER ESTIMATES.
      DO 63 I=1,INFM1
63    T(I)=T(I)+C(I,INF)
      SSE1=SSE2
C-----
64    CALL ACOM
C-----
C-----COMPUTE SSE
      DO 76 J=1,IX
      E(J)=X(IX,J)+A(11)*X(IM2,J)-A(8)*X(IM1,J)-A(12)*X(IM3,J)
      IF (IA) 65,68,66
65    K=607
      GO TO 139
66    DO 67 I=1,IA
      M=3*I-2
67    E(J)=E(J)+T(I)*(A(1)*X(M+1,J)-X(M,J)-A(2)*X(M+2,J))
68    IF (IB) 69,72,70
69    K=617
      GO TO 139
70    DO 71 I=1,IB
      M=(3*IA)+(3*I-2)
      K=IA+I
71    E(J)=E(J)+T(K)*(A(5)*X(M+1,J)-X(M,J)-A(6)*X(M+2,J))
72    IF (ID) 73,76,74
73    K=627
      GO TO 139
74    DO 75 I=1,ID
      M=IM3+I
      N=IN3+I
75    E(J)=E(J)-T(N)*X(M,J)
76    CONTINUE
      SSE2=E(IX)+A(11)*E(IM2)-A(8)*E(IM1)-A(12)*E(IM3)
      IF (IA) 77,80,78
77    K=641
      GO TO 139
78    DO 79 I=1,IA
      M=3*I-2
79    SSE2=SSE2+T(I)*(A(1)*E(M+1)-E(M)-A(2)*E(M+2))
80    IF (IB) 81,84,82
81    K=653
      GO TO 139
82    DO 83 I=1,IB
```

```
      M=(3*IA)+(3*I-2)
      K=IA+I
83    SSE2=SSE2+T(K)*(A(5)*E(M+1)-E(M)-A(6)*E(M+2))
84    IF (ID) 85,88,86
85    K=665
      GO TO 139
86    DO 87 I=1,ID
      M=IM3+I
      N=IN3+I
87    SSE2=SSE2-T(N)*E(M)
88    DF=IDF
      ESSM=SSE2/DF
      TMESS=X(IX,IX)-SSE2
      RSQR=TMESS/X(IX,IX)
C-----
C-----IY = ITERATION ON TEST FOR COEFFICIENT CHANGES.
C-----IV = ITERATION ON TEST FOR SS ERROR CHANGES
C-----IDF = DEGREES OF FREEDOM.
C-----X = SUM OF SQUARES MATRIS
C-----SSE2 = SUM OF SQUARES FOR ERROR.
C-----
C-----CHECK FOR REDUCTION IN SS ERROR
C-----IF SS ERROR IS SMALL ENOUGH THEN GO ON,
C-----  OTHERWISE HALVE THE INCREMENT IN PARAMETER CHANGES AND ITERATE.
      IF (SSE1-SSE2) 89,89,92
89    IV=2*IV
      IF (IV-9999) 90,90,92
90    DO 91 I=1,INFM1
C-----ADD 1/2 OF THETA CHANGE EACH ITERATION UNTIL SS ERROR IS REDUCED F
      C(I,INF)=C(I,INF)/2.
91    T(I)=T(I)-C(I,INF)
      GO TO 64
C-----
92    CONTINUE
C-----
C-----T(I) = THETA
C-----C(I,INF) = DTHETA
C-----E(I) = TEST
C-----VB(I) = VARIANCE
C-----FTSQR(I) = STD. ERROR
C-----STUDT(I) = STUDENT T
      SMS=IV*IV
      DO 95 I=1,INFM1
      IF (C(I,I)) 93,94,93
93    VB(I)=C(I,I)*ESSM
      TSQR(I)=T(I)*T(I)
      E(I)=((C(I,INF)*C(I,INF))*SMS)/VB(I)
      FTSQR(I)=TSQR(I)/VB(I)
      STUDT(I)=SQRT(FTSQR(I))
      FTSQR(I)=SQRT(VB(I))
      GO TO 95
94    VB(I)=0.
      C(I,INF)=0.
      E(I)=0.
      FTSQR(I)=0.
      STUDT(I)=0.0
```

```
95     CONTINUE
C-----
       ITEST=0
       DO 97 I=1,INFM1
       IF (E(I)-TEST) 97,97,96
96     ITEST=1
97     CONTINUE
C-----
       IF (ISS1-IY) 98,101,98
98     IY=IY+1
       DO 99 I=1,INFM1
99     VALUES(I,IY)=T(I)
       WRITE (6,156) IY,IV,IDF,X(IX,IX),SSE2,ESSM,TMESS,RSQR
C-----
C-----IF ITEST IS TOO LARGE GO THROUGH ANOTHER ITERATION
C-----THAT IS, IF ANY PARAMETER CHANGE IS LARGER THAN ALLOWABLE MINIMUM
       IF (ITEST) 100,101,20
100    K=727
       GO TO 139
C-----
C-----PRINT PARAMETER ESTIMATES
101    WRITE (6,175)
C-----IDENT(I) IS THE LIST OF  PARAMETER IDENTIFIERS.
       IADD=10000000000000B
       IDENT(1)=6HX 1
       DO 102 II=2,IA
102    IDENT(II)=IDENT(II-1)+IADD
       IDENT(IA+1)=6HZ 1
       LL=IA+2
       DO 103 II=LL,IL
103    IDENT(II)=IDENT(II-1)+IADD
       IDENT(IL+1)=6HLAMBDA
       IDENT(IL+2)=6HMU
       IDENT(IL+3)=6HBETA
       IDENT(IL+4)=6HD 1
       LL=IL+5
       DO 104 II=LL,INFM1
104    IDENT(II)=IDENT(II-1)+IADD
C-----
C-----PRINT VALUES FOR EACH ITERATION
       PRINT 140
       PRINT 141, (IDENT(I),I=1,INFM1)
       DO 105 J=1,IY
       K=J-1
105    PRINT 142, K,(VALUES(I,J),I=1,INFM1)
       PRINT 175
       WRITE (6,176)
       WRITE (6,157)
       DO 106 I=1,INFM1
106    WRITE (6,158) I,IDENT(I),T(I),C(I,INF),E(I),VB(I),FTSQR(I),STUDT(I
      1)
       WRITE (6,159) INFM1,INFM1
C-----
C-----PRINT THE INVERSE MATRIX
       DO 107 I=1,INFM1
107    WRITE (6,160) (C(I,J),J=1,I)
```

```
C-----
C-----CALCULATE THE CONSTANT TERM
      CONST=0.
      IF (ISS2-1) 121,108,121
108   CONST=XM(IX)
      IF (IA) 109,112,110
109   K=738
      GO TO 139
C-----
110   DO 111 I=1,IA
111   CONST=CONST+T(I)*(A(1)*XM(3*I-1)-XM(3*I-2)-A(2)*XM(3*I))
112   IF (IB) 113,116,114
113   K=743
      GO TO 139
C-----
114   DO 115 I=1,IB
      M=(3*IA)+(3*I-2)
      K=IA+I
115   CONST=CONST+T(K)*(A(5)*XM(M+1)-XM(M)-A(6)*XM(M+2))
116   CONST=CONST+A(11)*XM(IM2)-A(8)*XM(IM1)-A(12)*XM(IM3)
      IF (ID) 117,120,118
117   K=760
      GO TO 139
C-----
118   DO 119 I=1,ID
      M=IM3+I
      N=IN3+I
119   CONST=CONST-T(N)*XM(M)
120   WRITE (6,161) CONST
C-----
C-----CALCULATE A0=CONST/(1-LAMBDA)(1-MU)(1-BETA)
      DENOM=(1-T(IL+1))*(1-T(IL+2))*(1-T(IL+3))
      A0=CONST/DENOM
      WRITE (6,162) A0
121   IF (IERR.NE.1) GO TO 131
C-----
C-----CALCULATE PREDICTED VALUES OF Y
C-----
      DO 128 I=4,NOBS
      Y(I,NVAR+1)=CONST
      IF (IA.EQ.0) GO TO 123
      DO 122 J=1,IA
122   Y(I,NVAR+1)=Y(I,NVAR+1)+T(J)*(Y(I,J)-A(1)*Y(I-1,J)+A(2)*Y(I-2,J))
123   IF (IB.EQ.0) GO TO 125
      DO 124 J=1,IB
      M=IA+J
124   Y(I,NVAR+1)=Y(I,NVAR+1)+T(M)*(Y(I,M)-A(5)*Y(I-1,M)+A(6)*Y(I-2,M))
125   IF (ID.EQ.0) GO TO 127
      DO 126 J=1,ID
      M=IL+J
      N=IN3+J
126   Y(I,NVAR+1)=Y(I,NVAR+1)+T(N)*Y(I,M)
127   Y(I,NVAR+1)=Y(I,NVAR+1)+A(8)*Y(I-1,IRHS)-A(11)*Y(I-2,IRHS)+A(12)*Y
     1(I-3,IRHS)
128   CONTINUE
C-----
```

```
      SSDIF=0.0
C-----
C-----PRINT OBSERVATIONS, PREDICTED VALUES AND DEVIATIONS.
      WRITE (6,176)
      WRITE (6,163)
      SDEV1=0.0
      DO 130 I=4,NOBS
      DEV1=Y(I,IRHS)-Y(I,NVAR+1)
      SDEV1=SDEV1+DEV1*DEV1
      N=I-3
      WRITE (6,164) N,Y(I,IRHS),Y(I,NVAR+1),DEV1
      IF (I.EQ.4) GO TO 129
      SSDIF=SSDIF+(DEV2-DEV1)*(DEV2-DEV1)
129   DEV2=DEV1
130   CONTINUE
C-----
      DUBWA=SSDIF/SDEV1
      WRITE (6,165) DUBWA
C-----
C-----PRINT MEANS OF VARIABLES
131   WRITE (6,166)
      IF (IA.EQ.0) GO TO 133
      DO 132 I=1,IA
      LL=3*(I-1)+1
      LU=3*I
132   WRITE (6,167) I,(XM(II),II=LL,LU)
133   IF (IB.EQ.0) GO TO 135
      DO 134 I=1,IB
      LL=3*IA+3*(I-1)+1
      LU=3*(IA+I)
134   WRITE (6,168) I,(XM(II),II=LL,LU)
135   LL=3*IL+1
      LU=LL+2
      WRITE (6,170) XM(IX),(XM(II),II=LL,LU)
      IF (ID.EQ.0) GO TO 137
      DO 136 I=1,ID
      II=3*(IL+1)+I
136   WRITE (6,169) I,XM(II)
137   WRITE (6,171) JOBNO
      NN=NN+1
      IF (NN-NJOBS) 3,1,138
C-----
138   CALL EXIT
C-----
C-----TERMINATE DUE TO AN ERROR.
139   WRITE (6,172) K
C-----INPUT-OUTPUT FORMATS.
C-----
140   FORMAT (1H0,*VALUES OF PARAMETER ESTIMATES AT EACH ITERATION*)
141   FORMAT (1H0,5X,9(A6,8X)//)
142   FORMAT (1H ,I3,9(1X,E13.6))
143   FORMAT (5I5)
144   FORMAT (48H1PROBLEM NUMBER - NO. VAR. - NO. OBS. - NO. JOBS/5X,I5,
     17X,I5,6X,I5,6X,I5)
145   FORMAT (7F10.4)
146   FORMAT (6H1JOBNO,2X,1HA,3H B ,3H C ,5H RHS ,7H NSKIP ,7H NITER ,6H
     1 CONS ,5H ERR ,6H CORR ,12H    TEST     ,8H NWSSCP ,/)
147   FORMAT (I5,6I2,3I1,F10.4,I1,4A10)
148   FORMAT (1X,I5,I3,I2,I3,I4,I6,I7,I6,I6,I5,4X,F10.8,I6,10X,4A10,//)
149   FORMAT (1H0,3X,21I6/)
150   FORMAT (20H  CORRELATION MATRIX)
151   FORMAT (1H I3,3X,21F6.3)
152   FORMAT (15(I2,2X))
153   FORMAT (8F10.5)
154   FORMAT (* ITER  STEP  IDF*,5X,*TSS*,14X,*SSE*,14X,*MSE*,14X,*SSR*,
     1 12X,*R SQUARE*//)
155   FORMAT (*-ITERATIVE CALCULATIONS FOR COEFFICIENT DETERMINATION*//)
156   FORMAT (1X,3I5,E15.8,4(2X,E15.8))
157   FORMAT (21X,5HTHETA11X,7HD THETA12X,4HTEST11X,8HVARIANCE9X,8HSTAND
     1 ER8X,9HSTUDENT T//)
158   FORMAT (1X,I5,3X,A6,6(2X,E15.8))
159   FORMAT (23H-SIZE OF INVERSE MATRIXI10,I10/15H INVERSE MATRIX//)
160   FORMAT (10(2X,E11.4))
161   FORMAT (//10H CONSTANT=E15.8//)
162   FORMAT (*0*,6X,*A0=*,E15.8)
163   FORMAT (10X,10HOBSERVED Y,7X,11HPREDICTED Y,7X,8HRESIDUAL)
164   FORMAT (1X,I4,3X,E15.8,2X,E15.8,2X,E15.8)
165   FORMAT (23H-    DURBIN-WATSON D     ,F10.8)
166   FORMAT (*-VARIABLE MEANS AT*,15X,*T*,15X,*T-1*,14X,*T-2*,14X,*T-3*
     1)
167   FORMAT (*0      VARIABLE NO. X*,I1,4X,3(2X,E15.8))
168   FORMAT (*0      VARIABLE NO. Z*,I1,4X,3(2XE15.8))
169   FORMAT (*0      VARIABLE NO. D*,I1,6X,E15.8)
170   FORMAT (*0      DEPENDENT VARIABLE*,1X,4(2X,E15.8))
171   FORMAT (13H END OF JOBNOI7/1H1)
172   FORMAT (6H ERRORI4)
173   FORMAT (1H ,10F10.4)
174   FORMAT (1H0,*ORIGINAL DATA HAVE*,I5,* VARIABLES FOR*,I5,* OBVERSAT
     1IONS.*)
175   FORMAT (1H0)
176   FORMAT (1H1)
      END
```

```
      SUBROUTINE ACOM
C-----
C-----ACOM CALCULATES COEFFICIENTS FROM THE PARAMETER ESTIMATES.
      DIMENSION A(12), T(35)
      COMMON A,T,IN1,IN2,IN3
1     A(1)=T(IN2)+T(IN3)
      A(2)=T(IN2)*T(IN3)
      A(3)=T(IN2)
      A(4)=T(IN3)
      A(5)=T(IN1)+T(IN3)
      A(6)=T(IN1)*T(IN3)
      A(7)=T(IN1)
      A(8)=A(1)+A(7)
      A(9)=T(IN1)+T(IN2)
      A(10)=T(IN1)*T(IN2)                                              B
      A(11)=A(9)*A(4)+A(10)                                            B
      A(12)=A(10)*A(4)                                                 B
      RETURN                                                           B
      END                                                              B
```

```
  01     7     27     1       THIS IS THE GENERATED TEST DECK FOR MRHS-1      PROBCARI
   52.625     59.250      .2888    45.567     46.2357     31.333    2701.12
   55.375     61.833      .2888    35.567     45.2354
   50.166     56.583      .2888    35.5674    45.234      31.333    2560.81
  50.0000    62.0000    23.0000    45.0000    24.0000    24.0000  933.2568
  50.0000    62.0000    23.0000    54.0000    19.0000    16.0000 1578.2488
  54.0000    57.0000    27.0000    56.0000    30.0000    36.0000 2250.1733
  55.0000    60.0000    54.0000    37.0000    35.0000    42.0000 2953.6896
  55.0000    28.0000    45.0000    37.0000    34.0000    42.0000 3425.0970
  47.0000    51.0000    27.0000    56.0000    14.0000    38.0000 3840.4712
  49.0000    62.0000    54.0000    25.0000     8.0000    39.0000 3954.8383
  61.0000    62.0000    45.0000    38.0000    12.0000    27.0000 3979.7188
  64.0000    62.0000    47.0000    35.0000    31.0000    36.0000 4062.9793
  60.0000    59.0000    48.0000    57.0000    30.0000    33.0000 4302.4975
  55.0000    61.0000     6.0000    54.0000    40.0000    45.0000 4445.2611
  61.0000    65.0000    24.0000    35.0000    32.0000    30.0000 4444.0656
  56.0000    64.0000    45.0000    37.0000    31.0000    26.0000 4449.2607
  59.0000    60.0000    58.0000    67.0000    24.0000    19.0000 4490.5426
  54.0000    61.0000    23.0000    45.0000    30.0000    26.0000 4221.8082
  54.0000    62.0000    34.0000    56.0000    20.0000    33.0000 4214.8520
  53.0000    63.0000    26.0000    35.0000    12.0000    38.0000 4007.9200
  53.0000     3.0000    26.0000    45.0000    14.0000    28.0000 3702.0653
  54.0000    62.0000    37.0000    48.0000    22.0000    42.0000 3955.2186
  55.0000    66.0000    56.0000    45.0000    32.0000    40.0000 4225.5609
  56.0000    69.0000    34.0000    67.0000    36.0000    36.0000 4525.6412
  54.0000    69.0000    37.0000    56.0000    43.0000    20.0000 4538.4640
  59.0000    64.0000    26.0000    75.0000    33.0000    22.0000 4652.2580
  66.0000    66.0000    47.0000    70.0000    34.0000    14.0000 4632.4896
   1 2 2 2 7 099111     0.00010TWO LAG MODEL WITH AUTOREGRESSIVE ERRORS
                                                                       CONCARD2
2.2748     2.9439     4.1488     5.4830     0.1549     0.6837     0.4563     5.4936
7.1704
```

```
PROBLEM NUMBER - NO. VAR. - NO. OBS. - NO. JOBS
      1            7          27          1

ORIGINAL DATA HAVE    7 VARIABLES FOR   27 OBVERSATIONS.
   52.6250    59.2500      .2888    45.5670    46.2357    31.3330 2701.1200
   55.3750    61.8330      .2888    35.5670    45.2354    -0.0000   -0.0000
   50.1660    56.5830      .2888    35.5674    45.2340    31.3330 2560.8100
   50.0000    62.0000    23.0000    45.0000    24.0000    24.0000  933.2568
   50.0000    62.0000    23.0000    54.0000    19.0000    16.0000 1578.2488
   54.0000    57.0000    27.0000    56.0000    30.0000    36.0000 2250.1733
   55.0000    60.0000    54.0000    37.0000    35.0000    42.0000 2953.6896
   55.0000    28.0000    45.0000    37.0000    34.0000    42.0000 3425.0970
   47.0000    51.0000    27.0000    56.0000    14.0000    38.0000 3840.4712
   49.0000    62.0000    54.0000    25.0000     8.0000    39.0000 3954.8383
   61.0000    62.0000    45.0000    38.0000    12.0000    27.0000 3979.7188
   64.0000    62.0000    47.0000    35.0000    31.0000    36.0000 4062.9793
   60.0000    59.0000    48.0000    57.0000    30.0000    33.0000 4302.4975
   55.0000    61.0000     6.0000    54.0000    40.0000    45.0000 4445.2611
   61.0000    65.0000    24.0000    35.0000    32.0000    30.0000 4444.0656
   56.0000    64.0000    45.0000    37.0000    31.0000    26.0000 4449.2607
   59.0000    60.0000    58.0000    67.0000    24.0000    19.0000 4490.5426
   54.0000    61.0000    23.0000    45.0000    30.0000    26.0000 4221.8082
   54.0000    62.0000    34.0000    56.0000    20.0000    33.0000 4214.8520
   53.0000    63.0000    26.0000    35.0000    12.0000    38.0000 4007.9200
   53.0000     3.0000    26.0000    45.0000    14.0000    28.0000 3702.0653
   54.0000    62.0000    37.0000    48.0000    22.0000    42.0000 3955.2186
   55.0000    66.0000    56.0000    45.0000    32.0000    40.0000 4225.5609
   56.0000    69.0000    34.0000    67.0000    36.0000    36.0000 4525.6412
   54.0000    69.0000    37.0000    56.0000    43.0000    20.0000 4538.4640
   59.0000    64.0000    26.0000    75.0000    33.0000    22.0000 4652.2580
   66.0000    66.0000    47.0000    70.0000    34.0000    14.0000 4632.4896
```

```
JOBNO  A B  C  RHS  NSKIP  NITER  CONS  ERR  CORR      TEST      NWSSCP

    1  2 2  2   7     0     99     1     1    1     .00010000     0          TWO LAG MODEL WITH AUTOREGRESSIVE ERRORS


       ION MATRIX

        1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18
  1   1.000
  2    .500 1.000
  3    .013  .444 1.000
  4    .207  .094  .049 1.000
  5    .358  .173  .072  .030 1.000
  6    .225  .333  .160 -.078  .019 1.000
  7    .328  .055 -.254  .103  .043 -.394 1.000
  8    .181  .360  .003 -.058  .085 -.005  .151 1.000
  9    .199  .343  .345  .024 -.033  .062 -.039  .342 1.000
 10    .205  .208  .291  .167  .126  .393 -.166  .169  .097 1.000
 11    .082  .094  .139  .356  .138  .086 -.179 -.110  .287  .203 1.000
 12    .230 -.131 -.012 -.179  .342  .094  .021 -.161  .028  .148  .087 1.000
 13    .451  .498  .485  .135  .129  .049  .115  .298  .549  .131  .176  .204 1.000
 14    .438  .467  .356  .106  .103  .053  .083  .353  .496  .189  .178  .195  .833 1.000
 15    .441  .428  .439  .139  .090  .080  .004  .113  .400  .191  .163  .135  .885  .729 1.000
 16    .410  .559  .228  .267  .331  .218 -.023  .136  .145  .338  .403  .086  .148  .033  .092 1.000
                                                                                              025 1.000
 18    .487  .528  .390  .146  .082 -.005  .731  .471  .632  .182  .725  .244  .892  .891  .723  .225  .058 1.000
```

```
ITERATIVE CALCULATIONS FOR COEFFICIENT DETERMINATION

ITER  STEP   IDF     TSS              SSE              MSE               SSP              R SQUARE

   2     1    14  .21607277E+08    .44189524E+08    .31563945E+07    -.22582246E+08    -.10451223E+01
   3     1    14  .21607277E+08    .29256298E+07    .20897356E+06     .18681647E+08     .86459979E+00
   4     1    14  .21607277E+08    .82866863E+06    .59190616E+05     .20778609E+08     .96164863E+00
   5     1    14  .21607277E+08    .76844337E+06    .54888812E+05     .20838834E+08     .96443590E+00
   6     1    14  .21607277E+08    .76838981E+06    .54884986E+05     .20838887E+08     .96443838E+00
   7     1    14  .21607277E+08    .76838592E+06    .54884709E+05     .20838891E+08     .96443856E+00


VALUES OF PARAMETER ESTIMATES AT EACH ITERATION

     X 1            X 2            Z 1            Z 2           LAMBDA         MU             BETA           D 1            D 2

                                                 548300E+01    .154900E+00    .683700E+00    .456300E+00    .549360E+01    .717040E+01
 1   .117564E+02    .481304E-01   .861885E+01   .950673E+01   -.305561E+00    .147477E+01   -.614552E+00    .101065E+02    .282291E+02
 2   .120899E+02   -.771418E+00   .803591E+01   .107038E+02    .406335E-01    .107422E+01   -.529207E+00    .600216E+01    .285559E+02
 3   .129914E+02   -.992371E+00   .810461E+01   .112168E+02    .248978E+00    .908576E+00   -.586733E+00    .686755E+01    .290232E+02
 4   .111774E+02   -.838607E+00   .820632E+01   .977143E+01    .233209E+00    .870667E+00   -.570842E+00    .800101E+01    .279886E+02
 5   .110969E+02   -.840292E+00   .824738E+01   .990005F+01    .236798E+00    .870752E+00   -.570829E+00    .795202E+01    .280459E+02
 6   .110476E+02   -.837233E+00   .825471E+01   .985452E+01    .234861E+00    .870863E+00   -.570463E+00    .797910E+01    .280104E+02
```

```
                 THETA           D THETA          TEST           VARIANCE        STAND ER        STUDENT T


 1   X 1     .11042581E+02   -.54293446E-01   .17478660E-04   .16865013E+03   .12986536E+02   .85030996E+00
                                                              .91909982E+01   .30316659E+01   .27616282E+00
 3   Z 1     .82547055E+01    .73225393E-02   .33669575E-05   .15925233E+02   .39906431E+01   .20685151E+01
 4   Z 2     .98545235E+01   -.45523835E-01   .78626799E-04   .26357674E+02   .51339725E+01   .19194734E+01
 5   LAMBDA  .23486122E+00   -.19365600E-02   .93783682E-04   .39988457E-01   .19997114E+00   .11744756E+01
 6   MU      .87086305E+00    .11133750E-03   .65312487E-05   .18979583E-02   .43565564E-01   .19989712E+02
 7   BETA   -.57046266E+00    .36636861E-03   .21881732E-04   .61341560E-02   .78320853E-01   .72836625E+01
 8   D 1     .79790991E+01    .27081265E-01   .19593688E-04   .37430160E+02   .61180193E+01   .13041965E+01
 9   D 2     .28010397E+02   -.35456578E-01   .25851905E-04   .48629643E+02   .69734958E+01   .40166938E+01

SIZE OF INVERSE MATRIX          9          9
INVERSE MATRIX


  .3073E-02
 -.1216E-04    .1675E-03
 -.4373E-03   -.3618E-04    .2902E-03
  .4058E-03   -.3737E-04   -.2104E-04    .4802E-03
  .1816E-04   -.2384E-06   -.5321E-05    .8755E-05    .7286F-06
                                                                    3458E-07
 -.3103E-05   -.3520E-07    .1635E-05   -.1003E-05   -.1718E-06    .9957F-08    .1118E-06
 -.4288E-03    .8377E-05    .3694E-04   -.2991E-03   -.4999E-05   -.8320E-07    .6960E-08    .6820E-03
  .6877E-03   -.6502E-04   -.9722E-04    .3553E-03    .8387E-05    .7012E-07   -.2018E-05   -.2305E-03    .8860E-03

CONSTANT= -.14312853E+04


      A0= -.92237572E+04
```

```
        ORSERVED Y       PREDICTED Y       RESIDUAL
   1    .93325680E+03    .12624443E+04   -.32918751E+03
   2    .15782488E+04    .17450934E+04   -.16684459E+03
   3    .22501733E+04    .18422301E+04    .40794325E+03
   4    .29536896E+04    .29101242E+04    .43565426E+02
   5    .34250970E+04    .34206667E+04    .44302566E+01
   6    .38404712E+04    .35118994E+04    .32857177E+03
   7    .39548383E+04    .38054180E+04    .14942032E+03
   8    .39797188E+04    .38874829E+04    .92235884E+02
   9    .40629793E+04    .43077802E+04   -.24480086E+03
  10    .43024975E+04    .43536190E+04   -.51121477E+02
  11    .44452611E+04    .45705113E+04   -.12525020E+03
  12    .44440656E+04    .41609302E+04    .28313539E+03
                            586613E+04    .19059936E+03
  14    .44905426E+04    .44868240E+04    .37185647E+01
  15    .42218082E+04    .43188772E+04   -.97069029E+02
  16    .42148520E+04    .42816810E+04   -.66829037E+02
  17    .40079200E+04    .41126563E+04   -.10473631E+03
  18    .37020653E+04    .38005997E+04   -.98534377E+02
  19    .39552186E+04    .41479429E+04   -.19272426E+03
  20    .42255609E+04    .43497192E+04   -.12415835E+03
  21    .45256412E+04    .46449464E+04   -.11930519E+03
  22    .45384640E+04    .43816030E+04    .15686101E+03
  23    .46522580E+04    .45865061E+04    .65751877E+02
  24    .46324896E+04    .46381615E+04   -.56719138E+01

   DURBIN-WATSON D    1.44999186

VARIABLE MEANS AT            T               T-1              T-2              T-3

     VARIABLE NO. X1       .55583333E+02    .54923583E+02    .54772542E+02

     VARIABLE NO. X2       .58333333E+02    .57940958E+02    .57850667E+02

     VARIABLE NO. Z1       .36333333E+02    .34387033E+02    .33315733E+02

     VARIABLE NO. Z2       .48958333E+02    .47523647E+02    .45880600E+02

     DEPENDENT VARIABLE    .38244324E+04    .37381124E+04    .35442684E+04    .34677124E+04
                                      02

     VARIABLE NO. D2       .31333333E+02
END OF JOBNO      1
```

Program NLINEAR

INTRODUCTION

NLINEAR is a generalized program for least-squares computation of nonlinear regressions, originally written by M. P. Lietzke (ORNL-3259, April 4, 1962) for an IBM 7090 computer. It is presented here in a modified form usable on the NCAR and SCOPE systems of a CDC 6400 computer.

The program is set up as a main program and a package of subroutines. The main program controls only input and output, so that the subroutines may be used separately in other FORTRAN programs to provide the complete nonlinear least squares analysis. A user-supplied, problem-specific subroutine must be included to define the functional form of the model to be fit and to define its partial derivatives with respect to the coefficients.

NLINEAR will provide a least squares fit to any function, linear or nonlinear, with up to eight coefficients and up to five independent variables. Weighting factors may either be defined or computed. The user must furnish a subroutine to define the function to be fitted and the partial derivatives with respect to the coefficients. The standard output of the calling program includes the values of the coefficients, the standard error in each coefficient, the variance of fit, and a point by point solution of the equation for each data point. The variance in the dependent variable is also computed for each data point. In addition, the inverse matrix may be printed out at the option of the user, as may the values of the parameters at each iteration.

PROGRAM OPERATION

The calling program reads the data as outlined in Table 1, calls subroutine NLLS to compute the regression, prints appropriate error messages if any are detected, prints the output, and checks to see if there is another set of data to be processed.

A listing of the complete program and associated subroutines is given below, as is a set of sample input and output.

Table 1. Input Data

Card No.	Field	Variable	Type	
1	1-72	Title	ALPHA	Up to 72 alphameric characters which will appear on each page o output.
2	1-4	NDP	INT	Number of data points $1 < NDP \leq 250$.
	5-6	NP	INT	Number of coefficients $1 \leq NP \leq 8$.
	7-8	NX	INT	Number of independent variables $1 \leq NX \leq 5$.
	9-12	NIT	INT	Number of iterations allowed on given case.
	13-22	EPS	REAL	Epsilon, the maximum allowed fractional difference between successive values of the coefficients at convergence.
	23	ITP	INT	= 0, do not list parameters at e iteration = 1, list parameters at each ite
	24	NINV	INT	= 0, do not list inverse matrix; = 1, list inverse matrix.
	The above variables are read with an (12A6/I4,2I2,I4,F10.0,2I1) for			
3	1-10	FACT	REAL	Step size of iterations
	Read with an (E 10.3) format.			
4	1-80	FMT	Alpha	Variable format for the data (cards 6 - on below)
	Read with an (10 A 8) format.			

Card No.	Field	Variable	Type	
5	1-10	GP1(1)	REAL	Guess on first coefficient
	11-20	GP1(2)	REAL	Guess on second coefficient.
	21-30	GP1(3)	REAL	Guess on third coefficient
	⋮	⋮	⋮	⋮
	61-70	GP1(6)	REAL	Guess on sixth coefficient
5	1-10	GP1(7)	REAL	Guess on seventh coefficient (omitted if NP < 7)
	Read with an (6 E 10.0) format			
6 - NDP + 5	...	X	REAL	independent variables = NX
		Y	REAL	dependent variable
		W	REAL	weighting factor (blank if none)
	Read according to (FMT) above.			

The above sequence may be repeated for as many data sets as desired. Two trailer cards must follow the last data set, the first containing ENDIT in columns 1-5, the second blank or any numeric characters.

Subroutine NLLS

The generalized least squares subroutine NLLS requires no common storage. To use the subroutine with any FORTRAN program it is necessary to include the following statements:

```
DIMENSION W(250), X(250,5), Y(250), GP1(8), STDE(8), YCALC(250), VARY(250),
A(8,8), B(8), XTX(8,8), TB(8), CONV(8), DERIV(8), C(8,8), VMAT(8,8).
CALL NLLS (W,X,Y,GP1,STDE,YCALC,VARY,NERR,EPS,NIT,NP,NDP,ITP,VARF,VARF1,C,
COREL,N).
```

The statement following the call of the subroutine NLLS should test the error indicator (NERR) and appropriate action should be taken if an error has occurred.

Note that the value of the independent variable is doubly subscripted.

The first index refers to the number of the data point and the second to the specified independent variable. For example, X(J,1) would refer to the jth data point of the first independent variable and X(J,5) to the jth data point of the fifth independent variable.

Subroutine NLLS calls a subroutine SUBRT which must be supplied by the user. This subroutine must evaluate the function to be fitted, the partial derivatives of the dependent variable with respect to the coefficients, the difference between the observed and calculated value of the dependent variable, and a statement on the weight to be assigned to each data point.

Subroutine NLLS does not have built-in scaling. Scaling must be done by the user, if necessary, on the data.

The following definitions apply:

W	Weighting factor (input, may be zero)
X	Independent variables
Y	Dependent variable
GP1	Initial guesses on the coefficients
STDE	Standard error in each coefficient
YCALC	Calculated value of Y for each data point using the converged values of the coefficients
VARY	Variance in Y
NERR	Error indicator 1 = no error 2 = singular matrix 3 = non-convergence within the specified number of iterations 4 = singular inverse matrix

EPS	Maximum allowed percentage difference between successive values of the coefficients at convergence
NIT	Number of iterations to be allowed
NP	Number of coefficients $(1 \leq NP \leq 8)$
NDP	Number of data points
VARF	Variance of fit
VARF1	$VARF/\sum_i W_i$
C	Inverse matrix

Subroutine SUBRT

SUBRT must evaluate the model being fit at the point called for, compute the difference between the calculated and the measured values, evaluate the first derivatives of the model with respect to each parameter, and compute or specify the weighting factor to be used.

An example of this user supplied subroutine for a simple exponential model follows, where

$$Y = (A_1) \quad (A_2)^X$$

so that

$$\frac{\partial y}{\partial A_1} = (A_2)^X$$

$$\frac{\partial y}{\partial A_2} = (A_1) \ (X) \ (A_2)^{X-1}$$

and assuming that all weights are unity.

```
SUBROUTINE SUBRT (J,W,X,U,GP1,DERIV,YC,F1,WI)
DIMENSION W(250),X(250,5),Y(250),GP1(8),DERIV(8)
YC=GP1(1)*GP1(2)**X(J,1)
GI=Y(J)-YC
DERIV(1)=GP1(2)**X(J,1)
DERIV(2)=GP1*X(J,1)*GP1(2)**(X(J,1)-1.)
WI=1.
RETURN
END
```

The following definitions apply:

J	Index on data point
W	Weighting factor (input, may be zero)
X	Independent variable
Y	Dependent variable (observed value)
GPI	Guess on value of coefficient. Thus GPI(1)=a_0 and GPI(2)=a_1 in the example above.
DERIV	Partial derivative of dependent variable with respect to coefficient.
YC	Value of dependent variable computed using the current guesses on the coefficients.
FI	Difference between the observed and calculated value of the dependent variable.
WI	Weight assigned to data point.

In case specific weights are read in for each data point then the statement

WI = W(J)

would be made in the subroutine. In the event weights are to be computed, then the appropriate statement would be included in SUBRT. If all weights are unity then WI = 1.0.

Any subroutine SUBRT written would have the same arguments and dimension statement as in the example.

SAMPLE RUN OF PROGRAM

A listing of program NLINEAR follows along with a sample input and the resultant output.

```
      PROGRAM NLINEAR
     1 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C PROGRAM MODIFIED FROM LIETZKE,M H 1962  A GENERALIZED LEAST SQUARES PROGRAM
C      NONLINEAR LEAST SQUARES
C FOR THE IBM 7090 COMPUTER. ORNL-3259
      DIMENSION          W(250),X(250,5),Y(250),GP(8),GP1(8),TITLE(12),
     1STDE(8),YCALC(250),VARY(250),COREL(8,8),C(8,8)
      INTEGER FMT(10)
      COMMON FACT
      TYPE INTEGER YES, NO

C DEFINITIONS
C EPS MAX ALLOWED PERCENTAGE DIFFERENCE BETWEEN SUCCESSIVE VALUES OF THE
C COFFICIENTS AT CONVERGENCE(EX. .002 -.005)
C NIT NO. OF ITERATIONS TO BE ALLOWED(EX. 50)
C NP NO. OF COFFICIENTS(1=NP=8)
C NUMBER OF X S TO BE READ IN (5 FOR THIS PROGRAM)
C NDP NUMBER OF DATA POINTS
C ITP 0 = DO NOT LIST PARAMETERS AT EACH ITERATION
C      1 = LIST PARAMETERS AT EACH ITERATION
C NINV 0 = DO NOT LIST INVERSE OF VARIANCE-COVARIANCE MATRIX
C       1 = LIST MATRIX
C FACT CONTROLS STEP SIZE(E.G. 0.5)
C FMT VARIABLE FORMAT

      NO=8HNO
      YES=8HYES

C READ TITLE CARD AND CONTROLS AND READ DATA FORMAT

   20 READ 1, TITLE,NDP,NP,NX,NIT,EPS,ITP,NINV
      IF (TITLE(1).EQ.5HENDIT) STOP
      READ 3, FACT
      READ 99, FMT

C  READ INITIAL ESTIMATES OF PARAMETERS
      READ 2, (GP(I),I=1,NP)

C   READ WEIGHTS ,Y, X VALUES.

      DO 18 I=1,NDP
   18 READ FMT,(X(I,J),J=1,NX),Y(I),W(I)
      PRINT 201 $ PRINT 202,TITLE $ PRINT 203,NDP $ PRINT 204,NP
      PRINT 205, NX $ PRINT 206,NIT $ PRINT 207, EPS
      IF(ITP.EQ.1)32,31
32    PRINT 208, YES
      GO TO 33
31    PRINT 208, NO
33    IF(NINV.EQ.1)34,35
34    PRINT 209,YES
      GO TO 36
35    PRINT 209, NO
36    CONTINUE
      PRINT 210, FACT $ PRINT 211,FMT

C SET ASIDE INITIAL ESTIMATES

      DO 21 I=1,NP                                              23120205
   21 GP1(I)=GP(I)                                              23120207
```

```
         CALL NLLS(W,X,Y,GP1,STDE,YCALC,VARY,
        1NERR,EPS,NIT,NP,NDP,ITP,VARF,VARF1,C,COREL,N)

      C  CHECK IF AN ERROR HAS OCCURED

         IF (NERR .LE. 4) 501,502
     501 GO TO (30,100,101,102),NERR
     502 NERR=NERR-4 $ PRINT 212,NERR $ GO TO 20

      C  WRITE OUTPUT TITLE, COEFFICIENTS, ST. ERRORS,

      30 WRITE (6,7) TITLE
         DO 130 I=1,NP                                                      231208
     130 WRITE (6,8) I,GP1(I),STDE(I)

      C  WRITE APPROPRIATE COLUMN HEADINGS

         GO TO (50,51,52,53,54),NX
      50 WRITE(6, 9) VARF,VARF1
         GO TO 55
      51 WRITE(6,10) VARF,VARF1
         GO TO 55
      52 WRITE(6,11) VARF,VARF1
         GO TO 55
      53 WRITE(6,12) VARF,VARF1
         GO TO 55
      54 WRITE(6,13) VARF,VARF1

      C  WRITE OBSERVED AND PREDICTED VALUES AND NO. OF ITERATIONS

      55 DO 56 I=1,NDP
      56 WRITE (6,14) Y(I),YCALC(I),VARY(I),(X(I,J),J=1,NX)
         WRITE (6,153) N

      C  GIVE INVERSE MATRIX IF REQUESTED

     160 IF (NINV) 60,20,60
      60 WRITE (6,15)
         DO 61 I=1,NP
      61 WRITE (6,16) (C(I,J),J=1,NP)

      C  WRITE CORRELATION MATRIX

         WRITE (6,200)
         DO 161 I=1,NP
     161 WRITE (6,16) (COREL(I,J),J=1,NP)                                   231210
         GO TO 20

      C  ERROR MESSAGE FOR SINGULAR MATRIX

     100 WRITE (6,19) TITLE
         WRITE (6,150)
         GO TO 20
      C  ERROR MESSAGE FOR NON CONVERGENCE
     101 WRITE (6,19) TITLE
         WRITE (6,151) NIT
         GO TO 20

      C  ERROR  MESSAGE FOR SINGULAR INVERSE
```

```
   102 WRITE (6,19) TITLE
       WRITE (6*152)
       GO TO 20
C  FORMATS
     1 FORMAT(12A6/I4,2I2,I4,F10.0,2I1)
2      FORMAT(6E10.0)
3      FORMAT(E10.3)
     7 FORMAT(1H112A6/32H0      PARAMETER          STD ERROR)
     8 FORMAT(3H0 A,I1,1PE14.5,E15.5)                                              2312090
     9 FORMAT(7H0  VARF,1PF15.5,10H      VARF1,E15.5/
      152H0      Y(OBS)          Y(CALC)          VAR(Y)          X(1))
    10 FORMAT(7H0  VARF,1PF15.5,10H      VARF1,E15.5/
      152H0      Y(OBS)          Y(CALC)          VAR(Y)          X(1)
       214H           X(2))
    11 FORMAT(7H0  VARF,1PF15.5,10H      VARF1,E15.5/
      152H0      Y(OBS)          Y(CALC)          VAR(Y)          X(1)
       228H           X(2)           X(3))
    12 FORMAT(7H0  VARF,1PF15.5,10H      VARF1,E15.5/
      152H0      Y(OBS)          Y(CALC)          VAR(Y)          X(1)
       242H           X(2)           X(3)           X(4))
    13 FORMAT(7H0  VARF,1PF15.5,10H      VARF1,E15.5/
      152H0      Y(OBS)          Y(CALC)          VAR(Y)          X(1)
       256H           X(2)           X(3)           X(4)          X(5))
    14 FORMAT(1H01P8E14.5)
    15 FORMAT(//20H0       INVERSE MATRIX)
16     FORMAT(1H0,10F9.3)
    19 FORMAT(1H112A6)
   99  FORMAT (10A8)
   150 FORMAT(16H0SINGULAR MATRIX)
   151 FORMAT(20H0DID NOT CONVERGE IN14,
      111H ITERATIONS)
   152 FORMAT(24H0SINGULAR INVERSE MATRIX)
   153 FORMAT(13H0CONVERGED IN14,11H ITERATIONS)
   200 FORMAT(//24H0      CORRELATION MATRIX)
201    FORMAT(1H1,*CONTROL VARIABLES*)
202    FORMAT(1H0,*TITLE*,2X,12A6)
203    FORMAT(1H0,*NDP    NUMBER OF DATA POINTS*,15X,I4)
204    FORMAT(1H0,*NP     NUMBER OF PARAMETERS*,16X,I2)
205    FORMAT(1H0,*NX     NUMBER OF X S*,23X,I2)
206    FORMAT(1H0,*NIT    NUMBER OF ITERATIONS ALLOWED*,8X,I4)
207    FORMAT(1H0,*EPS    ALLOWABLE DIFFERENCE IN ESTIMATES    *,E13.4)
208    FORMAT(1H0,*ITP    PRINT CONTROL ITERATIONS*,13X,A8)
209    FORMAT(1H0,*NINV   PRINT INVERSE*,24X,A8)
210    FORMAT(1H0,*FACT   ADJUSTMENT CORRECTION FACTOR*,9X,F5.3)
211    FORMAT(1H0,*FMT    VARIABLE FORMAT*,22X,5A8/5A8)
   212 FORMAT(1H0*DERIVATIVE WRT PARAM NO *I5* IS ZERO FOR ALL DATA PTS*)
       END
      SUBROUTINE NLLS(W,X,Y,GP1,STDE,YCALC,VARY,
     1NERR,EPS,NIT,NP,NDP,ITP,VARF,VARF1,C,COREL,N)

      DIMENSION W(250),X(250,5),Y(250),GP1(8),
     1STDE(8),YCALC(250),VARY(250),COREL(8,8),
     2A(8,8),B(8),XTX(8,8),TB(8),CONV(8),DERIV(8),C(8,8),
     3VMAT(8,8),R(8)
      COMMON FACT

C     SET NERR AS ERROR MESSAGE CONTROL

      NERR=1                                                            14800604

C   ZERO OUT VARIABLES

      DO 118 N=1,NIT                                                    23120213
      DO 110 L=1,NP                                                     23120219
      B(L)=0.0                                                          23120221
      DO 110 M=1,NP                                                     23120223
  110 A(L,M)=0.0                                                        23120225

C   SEE IF ESTIMATES OF COEFFICIENTS TO BE WRITTEN AT EACH ITERATION

      IF (ITP)21,21,20
   20 WRITE (6,2) N,(GP1(I),I=1,NP)

C   CALL USERS SUBROUTINE TO DEFINE FUNCTION AND PARTIAL DERIVATIVES

   21 DO 111 J=1,NDP

      CALL SUBRT(J,W,X,Y,GP1,DERIV,YC,FI,WI)                            14800303

C   MODIFY THE COEFFICIENTS

   50 DO 111 L=1,NP                                                     23120305
      B(L)=B(L)+DERIV(L)*FI*WI                                          23120307

C   SET UP MATRIX OF SS AND CP AND SET IT ASIDE TO BE INVERTED

      DO 111 M=L,NP                                                     23120309
  111 A(L,M)=A(L,M)+DERIV(L)*DERIV(M)*WI                                23120311
      DO 112 M=2,NP                                                     23120313
      K=M-1                                                             23120315
      DO 112 I=1,K                                                      23120317
  112 A(M,I)=A(I,M)                                                     23120319
      DO 212 L=1,NP                                                     23120321
      DO 212 M=1,NP                                                     23120323
  212 XTX(L,M)=A(L,M)                                                   23120325

      CALL INVERT(XTX,NP,1.0E-30,NEROR,DELTA)

C   SEE IF MATRIX WAS SINGULAR

      IF(NEROR)113,114,113
  113 NERR=2                                                            14800506
      GO TO 200

C   ALTER ESTIMATES OF COEFFICIENTS AND CHECK FOR CONVERGENCE
```

```
  114 DO 230 I=1,NP
      DO 230 J=1,NP
  230 C(I,J)=XTX(I,J)
  240 DO 215 I=1,NP
      R(I)=0.0
      DO 215 J=1,NP
  215 R(I)=R(I)+XTX(I,J)*R(J)
      DO 216 I=1,NP
      BETA=FACT*GP1(I)
      COR=SIGN (AMIN1(ABS (R(I)),ABS (BETA)),R(I))
  216 TB(I)=GP1(I)+COR
      DO 116 I=1,NP                                                  2312051
      CONV(I)=ABS (GP1(I)/TB(I)-1.0)
      IF(CONV(I)-EPS)116,116,117                                     2312051
  116 CONTINUE                                                       2312051
      GO TO 120                                                      2312052
  117 DO 118 I=1,NP                                                  2312052
  118 GP1(I)=TB(I)                                                   2312052
      NERR=3
      GO TO 200                                                      1480060

C  GO BACK TO USERS SUBROUTINE THEN CALCULATE ST. DEV.,VARANCE, E.T.C.

  120 DO 121 I=1,NP                                                  2312060
  121 GP1(I)=TB(I)                                                   2312060
      SUMW=0.0                                                       2312061
      VARF=0.0                                                       2312061
      DO 122 J=1,NDP                                                 2312061

      CALLSUBRT(J,W,X,Y,GP1,DERIV,YC,F1,WI)                          1480061

      VARF=VARF+WI*F1**2
  122 SUMW=SUMW+WI                                                   2312062
      VARF=VARF/(FLOAT (NDP)-FLOAT (NP))
      VARF1=VARF/SUMW                                                2312062
  127 DO 128 I=1,NP                                                  2312080
      DO 128 J=1,NP                                                  2312080
      COREL(I,J)=C(I,J)/SQRT (C(I,I)*C(J,J))
  128 VMAT(I,J)=C(I,J)*VARF                                          2312080
      DO 129 I=1,NP                                                  2312080
  129 STDE(I)=SQRT (VMAT(I,I))
      DO 136 J=1,NDP                                                 1480091

      CALL SUBRT(J,W,X,Y,GP1,DERIV,YC,F1,WI)                         1480091

      VY=0.0                                                         2312091
      YCALC(J)=YC                                                    14800920
      DO 135 I=1,NP                                                  23120921
      DO 135 K=1,NP                                                  23120923
  135 VY=VY+DERIV(I)*DERIV(K)*VMAT(I,K)
  136 VARY(J)=VY

  200 RETURN                                                         14801003

    2 FORMAT(10H0ITERATIONI5,5X,10HPARAMETERS/(1H01P7E15,7))

      END                                                            14800007
      SUBROUTINE INVERT(A,N,EPS,NEROR,DELTA)
C     MATRIX INVERSION BY GAUSS-JORDAN ELIMINATION
      DIMENSION A(8,8),B(8),C(8),LZ(8)
      DELTA=1.0
      NEROR=0
      DO 10 J=1,N
   10 LZ(J)=J
      DO 20 I=1,N
      K=I
      Y=A(I,I)
      L=I-1
      LP=I+1
      IF(N-LP)14,11,11
   11 DO 13 J=LP,N
      W=A(I,J)
      IF(ABS (W)-ABS (Y))13,13,12
   12 K=J
      Y=W
   13 CONTINUE
   14 DELTA=DELTA*Y
      DO 15 J=1,N
      C(J)=A(J,K)
      A(J,K)=A(J,I)
      A(J,I)=-C(J)/Y
      A(I,J)=A(I,J)/Y
   15 B(J)=A(I,J)
      A(I,I)=1.0/Y
      J=LZ(I)
      LZ(I)=LZ(K)
      LZ(K)=J
      DO 19 K=1,N
      IF(I-K)16,19,16
   16 DO 18 J=1,N
      IF(I-J)17,18,17
   17 A(K,J)=A(K,J)-B(J)*C(K)
   18 CONTINUE
   19 CONTINUE
   20 CONTINUE
      IF(ABS (DELTA)-EPS)80,80,81
   80 NEROR=1
      GO TO 82
   81 DO 200 I=1,N
      IF(I-LZ(I))100,200,100
  100 K=I+1
      IF(I-N)800,200,200
  800 DO 500 J=K,N
      IF(I-LZ(J))500,600,500
  600 M=LZ(I)
      LZ(I)=LZ(J)
      LZ(J)=M
      DELTA=-DELTA
      DO 700 L=1,N
      C(L)=A(I,L)
      A(I,L)=A(J,L)
  700 A(J,L)=C(L)
  500 CONTINUE
```

```
       200 CONTINUE
        82 RETURN
           END
```

```
      SUBROUTINE SUBRT (J,W,X,Y,GP1,DERIV,YC,F1,WI)
      DIMENSION W(250),X(250,5),Y(250),GP1(8),DERIV(8)
      YC=GP1(1)*GP1(2)**X(J,1)
      F1=Y(J)-YC
      DERIV(1)=GP1(2)**X(J,1)
      DERIV(2)=GP1(1)*X(J,1)*GP1(2)**(X(J,1)-1.)
      WI=1.
      RETURN
      END
```

INPUT DECK FOR SAMPLE RUN OF NLINEAR

```
    EXAMPLE - EXPONENTIAL FIT OF DUCK WEED GROWTH
  14 2 1  30       .00500
        0.5
(3F5.0)
        79.        1.4
    0  100
    1  127
    2  171
    3  233
    4  323
    5  452
    6  654
    7  918
    8 1406
    9 2150
   10 2800
   11 4140
   12 5760
   13 8250
ENDIT
00000
```

```
CONTROL VARIABLES

TITLE      EXAMPLE - EXPONENTIAL FIT OF DUCK WEED GROWTH

NDP    NUMBER OF DATA POINTS                      14

NP     NUMBER OF PARAMETERS                      2
NX     NUMBER OF X S                             1

NIT    NUMBER OF ITERATIONS ALLOWED               30

EPS    ALLOWABLE DIFFERENCE IN ESTIMATES               .5000E-02

ITP    PRINT CONTROL ITERATIONS                  NO

NINV   PRINT INVERSE                             NO

FACT   ADJUSTMENT CORRECTION FACTOR               .500

FMT    VARIABLE FORMAT                           (3F5.0)
```

```
  EXAMPLE - EXPONENTIAL FIT OF DUCK WEED GROWTH

     PARAMETER          STD ERROR

 A1    8.38948E+01     3.59894E+00

 A2    1.42329E+00     5.04684E-03

  VARF     3.23843E+03      VARF1     2.31317E+02

    Y(OBS)          Y(CALC)         VAR(Y)          X(1)

   1.00000E+02    8.38948E+01    1.29524E+01    0.

   1.27000E+02    1.19407E+02    2.21084E+01    1.00000E+00

   1.71000E+02    1.69950E+02    3.71463E+01    2.00000E+00

   2.33000E+02    2.41889E+02    6.12440E+01    3.00000E+00

   3.23000E+02    3.44279E+02    9.86746E+01    4.00000E+00

   4.52000E+02    4.90007E+02    1.54493E+02    5.00000E+00

   6.54000E+02    6.97422E+02    2.33233E+02    6.00000E+00

   9.18000E+02    9.92634E+02    3.35733E+02    7.00000E+00

   1.40600E+03    1.41281E+03    4.53302E+02    8.00000E+00

   2.15000E+03    2.01083E+03    5.60498E+02    9.00000E+00

   2.80000E+03    2.86200E+03    6.16632E+02    1.00000E+01

   4.14000E+03    4.07346E+03    6.15447E+02    1.10000E+01

   5.76000E+03    5.79771E+03    8.08301E+02    1.20000E+01

   8.25000E+03    8.25183E+03    2.46155E+03    1.30000E+01

CONVERGED IN    3 ITERATIONS
```

Furnival's SCREEN

INTRODUCTION

This program was developed as a partial solution for the problem of selecting from a large group of independent variables a small number to be used as predictors in a regression equation. This program enables the user to compute all possible regressions within given constraints. Obviously it is not feasible to compute all possible regressions when the number of independent variables is large, but it is perfectly feasible to compute more than the one regression produced by a step-wise program. This program allows the user to restrict regression computation to combinations of independent variables which meet one or more of four optional constraints.

1. A number of independent variables may be fixed or forced to appear in every regression. This is designed for use when the investigator is sure that one or more variables must appear in the final regression equation.
2. The maximum number of independent variables appearing in any regression may be limited to less than the total number of independent variables.
3. Independent variables may be placed in sets such that if one variable in a set is present in a regression, all variables in that set will be present. This constraint effectively reduces the dimensions of a screening problem because a set of variables is treated essentially as a single variable by the program. R-squares are computed only for those regressions which either include every variable of a set or will fit every variable of a set.
4. Variables may be placed in groups such that if one member of a group is present in a regression no other member of a group will be present. The members of a group may be either individual variables or sets of

variables. This constraint is especially useful when one wishes to screen a number of possible transforms of each of several basic variables with the intention of retaining no more than one transform of each variable in the final prediction equation.

The description of the constraints above give ample evidence as to the value and the utility of this program in reducing a large number of measured variables to a smaller workable group. The output does not give the regression coefficient, any measure of the correlation between independent variables, i.e., partial r-squares, or other statistics about the data which would be necessary in a final analysis. Thus this program should be viewed as but one step in obtaining a prediction equation from a raw set of independent variables. One other utility of this program is the fact that it allows the user to compute regressions against up to four dependent variables.

The primary limitation of the program is that the number of regressions to be computed must be less than the quantity

$$\frac{15,000}{NY + 1}$$

where NY is the number of independent variables. Therefore, when none of the four optional constraints given above is employed, the maximum number of independent variables is limited to 12 for one or two, 11 for three through six, and 10 for seven or eight dependent variables. With the inclusion of one or more constraints decreasing the number of regressions below the allowable maximum, then the total number of dependent and independent variables may be as large as 50.

The program is written in FORTRAN and operates on the CDC 6400 computer utilizing the NCAR or SCOPE compilers.

Input Control Cards With Options for Constraints

Card preparation.

Card	Column	Description
1	1-72	Job title, alphanumeric.
2	1	Label card (b = = labels furnished by program, 1 = labels on card 5).
	2	Set card (b = all sets have one variable, 1 = set sizes on card 7).
	3	Group card (b = all groups have one set, 1 = group sizes on card 8).
	4	Intercept flag (b = intercepts, 1 = no intercept).
	5	Format card (b = no second format card, 1 = more formats on card 4).
	6-10	Number of observations.
	19-20	Number (of sets) of independent variables not fixed.
	30	Number of dependent variables (b = 1).
	39-40	Number (of sets) of fixed independent variables (b = 0).
	49-50	Large number (of sets) of independent variables not fixed to be included in any regression (b = no restriction).
	59-60	Total number of variables before transformations (b = same as after transformations).
	61-62	Number of transformations per observation (b = none).
3	1	Left parenthesis.
	2-72	Format of data (only E or F type formats are permitted). Followed by right parenthesis.
4	1-72	Continuation of data format (omit card 4 if column 5 of card 2 is blank).

Card	Column	Description
5	1- 4	Alphanumeric label for the first set of independent variables.
	5- 8	Same for second set. Continue for labels of all independent variable(s); omit if column 1 of card 2 is blank.
6	1-72	Continuation of labels if needed.
7	1- 2	Number of variables in first set of independent variables.
	3- 4	Same for second set. Continue for all variables in remaining sets using two columns for each set. (Omit card 7 if column 2 of card 2 is blank.)
8	1- 2	Number of groups of independent variables.
	3- 4	Number of sets of independent variables in first group.
	5- 6	Number of sets of independent variables in second group.
		Continue for all groups of variables using two columns for each group. (Omit card 8 if column 3 of card 2 is blank.)
9		Transformation control cards. (Omit these cards if columns 61-62 of card 2 is blank or zero.) For each transformation there must be one transformation control card. At the end of the transformation procedure, the dependent variables must follow all the independent variables. A list of transformations is included below.
	1- 2	Number of the transformation from the list (below).
	3- 4	Number of the resultant (transformed) variable.
	5- 6	Number of the first variable in the transformation (on the right of the "=" sign).
	7- 8	Number of the second variable (if two variables are involved in the transformation).
	or	
	7-14	Constant term (c) punched with decimal point.

Card	Column	Description
10		Your observational data deck.
11	1- 4	Punch DONE or QUIT. Punch QUIT if this is the last or only data set to be processed. If another data set and set of control cards follow, punch DONE.

A few points on the above control cards may need further clarification. For example, on the second control card, the number of variables referred to in columns 39-40 refer to the total number of variables, both dependent and independent. In connection with the labels, the labeling of independent variables is by sets, and one set receives one label regardless of the number of varialbes in the set. A label must be supplied for each dependent variable also. Card 7 also needs some additional explanation. The rules that must be followed are as follows:

1. Each independent variable whether fixed or unfixed must be assigned to a set.
2. Fixed variables and variables that are not fixed cannot be assigned to the same set.
3. The number of variables in each set must be punched on card 7 even though a particular set may contain only one variable.
4. The number of variables are punched in the order in which the sets occur in the observation vector.

The group constraint card, control card 8, is similar to that described for the set constraint. The major difference is that fixed variables are ignored in the group constraint; group assignment begins with the first variable not fixed. Basic rules to follow are listed as follows, the parenthesized material is meaningful only when the set constraint is also used:

1. Each (set of) independent variable that is not fixed must be assigned to a group.
2. The number (of sets) of variables in each group must be punched on card 8 even though some groups may contain only one (set of variables).
3. The number (of sets) of variables are punched in the order in which groups occur in the observation vector.

The following transformations may be called by the numbers given on the left (enter this number in columns 1-2 of card 9).

t = resultant variable (columns 3-4 of card 9).

u = first variable in the transformation (columns 5-6 of card 9).

v = second variable in the transformation, if used (columns 7-8 of card 9).

c = constant, punched with decimal point (columns 7-14 of card 9).

01	$t = u$
02	$t = c * u$
03	$t = c + u$
04	$t = u + v$
05	$t = u * v$
06	$t = u/v$
07	$t = 1/u$
08	$t = u^c$
09	$t = \ln_e u$
10	$t = e^u$
11	$t = \log_{10} u$

12	t = original value of u regardless of previous transformations on u
13	t = sin (u)
14	t = cos (u)
15	t = u/c
16	t = $\|u\|$
17	call user written subroutine TR$\overline{\text{O}}$BS

SAMPLE RUN OF PROGRAM

Following is a listing of SCREEN, a sample input deck and the output generated.

```
      PROGRAM SCREEN
     1 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
CMAIN SCREEN PROGRAM 5/12/64
      DIMENSION NR(30),NV(30),Z(50,17) ,ND(51),A(14000),NS(31),NDA(31),
     1SS(10),R(50,9),X(50),T(30),FMT(24),ISF(30),RA(10),IT(17),NW(50),
     2NRX(30),SC(51),TY(10),MSC(31),D(31),NG(30),NDG(18),TT(12),NAD(31),
     3KNT(30),NK(17),RN(30),GN(30),NO(17),C(20),NT(30)
      COMMON/BANK/NR,NV,IT,ND,A,NS,NDA,SS,R,X,T,FMT,Z,ISF,RA,NRX,SC,IA,
     1IB,IN,IM,KC,LM,PZ,NF,NFA,KNT,N,NP,NY,KRT,NSET,NOB,NCP,MSC,NGX,DF,
     2D,TY,NXS,NG,NVR,IS,IE,NYT,TT,NAD,NDG,NO,C,NT,NIF,ZERO,RN,GN,NK,
     3NYY,NVS,LA,NW,LMA
      COMMON/L1/DONE
      INTEGER DONE
      XADF(I,J,NN)=NN-((NP-I+1)*(NP-I))/2-(NP-J)
C
C
    1 CALL ROMNT
CTEST FOR CONTROL CARD ERROR
      IF(PZ) 3,4,3
    3 GO TO 1
    4 CALL INITL
C
C
CSKIP MEAN FOR ZERO INTERCEPT
C
      IF(NIF) 6,5,6
CSUMS OF SQUARES AROUND MEAN
C
    5 IS=1
      IE=1
      CALL MEAN
    6 DO 37 I=NYT,NP
                         II=XADF(I,I,NAD(1))
      K=I-NYT+1
   37 SS(K)=A(II)
    7 IF(NF) 9,14,9
C
CFIXED VARIABLES
C
    9 IS=2
      DO 10 M=1,NF
      IE=ND(IS)
      J=NF-M+1
      CALL MEAN
      IS=IE+1
CCOMPUTE R SQUARES FOR FIXED SETS
      DO 43 I=NYT,NP
                         II=XADF(I,I,NAD(1))
      K=I-NYT+1
   43 R(J,K+1)=1.0-A(II)/SS(K)
      R(J,1)=DF
   10 NW(J)=0
C
CFIRST X AFTER FIXED,SKIP COUNTER
C
   14 D(1)=DF
      IM=1
      IN=0
      KC=-LM
      LB=NSET
      LA=LB
      NV(1)=1
      GO TO 61
C
CCOUNTER LOOP
C
   38 J=1
   39 DO 40 I=J,KRT
      IF(KNT(I)) 40,41,40
   41 KNT(I)=1
      GO TO 42
   40 KNT(I)=0
      GO TO 600
   42 IN=NXS-I
      KC=NV(IN)-LM
CTEST FOR LIMIT AND OLD ILLEGAL GROUP COMBINATION
      IF(KC) 20,21,20
   20 IF(ISF(IN)) 21,193,21
CSTEP COUNTER
   21 J=I
      GO TO 39
C
CTEST FOR AND WRITE FULL BLOCK
C
  193 IM=IN+1
      NV(IM)=NV(IN)+1
      IF(LB+1-50) 197,197,196
  196 WRITE(1)LB
      WRITE(1)(NW(L),(R(L,M),M=1,NYY),L=1,LB)
      LB=0
  197 LB=LB+1
      LA=LB
C
CMAJOR REDUCTION LOOP
C
   61 DO 127 IAB=IM,NXS
      IA=IAB
      N=NS(IA)
      IB=IA+1
      ISF(IA)=0
CCHECK FOR NEW ILLEGAL GROUP COMBINATION
      MOT=IA-IN
      IF(MOT-MSC(IM)) 66,66,68
   66 ISF(IA)=1
      LB=LB-1
      LA=LB
      GO TO 127
   68 CALL MJLOOP
C
CLOAD BLOCKS,STEP LOADING INDEX,AND COUNT REGRESSIONS
C
      NW(LA)=NV(IA)
      R(LA,1)=D(IB)
      LA=LA-1
      K=NV(IA)
      NO(K)=NO(K)+1
  127 NV(IB)=NV(IA)
      GO TO 38
C
CWRITE PARTIAL BLOCK AT END OF COMPUTATIONS
```

```
      C
        600 WRITE(1)LB
            WRITE(1)(NW(L),(R(L,M),M=1,NYY),L=1,LB)
      C
      COVERALL R SQUARES
      C
            IF(NXS-LM) 601,201,601
        601 IS=NFA+2
            IE=NP-NY
            CALL MEAN
            DO 116 I=NYT,NP
                        II=XADF(I,I,NAD(I))
            K=I-NYT+1
        116 RA(K)=1.0-A(II)/SS(K)
      C
      CMAIN PRINT LOOP
      C
        201 NVA=99
            WRITE(1)NVA
            REWIND 1
            CALL MPRINT
      C --- CHECK FOR QUIT CARD SIGNIFYING END OF RUN
            IF(DONE.EQ.4HQUIT) STOP
            GO TO 1
            END
```

```
               FORTRAN EXTENDED VERSION 2.0            03/14/70          *01

$IBFTC BAT11    DECK
       SUBROUTINE CTEST(I,J,K,L)
       DIMENSION NR(30),NV(30),Z(50,17) ,ND(51),A(14000),NS(31),NDA(31),
      1SS(10),R(50,9),X(50),T(30),FMT(24),ISF(30),RA(10),IT(17),NW(50),
      2NRX(30),SC(51),TY(10),MSC(31),D(31),NG(30),NDG(18),TT(12),NAD(31),
      3KNT(30),NK(17),RN(30),GN(30),NO(17),C(20),NT(30)
       COMMON/BANK/NR,NV,IT,ND,A,NS,NDA,SS,R,X,T,FMT,Z,ISF,RA,NRX,SC,IA,
      1IB,IN,IM,KC,LM,PZ,NF,NFA,KNT,N,NP,NY,KRT,NSET,NOB,NCP,MSC,NGX,DF,
      2D,TY,NXS,NG,NVR,IS,IE,NYT,TT,NAD,NDG,NO,C,NT,NIF,ZERO,RN,GN,NK,
      3NYY,NVS,LA,NW,LMA
C
     1 DATA ASK/1H*/
       IF(I-K) 2,2,4
     2 IF(J-1) 5,5,4
     4 C(L)=ASK
       PZ=1
     5 RETURN
       END
```

```
$IBFTC BAT12    DECK
      SUBROUTINE MATOUT (A,NVS,NY,T,TY,TT)
      DIMENSION A(14000),T(30),TY(10),TT(12)
      A(1)=A(1)
      NVS=NVS
      NY=NY
      T(1)=T(1)
      TY(1)=TY(1)
      TT(1)=TT(1)
      RETURN
      END
```

```
$IBFTC BAI13   DECK
      SUBROUTINE INITL
      DIMENSION NR(30),NV(30),Z(50,17) ,ND(51),A(14000),NS(31),NDA(31),
     1SS(10),R(50,9),X(50),T(30),FMT(24),ISF(30),RA(10),IT(17),NW(50),
     2NRX(30),SC(51),TY(10),MSC(31),D(31),NG(30),NDG(18),TT(12),NAD(31),
     3KNT(30),NK(17),RN(30),GN(30),NO(17),C(20),NT(30)
      COMMON/BANK/NR,NV,IT,ND,A,NS,NDA,SS,R,X,T,FMT,Z,ISF,RA,NRX,SC,IA,
     1IB,IN,IM,KC,LM,PZ,NF,NFA,KNT,N,NP,NY,KPT,NSET,NOB,NCP,MSC,NGX,DF,
     2D,TY,NXS,NG,NVR,IS,IE,NYT,TT,NAD,NDG,NO,C,NT,NIF,ZERO,RN,GN,NK,
     3NYY,NVS,LA,NW,LMA
      XADF(I,J,NN)=NN-((NP-I+1)*(NP-I))/2-(NP-J)
C
    1 REWIND 1
      DF=1.0
C
CCOMPUTE NUMBER OF FIXED XS
C
      NFA=0
      IF (NF) 9,11,9
    9 DO 10 I=1,NF
   10 NFA=NFA+NRX(I)
C
CINDEX OF FIRST Y
C
   11 NYT=NP-NY+1
C
C
CMOVE SET SIZES UP,ZERO COUNTR,COMPUTE ORDERS AND LAST ADDRESSES OF SUB-
CMATRICES
C
      KRT=NXS-1
      NN=(NP+1)*NP
      NAD(1)=NN/2
      NS(1)=NVS-NFA
      DO 33 I=1,NXS
      KNT(I)=0
      INF=I+NF
      NR(I)=NRX(INF)
      NS(I+1)=NS(I)-NR(I)
      NN=(NS(I)-1)*NS(I)
   33 NAD(I+1)=NAD(I)+NN/2
C
CCOMPUTE DETERMINANT DIVIDERS
C
      DO 35 I=1,NP
                    II=XADF(I,I,NAD(1))
   35 SC(I)=A(II)
C
CCOMPUTE NUMBER SETS REMAINING IN GROUP AND INDEX OF FIRST X IN NEXT
CGROUP
C
      M=0
      NZ=NFA+2
      MSC(1)=-1
      L=0
      DO 20 I=1,NGX
      JE=NG(I)
      DO 21 J=1,JE
      L=L+1
      MSC(L+1)=NG(I)-J
   21 NZ=NZ+NR(L)
      DO 20 K=1,JE
      M=M+1
   20 NDA(M)=NZ
C
CCOMPUTE INDEX OF LAST X IN SET
C
      L=1
      NZ=1
      DO 22 I=1,NSET
      NZ=NZ+NRX(I)
      JE=NRX(I)
      DO 22 J=1,JE
      L=L+1
   22 ND(L)=NZ
      DO 23 I=NYT,NP
   23 ND(I)=NP
C
CZERO REGRESSION COUNTS
C
      DO 24 I=1,LM
   24 NO(I)=0
C
CTERMINAL INDICES FOR IDENTIFICATION
CSET DIGITS FOR FIXED TAPE WRITE,COMPUTE NUMBER OF FIXED XS
      IT(1)=NXS+1
      LMA=LM-1
      NYY=NY+1
      DO 25 I=1,LMA
      NGY=NGX-I+1
   25 IT(I+1)=IT(I)-NG(NGY)
      DO 26 I=LM,17
   26 IT(I+1)=1
      RETURN
      END
```

```
$IBFTC BAT14    DECK
       SUBROUTINE MEAN
       DIMENSION NR(30),NV(30),Z(50,17)  ,ND(51),A(14000),NS(31),NDA(31),
      1SS(10),R(50,9),X(50),T(30),FMT(24),ISF(30),RA(10),IT(17),NW(50),
      2NRX(30),SC(51),TY(10),MSC(31),D(31),NG(30),NDG(18),TT(12),NAD(31),
      3KNT(30),NK(17),RN(30),GN(30),NO(17),C(20),NT(30)
       COMMON/BANK/NR,NV,IT,ND,A,NS,NDA,SS,R,X,T,FMT,Z,ISF,RA,NRX,SC,IA,
      1IB,IN,IM,KC,LM,PZ,NF,NFA,KNT,N,NP,NY,KRT,NSET,NOB,NCP,MSC,NGX,DF,
      2D,TY,NXS,NG,NVR,IS,IE,NYT,TT,NAD,NDG,NO,C,NT,NIF,ZERO,RN,GN,NK,
      3NYY,NVS,LA,NW,LMA
       XADF(I,J,NN)=NN-((NP-I+1)*(NP-I))/2-(NP-J)
C
C
       DO 50 I=IS,IE
       JS=I+1
       DO 41 J=JS,NP
                          IJ=XADF(I,J,NAD(1))
                          II=XADF(I,I,NAD(1))
       B=A(IJ)/A(II)
       DO 41 K=J,NP
                          JK=XADF(J,K,NAD(1))
                          IK=XADF(I,K,NAD(1))
    41 A(JK)=A(JK)-B*A(IK)
CCOMPUTE DETERMINANTS
    50 DF=DF*A(II)/SC(I)
       RETURN
       END
```

```
$IBFTC BAT18    DECK
      SUBROUTINE MILOOP
      DIMENSION NR(30),NV(30),Z(50,17) ,ND(51),A(14000),NS(31),NDA(31),
     1SS(10),R(50,9),X(50),T(30),FMT(24),ISF(30),RA(10),IT(17),NW(50),
     2NRX(30),SC(51),TY(10),MSC(31),D(31),NG(30),NDG(18),TT(12),NAD(31),
     3KNT(30),NK(17),RN(30),GN(30),NO(17),C(20),NT(30)
      COMMON/BANK/NR,NV,IT,ND,A,NS,NDA,SS,R,X,T,FMT,Z,ISF,RA,NRX,SC,IA,
     1IB,IN,IM,KC,LM,PZ,NF,NFA,KNT,N,NP,NY,KRT,NSET,NOB,NCP,MSC,NGX,DF,
     2D,TY,NXS,NG,NVR,IS,IE,NYT,TT,NAD,NDG,NO,C,NT,NIF,ZERO,RN,GN,NK,
     3NYY,NVS,LA,NW,LMA
    1 XADF(I,J,NN)=NN-((NP-I+1)*(NP-I))/2-(NP-J)
C
C
   80 IS=NP-N+1
      JE=ND(IS)
      IE=JE
      JT=NDA(IA)
      D(IB)=D(IM)
      IF(KC+2) 40,40,29
C
C
CNOT LAST SET
C
   40 DO 25 I=IS,IE
      JS=I+1
      IF(IS-IE) 11,10,11
CFIRST OF ONE
   10 L=IM
      GO TO 18
   11 IF(I-IS) 13,12,13
CFIRST OF MORE THAN ONE
   12 L=IM
      GO TO 16
   13 IF(I-IE) 15,14,15
CLAST
   14 L=IB
      GO TO 18
CMIDDLE
   15 L=IB
      GO TO 16
C
C
CROWS IN SET
C
   16 DO 17 J=JS,IE
                        IJL=XADF(I,J,NAD(L))
                        IIL=XADF(I,I,NAD(L))
      B=A(IJL)/A(IIL)
CCOLUMNS IN SET
      DO 23 K=J,JE
                        JKB=XADF(J,K,NAD(IB))
                        JKL=XADF(J,K,NAD(L))
                        IKL=XADF(I,K,NAD(L))
   23 A(JKB)=A(JKL)-B*A(IKL)
CCOLUMNS NOT IN SET
      DO 17 K=JT,NP
                        JKB=XADF(J,K,NAD(IB))
                        JKL=XADF(J,K,NAD(L))
                        IKL=XADF(I,K,NAD(L))
   17 A(JKB)=A(JKL)-B*A(IKL)
```

```
C
CROWS NOT IN SET
C
   18 DO 22 J=JT,NP
                            IJL=XADF(I,J,NAD(L))
                            IIL=XADF(I,I,NAD(L))
      B=A(IJL)/A(IIL)
      IF(KC+2) 26,24,26
CNOT NEXT TO LAST SET ,DO ALL COLUMNS
   26 KS=J
      GO TO 21
CNEXT TO LAST SET,PREPARE TO SKIP TO YS
   24 KS=NYT
C
CTEST FOR MORE XS
      KE=ND(J)
      IF(KE-NP) 19,26,19
CCOLUMNS IN NEXT SET
   19 DO 20 K=J,KE
                            JKB=XADF(J,K,NAD(IB))
                            JKL=XADF(J,K,NAD(L))
                            IKL=XADF(I,K,NAD(L))
   20 A(JKB)=A(JKL)-B*A(IKL)
CCOLUMNS OF YS OR ALL COLUMNS
   21 DO 22 K=KS,NP
                            JKB=XADF(J,K,NAD(IB))
                            JKL=XADF(J,K,NAD(L))
                            IKL=XADF(I,K,NAD(L))
   22 A(JKB)=A(JKL)-B*A(IKL)
CCOMPUTE DETERMINANTS
   25 D(IB)=D(IB)*A(IIL)/SC(I)
      GO TO 55
C
C
CLAST SET
C
   29 DO 50 I=IS,IE
CTEST FOR LAST X OF SET
      IF(I-IE) 30,33,30
C
CROWS OF XS
C
   30 JS=I+1
      DO 32 J=JS,IE
                            IJM=XADF(I,J,NAD(IM))
                            IIM=XADF(I,I,NAD(IM))
      B=A(IJM)/A(IIM)
CCOLUMNS OF XS
      DO 31 K=J,IE
                            JKM=XADF(J,K,NAD(IM))
                            IKM=XADF(I,K,NAD(IM))
   31 A(JKM)=A(JKM)-B*A(IKM)
CCOLUMNS OF YS
      DO 32 K=NYT,NP
                            JKM=XADF(J,K,NAD(IM))
                            IKM=XADF(I,K,NAD(IM))
   32 A(JKM)=A(JKM)-B*A(IKM)
C
CROWS AND COLUMNS OF YS
C
```

```
      33 DO 37 J=NYT,NP
                              IJM=XADF(I,J,NAD(IM))
                              IIM=XADF(I,I,NAD(IM))
         B=A(IJM)/A(IIM)
         IF(I-IS) 35,34,35
CFIRST X,YS ABOVE
      34 L=IM
         GO TO 36
CNOT FIRST X,YS BELOW
      35 L=IB
      36 DO 37 K=J,NP
                              JKB=XADF(J,K,NAD(IB))
                              JKL=XADF(J,K,NAD(L))
                              IKM=XADF(I,K,NAD(IM))
      37 A(JKB)=A(JKL)-B*A(IKM)
CCOMPUTE DETERMINANT
      50 D(IB)=D(IB)*A(IIM)/SC(I)
C
C
CR SQUARES
C
      55 DO 60 L=NYT,NP
                              LL=XADF(L,L,NAD(IB))
         K=L-NYT+1
      60 R(LA,K+1)=1.0-A(LL)/SS(K)
         RETURN
         END
```

```
$IBFTC BATIS    DECK
      SUBROUTINE MPRINT
      DIMENSION NR(30),NV(30),Z(50,17) ,ND(51),A(14000),NS(31),NDA(31),
     1SS(10),R(50,9),X(50),T(30),FMT(24),ISF(30),RA(10),IT(17),NW(50),
     2NRX(30),SC(31),TY(10),MSC(31),D(31),NG(30),NDG(18),TT(12),NAD(31),
     3KNT(30),NK(17),RN(30),GN(30),NO(17),C(20),NT(30)
      COMMON/BANK/NR,NV,IT,ND,A,NS,NDA,SS,R,X,T,FMT,Z,ISF,RA,NRX,SC,IA,
     1IB,IN,IM,KC,LM,PZ,NF,NFA,KNT,N,NP,NY,KRT,NSET,NOB,NCP,MSC,NGX,DF,
     2D,TY,NXS,NG,NVR,IS,IE,NYT,TT,NAD,NDG,NO,C,NT,NIF,ZERO,RN,GN,NK,
     3NYY,NVS,LA,NW,LMA
      DATA BLANK/1H /
C
CPRINT TITLE AND COLUMN HEADINGS
C
    1 WRITE(6,2)(TT(I),I=1,12)
    2 FORMAT(1H112A6)
      WRITE(6,501)(TY(JT),JT=1,NY)
  501 FORMAT(25H0REGRESSION SCREEN OUTPUT/46H COEFFICIENTS OF COLINEARIT
     1Y AND DETERMINATION//28X,28H IDENTIFICATION OF VARIABLES,20X,6HC O
     2F C,2X,8A6)
C
CSET TERMINAL INDICES
C
      IT1=IT(1)
      IT2=IT(2)
      IT3=IT(3)
      IT4=IT(4)
      IT5=IT(5)
      IT6=IT(6)
      IT7=IT(7)
      IT8=IT(8)
      IT9=IT(9)
      IT10=IT(10)
      IT11=IT(11)
      IT12=IT(12)
      IT13=IT(13)
      IT14=IT(14)
      IT15=IT(15)
      IT16=IT(16)
      IT17=IT(17)
C
CCOMPUTE STORAGE ADDRESSES
C
      NDG(1)=NF*NYY-NY
      DO 3 I=1,LM
    3 NDG(I+1)=NDG(I)+NO(I)*NYY
C
CRELOAD TAPE
C
    7 READ(1)LB
      IF(LB-99) 4,8,4
    4 READ(1)(NW(L),(R(L,M),M=1,NYY),L=1,LB)
      DO 6 L=1,LB
      J=NW(L)+1
      K=NDG(J)
      DO 5 M=1,NYY
      A(K)=R(L,M)
    5 K=K+1
    6 NDG(J)=NDG(J)-NYY
      GO TO 7
```

```
C
CINITIALIZE FOR BLOCK PRINT
C
    8 KA=0
      KB=0
      KD=1
      L=0
      IF(NF) 9,14,9
C
CLABEL FIXED SETS
C
    9 KD=0
      DO 12 I=1,NF
      M=17-I
      DO 10 J=1,M
   10 Z(I,J)=BLANK
      DO 11 J=1,I
      K=M+J
   11 Z(I,K)=T(J)
C
      DO 12 J=1,NYY
      KA=KA+1
   12 R(I,J)=A(KA)
C
CREINDEX LABELS
   14 D(1)=BLANK
      DO 13 I=1,NXS
      J=I+NF
   13 D(I+1)=T(J)
C
CIDENTIFICATION LOOP
C
        DO 40 I17=1,IT17
        IS16=I17+MSC(I17)+1
        DO 41 I16=IS16,IT16
        IS15=I16+MSC(I16)+1
        DO 42 I15=IS15,IT15
        IS14=I15+MSC(I15)+1
        DO 43 I14=IS14,IT14
        IS13=I14+MSC(I14)+1
        DO 45 I13=IS13,IT13
        IS12=I13+MSC(I13)+1
        DO 46 I12=IS12,IT12
        IS11=I12+MSC(I12)+1
        DO 47 I11=IS11,IT11
        IS10=I11+MSC(I11)+1
        DO 48 I10=IS10,IT10
        IS9=I10+MSC(I10)+1
        DO 49 I9=IS9,IT9
        IS8=I9+MSC(I9)+1
        DO 50 I8=IS8,IT8
        IS7=I8+MSC(I8)+1
        DO 51 I7=IS7,IT7
        IS6=I7+MSC(I7)+1
        DO 52 I6=IS6,IT6
        IS5=I6+MSC(I6)+1
        DO 53 I5=IS5,IT5
        IS4=I5+MSC(I5)+1
        DO 54 I4=IS4,IT4
        IS3=I4+MSC(I4)+1
      DO 55 I3=IS3,IT3
      IS2=I3+MSC(I3)+1
      DO 56 I2=IS2,IT2
      IS1=I2+MSC(I2)+1
      DO 21 I1=IS1,IT1
      IF(I1-1) 16,30,16
   30 IF(NF) 31,21,31
   31 L=NF
      GO TO 20
   16 L=L+1
      KB=KB+1
C
CLABELS
C
      Z(L,1)=D(I17)
      Z(L,2)=D(I16)
      Z(L,3)=D(I15)
      Z(L,4)=D(I14)
      Z(L,5)=D(I13)
      Z(L,6)=D(I12)
      Z(L,7)=D(I11)
      Z(L,8)=D(I10)
      Z(L,9)=D(I9)
      Z(L,10)=D(I8)
      Z(L,11)=D(I7)
      Z(L,12)=D(I6)
      Z(L,13)=D(I5)
      Z(L,14)=D(I4)
      Z(L,15)=D(I3)
      Z(L,16)=D(I2)
      Z(L,17)=D(I1)
C
CLOAD PRINT BLOCK
C
      DO 23 I=1,NYY
      KA=KA+1
   23 R(L,I)=A(KA)
C
CPRINT FULL BLOCK
C
   22 IF(KB-NO(KD)) 24,25,24
   24 IF(L-50) 21,20,21
   25 KB=0
   20 GO TO (61,62,63,64,65,66,67,68),NY
   61 WRITE(6,71)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
   62 WRITE(6,72)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
   63 WRITE(6,73)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
   64 WRITE(6,74)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
   65 WRITE(6,75)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
   66 WRITE(6,76)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
   67 WRITE(6,77)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
   68 WRITE(6,78)(KD,(Z(K,M),M=1,17),(R(K,MA),MA=1,NYY),K=1,L)
      GO TO 69
```

```
 71 FORMAT  (I3,17A4,2XE9.2, F6.3)

 72 FORMAT  (I3,17A4,2XE9.2,2F6.3)
 73 FORMAT  (I3,17A4,2XE9.2,3F6.3)
 74 FORMAT  (I3,17A4,2XE9.2,4F6.3)
 75 FORMAT  (I3,17A4,2XE9.2,5F6.3)
 76 FORMAT  (I3,17A4,2XE9.2,6F6.3)
 77 FORMAT  (I3,17A4,2XE9.2,7F6.3)
 78 FORMAT  (I3,17A4,2XE9.2,8F6.3)
 69 L=0
    IF(KB) 21,28,21
 28 KD=KD+1
    WRITE(6,39)
 39 FORMAT (1H0)
 21 CONTINUE
 56 CONTINUE
 55 CONTINUE
 54 CONTINUE
 53 CONTINUE
 52 CONTINUE
 51 CONTINUE
 50 CONTINUE
 49 CONTINUE
 48 CONTINUE
 47 CONTINUE
 46 CONTINUE
 45 CONTINUE
 43 CONTINUE
 42 CONTINUE
 41 CONTINUE
 40 CONTINUE
CPRINT OVERALL R SQUARES
C
      IF(NXS-LM) 414,418,414
  414 NI=NVS-NY
      WRITE(6,415)NI,DF,(RA(I),I=1,NY)
  415 FORMAT(1H0,20X,4H ALL,I3,22H INDEPENDENT VARIABLES,23X,E9.2,8F6.3)
  418 RETURN
      END
```

```
$IBFTC BAT16    DECK
       SUBROUTINE RDMNT
       DIMENSION NR(30),NV(30),Z(50,17) ,ND(51),A(14000),NS(31),NDA(31),
      1SS(10),R(50,9),X(50),T(30),FMT(24),ISF(30),RA(10),IT(17),NW(50),
      2NRX(30),SC(51),TY(10),MSC(31),D(31),NG(30),NDG(18),TT(12),NAD(31),
      3KNT(30),NK(17),RN(30),GN(30),NO(17),C(20),NT(30)
       DIMENSION QT(30),TQ(8)
       COMMON/BANK/NR,NV,IT,ND,A,NS,NDA,SS,R,X,T,FMT,Z,ISF,RA,NRX,SC,IA,
      1IB,IN,IM,KC,LM,PZ,NF,NFA,KNT,N,NP,NY,KRT,NSET,NOB,NCP,MSC,NGX,DF,
      2D,TY,NXS,NG,NVR,IS,IE,NYT,TT,NAD,NDG,NO,C,NT,NIF,ZERO,RN,GN,NK,
      3NYY,NVS,LA,NW,LMA
       DATA BLANK/1H /
       DATAQT/4H   1,4H   2,4H   3,4H   4,4H   5,4H   6,4H   7,4H   8,4H
      1  9,4H  10,4H  11,4H  12,4H  13,4H  14,4H  15,4H  16,4H  17,4H  18
      2,4H  19,4H  20,4H  21,4H  22,4H  23,4H  24,4H  25,4H  26,4H  27,4H
      3  28,4H  29,4H  30/
       DATATQ/4H  Y1,4H  Y2,4H  Y3,4H  Y4,4H  Y5,4H  Y6,4H  Y7,4H  Y8/
       XADF(I,J,NN)=NN-((NP-I+1)*(NP-I))/2-(NP-J)
C
CREAD TITLE,CONSTANTS
C
  300 PZ=0
      READ (5,205)(TT(I),I=1,12)
  205 FORMAT (12A6)
      WRITE(6,206)(TT(I),I=1,12)
  206 FORMAT(1H1,12A6)
      DATAZERO/1H /
      DO 11 I=1,20
   11 C(I)=BLANK
    1 READ (5,200)NTF,NRF,NGF,NIF,NFF,NOB,NXS,NY,NF,LM,NVR,NTRAN
  200 FORMAT(5I1,I5,5I10,I2)
   57 IF(NY) 19,18,19
   18 NY=1
   19 NSET=NF+NXS
C
CSET OR READ SET LABELS
C
   22 READ(5,203)(FMT(I),I=1,12)
      IF(NFF) 23,24,23
   23 READ(5,203)(FMT(I),I=13,24)
  203 FORMAT(12A6)
   24 IF(NTF) 27,26,27
   26 DO 2160 I=1,30
 2160 T(I)=QT(I)
      DO 2161 I=1,8
 2161 TY(I)=TQ(I)
      GO TO 28
   27 READ(5,201)(T(I),I=1,NSET),(TY(I),I=1,NY)
  201 FORMAT(18A4/18A4)
C
CSET OR READ SET SIZES
C
   28 IF(NRF) 31,29,31
   29 DO 30 I=1,NSET
   30 NRX(I)=1
      GO TO 32
   31 READ(5,202)(NRX(I),I=1,NSET)
  202 FORMAT(36I2)
C
CSET OR READ GROUP SIZES
```

```
      C
   32 IF(NGF) 76,74,76
   74 NGX=NXS
      DO 75 I=1,NGX
   75 NG(I)=1
      GO TO 54
   76 READ(5,207)NGX,(NG(I),I=1,NXS)
  207 FORMAT(36I2)
   54 IF(LM) 81,6,81
    6 LM=NGX
      C
      CSUM SETS BY GROUP AND XS BY SET
      C
   81 MXY=0
      DO 77 I=1,NGX
   77 MXY=MXY+NG(I)
      NVS=NY
      DO 47 I=1,NSET
   47 NVS=NVS+NRX(I)
      NP=NVS+1
      IF(NVR) 71,70,71
      C
      CCOMPUTE NUMBER OF REGRESSIONS,LOGICAL PRODUCTS OR BINOMIAL COEFF
   70 NVR=NVS
      C
   71 RGK=0.0
      KGT=NGX-1
      IF(NGF) 72,90,72
      C
      CSUM LOGICAL PRODUCTS FOR GROUPS
      C
      CINITIALIZE AND COMPUTE NUMBER OF REGRESSIONS WITH SINGLE X
   72 DO 50 I=1,NGX
      GN(I)=NG(I)
      RN(I)=GN(I)
      RGK=RGK+RN(I)
      NT(I)=1
   50 KNT(I)=0
      CCOUNTER LOOP
   38 J=1
   39 DO 40 I=J,KGT
      IF(KNT(I)) 40,41,40
   41 KNT(I)=1
      GO TO 42
   40 KNT(I)=0
      GO TO 98
   42 IN=NGX-I
      IM=IN+1
      CTEST FOR LIMIT
      KC=NT(IM)-LM
      IF(KC) 193,21,193
      CSTEP COUNTER
   21 J=I
      GO TO 39
  193 DO 51 I=IM,NGX
      RN(I)=RN(IN)*GN(I)
      RGK=RGK+RN(I)
   51 NT(I)=NT(IN)+1
      GO TO 38
      C
      CSUM BINOMIAL COEFF FOR SETS
      C
         90 DO 95 I=1,LM
            E=NXS
            F=1.0
            G=1.0
            DO 94 J=1,I
            G=G*E/F
            E=E-1.0
         94 F=F+1.0
         95 RGK=RGK+G
      C
      CTEST CONTROL CONSTANTS
      C
         98 XP=NY+1
            XPP=14000./XP
            DATA QQQQQ/1H*/
            IF(RGK-XPP) 96,96,97
         97 C(13)=QQQQQ
            PZ=1
         96 CALL CTEST(NTF,0,1,1)
            CALL CTEST(NRF,0,1,2)
            CALL CTEST(NGF,0,1,3)
              CALL CTEST(NIF,0,1,14)
            CALL CTEST(NOB,4,99999,15)
            CALL CTEST(NXS,2,30,4)
            CALL CTEST(NFF,0,1,5)
            CALL CTEST(NY,1,8,6)
            CALL CTEST(NF,0,17,7)
            CALL CTEST(NSET,2,30,16)
            CALL CTEST(LM,2,17,8)
            CALL CTEST(NGX,LM,NXS,10)
            CALL CTEST(NVR,0,50,11)
            CALL CTEST(NVS,0,50,9)
            CALL CTEST(MXY,NXS,NXS,12)
      C
      CPRINT CONTROL INFORMATION
      C
            WRITE(6,101)C(1),NTF
        101 FORMAT(*6HUCONSTANTS READ OR COMPUTED FROM CONTROL CARDS//1H ,A1,I
           19,42H=NTF (LABEL CARD FLAG,MUST BE ZERO OR ONE))
            WRITE(6,102)C(2),NRF
        102 FORMAT(1H ,A1,I9,40H=NRF  (SET CARD FLAG,MUST BE ZERO OR ONE))
            WRITE(6,120)C(3),NGF
        120 FORMAT(1H ,A1,I9,42H=NGF  (GROUP CARD FLAG,MUST BE ZERO OR ONE))
            WRITE(6,124)C(14),NIF
        124 FORMAT(1H ,A1,I9,41H=NIF  (INTERCEPT FLAG,MUST BE ZERO OR ONE))
            WRITE(6,105)C(5),NFF
        105 FORMAT(1H ,A1,I9,43H=NFF  (FORMAT CARD FLAG,MUST BE ZERO OR ONE))
            WRITE(6,103)C(15),NOB
        103 FORMAT(1H ,A1,I9,56H=NOB  (NUMBER OF OBSERVATIONS,MUST BE GREATER T
           1HAN THREE))
            WRITE (6,104)C(4),NXS
        104 FORMAT(1H ,A1,I9,98H=NXS (NUMBER OF SETS OF INDEPENDENT VARIABLES
           1NOT FIXED,MUST BE GREATER THAN ONE AND LESS THAN 31))
            WRITE(6,106)C(6),NY
        106 FORMAT(1H  A1,I9,59H=NY   (NUMBER OF DEPENDENT VARIABLES,MUST BE LE
           1SS THAN NINE))
            WRITE(6,107)C(7),NF
        107 FORMAT(1H A1,I9, 73H=NF   (NUMBER OF SETS OF FIXED INDEPENDENT VARI
     1ABLES,MUST BE LESS THAN 18))
      WRITE(6,150)C(16),NSET
  150 FORMAT(1H ,A1,I9,101H=NSET(TOTAL NUMBER OF SETS OF INDEPENDENT VAR
     1IABLES,NF+NXS,MUST BE GREATER THAN ONE AND LESS THAN 31))
      WRITE(6,108)C(8),LM
  108 FORMAT(1H A1,I9,114H=LM  (LARGEST NUMBER OF SETS NOT FIXED TO BE I
     1NCLUDED IN ANY REGRESSION,MUST BE GREATER THAN ONE AND LESS THAN 1
     28))
      WRITE(6,116)C(10),NGX
  116 FORMAT (1H A1,I9,89H=NGX (NUMBER OF GROUPS,MUST BE EQUAL TO OR GRE
     1ATER THAN LM AND LESS THAN OR EQUAL TO NXS))
      WRITE(6,110)C(11),NVR
  110 FORMAT(1H A1,I9,76H=NVR (TOTAL NUMBER OF VARIABLES BEFORE TRANSFOR
     1MATIONS,MUST BE LESS THAN 51))
      WRITE(6,109)C(9),NVS
  109 FORMAT(1H A1,I9,75H=NVS (TOTAL NUMBER OF VARIABLES AFTER TRANSFORM
     1ATIONS,MUST BE LESS THAN 51))
      WRITE(6,122)RGK
  122 FORMAT(1H ,1X,F9.0,69H=RGK (NUMBER OF REGRESSIONS,MUST BE LESS THA
     1N 14,000 DIVIDED BY NY+1))
      WRITE(6,114)
  114 FORMAT(13H0INPUT FORMAT)
      WRITE(6,115)(FMT(I),I=1,12)
  115 FORMAT(1H 12A6)
      WRITE(6,123)(TY(I),I=1,NY)
  123 FORMAT(22H0DEPENDENT VARIABLE(S)/1H ,8A4)
      WRITE(6,111)
  111 FORMAT (43H0SET LABELS AND NUMBER OF VARIABLES PER SET)
      WRITE(6,112)(T(I),I=1,NSET)
  112 FORMAT(1H 30A4)
      WRITE(6,113)(NRX(I),I=1,NSET)
  113 FORMAT(1H 30I4)
      WRITE(6,119)C(12),(NG(I),I=1,NGX)
  119 FORMAT (50H0NUMBER OF SETS PER GROUP,SUM MUST BE EQUAL TO NXS/1H A
     11,I3,29I4)
      IF(PZ) 60,61,60
CPRINT CONTROL CARD ERROR
C
   60 WRITE(6,117)
  117 FORMAT(55H0CONTROL CARD ERROR,CHECK PREVIOUS OUTPUT FOR ASTERISKS)
      GO TO 64
C
CZERO MATRIX
C
   61 K=0
      DO 62 I=1,NP
      DO 62 J=I,NP
      K=K+1
   62 A(K)=0.0
C
CDATA INPUTAND OUTPUT OF MATRIX
C
   64 CALL DATAIN (A,X,FMT,NOB,NVR,NVS,PZ,NTRAN)
      IF(PZ) 45,46,45
   46 CALL MATOUT (A,NVS,NY,T,TY,TT)
   45 RETURN
      END
```

```
$IBFTC BAT17   DECK
      SUBROUTINE TROBS(X)
      DIMENSION X(75)
       WRITE(6,1)
05  1 FORMAT(40H0DUMMY SUBROUTINE TROBS HAS BEEN CALLED.)
      RETURN
      END
```

```
$IBFTC BAT19    DECK
      SUBROUTINE DATAIN  (A,X,FMT,NOB,NVR,NVS,PZ,NTRAN)
      DIMENSION A(14000),X(50),FMT(24)
      COMMON/BANK/DUMMY,NY
      COMMON/L1/DONE
        DIMENSION YTOT(8),YSQTOT(8),DUMMY(15766)
      INTEGER DONE,WMAR
      FNOB=NOB
      IF (PZ) 20,501,20
  501 IF (NTRAN) 502,20,502
   20 WRITE(6,1)
    1 FORMAT (36H0LISTING OF FIRST THREE SETS OF DATA)
      NP=NVS+1
       JJ=NVR-NY
       DO 101 I=1,8
       YSQTOT(I)=0.0
101    YTOT(I)=0.0
      DO 5 L=1,NOB
C
CREAD DATA
C
      READ(5,FMT)(X(I),I=1,NVR)
       DO 102 JJJ=1,NY
       JK=JJ+JJJ
       YSQTOT(JJJ)=YSQTOT(JJJ)+X(JK)*X(JK)
102     YTOT(JJJ)=YTOT(JJJ)+X(JK)
CSKIP MATRIX AND TRANSFORMATION AFTER CONTROL CARD ERROR
      IF(PZ) 5,2,5
    2 IF(L-4)6,12,14
CPRINT HEADING FOR LAST SET OF DATA
   12 WRITE(6,13)
   13 FORMAT(28H0LISTING OF LAST SET OF DATA/100H IF MISSING,PROGRAM HAS
     1 READ PAST DATA CARDS-CHECK NUMBER OF OBSERVATIONS,FORMAT CARD AND
     2 DATA CARDS)
      DATA WMAR/4HDONE/
   14 IF(L-NOB) 7,6,7
CPRINT FIRST THREE AND LAST SETS OF DATA
    6 WRITE(6,9)
    9 FORMAT(1H )
      WRITE(6,8)(X(I),I=1,NVS)
    8 FORMAT(1H ,8E15.7)
C
CCOMPUTE MOMENTS
C
    7 A(1)=A(1)+1.0
      DO 3 K=2,NP
    3 A(K)=A(K)+X(K-1)
      K=NP
      DO 4 I=2,NP
      DO 4 J=I,NP
      K=K+1
    4 A(K)=A(K)+X(I-1)*X(J-1)
    5 CONTINUE
      GO TO 505
  502 CALL TRNSX(X,NVR,NVS,FMT,NTRAN,NOB,A,YTOT,YSQTOT)
  505 DO 400 I=1,NY
400     YSQTOT(I)=YSQTOT(I)-YTOT(I)*YTOT(I)/FNOB
      WRITE(6,300)(I,YSQTOT(I),I=1,NY)
300    FORMAT(1H1,5X,24HCORRECTED SUM OF SQUARES/(/10X,1HY,I1,1H=,E15.7)
     1)
```

```
      C
      C --- CHECK FOR DONE OR QUIT CARD
      C
        10 READ(5,204)DONE
       204 FORMAT(A4)
           IF(DONE.EQ.4HMAR.OR.DONE.EQ.4HQUIT) GO TO 44
        43 IF(PZ) 10,11,11
        11 WRITE(6,118)
       118 FORMAT(101H0DONE CARD NOT FOUND AT END OF DATA,CHECK NUMBER OF OBS
          1ERVATIONS,FORMAT CARD,DATA CARDS,AND DONE CARD)
           PZ=-1
           GO TO 10
        44 RETURN
           END
```

```
$IBFTC BAT10    DECK
      SUBROUTINE TRNSX(X,NVR,NVS,FMT,NTRAN,NOB,A,YTOT,YSQTOT)
      COMMON/BANK/DUMMY,NY
      DIMENSION A(14000),X(50),FMT(24)
      DIMENSION IFORM(50),IA(50),IB(50),CONS(50),D(50)
      DIMENSION YTOT(8),YSQTOT(8),DUMMY(15766)
       DO 5 I=1,NTRAN
    5 READ (5,4)IFORM(I),IA(I),IB(I),CONS(I)
    4 FORMAT(3I2,F8.6)
      MI=1
      NP=NVS+1
      JJ=NVS-NY
      DO 101 I=1,8
      YSQTOT(I)=0.0
  101 YTOT(I)=0.0
      DO 105 LI=1,NOB
   71 READ (5,FMT)(X(I),I=1,NVR)
      DO 9 J=1,NVR
    9 D(J)=X(J)
      DO 30 I=1,NTRAN
      ITRAN=IFORM(I)
      M=IA(I)
      L=IB(I)
      K=CONS(I)
      GO TO (11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28),ITR
     1N
   11 X(M)=X(L)
      GO TO 30
   12 X(M)=CONS(I)*X(L)
      GO TO 30
   13 X(M)=X(L)+CONS(I)
      GO TO 30
   14 X(M)=X(L)+X(K)
      GO TO 30
   15 X(M)=X(L)*X(K)
      GO TO 30
   16 X(M)=X(L)/X(K)
      GO TO 30
   17 X(M)=1./X(L)
      GO TO 30
   18 X(M)=X(L)**CONS(I)
      GO TO 30
   19 X(M)=ALOG (X(L))
      GO TO 30
   20 X(M)=EXP (X(L))
      GO TO 30
   21 X(M)=ALOG(X(L))*.43429448
      GO TO 30
   22 X(M)=D(L)
      GO TO 30
   23 X(M)=SIN (X(L))
      GO TO 30
   24 X(M)=COS (X(L))
      GO TO 30
   25 X(M)=X(L)/CONS(I)
      GO TO 30
   26 X(M)=ABS (X(L))
      GO TO 30
   27 CALL TROBS(X)
      GO TO 30
   28 X(M)=X(L)-X(K)
   30 CONTINUE
C
      GO TO (31,114),MI
   31 WRITE(6,32)(D(I),I=1,NVR)
   32 FORMAT (32H0FIRST UNTRANSFORMED OBSERVATION/(1X,9E14.7))
      WRITE(6,33)(X(I),I=1,NVS)
   33 FORMAT (30H0FIRST TRANSFORMED OBSERVATION/(1X,9E14.7))
CPRINTHEADING FOR LAST SET OF DATA
      WRITE(6,113)
  113 FORMAT (28H0LISTING OF LAST SET OF DATA/100H IF MISSING,PROGRAM HA
     1S READ PAST DATA CARDS-CHECK NUMBER OF OBSERVATIONS,FORMAT CARD AN
     2D DATA CARDS)
      MI=2
  114 IF(LI-NOB)107,106,107
CPRINTLAST SET OF DATA
  106 WRITE(6,109)(D(I),I=1,NVR)
  109 FORMAT (14H UNTRANSFORMED/(1X,9E14.7))
      WRITE(6,108)(X(I),I=1,NVS)
  108 FORMAT (12H TRANSFORMED/(1X,9E14.7))
C
CCOMPUTE MOMENTS
C
  107 DO 102 JJJ=1,NY
      JK=JJ+JJJ
      YSQTOT(JJJ)=YSQTOT(JJJ)+X(JK)*X(JK)
  102 YTOT(JJJ)=YTOT(JJJ)+X(JK)
      A(1)=A(1)+1.0
      DO 103 KI=2,NP
  103 A(KI)=A(KI)+X(KI-1)
      KI=NP
      DO 104 II=2,NP
      DO 104 JI=II,NP
      KI=KI+1
  104 A(KI)=A(KI)+X(II-1)*X(JI-1)
  105 CONTINUE
      RETURN
      END
```

INPUT DECK FOR SAMPLE RUN OF SCREEN

```
TEST RUN OF SCREEN
1       15          2                                                   4 4
(F4.1,2(F5.1),1X,F3.0)
CPAMBSA POUD
0202022.0
03010102
010203
010304
20.8 22.1  0.4 147
25.1 17.6  1.6 106
19.3 17.3  0.2 133
30.5 38.9  2.3 202
29.7 36.1  5.0 378
29.4 32.9  4.0 256
30.9 29.7  5.0 228
29.2 26.0  2.8 199
22.3 27.5  2.8 317
38.9 36.5  2.1 274
20.1 19.8  2.4 146
35.4 32.2  1.9 176
27.7 34.3  4.1 325
21.1 21.9  0.0 110
27.6 33.9  0.5 242
QUIT
```

```
TEST RUN OF SCREEN

CONSTANTS READ OR COMPUTED FROM CONTROL CARDS

         1=NIF  (LABEL CARD FLAG,MUST BE ZERO OR ONE)
        -0=NRF  (SET CARD FLAG,MUST BE ZERO OR ONE)
        -0=NGF  (GROUP CARD FLAG,MUST BE ZERO OR ONE)
        -0=NIF  (INTERCEPT FLAG,MUST BE ZERO OR ONE)
        -0=NFF  (FORMAT CARD FLAG,MUST BE ZERO OR ONE)
        15=NOB  (NUMBER OF OBSERVATIONS,MUST BE GREATER THAN THREE)
         2=NXS  (NUMBER OF SETS OF INDEPENDENT VARIABLES NOT FIXED,MUST BE GREATER THAN ONE AND LESS THAN 31)
         1=NY   (NUMBER OF DEPENDENT VARIABLES,MUST BE LESS THAN NINE)
        -0=NF   (NUMBER OF SETS OF FIXED INDEPENDENT VARIABLES,MUST BE LESS THAN 18)
         2=NSET (TOTAL NUMBER OF SETS OF INDEPENDENT VARIABLES,NF+NXS,MUST BE GREATER THAN ONE AND LESS THAN 31)
         2=LM   (LARGEST NUMBER OF SETS NOT FIXED TO BE INCLUDED IN ANY REGRESSION,MUST BE GREATER THAN ONE AND LESS THAN 18)
         2=NGA  (NUMBER OF GROUPS,MUST BE EQUAL TO OR GREATER THAN LM AND LESS THAN OR EQUAL TO NXS)
         4=NVR  (TOTAL NUMBER OF VARIABLES BEFORE TRANSFORMATIONS,MUST BE LESS THAN 51)
         3=NVS  (TOTAL NUMBER OF VARIABLES AFTER TRANSFORMATIONS,MUST BE LESS THAN 51)
        3.=RGN  (NUMBER OF REGRESSIONS,MUST BE LESS THAN 14,000 DIVIDED BY NY+1)

INPUT FORMAT
(F4.1,2(F5.1),1X,F3.0)
DEPENDENT VARIABLE(S)
POUD

SET LABELS AND NUMBER OF VARIABLES PER SET
CPAMBSA
   1    1

NUMBER OF SETS PER GROUP,SUM MUST BE EQUAL TO NXS
   1    1

FIRST UNTRANSFORMED OBSERVATION
  .2080000E+02  .2410000E+02  .4000000E+00  .1470000E+03

FIRST TRANSFORMED OBSERVATION
  .2280000E+02  .4000000E+00  .1470000E+03

LISTING OF LAST SET OF DATA
IF MISSING,PROGRAM HAS READ PAST DATA CARDS-CHECK NUMBER OF OBSERVATIONS,FORMAT CARD AND DATA CARDS
UNTRANSFORMED
  .2760000E+02  .3390000E+02  .5000000E+00  .2420000E+03
TRANSFORMED
  .2960000E+02  .5000000E+00  .2420000E+03

     CORRECTED SUM OF SQUARES

         Y1=   .9608093E+05


TEST RUN OF SCREEN

REGRESSION SCREEN OUTPUT
COEFFICIENTS OF COLINEARITY AND DETERMINATION

                       IDENTIFICATION OF VARIABLES                         C OF C   POUD
1                                                               CPAM      .35E-01   .182
1                                                               BSA       .32E+00   .496


2                                                          CPAMBSA        .91E-02   .515
```

MATEXP

INTRODUCTION

MATEXP is a general purpose program for the solution of systems of ordinary differential equations by the matrix exponential method. It was written in 1967 by S. J. Ball and R. K. Adams for the IBM 7090 and has been converted for use on both the NCAR and SCOPE systems of the CDC 6400 computer.

MATEXP has several advantages over standard numerical integration routines. It gives virtually exact solutions to constant-coefficient homogeneous equations and to non-homogeneous equations for which the forcing functions are constant during the computation interval. The method has also been extended to nonlinear equations and equations with time varying coefficients. This use makes it particularly effective for systems analysis in both the engineering and ecological areas. It has also been very effective when used to calculate the sensitivities of the time response of a system to changes in parameter values.

METHOD OF OPERATION

A detailed explanation of the program operation is described in ORNL-TM-1933 entitled *MATEXP, a general purpose digital computer program for solving ordinary differential equations by the Matrix Exponential Method* by S. J. Ball and R. K. Adams. With reference to this article, only the types of systems which the program is capable of solving will be mentioned here.

The most basic type of system is the linear, homogeneous system of the type:

$$\frac{dX}{dt} = AX$$

where X is the column vector of state variables and A represents the matrix of coefficients. If the system is non-homogeneous but linear, it is expressed as

$$\frac{dX}{dt} = AX + Z,$$

where Z is the disturbance, or forcing function, vector. The methods of operation are very similar for these two systems. However, if either the coefficient matrix or the disturbance vector becomes nonlinear, the method of solution is changed somewhat and a user supplied subroutine is required which will recalculate the necessary coefficients, or forcing functions, at each time interval. This subroutine is entitled DSTRB in the ORNL report. An example of the nonlinear problem is shown in the program listing.

INPUT REQUIREMENTS

The MATEXP program consists of the main program and two subroutines OUTPUT and DISTRB, plus any other subroutines called by DISTRB. Even if DISTRB is not used, a dummy must be included.

For each case run on MATEXP, the data will include (if appropriate):

1. MATEXP Control Card
2. Coefficient matrix (A)
3. Initial Condition Vector (XIC)
4. Any data read in by subroutine DISTRB
5. Fixed forcing function vector (Z).

Input Data Formats - MATEXP Main Program

1. Control Card

Column	1-2		6-7		11-20	21-30	31-40	41-50	51-60	61-62
Format	I2	3X	I2	3X	F10.0	F10.0	F10.0	F10.0	F10.0	I2
Input	NE		LL		P	TZERO	T	TMAX	PLTINC	MATYES

Control Card - (continued)

Column	63-64	65-66	67-69	70	71-72	73-74	75-80
Format	I2	I2	I3	I1	I2	I2	F6.0
Input	ICSS	JFLAG	ITMAX	LASTCC	IIZ	ICONTR	VAR

NE	number of equations
LL	coefficient matrix tag number
P	precision of C and HP (recommend 10^{-6} or less).
TZERO	zero time
T	computation time interval
TMAX	maximum time
PLTINC	printing time interval
MATYES	coefficient matrix (A) control flag:

1 = use previous A and T

2 = read new coefficients to alter A

3 = read entire new A (nonzero values)

4 = DISTRB to calculate entire new A

5 = read some, DISTRB to calculate others

6 = IDSTRB to alter some A elements

ICSS initial condition vector (XIC) flag:

1 = read in all new nonzero values

2 = read new values to alter previous vector

3 = use previous vector

4 = vector = 0

5 = use last value of X vector from previous run

JFLAG forcing function (Z) flag:

1 thru 4 = same as for ICSS for constant Z

5 = call DISTRB at each time step for variable Z

ITMAX	maximum number of terms in series approximation of exp (AT)
LASTCC	nonzero for last case (blank otherwise).
IIZ	row of Z if only one nonzero, otherwise = 0
ICONTR	for internal control options:

0 = read new control card for next case

1 = go to 212 call DISTRB for new A or T

-1 = go to 215 call DISTRB for new initial conditions

VAR	maximum allowable value of largest coefficient matrix element *T (Recommend VAR = 1.0)

2. Coefficient Matrix A Format 4(2I3, E12.3) - Include if MATYES = 2, 3, or 5.

Column	1-3	4-6	7-18	
Format	I3	I3	E12.3	Repeat
Input	Row No.	Col. No.	COEFFICIENT	4 per card

Notes: 1. All row and column number entries on a card must be nonzero.

2. Insert blank card after all coefficient matrix data is read in.

3. Data can be entered in floating point (F) format with decimal point.

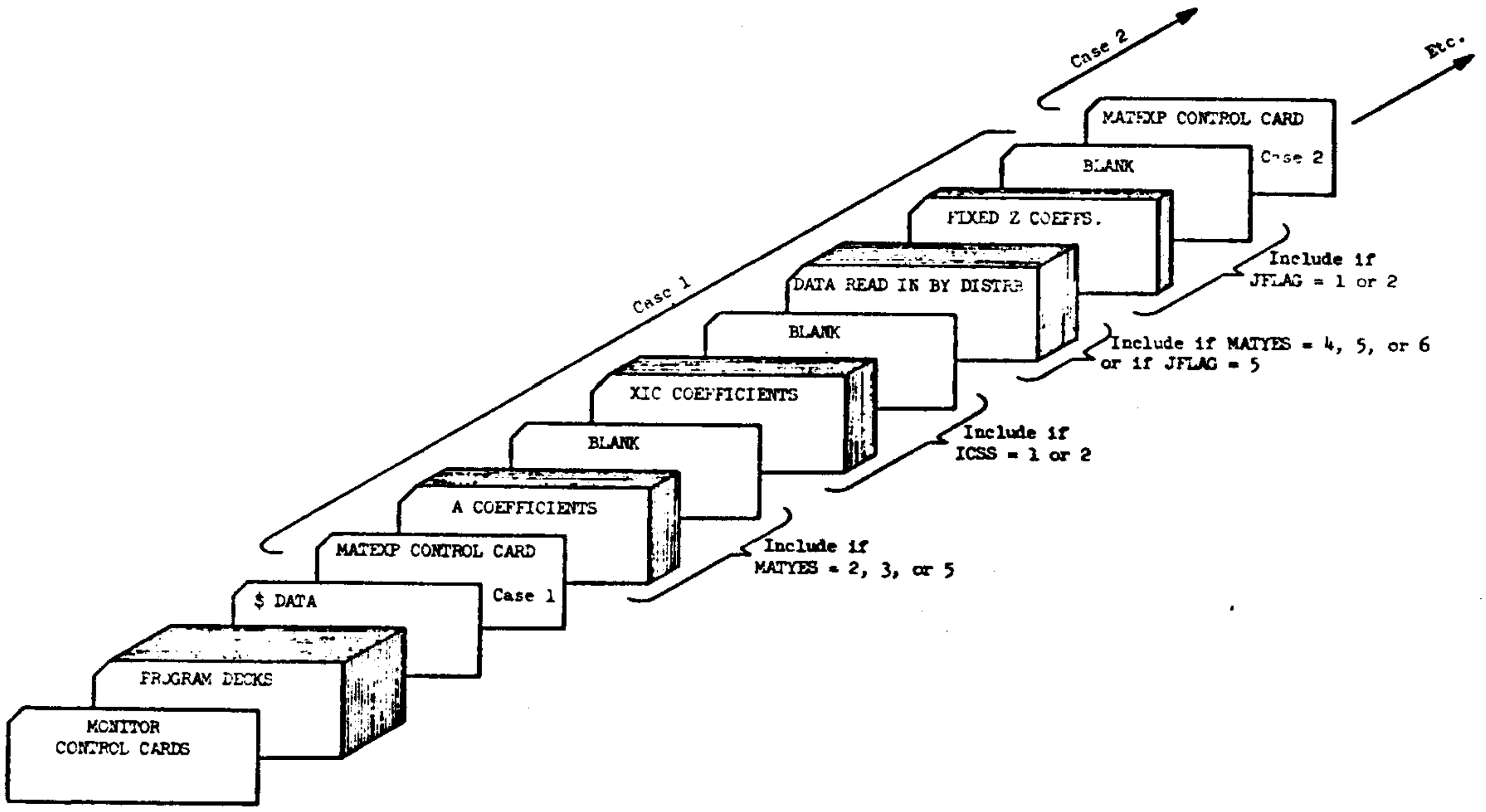
3. Initial Condition Vector XIC Format (I2, 5(I3, E12.3)) - Include if ICSS = 1 or 2

Column	1-2	3-5	6-17	
Format	I2	I3	E12.3	Repeat Cols. 3-17,
Input	MM	Row No.	I.C. Value	5 per card

Notes: 1. All row number entries on a card must be nonzero.

2. Insert blank card after all XIC data is read in.

3. Data can be entered in F format.



MATEXP Data Arrangement

4. Disturbance Vector Z Format (I2, 5(I3, E12.3)) - Include if JFLAG = 1 or 2

Column	1-2	3-5	6-17	
Format	I2	I3	E12.3	Repeat Cols. 3-17,
Input	KK	Row No.	Z Value	5 per card

Note: See notes under 3.

To aid in understanding the MATEXP program, Fig. 1 summarizes the data arrangement.

SAMPLE LISTING AND OUTPUT

The listing shown is the solution for a sample ecology problem with nonlinear coefficients designed by B. C. Patten. A detailed description of this model is also included with the ORNL report.

```
      PROGRAM MATEXP
C-----PROGRAM MATEXP  FOR THE 7090 - FORTRAN 4
C-----
C-----THIS PROGRAM CALCULATES THE SOLUTION OF A MATRIX OF FIRST
C-----ORDER, SIMULTANEOUS DIFFERENTIAL EQUATIONS W/ CONSTANT COEFFICIENT
C-----OF THE FORM DX/DT = AX + Z.
C-----
C-----THE METHOD IS PAYNTER-S MATRIX EXPONENTIAL METHOD
C-----
C-----THE SOLUTION IS GIVEN FOR INCREMENTS OF THE INDEPENDENT
C-----VARIABLE (T) FROM TZERO THROUGH TMAX
C-----
C-----COMPUTES MATRICES C = EXP(A*T) AND
C-----                 HP = (C-I)*A INVERSE
C-----SOLUTION X(N*T) = C*X((N-1)*T)+HP*Z((N-1)*T)
C-----SERIES CALCULATION OF C AND HP MONITORED TO
C-----      ASSURE SPECIFIED SIGNIFICANCE.
C-----IF T IS REDUCED FOR C AND HP CALCS.,
C-----ORIGINAL ARGUEMENTS ARE RESTORED BY -
C-----      C(2*T)=C(T)*C(T)
C-----      HP(2*T)=HP(T)+C(T)*HP(T)
C-----
C-----OUTPUT FROM THE PROGRAM IS PRINTED AT INTERVALS PLTINC.
C-----THE PROGRAM USES SUBROUTINES DISTRB AND OUTPUT
C-----
C-----INPUT FOR THE PROGRAM CONSISTS OF
C-----      ONE CONTROL CARD
C-----      THE COEFFICIENT MATRIX A (UP TO 60 X 60)
C-----      THE INITIAL CONDITION VECTOR X
C-----      A FIXED DISTURBANCE VECTOR Z
C-----
C-----A VARYING Z CAN BE GENERATED BY DISTRB
C-----VARIABLE COEFFICIENT EQUATIONS MAY BE SOLVED BY APPROPRIATE
C-----FUDGING OF THE DISTURBANCE FUNCTION SUBROUTINE.
C-----
C-----CONTROL CARD INPUT INFORMATION
C-----      NE=NO. OF EQUATIONS (I2)
C-----      LL=COEFF. MATRIX TAG NO.  (I2)
C-----      P=PRECISION OF C AND HP (F10.0) -  RECOMMEND 1.0E-6 OR LESS
C-----      TZERO=ZERO TIME  (F10.0)
C-----      T=COMPUTATION TIME INTERVAL (F10.0)
C-----      TMAX=MAXIMUM TIME  (F10.0)
C-----      PLTINC=PRINTING TIME INTERVAL (F10.0)
C-----      MATYES=COEFF. MATRIX (A) CONTROL FLAG (I2)
C-----        1=USE PREVIOUS A AND T
C-----        2=READ NEW COEFF.S TO ALTER A
C-----        3=READ ENTIRE NEW A  (NON-ZERO VALUES)
C-----        4=DISTRB TO CALC. ENTIRE NEW A
C-----        5=READ SOME, DISTRB TO CALC. OTHERS
C-----        6=DISTRB TO ALTER SOME A ELEMENTS
C-----      ICSS=INITIAL CONDITION VECTOR (XIC) FLAG (I2)
C-----        1=READ IN ALL NEW NON-ZERO VALUES
C-----        2=READ NEW VALUES TO ALTER PREVIOUS VECTOR
C-----        3=USE PREVIOUS VECTOR
C-----        4=VECTOR=0
C-----        5=USE LAST VALUE OF X VECTOR FROM PREVIOUS RUN
```

```
C-----     JFLAG=FORCING FUNCTION (Z) FLAG (I2)
C-----       1 THRU 4=SAME AS FOR ICSS FOR CONSTANT Z
C-----       5=CALL DISTRB AT EACH TIME STEP FOR VARIABLE Z
C-----ITMAX = MAX. NO. OF TERMS IN SERIES APPROX.
C-----     OF EXP(AT).  (I3)
C-----LASTCC = NON-ZERO FOR LAST CASE (I1)
C-----I1Z = ROW NO. OF Z IF ONLY ONE NON-ZERO,
C-----     OTHERWISE =0    (I2)
C-----ICONTR - FOR INTERNAL CONTROL OPTIONS   (I2) -
C-----     0=READ NEW CONTROL CARD FOR NEXT CASE
C-----     1=GO TO 212 CALL DISTRB FOR NEW A OR T
C-----    -1=GO TO 215 CALL DISTRB FOR NEW I.C.-S
C-----VAR = MAX. ALLOWABLE VALUE OF LARGEST COEFF. MATRIX ELEMENT * T
C-----(RECOMMEND VAR=1.0)   (F6.0)
C-----
      DIMENSION A(60,60), C(60,60), HP(60,60), QPT(60,60), X(60), Y(60),
     1 Z(60), XIC(60), TQP(60)
C-----
      COMMON C,HP,A,QPT,X,Z,Y,ITMAX,KK,LL,MM,JJFLAG,XIC,NI,TIME,TMAX,TZE
     1RO,NE,TQP,T,I1Z,ICONTR,PLTINC,MATYES,ICSS,JFLAG,PLT
C-----
C-----K=CASE NUMBER
C-----NI=0 ON 1-ST PASS.  SET TO 1 ON 1-ST CALL OF OUTPUT.
      K=1
      NI=0
C-----
1     READ (5,94) NE,LL,P,TZERO,T,TMAX,PLTINC,MATYES,ICSS,JFLAG,ITMAX,LA
     1STCC,I1Z,ICONTR,VAR
      IF(NE.EQ.0) STOP
C-----
C-----COEFFICIENT MATRIX INPUT
      GO TO (7,4,2,2,2,7), MATYES
2     DO 3 I=1,NE
      DO 3 J=1,NE
3     A(I,J)=0.0
      IF (MATYES-4) 4,7,4
4     DO 6 I=1,1379
C-----MATRIX ELEMENTS  5(ROW, COLUMN, VALUE)
C-----ALL I AND J ENTRIES ON CARD MUST BE NON-ZERO.
C-----A BLANK CARD IS REQUIRED AFTER ALL ELEMENTS ARE READ IN.
      READ (5,95) I1,J1,D1,I2,J2,D2,I3,J3,D3,I4,J4,D4
      IF (I1) 7,7,5
5     A(I1,J1)=D1
      A(I2,J2)=D2
      A(I3,J3)=D3
6     A(I4,J4)=D4
C-----
C-----INITIAL CONDITION VECTOR XIC INPUT
7     GO TO (8,10,15,13,15), ICSS
8     DO 9 I=1,NE
9     XIC(I)=0.0
10    DO 12 I=1,15
C-----ALL ROW (I) ENTRIES MUST BE NON-ZERO
C-----A BLANK CARD IS REQUIRED AFTER ALL ELEMENTS ARE READ IN.
      READ (5,96) MM,I11,D11,I12,D12,I13,D13,I14,D14,I15,D15
      IF (I11) 15,15,11
```

```
11     XIC(I11)=D11
       XIC(I12)=D12
       XIC(I13)=D13
       XIC(I14)=D14
12     XIC(I15)=D15
C-----
13     MM=0
       DO 14 I=1,NE
14     XIC(I)=0.0
15     IF (ICSS-5) 16,18,16
16     DO 17 I=1,NE
17     X(I)=XIC(I)
18     IF (MATYES-3) 20,20,19
19     CALL DISTRB
20     JJFLAG=0
C-----QPTMP = MAX. PERMISSIBLE ELEMENT OF QPT FOR  8 DECIMAL COMPUTER
C-----MATRIX CALC. LOSES SIGNIFICANCE IF LARGEST
C-----  ELEMENT IN SERIES APPROX. MATRIX QPT IS
C-----  GREATER THAN P*1.0E8
       QPTMP=P*1.0E8
C-----
       WRITE (6,97) K,NE,P,T,PLTINC,MATYES,ICSS,JFLAG,ICONTR,ITMAX,I1Z,VA
      1R,QPTMP
C-----
C-----
       PLTINC=PLTINC*0.9999
C-----
       JFK=0
       IF (MATYES-1) 66,66,21
C-----SCAN MATRIX FOR MAX. AND MIN. NON-ZERO ELEMENTS.
21     IMAX=1
       JMAX=1
       AMAX=ABS(A(1,1))
       DO 23 I=1,NE
       DO 23 J=1,NE
       IF (AMAX-ABS(A(I,J))) 22,23,23
22     AMAX=ABS(A(I,J))
       IMAX=I
       JMAX=J
23     CONTINUE
       IMIN=IMAX
       JMIN=JMAX
       AMIN=AMAX
       DO 26 I=1,NE
       DO 26 J=1,NE
       IF (A(I,J)) 24,26,24
24     IF (ABS(A(I,J))-AMIN) 25,26,26
25     AMIN=ABS(A(I,J))
       IMIN=I
       JMIN=J
26     CONTINUE
       RATIO=AMAX/AMIN
C-----AMIN = MINIMUM NON-ZERO ELEMENT
       ISTOR=0
       ADT=AMAX*T
       DO 28 I=1,11
      IF (VAR-ADT) 27,29,29
 27   ISTOR=ISTOR+1
 28   ADT=ADT*0.5
 29   T=ADT/AMAX
C-----COMPUTATION INTERVAL T IS HALVED ISTOR
C-----  TIMES (10=MAX.) SO MAX. ELEMENT IN A*T
C-----  IS LESS THAN VAR.
      WRITE (6,98) IMAX,JMAX,A(IMAX,JMAX),ADT,T,IMIN,JMIN,A(IMIN,JMIN),R
     1ATIO
C-----
      IF (ISTOR-10) 31,30,30
 30   WRITE (6,99)
      GO TO 91
C-----CALCULATION OF MATRIX EXPONENTIALS C AND HP
 31   DO 32 I=1,NE
      DO 32 J=1,NE
 32   C(I,J)=0.
C-----
      DO 33 I=1,NE
 33   C(I,I)=1.
C-----
C-----SKIP HP CALCS. FOR HOMOGENEOUS EQUATIONS
      IF (JFLAG-4) 34,37,34
 34   DO 35 I=1,NE
      DO 35 J=1,NE
 35   HP(I,J)=0.
C-----
      DO 36 I=1,NE
 36   HP(I,I)=T
C-----
 37   PE=0.0
C-----
      DO 38 I=1,NE
      DO 38 J=1,NE
 38   QPT(I,J)=C(I,J)
C-----
C-----NOW FORM THE MATRIX EXPONENTIALS C=EXP(A*T) AND HP=((C-I)*A INVERS
C-----
      AL=1.0
C-----
      DO 50 KL=1,ITMAX
C-----
      KLM=KL
      ALL=T/AL
      AL=AL+1.0
      TALLL=T/AL
C-----
      DO 40 I=1,NE
C-----
C-----
      DO 39 J=1,NE
      TQP(J)=0.0
      DO 39 KX=1,NE
 39   TQP(J)=TQP(J)+QPT(I,KX)*A(KX,J)
C-----
      DO 40 J=1,NE
```

```
40     QPT(I,J)=TQP(J)*ALL
C-----
C-----QPT=MATRIX TERM IN SERIES APPROX. =((A*T)**K)/K FACTORIAL
C-----
       DO 41 I=1,NE
       DO 41 J=1,NE
41     C(I,J)=C(I,J)+QPT(I,J)
C-----
       IF (JFLAG-4) 42,45,42
C-----
42     IF (ITMAX-KL) 45,45,43
43     DO 44 I=1,NE
       DO 44 J=1,NE
44     HP(I,J)=HP(I,J)+QPT(I,J)*TALLL
C-----
C-----
C-----FIND MAX ABS ELEMENT IN QPT AND CALL IT PMK
C-----LARGEST QPT ELEMENT USUALLY IN ROW IMAX, COLUMN JMAX
45     PMK=ABS(QPT(IMAX,JMAX))
       IF (QPTMP-PMK) 53,53,46
46     IF (PMK-P) 47,47,50
C-----SCAN OTHER QPT ELEMENTS ONLY WHEN QPT(IMAX, JMAX) IS LESS THAN P
47     DO 48 I=1,NE
       DO 48 J=1,NE
48     PMK=AMAX1(PMK,ABS(QPT(I,J)))
       IF (PMK-P) 49,49,50
C-----
C-----PRESENT MAX. QPT ELEMENT SHOULD BE LESS THAN
C-----  HALF PREVIOUS MAX. TO INSURE CONVERGENCE
49     IF (PE-2.*PMK) 50,51,51
50     PE=PMK
C-----
51     WRITE (6,100) KLM
C-----
C-----
       IF (ITMAX-1) 66,66,52
52     IF (KLM-ITMAX) 56,53,53
C-----
53     T=T*0.5
       JFK=JFK+1
       IF (JFK-7) 55,54,54
54     WRITE (6,101) PMK
       GO TO 91
55     WRITE (6,102) KLM,PMK,T
       GO TO 31
56     ISTOR=ISTOR+JFK
C-----ORIGINAL ARGUMENTS OF C AND HP MATRICES RESTORED IF ISTOR GREATER
       IF (ISTOR) 66,66,57
57     WRITE (6,103) ISTOR
       DO 65 KR=1,ISTOR
       IF (JFLAG-4) 58,61,58
C-----SKIP HP CALCS. FOR HOMOGENEOUS EQUATIONS
58     DO 60 I=1,NE
       DO 59 J=1,NE
       TOP(J)=0.0
       DO 59 KX=1,NE
```

```
59    TQP(J)=TQP(J)+HP(I,KX)*C(KX,J)
      DO 60 J=1,NE
60    HP(I,J)=TQP(J)+HP(I,J)
C-----
61    DO 62 I=1,NE
      DO 62 J=1,NE
62    QPT(I,J)=0.0
      DO 63 I=1,NE
      DO 63 J=1,NE
      DO 63 KX=1,NE
63    QPT(I,J)=QPT(I,J)+C(I,KX)*C(KX,J)
      DO 64 I=1,NE
      DO 64 J=1,NE
64    C(I,J)=QPT(I,J)
65    T=2.0*T
C-----
C-----C(I,J) IS THE MATRIX EXPONENTIAL C=EXP(A*T)
C-----AND HP(I,J) IS THE ((C-I)*A INVERSE) MATRIX
C-----NOW WE READ (OR CALL SUBROUTINE FOR) DISTURBANCE VECTOR
C-----
66    TIME=TZERO
      PLT=0.
      GO TO (69,71,76,74,67), JFLAG
67    IF (MATYES-3) 68,68,76
68    CALL DISTRB
      I1Z=I1Z
      GO TO 76
C-----
69    DO 70 I=1,NE
70    Z(I)=0.0
71    DO 73 I=1,15
C-----ALL ROW (I) ENTRIES MUST BE NON-ZERO
C-----A BLANK CARD IS REQUIRED AFTER ALL ELEMENTS ARE READ IN.
      READ (5,96) KK,I21,D21,I22,D22,I23,D23,I24,D24,I25,D25
      IF (I21) 76,76,72
72    Z(I21)=D21
      Z(I22)=D22
      Z(I23)=D23
      Z(I24)=D24
73    Z(I25)=D25
C-----
74    KK=0
      DO 75 I=1,NE
75    Z(I)=0.
C-----
C-----ON 1-ST CALL OF OUTPUT NI SET TO 1
76    CALL OUTPUT
C-----
C-----NOW COMES THE EQUATION SOLUTION BASED ON
C-----X(NT)=M*X(NT-1)+((M-I)A INV.)*Z(NT-1)
C-----
77    IF (JFLAG-4) 82,78,80
78    DO 79 I=1,NE
      Y(I)=C(I,1)*X(1)
      DO 79 J=2,NE
79    Y(I)=Y(I)+C(I,J)*X(J)
```

```
      IF (I1Z) 87,87,83
80    IF (JJFLAG) 81,82,81
81    CALL DISTRB
82    IF (I1Z) 85,85,78
C-----ONLY ONE Z-TERM CALC. IF I1Z IS GREATER THAN ZERO
83    DO 84 I=1,NE
84    Y(I)=Y(I)+HP(I,I1Z)*Z(I1Z)
      GO TO 87
85    DO 86 I=1,NE
      Y(I)=C(I,1)*X(1)+HP(I,1)*Z(1)
      DO 86 J=2,NE
86    Y(I)=Y(I)+C(I,J)*X(J)+HP(I,J)*Z(J)
87    DO 88 I=1,NE
88    X(I)=Y(I)
C-----
C-----ONE TIME INCREMENT OF THE SOLUTION HAS JUST BEEN FOUND
C-----NOW PLOT AND PRINT IF PLTINC INTERVAL HAS ELAPSED
C-----
      JJFLAG=1
      TIME=TIME+T
      PLT=PLT+T
      IF (PLT-PLTINC) 90,89,89
89    CALL OUTPUT
      PLT=0.
90    IF (TIME-TMAX) 77,91,91
91    IF (LASTCC) 93,92,93
92    K=K+1
      NI=0
      PLT=0.0
      IF (ICONTR) 68,1,19
93    STOP
C-----
94    FORMAT (2(I2,3X),5F10.0,3I2,I3,I1,2I2,F6.0)
95    FORMAT (4(2I3,E12.3))
96    FORMAT (I2,5(I3,E12.3))
97    FORMAT (12H1MATEXP CASE,I3/17H NO. OF EQUATIONS,I3/20H SPECIFIED P
     1RECISION,F12.8/6H TIME ,8HINTERVAL,F18.8/15H PLOT INCREMENT,F17.8/
     2/16H CONTROL FLAGS -/1H ,5X,6HMATYES,I4/1H ,5X,4HICSS,I6/1H ,5X,5H
     3JFLAG,I5/1H ,5X,6HICONTR,I4/34H0MAX. TERMS IN EXPONENTIAL APPROX.,
     4I5/13H SINGLE Z ROW,I4/20H MAX. ALLOWABLE A*DT,F9.3/27H MAX. ALLOW
     5ABLE QPT ELEMENT,F11.3)
98    FORMAT (31H0MAX.COEFF. MATRIX ELEMENT = A(,I2,1H,,I2,3H) =,E15.4/1
     13H MAX. A*DT = ,F12.8,2X,14HWITH DELTA T =,F15.8/30H0MINIMUM NON-Z
     2ERO ELEMENT = A(,I2,1H,,I2,3H) =,E15.4/18H RATIO AMAX/AMIN =,E15.4
     3)
99    FORMAT (34H0A*DT STILL GREATER THAN ALLOWABLE,19H AFTER 10 HALVING
     1S.)
100   FORMAT (44H0NO. OF TERMS IN SERIES APPROX. OF MATEXP = ,I2)
101   FORMAT (32H07 TRIES AT HALVING T N.G., PMK=,F12.6)
102   FORMAT (21H0MAX. ELEMENT IN TERM,I3,8H0F QPT =,E11.3/35H TRY HALVE
     1D TIME INTERVAL DELTA T =,F15.8)
103   FORMAT (26H0TOTAL NO. OF T HALVINGS =,I3)
      END
      SUBROUTINE VARCO (XTR,TX)
C-----FOR USE WITH DISTRB AND MATEXP FOR
C-----VARIABLE Z-S.  GIVES 1-ST ORDER EXTRAP.
C-----FOR AVG. X AND TIME, PLUS RESTART
C-----ON 1-ST INTERVAL.  DISTRB FORM =
C-----       CALC. MATRIX COEFF.-S, ETC. IF NI=0
C-----       CALL VARCO(XTR,TX)
C-----       CALC. Z-S USING XTR(I)-S AND TX (TIME).
C-----
      DIMENSION A(60,60), C(60,60), HP(60,60), QPT(60,60), X(60), Y(60),
     1 Z(60), XIC(60), TQP(60)
      COMMON C,HP,A,QPT,X,Z,Y,ITMAX,KK,LL,MM,JJFLAG,XIC,NI,TIME,TMAX,TZE
     1RO,NE,TQP,T,IIZ,ICONTR,PLTINC,MATYES,ICSS,JFLAG,PLT
      DIMENSION XTR(60), XL(60)
C-----
      IF (NI) 1,1,3
C-----FIRST ENTRY
1     NV=1
      TX=TZERO+0.5*T
      DO 2 I=1,NE
2     XTR(I)=XIC(I)
      GO TO 8
3     IF (NV) 6,6,4
C-----SECOND ENTRY
4     NV=0
      TIME=TZERO
      PLT=0.0
      DO 5 I=1,NE
      XL(I)=XIC(I)
      XTR(I)=0.5*(XL(I)+X(I))
5     X(I)=XIC(I)
      GO TO 8
C-----ENTRIES AFTER SECOND
6     TX=TIME+0.5*T
      DO 7 I=1,NE
      XTR(I)=X(I)+0.5*(X(I)-XL(I))
7     XL(I)=X(I)
8     RETURN
      END
```

```
      SUBROUTINE DFG (XD,ZD)
C-----
C-----EQUIVALENT TO 8 DFG-S WITH UP TO 32
C-----POINTS EACH.  CALLED BY DISTRB.
C-----
C-----INPUTS ARE
C-----  NDFGS  NO. OF DFG-S USED
C-----  NPTS  NO. OF POINTS IN EACH DFG
C-----  XP   INDEPENDENT VARIABLE DFG POINTS
C-----  ZP   DEPENDENT VARIABLE DFG POINTS
C-----
```

```
C-----XD IS THE INPUT VARIABLE AND ZD THE OUTPUT
C-----
      DIMENSION A(60,60), C(60,60), HP(60,60), QPT(60,60), X(60), Y(60),
     1 Z(60), XIC(60), TQP(60)
      COMMON C,HP,A,QPT,X,Z,Y,ITMAX,KK,LL,MM,JJFLAG,XIC,NI,TIME,TMAX,TZE
     1RO,NE,TQP,T,IIZ,ICONTR,PLTINC,MATYES,ICSS,JFLAG,PLT
      DIMENSION XP(32,8), ZP(32,8), SL(32,8), NPTS(8), JP(8), ZD(8), XD(
     18)
C-----
C-----
      IF (NI) 6,1,6
C-----FIRST CALL COMP.
C-----
1     READ (5,18) NDFGS,NPTS
      DO 2 I=1,NDFGS
      NP=NPTS(I)
      READ (5,19) (XP(J,I),ZP(J,I),J=1,NP)
2     WRITE (6,20) I,(XP(J,I),ZP(J,I),J=1,NP)
      DO 3 I=1,NDFGS
      M=NPTS(I)-1
      DO 3 J=1,M
3     SL(J,I)=(ZP(J+1,I)-ZP(J,I))/(XP(J+1,I)-XP(J,I))
C-----
      DO 5 I=1,NDFGS
      DO 4 J=2,32
      IF (XD(I)-XP(J,I)) 5,5,4
4     CONTINUE
5     JP(I)=J
C-----
C-----CALCS. MADE EACH TIME
6     DO 17 I=1,NDFGS
      J=JP(I)
7     IF (XD(I)-XP(J,I)) 8,14,12
8     IF (XD(I)-XP(J-1,I)) 9,11,16
9     J=J-1
      IF (J-1) 10,10,8
10    J=2
      GO TO 15
11    ZD(I)=ZP(J-1,I)
      GO TO 17
12    J=J+1
      IF (NPTS(I)-J) 13,7,7
13    J=NPTS(I)
      GO TO 15
14    ZD(I)=ZP(J,I)
      GO TO 17
15    WRITE (6,21) I
C-----
16    ZD(I)=ZP(J-1,I)+SL(J-1,I)*(XD(I)-XP(J-1,I))
C-----JP(I) STORES VALUE OF XD LOCATION
C-----  TO USE AS FIRST TRY NEXT TIME.
17    JP(I)=J
C-----
      RETURN
C-----
18    FORMAT (I2,8X,8I3)
```

```
19     FORMAT (8E10.3)
20     FORMAT (4HODFG,I3,17H XP AND ZP INPUTS/(1H0,4(2E12.4,4X)))
21     FORMAT (4HODFG,I3,16H RANGE EXCEEDED.)
       END
```

```
      SUBROUTINE TRLG (XT,W,ZT)
C-----
C-----VARIABLE TRANSPORT LAG GENERATOR -  FORTRAN IV
C-----
C-----USES UP TO 300 POINT APPROXIMATION FOR
C-----UP TO 6 VARIABLES.  USES INVENTORY CALC.
C-----
C-----INPUTS FOR EACH LAG (TOTAL = NLAGS)
C-----  1. INPUT FUNCTION XT(I)
C-----  2. MASS FLOWRATE W(I)
C-----  3. INITIAL VALUE OF LAG TIME TI(I)
C-----  4. MINIMUM EXPECTED VALUE OF MASS FLOW WMIN(I)
C-----
C-----OUTPUTS ARE LAGGED FUNCTIONS ZT(I)
C-----
      DIMENSION XT(6), W(6), TI(6), WMIN(6), ZT(6), XS(300,6), PS(300,6)
     1, KT(6), JT(6), XJMP(6), JMP(6), NJMP(6)
C-----
      COMMON NI,T
C-----NI = 1-ST CALL FLAG  (= 0 ON 1-ST CALL)
C-----T = COMPUTATION TIME INTERVAL
C-----
      IF (NI) 5,1,5
C-----FIRST CALL COMP.
1     READ (5,27) NLAGS,TI,WMIN
      WRITE (6,28) TI,WMIN
      DO 4 I=1,NLAGS
      XJMP(I)=1.0
      XS(1,I)=XT(I)
      PS(1,I)=W(I)*TI(I)
      XNSP=PS(1,I)/(WMIN(I)*T)
      DO 2 M=1,10
      P1=XJMP(I)*XNSP
      IF (300.0-P1) 2,3,3
2     XJMP(I)=XJMP(I)+1.0
C-----
3     JMP(I)=IFIX(XJMP(I))
      KT(I)=2
      JT(I)=1
4     NJMP(I)=1
      NV=-1
C-----
C-----CALCS. MADE EACH TIME
5     NV=NV+1
C-----** NOTE - IF A RESTART FEATURE IS USED (WHERE THE INITIAL TIME
C-----STEP CALCULATION IS REPEATED), THE FLAG NV AND STATEMENT 33 WILL
```

```
C-----OMIT THE TRLG CALC.  THIS 1-ST CALL OMISSION MAY BE DELETED BY
C-----REMOVING STATEMENT 33.
      IF (NV) 6,26,6
6     DO 25 I=1,NLAGS
      IF (NJMP(I)-JMP(I)) 7,8,8
7     NJMP(I)=NJMP(I)+1
      GO TO 25
8     NJMP(I)=1
      K=KT(I)
      J=JT(I)
      XS(K,I)=XT(I)
      PS(K,I)=XJMP(I)*W(I)*T
C-----J=NO. OF ELEMENT AT EXIT.  K=NO. AT ENTRANCE
      IF (PS(J,I)-PS(K,I)) 12,9,21
9     ZT(I)=XS(J,I)
      IF (J-300) 11,10,10
10    JT(I)=1
      GO TO 22
11    JT(I)=J+1
      GO TO 22
C-----
12    COLLT=XS(J,I)
      COLLP=PS(J,I)
      DO 16 M=1,300
      IF (J-300) 14,13,13
13    J=0
14    J=J+1
      PQ=COLLP+PS(J,I)
C-----
      IF (PQ-PS(K,I)) 15,17,20
15    COLLT=(COLLT*COLLP+XS(J,I)*PS(J,I))/PQ
C-----
16    COLLP=COLLP+PS(J,I)
C-----
17    ZT(I)=(COLLT*COLLP+XS(J,I)*PS(J,I))/PQ
C-----
      IF (J-300) 19,18,18
18    JT(I)=1
      GO TO 22
19    JT(I)=J+1
      GO TO 22
C-----
20    PS(J,I)=PQ-PS(K,I)
      ZT(I)=(COLLT*COLLP+XS(J,I)*PS(J,I))/(COLLP+PS(J,I))
      JT(I)=J
      GO TO 22
C-----
21    ZT(I)=XS(J,I)
      PS(J,I)=PS(J,I)-PS(K,I)
C-----
22    IF (K-300) 24,23,23
23    KT(I)=1
      GO TO 25
24    KT(I)=K+1
25    CONTINUE
C-----
```

```
26    RETURN
C-----
27    FORMAT (I2/(6E10.3))
28    FORMAT (26HOTRLG INPUTS - TI AND WMIN/(1H0,6E18.5))
      END
```

```
      SUBROUTINE FTLAG (XT,ZT,NLAGS,TI)
C-----FIXED TRANSPORT LAG SUBROUTINE
      DIMENSION A(60,60), C(60,60), HP(60,60), QPT(60,60), X(60), Y(60),
     1 Z(60), XIC(60), TQP(60)
      COMMON C,HP,A,QPT,X,Z,Y,ITMAX,KK,LL,MM,JJFLAG,XIC,NI,TIME,TMAX,TZE
     1RO,NE,TQP,T,I1Z,ICONTR,PLTINC,MATYES,ICSS,JFLAG,PLT
      DIMENSION XT(15), ZT(15), XS(15,200), TI(15), KT(15), XFR(15), OMX
     1(15), NSP(15)
      IF (NI.NE.0) GO TO 3
C-----FIRST CALL CALCS.
      DO 2 I=1,NLAGS
      KT(I)=3
      XSAMP=TI(I)/T
      NSAMP=IFIX(XSAMP)
      XSAMI=FLOAT(NSAMP)
      XFR(I)=XSAMP-XSAMI
      OMX(I)=1.0-XFR(I)
      NSP(I)=NSAMP+1
      IF (NSP(I).GT.200) WRITE (6,11) I,NSP(I)
C-----SET ALL INITIAL VALUES TO ZERO
      NPTS=NSP(I)
      DO 1 J=1,NPTS
1     XS(I,J)=0.0
2     CONTINUE
3     DO 10 I=1,NLAGS
      K=KT(I)
      NS=NSP(I)
      IF (NS-K) 6,4,4
4     ZT(I)=XS(I,K-1)*OMX(I)+XS(I,K-2)*XFR(I)
5     XS(I,K-2)=XT(I)
      K=K+1
      GO TO 10
6     IF (NS-K+2) 9,8,7
7     ZT(I)=XS(I,NS)*OMX(I)+XS(I,NS-1)*XFR(I)
      GO TO 5
8     ZT(I)=XS(I,1)*OMX(I)+XS(I,NS)*XFR(I)
      GO TO 5
9     K=3
      GO TO 4
10    KT(I)=K
      RETURN
C-----
11    FORMAT (4HOLAG,I3,6H NEEDS,I5,9H SAMPLES.)
      END
      SUBROUTINE DISTRB
C-----
C-----DISTRB SUBROUTINE FOR BERNIE-S NONLINEAR WORMS
C-----
      DIMENSION A(60,60), C(60,60), HP(60,60), QPT(60,60), X(60), Y(60),
     1 Z(60), XIC(60), TQP(60)
      COMMON C,HP,A,QPT,X,Z,Y,ITMAX,KK,LL,MM,JJFLAG,XIC,NI,TIME,TMAX,TZE
     1RO,NE,TQP,T,I1Z,ICONTR,PLTINC,MATYES,ICSS,JFLAG,PLT
C-----
C-----XIC DATA READ IN NORMALLY
      IF (NI.EQ.0.AND.ICONTR.EQ.0) GO TO 2
      IF (NI.NE.0) GO TO 4
C-----A MATRIX TO BE RECALCULATED
      NI=1
      ICONTR=0
      TMAX=TMAXS
      Z(1)=Z1S
      Z(2)=Z2S
      Z(3)=0.0
      Z(4)=0.0
      TAU21=T21P*X(2)
      TAU32=T32P*X(3)
      TAU43=T43P*X(4)
      DO 1 I=2,4
1     XIC(I)=X(I)
      GO TO 3
C-----INITIAL CALCS. - MATRIX ELEMENTS
C-----READ IN PARAMETERS RHO,TAU,GMU, AND GLAM. ALSO Z
2     READ (5,8) RHO01,RHO02,RHO03,RHO04,RHO05
      READ (5,8) TAU21,TAU32,TAU43
      READ (5,8) GMU51,GMU52,GMU53,GMU54
      READ (5,8) GLAM01
      READ (5,8) Z(1),Z(2)
C-----CALC. FIXED MATRIX ELEMENTS
      A(4,4)=-(RHO04+GMU54)
      A(5,1)=GMU51
      A(5,2)=GMU52
      A(5,3)=GMU53
      A(5,4)=GMU54
      A(5,5)=-RHO05
C-----X(2), 3, AND 4 AND Z1,2 AND TMAX SAVED
      X2S=XIC(2)
      X3S=XIC(3)
      X4S=XIC(4)
      Z1S=Z(1)
      Z2S=Z(2)
      TMAXS=TMAX
C-----RECALC VARIABLE A-S
3     A(1,1)=-(RHO01+TAU21+GMU51+GLAM01)
      A(2,1)=TAU21
      A(2,2)=-(RHO02+TAU32+GMU52)
      A(3,2)=TAU32
      A(3,3)=-(RHO03+TAU43+GMU53)
      A(4,3)=TAU43
C-----CALC. TAU-PRIMES
      T21P=TAU21/XIC(2)
      T32P=TAU32/XIC(3)
      T43P=TAU43/XIC(4)
      GO TO 7
C-----CALC. FUDGED Z-S
4     Z(1)=Z1S-(T21P*X(2)-TAU21)*X(1)
      Z(2)=Z2S+(T21P*X(2)-A(2,1))*X(1)-(T32P*X(3)-TAU32)*X(2)
      Z(3)=(T32P*X(3)-A(3,2))*X(2)-(T43P*X(4)-TAU43)*X(3)
      Z(4)=(T43P*X(4)-A(4,3))*X(3)
C-----RECALC. MATEXP WHEN ABS(X/XO).GT.XLIM
      XLIM=1.1
      DO 5 I=2,4
      XMAT=ABS(X(I)/XIC(I))-XLIM
      IF (XMAT.GE.0.0) GO TO 6
5     CONTINUE
      GO TO 7
C-----RESTART W/ NEW MATEXP-S  IF X OUT OF LINEAR RANGE
6     ICONTR=1
      WRITE (6,9) I,XMAT
      TMAX=TIME
      TZERO=TIME+T
7     RETURN
C-----
8     FORMAT (8F10.0)
9     FORMAT (2HOX,I2,*OUTOFLINEARRANGE-XMAT=*,F6.5)
      END
```

```
      SUBROUTINE OUTPUT
      DIMENSION A(60,60), C(60,60), HP(60,60), QPT(60,60), X(60), Y(60),
     1 Z(60), XIC(60), TQP(60)
      COMMON C,HP,A,QPT,X,Z,Y,ITMAX,KK,LL,MM,JJFLAG,XIC,NI,TIME,TMAX,TZE
     1RO,NE,TQP,T,I1Z,ICONTR,PLTINC,MATYES,ICSS,JFLAG,PLT
      IF (NI-1) 1,7,8
1     NC=10
      DO 2 NCM=1,51,10
      WRITE (6,12) LL,((A(I,J),J=NCM,NC),I=1,NE)
      IF (NE-NC) 3,3,2
2     NC=NC+10
3     NC=10
      DO 4 NCM=1,51,10
      WRITE (6,13) ((C(I,J),J=NCM,NC),I=1,NE)
      IF (NE-NC) 5,5,4
4     NC=NC+10
5     NC=10
      DO 6 NCM=1,51,10
      WRITE (6,14) ((HP(I,J),J=NCM,NC),I=1,NE)
      IF (NE-NC) 7,7,6
6     NC=NC+10
7     WRITE (6,15)
      NI=2
8     WRITE (6,16) TIME
      NF=1
```

```
9       NL=MINO(NF+4,NE)
        WRITE (6,17) (X(I),I=NF,NL)
        IF (JFLAG.NE.5) GO TO 10
        WRITE (6,18) (Z(I),I=NF,NL)
10      IF (NL.EQ.NE) GO TO 11
        NF=NF+5
        WRITE (6,19)
        GO TO 9
11      RETURN
C-----
12      FORMAT (2H0A,I2/(1H ,10E11.3))
13      FORMAT (2H0C/(1H ,10E11.3))
14      FORMAT (3H0HP/(1H ,10E11.3))
15      FORMAT (1H-,*TIME*,6X,*0*,23X,*SOLUTIONVECTOR*,21X,*0*,22X,*DISTUR
       1BANCEVECTOR*/)
16      FORMAT (1H ,E10.3,*0*)
17      FORMAT (1H+,15X,5E11.3,*0*)
18      FORMAT (1H+,75X,5E11.3)
19      FORMAT (12X,*0*)
        END
```

SAMPLE DATA DECK FOR MATEXP

```
05          .000001   0.0        0.01        3.0         0.01         4 1 5 320 0 0   1
    1 3421.26        2 213.44        3 62.06         4 8.87          5 24.38

2.23        8.86      5.10       1.466       188.6
0.84        1.79      0.339
1.01        5.13      0.74       0.676
2.0
20810.      486.
```

```
MATEXP CASE  1
NO. OF EQUATIONS  5
SPECIFIED PRECISION      .00000100
TIME INTERVAL            .01000000
PLOT INCREMENT           .01000000

CONTROL FLAGS -
     MATYES   4
     ICSS     1
     JFLAG    5
     ICONTR   0

MAX. TERMS IN EXPONENTIAL APPROX.    32
SINGLE Z ROW    0
MAX. ALLOWABLE A*DT     1.000
MAX. ALLOWABLE QPT ELEMENT     100.000

MAX.COEFF. MATRIX ELEMENT = A( 5, 5) =     -1.8860E+02
MAX. A*DT =    .94300000  WITH DELTA T =       .00500000

MINIMUM NON-ZERO ELEMENT = A( 4, 3) =      3.3900E-01
RATIO AMAX/AMIN =      5.5634E+02

NO. OF TERMS IN SERIES APPROX. OF MATEXP = 10

TOTAL NO. OF T HALVINGS =  1

A-O
 -6.080E+00  0.          0.          0.          0.          0.          0.          0.          0.          0.
  8.400E-01 -1.578E+01   0.          0.          0.          0.          0.          0.          0.          0.
  0.          1.790E+00 -6.179E+00  0.          0.          0.          0.          0.          0.          0.
  0.          0.          3.390E-01 -2.142E+00  0.          0.          0.          0.          0.          0.
  1.010E+00   5.130E+00  7.400E-01  6.760E-01 -1.886E+02  0.          0.          0.          0.          0.
C
  9.410E-01  0.          0.          0.          0.          0.          0.          0.          0.          0.
  7.533E-03  8.540E-01  0.          0.          0.          0.          0.          0.          0.          0.
  6.849E-05  1.604E-02  9.401E-01  0.          0.          0.          0.          0.          0.          0.
  7.880E-08  2.801E-05  3.252E-03  9.788E-01  0.          0.          0.          0.          0.          0.
  4.484E-03  2.088E-02  3.205E-03  2.999E-03  1.517E-01  0.          0.          0.          0.          0.
HP
  9.702E-03  0.          0.          0.          0.          0.          0.          0.          0.          0.
  3.907E-05  9.251E-03  0.          0.          0.          0.          0.          0.          0.          0.
  2.337E-07  8.323E-05  9.697E-03  0.          0.          0.          0.          0.          0.          0.
  2.000E-10  9.526E-08  1.649E-05  9.894E-03  0.          0.          0.          0.          0.          0.
  2.925E-05  1.412E-04  2.112E-05  1.956E-05  4.498E-03  0.          0.          0.          0.          0.

TIME                                SOLUTIONVECTOR                                                    DISTURBANCEVECTOR

0.            3.421E+03  2.134E+02  6.206E+01  8.870E+00  2.438E+01     2.081E+04  4.860E+02  0.          0.          0.
1.000E-02     3.421E+03  2.134E+02  6.205E+01  8.890E+00  2.440E+01     2.081E+04  4.860E+02  0.          0.          0.
2.000E-02     3.421E+03  2.133E+02  6.203E+01  8.910E+00  2.440E+01     2.081E+04  4.851E+02 -1.362E-01  4.778E-02  0.
3.000E-02     3.422E+03  2.132E+02  6.201E+01  8.930E+00  2.440E+01     2.081E+04  4.842E+02 -2.830E-01  9.551E-02  0.
4.000E-02     3.422E+03  2.132E+02  6.199E+01  8.951E+00  2.440E+01     2.081E+04  4.834E+02 -4.401E-01  1.432E-01  0.
5.000E-02     3.422E+03  2.131E+02  6.197E+01  8.971E+00  2.440E+01     2.081E+04  4.826E+02 -6.072E-01  1.908E-01  0.
6.000E-02     3.422E+03  2.130E+02  6.195E+01  8.991E+00  2.440E+01     2.081E+04  4.819E+02 -7.837E-01  2.383E-01  0.
7.000E-02     3.422E+03  2.130E+02  6.192E+01  9.011E+00  2.439E+01     2.082E+04  4.812E+02 -9.694E-01  2.857E-01  0.
8.000E-02     3.422E+03  2.129E+02  6.190E+01  9.031E+00  2.439E+01     2.082E+04  4.806E+02 -1.164E+00  3.330E-01  0.
9.000E-02     3.422E+03  2.129E+02  6.187E+01  9.051E+00  2.439E+01     2.082E+04  4.800E+02 -1.367E+00  3.802E-01  0.
1.000E-01     3.422E+03  2.128E+02  6.184E+01  9.071E+00  2.439E+01     2.082E+04  4.795E+02 -1.577E+00  4.272E-01  0.
1.100E-01     3.422E+03  2.128E+02  6.182E+01  9.090E+00  2.439E+01     2.082E+04  4.791E+02 -1.796E+00  4.740E-01  0.
1.200E-01     3.422E+03  2.127E+02  6.179E+01  9.110E+00  2.439E+01     2.082E+04  4.787E+02 -2.021E+00  5.206E-01  0.
1.300E-01     3.423E+03  2.127E+02  6.175E+01  9.130E+00  2.439E+01     2.082E+04  4.784E+02 -2.254E+00  5.671E-01  0.
```

```
1.400E-01     3.423E+03  2.127E+02  6.172E+01  9.150E+00  2.439E+01     2.082E+04  4.781E+02 -2.492E+00  6.133E-01  0.
1.500E-01     3.423E+03  2.126E+02  6.169E+01  9.169E+00  2.439E+01     2.082E+04  4.778E+02 -2.737E+00  6.594E-01  0.
1.600E-01     3.423E+03  2.126E+02  6.165E+01  9.189E+00  2.439E+01     2.082E+04  4.776E+02 -2.988E+00  7.052E-01  0.
1.700E-01     3.423E+03  2.126E+02  6.162E+01  9.208E+00  2.439E+01     2.082E+04  4.775E+02 -3.243E+00  7.507E-01  0.
1.800E-01     3.423E+03  2.126E+02  6.158E+01  9.227E+00  2.439E+01     2.082E+04  4.774E+02 -3.504E+00  7.960E-01  0.
1.900E-01     3.423E+03  2.126E+02  6.155E+01  9.247E+00  2.439E+01     2.082E+04  4.774E+02 -3.769E+00  8.411E-01  0.
2.000E-01     3.423E+03  2.126E+02  6.151E+01  9.266E+00  2.439E+01     2.082E+04  4.774E+02 -4.039E+00  8.859E-01  0.
2.100E-01     3.423E+03  2.125E+02  6.147E+01  9.285E+00  2.439E+01     2.082E+04  4.774E+02 -4.312E+00  9.304E-01  0.
2.200E-01     3.423E+03  2.125E+02  6.143E+01  9.304E+00  2.439E+01     2.082E+04  4.775E+02 -4.589E+00  9.746E-01  0.
2.300E-01     3.423E+03  2.125E+02  6.139E+01  9.323E+00  2.439E+01     2.082E+04  4.776E+02 -4.869E+00  1.019E+00  0.
2.400E-01     3.424E+03  2.125E+02  6.135E+01  9.342E+00  2.439E+01     2.082E+04  4.778E+02 -5.151E+00  1.062E+00  0.
2.500E-01     3.424E+03  2.125E+02  6.131E+01  9.360E+00  2.439E+01     2.082E+04  4.781E+02 -5.437E+00  1.106E+00  0.
2.600E-01     3.424E+03  2.125E+02  6.127E+01  9.379E+00  2.439E+01     2.082E+04  4.783E+02 -5.724E+00  1.149E+00  0.
2.700E-01     3.424E+03  2.125E+02  6.123E+01  9.397E+00  2.439E+01     2.082E+04  4.786E+02 -6.014E+00  1.191E+00  0.
2.800E-01     3.424E+03  2.126E+02  6.119E+01  9.416E+00  2.439E+01     2.082E+04  4.790E+02 -6.305E+00  1.234E+00  0.
2.900E-01     3.424E+03  2.126E+02  6.115E+01  9.434E+00  2.439E+01     2.082E+04  4.793E+02 -6.597E+00  1.276E+00  0.
3.000E-01     3.424E+03  2.126E+02  6.111E+01  9.452E+00  2.439E+01     2.082E+04  4.798E+02 -6.890E+00  1.318E+00  0.
3.100E-01     3.424E+03  2.126E+02  6.107E+01  9.470E+00  2.439E+01     2.082E+04  4.802E+02 -7.184E+00  1.359E+00  0.
3.200E-01     3.424E+03  2.126E+02  6.103E+01  9.488E+00  2.439E+01     2.082E+04  4.807E+02 -7.479E+00  1.400E+00  0.
3.300E-01     3.424E+03  2.126E+02  6.099E+01  9.506E+00  2.439E+01     2.082E+04  4.812E+02 -7.773E+00  1.441E+00  0.
3.400E-01     3.424E+03  2.127E+02  6.094E+01  9.523E+00  2.439E+01     2.082E+04  4.817E+02 -8.068E+00  1.482E+00  0.
3.500E-01     3.424E+03  2.127E+02  6.090E+01  9.541E+00  2.439E+01     2.082E+04  4.823E+02 -8.363E+00  1.522E+00  0.
3.600E-01     3.424E+03  2.127E+02  6.086E+01  9.558E+00  2.440E+01     2.082E+04  4.829E+02 -8.657E+00  1.562E+00  0.
3.700E-01     3.424E+03  2.127E+02  6.082E+01  9.576E+00  2.440E+01     2.082E+04  4.835E+02 -8.950E+00  1.601E+00  0.
3.800E-01     3.424E+03  2.128E+02  6.078E+01  9.593E+00  2.440E+01     2.082E+04  4.842E+02 -9.242E+00  1.641E+00  0.
3.900E-01     3.424E+03  2.128E+02  6.074E+01  9.610E+00  2.440E+01     2.082E+04  4.849E+02 -9.533E+00  1.680E+00  0.
4.000E-01     3.424E+03  2.128E+02  6.070E+01  9.627E+00  2.440E+01     2.082E+04  4.856E+02 -9.823E+00  1.718E+00  0.
4.100E-01     3.424E+03  2.129E+02  6.066E+01  9.644E+00  2.440E+01     2.082E+04  4.863E+02 -1.011E+01  1.756E+00  0.
4.200E-01     3.424E+03  2.129E+02  6.062E+01  9.661E+00  2.440E+01     2.082E+04  4.870E+02 -1.040E+01  1.794E+00  0.
4.300E-01     3.424E+03  2.130E+02  6.058E+01  9.677E+00  2.440E+01     2.082E+04  4.878E+02 -1.068E+01  1.832E+00  0.
4.400E-01     3.424E+03  2.130E+02  6.054E+01  9.694E+00  2.440E+01     2.082E+04  4.885E+02 -1.097E+01  1.869E+00  0.
4.500E-01     3.424E+03  2.130E+02  6.050E+01  9.710E+00  2.440E+01     2.082E+04  4.893E+02 -1.125E+01  1.906E+00  0.
4.600E-01     3.424E+03  2.131E+02  6.046E+01  9.727E+00  2.440E+01     2.082E+04  4.901E+02 -1.153E+01  1.943E+00  0.
4.700E-01     3.424E+03  2.131E+02  6.042E+01  9.743E+00  2.440E+01     2.081E+04  4.909E+02 -1.180E+01  1.979E+00  0.
4.800E-01     3.424E+03  2.132E+02  6.039E+01  9.759E+00  2.441E+01     2.081E+04  4.918E+02 -1.207E+01  2.015E+00  0.

X 4OUTOFLINEARRANGE-XMAT=.00020
 4.900E-01    3.424E+03  2.132E+02  6.035E+01  9.775E+00  2.441E+01     2.081E+04  4.926E+02 -1.235E+01  2.051E+00  0.
```

```
MATEXP CASE  2
NO. OF EQUATIONS  5
SPECIFIED PRECISION      .00000100
TIME INTERVAL            .01000000
PLOT INCREMENT           .00999900

CONTROL FLAGS -
     MATYES    4
     ICSS      1
     JFLAG     5
     ICONTR    0

MAX. TERMS IN EXPONENTIAL APPROX.    32
SINGLE Z ROW    0
MAX. ALLOWABLE A*DT     1.000
MAX. ALLOWABLE OPT ELEMENT    100.000

MAX.COEFF. MATRIX ELEMENT = A( 5, 5) =     -1.8860E+02
MAX. A*DT =     .94300000  WITH DELTA T =       .00500000

MINIMUM NON-ZERO ELEMENT = A( 4, 3) =      3.7358E-01
RATIO AMAX/AMIN =      5.0485E+02

NO. OF TERMS IN SERIES APPROX. OF MATEXP = 10

TOTAL NO. OF T HALVINGS =  1

TIME                              SOLUTIONVECTOR                                                         DISTURBANCEVECTOR

 4.900E-01       3.424E+03  2.132E+02  6.035E+01  9.775E+00  2.441E+01       2.081E+04  4.860E+02  0.          0.          0.
 5.000E-01       3.424E+03  2.133E+02  6.031E+01  9.791E+00  2.441E+01       2.081E+04  4.860E+02  0.          0.          0.
 5.100E-01       3.424E+03  2.133E+02  6.027E+01  9.806E+00  2.441E+01       2.081E+04  4.869E+02 -2.640E-01  3.649E-02  0.
 5.200E-01       3.424E+03  2.134E+02  6.024E+01  9.822E+00  2.441E+01       2.081E+04  4.877E+02 -5.250E-01  7.269E-02  0.
 5.300E-01       3.424E+03  2.134E+02  6.020E+01  9.837E+00  2.441E+01       2.081E+04  4.886E+02 -7.828E-01  1.086E-01  0.
 5.400E-01       3.424E+03  2.135E+02  6.017E+01  9.853E+00  2.441E+01       2.081E+04  4.895E+02 -1.037E+00  1.442E-01  0.
 5.500E-01       3.424E+03  2.135E+02  6.013E+01  9.868E+00  2.441E+01       2.081E+04  4.903E+02 -1.289E+00  1.795E-01  0.
 5.600E-01       3.424E+03  2.136E+02  6.010E+01  9.883E+00  2.441E+01       2.081E+04  4.912E+02 -1.537E+00  2.146E-01  0.
 5.700E-01       3.423E+03  2.136E+02  6.007E+01  9.898E+00  2.441E+01       2.081E+04  4.921E+02 -1.782E+00  2.493E-01  0.
 5.800E-01       3.423E+03  2.137E+02  6.003E+01  9.913E+00  2.442E+01       2.080E+04  4.930E+02 -2.023E+00  2.838E-01  0.
 5.900E-01       3.423E+03  2.137E+02  6.000E+01  9.928E+00  2.442E+01       2.080E+04  4.939E+02 -2.260E+00  3.181E-01  0.
 6.000E-01       3.423E+03  2.138E+02  5.997E+01  9.943E+00  2.442E+01       2.080E+04  4.948E+02 -2.495E+00  3.520E-01  0.
 6.100E-01       3.423E+03  2.138E+02  5.994E+01  9.958E+00  2.442E+01       2.080E+04  4.957E+02 -2.725E+00  3.857E-01  0.
 6.200E-01       3.423E+03  2.139E+02  5.991E+01  9.972E+00  2.442E+01       2.080E+04  4.966E+02 -2.952E+00  4.192E-01  0.
 6.300E-01       3.423E+03  2.139E+02  5.988E+01  9.987E+00  2.442E+01       2.080E+04  4.975E+02 -3.176E+00  4.524E-01  0.
 6.400E-01       3.423E+03  2.140E+02  5.985E+01  1.000E+01  2.442E+01       2.080E+04  4.984E+02 -3.396E+00  4.853E-01  0.
 6.500E-01       3.423E+03  2.140E+02  5.982E+01  1.002E+01  2.442E+01       2.080E+04  4.992E+02 -3.612E+00  5.181E-01  0.
 6.600E-01       3.423E+03  2.141E+02  5.979E+01  1.003E+01  2.442E+01       2.080E+04  5.001E+02 -3.824E+00  5.505E-01  0.
 6.700E-01       3.423E+03  2.141E+02  5.976E+01  1.004E+01  2.442E+01       2.080E+04  5.010E+02 -4.033E+00  5.828E-01  0.
 6.800E-01       3.423E+03  2.142E+02  5.973E+01  1.006E+01  2.442E+01       2.080E+04  5.019E+02 -4.239E+00  6.148E-01  0.
 6.900E-01       3.423E+03  2.142E+02  5.971E+01  1.007E+01  2.443E+01       2.080E+04  5.027E+02 -4.440E+00  6.467E-01  0.
 7.000E-01       3.422E+03  2.143E+02  5.968E+01  1.009E+01  2.443E+01       2.080E+04  5.036E+02 -4.639E+00  6.783E-01  0.
 7.100E-01       3.422E+03  2.143E+02  5.966E+01  1.010E+01  2.443E+01       2.080E+04  5.044E+02 -4.833E+00  7.097E-01  0.
 7.200E-01       3.422E+03  2.144E+02  5.963E+01  1.011E+01  2.443E+01       2.080E+04  5.053E+02 -5.024E+00  7.409E-01  0.
 7.300E-01       3.422E+03  2.144E+02  5.961E+01  1.013E+01  2.443E+01       2.079E+04  5.061E+02 -5.211E+00  7.718E-01  0.
 7.400E-01       3.422E+03  2.145E+02  5.958E+01  1.014E+01  2.443E+01       2.079E+04  5.069E+02 -5.395E+00  8.026E-01  0.
 7.500E-01       3.422E+03  2.145E+02  5.956E+01  1.015E+01  2.443E+01       2.079E+04  5.078E+02 -5.576E+00  8.333E-01  0.
 7.600E-01       3.422E+03  2.146E+02  5.954E+01  1.017E+01  2.443E+01       2.079E+04  5.086E+02 -5.753E+00  8.637E-01  0.
 7.700E-01       3.422E+03  2.146E+02  5.951E+01  1.018E+01  2.443E+01       2.079E+04  5.094E+02 -5.926E+00  8.939E-01  0.
 7.800E-01       3.422E+03  2.147E+02  5.949E+01  1.019E+01  2.443E+01       2.079E+04  5.102E+02 -6.097E+00  9.240E-01  0.
 7.900E-01       3.422E+03  2.147E+02  5.947E+01  1.021E+01  2.443E+01       2.079E+04  5.109E+02 -6.264E+00  9.539E-01  0.
 8.000E-01       3.421E+03  2.148E+02  5.945E+01  1.022E+01  2.443E+01       2.079E+04  5.117E+02 -6.427E+00  9.837E-01  0.
 8.100E-01       3.421E+03  2.148E+02  5.943E+01  1.023E+01  2.443E+01       2.079E+04  5.125E+02 -6.588E+00  1.013E+00  0.
 8.200E-01       3.421E+03  2.149E+02  5.941E+01  1.025E+01  2.444E+01       2.079E+04  5.132E+02 -6.745E+00  1.043E+00  0.
 8.300E-01       3.421E+03  2.149E+02  5.939E+01  1.026E+01  2.444E+01       2.079E+04  5.140E+02 -6.899E+00  1.072E+00  0.
```

```
8.400E-01    3.421E+03  2.149E+02  5.937E+01  1.027E+01  2.444E+01    2.079E+04  5.147E+02 -7.050E+00  1.101E+00  0.
8.500E-01    3.421E+03  2.150E+02  5.935E+01  1.029E+01  2.444E+01    2.079E+04  5.154E+02 -7.199E+00  1.130E+00  0.
8.600E-01    3.421E+03  2.150E+02  5.933E+01  1.030E+01  2.444E+01    2.079E+04  5.161E+02 -7.344E+00  1.159E+00  0.
8.700E-01    3.421E+03  2.151E+02  5.932E+01  1.031E+01  2.444E+01    2.079E+04  5.168E+02 -7.486E+00  1.188E+00  0.
8.800E-01    3.421E+03  2.151E+02  5.930E+01  1.032E+01  2.444E+01    2.078E+04  5.175E+02 -7.626E+00  1.216E+00  0.
8.900E-01    3.421E+03  2.152E+02  5.928E+01  1.034E+01  2.444E+01    2.078E+04  5.182E+02 -7.762E+00  1.245E+00  0.
9.000E-01    3.420E+03  2.152E+02  5.926E+01  1.035E+01  2.444E+01    2.078E+04  5.189E+02 -7.896E+00  1.273E+00  0.
9.100E-01    3.420E+03  2.152E+02  5.925E+01  1.036E+01  2.444E+01    2.078E+04  5.195E+02 -8.028E+00  1.301E+00  0.
9.200E-01    3.420F+03  2.153E+02  5.923E+01  1.037E+01  2.444E+01    2.078E+04  5.201E+02 -8.157E+00  1.329E+00  0.
9.300E-01    3.420E+03  2.153E+02  5.922E+01  1.039E+01  2.444E+01    2.078E+^4  5.208E+02 -8.283E+00  1.357E+00  0.
9.400E-01    3.420E+03  2.154E+02  5.920E+01  1.040E+01  2.444E+01    2.078E+04  5.214E+02 -8.407E+00  1.385E+00  0.
9.500E-01    3.420E+03  2.154E+02  5.919E+01  1.041E+01  2.444E+01    2.078E+04  5.220E+02 -8.528E+00  1.413E+00  0.
9.600E-01    3.420E+03  2.154E+02  5.917E+01  1.042E+01  2.444E+01    2.078E+04  5.226E+02 -8.648E+00  1.441E+00  0.
9.700E-01    3.420E+03  2.155E+02  5.916E+01  1.044E+01  2.444E+01    2.078E+04  5.232E+02 -8.765E+00  1.468E+00  0.
9.800E-01    3.420E+03  2.155E+02  5.915E+01  1.045E+01  2.444E+01    2.078E+04  5.237E+02 -8.879E+00  1.496E+00  0.
9.900E-01    3.420E+03  2.155E+02  5.913E+01  1.046E+01  2.444E+01    2.078E+04  5.243E+02 -8.992E+00  1.523E+00  0.
1.000E+00    3.420E+03  2.156E+02  5.912E+01  1.047E+01  2.445E+01    2.078E+04  5.248E+02 -9.103E+00  1.550E+00  0.
1.010E+00    3.419F+03  2.156E+02  5.911E+01  1.049E+01  2.445E+01    2.078E+04  5.254E+02 -9.212E+00  1.578E+00  0.
1.020E+00    3.419E+03  2.156E+02  5.910E+01  1.050E+01  2.445E+01    2.078E+04  5.259E+02 -9.319E+00  1.605E+00  0.
1.030E+00    3.419E+03  2.157E+02  5.908E+01  1.051E+01  2.445E+01    2.078E+04  5.264E+02 -9.424E+00  1.632E+00  0.
1.040E+00    3.419E+03  2.157E+02  5.907E+01  1.052E+01  2.445E+01    2.078E+04  5.269E+02 -9.527E+00  1.659E+00  0.
1.050E+00    3.419E+03  2.157E+02  5.906E+01  1.053E+01  2.445E+01    2.078E+04  5.274E+02 -9.628E+00  1.686E+00  0.
1.060E+00    3.419E+03  2.158E+02  5.905E+01  1.055E+01  2.445F+01    2.078E+04  5.279E+02 -9.728E+00  1.712E+00  0.
1.070E+00    3.419E+03  2.158E+02  5.904E+01  1.056E+01  2.445E+01    2.078E+04  5.284E+02 -9.827E+00  1.739E+00  0.
1.080E+00    3.419E+03  2.158E+02  5.903E+01  1.057E+01  2.445E+01    2.078E+04  5.289E+02 -9.924E+00  1.766E+00  0.
1.090E+00    3.419E+03  2.158E+02  5.902E+01  1.058E+01  2.445E+01    2.077E+04  5.293E+02 -1.002E+01  1.792E+00  0.
1.100E+00    3.419F+03  2.159E+02  5.901E+01  1.059E+01  2.445E+01    2.077E+04  5.298E+02 -1.011E+01  1.819E+00  0.
1.110E+00    3.419F+03  2.159E+02  5.900E+01  1.060E+01  2.445E+01    2.077E+04  5.302E+02 -1.021E+01  1.845E+00  0.
1.120E+00    3.418E+03  2.159E+02  5.899E+01  1.062E+01  2.445E+01    2.077E+04  5.306E+02 -1.030E+01  1.872E+00  0.
1.130E+00    3.418E+03  2.160E+02  5.898E+01  1.063E+01  2.445E+01    2.077E+04  5.310E+02 -1.039E+01  1.898E+00  0.
1.140E+00    3.418E+03  2.160E+02  5.897E+01  1.064E+01  2.445E+01    2.077E+04  5.314E+02 -1.048E+01  1.924E+00  0.
1.150E+00    3.418E+03  2.160E+02  5.896E+01  1.065E+01  2.445E+01    2.077E+04  5.318E+02 -1.056E+01  1.950E+00  0.
1.160E+00    3.418E+03  2.160E+02  5.895E+01  1.066E+01  2.445E+01    2.077E+04  5.322E+02 -1.065E+01  1.976E+00  0.
1.170E+00    3.418E+03  2.161E+02  5.894E+01  1.068E+01  2.445E+01    2.077E+04  5.326E+02 -1.074E+01  2.002E+00  0.
1.180E+00    3.418E+03  2.161E+02  5.893E+01  1.069E+01  2.445E+01    2.077E+04  5.330E+02 -1.082E+01  2.028E+00  0.
1.190E+00    3.418E+03  2.161E+02  5.892E+01  1.070E+01  2.445E+01    2.077E+04  5.334E+02 -1.091E+01  2.054E+00  0.
1.200E+00    3.418E+03  2.161E+02  5.891E+01  1.071E+01  2.445E+01    2.077E+04  5.337E+02 -1.099E+01  2.080E+00  0.
1.210E+00    3.418E+03  2.161E+02  5.890E+01  1.072E+01  2.445E+01    2.077E+04  5.341E+02 -1.107E+01  2.106E+00  0.
1.220E+00    3.418E+03  2.162E+02  5.889E+01  1.073E+01  2.445E+01    2.077E+04  5.344E+02 -1.115E+01  2.132E+00  0.
1.230E+00    3.418F+03  2.162E+02  5.888E+01  1.074E+01  2.445E+01    2.077E+04  5.348E+02 -1.123E+01  2.158E+00  0.
1.240E+00    3.418E+03  2.162E+02  5.888E+01  1.076E+01  2.445E+01    2.077E+04  5.351E+02 -1.132E+01  2.183E+00  0.
X 40UTOFLINEARRANGE-XMAT=.00042
 1.250E+00    3.418E+03  2.162E+02  5.887E+01  1.077E+01  2.445F+01    2.077E+04  5.354E+02 -1.140F+01  2.209E+00  0.
```

```
MATEXP CASE  3
NO. OF EQUATIONS  5
SPECIFIED PRECISION     .00000100
TIME INTERVAL           .01000000
PLOT INCREMENT          .00999800

CONTROL FLAGS -
     MATYES   4
     ICSS     1
     JFLAG    5
     ICONTR   0

MAX. TERMS IN EXPONENTIAL APPROX.   32
SINGLE Z ROW   0
MAX. ALLOWABLE A*DT     1.000
MAX. ALLOWABLE OPT ELEMENT    100.000

MAX.COEFF. MATRIX ELEMENT = A( 5, 5) =     -1.8860E+02
MAX. A*DT =    .94300000  WITH DELTA T =       .00500000

MINIMUM NON-ZERO ELEMENT = A( 4, 3) =     4.1153E-01
RATIO AMAX/AMIN =     4.5829E+02

NO. OF TERMS IN SERIES APPROX. OF MATEXP = 10

TOTAL NO. OF T HALVINGS =  1


TIME                           SOLUTIONVECTOR                                                 DISTURBANCEVECTOR

 1.250E+00        3.418E+03  2.162E+02  5.887E+01  1.077E+01  2.445E+01        2.081E+04  4.860E+02  0.          0.          0.
 1.260E+00        3.418E+03  2.162E+02  5.886E+01  1.078E+01  2.445E+01        2.081E+04  4.860E+02  0.          0.          0.
 1.270E+00        3.417E+03  2.163E+02  5.885E+01  1.079E+01  2.445E+01        2.081E+04  4.863E+02 -7.825E-02  2.580E-02  0.
 1.280E+00        3.417E+03  2.163E+02  5.884E+01  1.080E+01  2.445E+01        2.081E+04  4.866E+02 -1.560E-01  5.154E-02  0.
 1.290E+00        3.417E+03  2.163E+02  5.883E+01  1.081E+01  2.445E+01        2.081E+04  4.869E+02 -2.334E-01  7.722E-02  0.
 1.300E+00        3.417E+03  2.163E+02  5.883E+01  1.082E+01  2.445E+01        2.081E+04  4.872E+02 -3.103E-01  1.029E-01  0.
 1.310E+00        3.417E+03  2.163E+02  5.882E+01  1.084E+01  2.445E+01        2.081E+04  4.875E+02 -3.869E-01  1.284E-01  0.
 1.320E+00        3.417E+03  2.164E+02  5.881E+01  1.085E+01  2.445E+01        2.081E+04  4.878E+02 -4.632E-01  1.539E-01  0.
 1.330E+00        3.417E+03  2.164E+02  5.880E+01  1.086E+01  2.445E+01        2.081E+04  4.881E+02 -5.392E-01  1.794E-01  0.
 1.340E+00        3.417E+03  2.164E+02  5.879E+01  1.087E+01  2.445E+01        2.081E+04  4.884E+02 -6.149E-01  2.048E-01  0.
 1.350E+00        3.417E+03  2.164E+02  5.878E+01  1.088E+01  2.445E+01        2.081E+04  4.886E+02 -6.904E-01  2.302E-01  0.
 1.360E+00        3.417E+03  2.164E+02  5.878E+01  1.089E+01  2.445E+01        2.081E+04  4.889E+02 -7.657E-01  2.555E-01  0.
 1.370E+00        3.417E+03  2.164E+02  5.877E+01  1.090E+01  2.446E+01        2.081E+04  4.892E+02 -8.408E-01  2.807E-01  0.
 1.380E+00        3.417E+03  2.164E+02  5.876E+01  1.092E+01  2.446E+01        2.081E+04  4.894E+02 -9.157E-01  3.059E-01  0.
 1.390E+00        3.417E+03  2.165E+02  5.875E+01  1.093E+01  2.446E+01        2.081E+04  4.897E+02 -9.905E-01  3.310E-01  0.
 1.400E+00        3.417E+03  2.165E+02  5.875E+01  1.094E+01  2.446E+01        2.081E+04  4.899E+02 -1.065E+00  3.561E-01  0.
 1.410E+00        3.417E+03  2.165E+02  5.874E+01  1.095E+01  2.446E+01        2.081E+04  4.902E+02 -1.140E+00  3.812E-01  0.
 1.420E+00        3.417E+03  2.165E+02  5.873E+01  1.096E+01  2.446E+01        2.081E+04  4.904E+02 -1.214E+00  4.061E-01  0.
 1.430E+00        3.417E+03  2.165E+02  5.872E+01  1.097E+01  2.446E+01        2.081E+04  4.907E+02 -1.289E+00  4.310E-01  0.
 1.440E+00        3.417E+03  2.165E+02  5.871E+01  1.098E+01  2.446E+01        2.081E+04  4.909E+02 -1.363E+00  4.559E-01  0.
 1.450E+00        3.417E+03  2.165E+02  5.871E+01  1.099E+01  2.446E+01        2.081E+04  4.912E+02 -1.438E+00  4.807E-01  0.
 1.460E+00        3.417E+03  2.166E+02  5.870E+01  1.100E+01  2.446E+01        2.081E+04  4.914E+02 -1.512E+00  5.055E-01  0.
 1.470E+00        3.416E+03  2.166E+02  5.869E+01  1.102E+01  2.446E+01        2.081E+04  4.916E+02 -1.587E+00  5.302E-01  0.
 1.480E+00        3.416E+03  2.166E+02  5.868E+01  1.103E+01  2.446E+01        2.081E+04  4.919E+02 -1.661E+00  5.548E-01  0.
 1.490E+00        3.416E+03  2.166E+02  5.867E+01  1.104E+01  2.446E+01        2.081E+04  4.921E+02 -1.736E+00  5.794E-01  0.
 1.500E+00        3.416E+03  2.166E+02  5.867E+01  1.105E+01  2.446E+01        2.080E+04  4.923E+02 -1.811E+00  6.039E-01  0.
 1.510E+00        3.416E+03  2.166E+02  5.866E+01  1.106E+01  2.446E+01        2.080E+04  4.926E+02 -1.885E+00  6.284E-01  0.
 1.520E+00        3.416E+03  2.166E+02  5.865E+01  1.107E+01  2.446E+01        2.080E+04  4.928E+02 -1.960E+00  6.528E-01  0.
 1.530E+00        3.416E+03  2.167E+02  5.864E+01  1.108E+01  2.446E+01        2.080E+04  4.930E+02 -2.035E+00  6.771E-01  0.
 1.540E+00        3.416E+03  2.167E+02  5.863E+01  1.109E+01  2.446E+01        2.080E+04  4.932E+02 -2.110E+00  7.014E-01  0.
 1.550E+00        3.416E+03  2.167E+02  5.863E+01  1.110E+01  2.446E+01        2.080E+04  4.935E+02 -2.185E+00  7.257E-01  0.
 1.560E+00        3.416E+03  2.167E+02  5.862E+01  1.111E+01  2.446E+01        2.080E+04  4.937E+02 -2.260E+00  7.498E-01  0.
 1.570E+00        3.416E+03  2.167E+02  5.861E+01  1.112E+01  2.446E+01        2.080E+04  4.939E+02 -2.336E+00  7.740E-01  0.
 1.580E+00        3.416E+03  2.167E+02  5.860E+01  1.113E+01  2.446E+01        2.080E+04  4.941E+02 -2.411E+00  7.980E-01  0.
 1.590E+00        3.416E+03  2.167E+02  5.859E+01  1.115E+01  2.446E+01        2.080E+04  4.943E+02 -2.487E+00  8.220E-01  0.
```

```
1.600E+00     3.416E+03  2.167E+02  5.858E+01  1.116E+01  2.446E+01     2.080E+04  4.946E+02 -2.562E+00  8.459E-01  0.
1.610E+00     3.416E+03  2.168E+02  5.858E+01  1.117E+01  2.446E+01     2.080E+04  4.948E+02 -2.638E+00  8.698E-01  0.
1.620E+00     3.416E+03  2.168E+02  5.857E+01  1.118E+01  2.446E+01     2.080E+04  4.950E+02 -2.714E+00  8.936E-01  0.
1.630E+00     3.416E+03  2.168E+02  5.856E+01  1.119E+01  2.446E+01     2.080E+04  4.952E+02 -2.790E+00  9.174E-01  0.
1.640E+00     3.416E+03  2.168E+02  5.855E+01  1.120E+01  2.446E+01     2.080E+04  4.954E+02 -2.866E+00  9.410E-01  0.
1.650E+00     3.416E+03  2.168E+02  5.854E+01  1.121E+01  2.446E+01     2.080E+04  4.956E+02 -2.942E+00  9.646E-01  0.
1.660E+00     3.416E+03  2.168E+02  5.853E+01  1.122E+01  2.446E+01     2.080E+04  4.959E+02 -3.019E+00  9.882E-01  0.
1.670E+00     3.416E+03  2.168E+02  5.853E+01  1.123E+01  2.446E+01     2.080E+04  4.961E+02 -3.095E+00  1.012E+00  0.
1.680E+00     3.416E+03  2.168E+02  5.852E+01  1.124E+01  2.446E+01     2.080E+04  4.963E+02 -3.172E+00  1.035E+00  0.
1.690E+00     3.416E+03  2.169E+02  5.851E+01  1.125E+01  2.446E+01     2.080E+0.  4.965E+02 -3.249E+00  1.058E+00  0.
1.700E+00     3.416E+03  2.169E+02  5.850E+01  1.126E+01  2.446E+01     2.080E+04  4.967E+02 -3.326E+00  1.082E+00  0.
1.710E+00     3.416E+03  2.169E+02  5.849E+01  1.127E+01  2.446E+01     2.080E+04  4.969E+02 -3.403E+00  1.105E+00  0.
1.720E+00     3.416E+03  2.169E+02  5.848E+01  1.128E+01  2.446E+01     2.080E+04  4.972E+02 -3.480E+00  1.128E+00  0.
1.730E+00     3.416E+03  2.169E+02  5.847E+01  1.129E+01  2.446E+01     2.080E+04  4.974E+02 -3.557E+00  1.151E+00  0.
1.740E+00     3.416E+03  2.169E+02  5.846E+01  1.130E+01  2.446E+01     2.080E+04  4.976E+02 -3.634E+00  1.174E+00  0.
1.750E+00     3.415E+03  2.169E+02  5.846E+01  1.131E+01  2.446E+01     2.080E+04  4.978E+02 -3.712E+00  1.197E+00  0.
1.760E+00     3.415E+03  2.169E+02  5.845E+01  1.132E+01  2.446E+01     2.080E+04  4.980E+02 -3.789E+00  1.220E+00  0.
1.770E+00     3.415E+03  2.170E+02  5.844E+01  1.133E+01  2.446E+01     2.080E+04  4.983E+02 -3.867E+00  1.243E+00  0.
1.780E+00     3.415E+03  2.170E+02  5.843E+01  1.134E+01  2.446E+01     2.080E+04  4.985E+02 -3.944E+00  1.266E+00  0.
1.790E+00     3.415E+03  2.170E+02  5.842E+01  1.135E+01  2.446E+01     2.080E+04  4.987E+02 -4.022E+00  1.288E+00  0.
1.800E+00     3.415E+03  2.170E+02  5.841E+01  1.137E+01  2.446E+01     2.080E+04  4.989E+02 -4.100E+00  1.311E+00  0.
1.810E+00     3.415E+03  2.170E+02  5.840E+01  1.138E+01  2.446E+01     2.080E+04  4.991E+02 -4.178E+00  1.333E+00  0.
1.820E+00     3.415E+03  2.170E+02  5.839E+01  1.139E+01  2.446E+01     2.080E+04  4.994E+02 -4.256E+00  1.356E+00  0.
1.830E+00     3.415E+03  2.170E+02  5.839E+01  1.140E+01  2.446E+01     2.080E+04  4.996E+02 -4.334E+00  1.378E+00  0.
1.840E+00     3.415E+03  2.170E+02  5.838E+01  1.141E+01  2.446E+01     2.080E+04  4.998E+02 -4.412E+00  1.401E+00  0.
1.850E+00     3.415E+03  2.170E+02  5.837E+01  1.142E+01  2.446E+01     2.080E+04  5.000E+02 -4.489E+00  1.423E+00  0.
1.860E+00     3.415E+03  2.171E+02  5.836E+01  1.143E+01  2.446E+01     2.080E+04  5.002E+02 -4.568E+00  1.445E+00  0.
1.870E+00     3.415E+03  2.171E+02  5.835E+01  1.144E+01  2.446E+01     2.080E+04  5.005E+02 -4.646E+00  1.467E+00  0.
1.880E+00     3.415E+03  2.171E+02  5.834E+01  1.145E+01  2.446E+01     2.080E+04  5.007E+02 -4.724E+00  1.489E+00  0.
1.890E+00     3.415E+03  2.171E+02  5.833E+01  1.146E+01  2.446E+01     2.080E+04  5.009E+02 -4.802E+00  1.511E+00  0.
1.900E+00     3.415E+03  2.171E+02  5.832E+01  1.147E+01  2.446E+01     2.080E+04  5.011E+02 -4.880E+00  1.533E+00  0.
1.910E+00     3.415E+03  2.171E+02  5.831E+01  1.148E+01  2.446E+01     2.080E+04  5.014E+02 -4.958E+00  1.555E+00  0.
1.920E+00     3.415E+03  2.171E+02  5.831E+01  1.149E+01  2.446E+01     2.080E+04  5.016E+02 -5.036E+00  1.576E+00  0.
1.930E+00     3.415E+03  2.171E+02  5.830E+01  1.149E+01  2.446E+01     2.080E+04  5.018E+02 -5.114E+00  1.598E+00  0.
1.940E+00     3.415E+03  2.172E+02  5.829E+01  1.150E+01  2.446E+01     2.080E+04  5.020E+02 -5.191E+00  1.620E+00  0.
1.950E+00     3.415E+03  2.172E+02  5.828E+01  1.151E+01  2.446E+01     2.080E+04  5.023E+02 -5.269E+00  1.641E+00  0.
1.960E+00     3.415E+03  2.172E+02  5.827E+01  1.152E+01  2.447E+01     2.080E+04  5.025E+02 -5.347E+00  1.663E+00  0.
1.970E+00     3.415E+03  2.172E+02  5.826E+01  1.153E+01  2.447E+01     2.080E+04  5.027E+02 -5.425E+00  1.684E+00  0.
1.980E+00     3.415E+03  2.172E+02  5.825E+01  1.154E+01  2.447E+01     2.080E+04  5.030E+02 -5.503E+00  1.705E+00  0.
1.990E+00     3.415E+03  2.172E+02  5.824E+01  1.155E+01  2.447E+01     2.080E+04  5.032E+02 -5.580E+00  1.726E+00  0.
2.000E+00     3.415E+03  2.172E+02  5.823E+01  1.156E+01  2.447E+01     2.080E+04  5.034E+02 -5.658E+00  1.748E+00  0.
2.010E+00     3.415E+03  2.173E+02  5.822E+01  1.157E+01  2.447E+01     2.080E+04  5.036E+02 -5.735E+00  1.769E+00  0.
2.020E+00     3.415E+03  2.173E+02  5.822E+01  1.158E+01  2.447E+01     2.080E+04  5.039E+02 -5.812E+00  1.790E+00  0.
2.030E+00     3.415E+03  2.173E+02  5.821E+01  1.159E+01  2.447E+01     2.080E+04  5.041E+02 -5.889E+00  1.810E+00  0.
2.040E+00     3.415E+03  2.173E+02  5.820E+01  1.160E+01  2.447E+01     2.080E+04  5.043E+02 -5.967E+00  1.831E+00  0.
2.050E+00     3.415E+03  2.173E+02  5.819E+01  1.161E+01  2.447E+01     2.080E+04  5.046E+02 -6.044E+00  1.852E+00  0.
2.060E+00     3.415E+03  2.173E+02  5.818E+01  1.162E+01  2.447E+01     2.080E+04  5.048E+02 -6.120E+00  1.873E+00  0.
2.070E+00     3.415E+03  2.173E+02  5.817E+01  1.163E+01  2.447E+01     2.080E+04  5.050E+02 -6.197E+00  1.893E+00  0.
2.080E+00     3.415E+03  2.173E+02  5.816E+01  1.164E+01  2.447E+01     2.080E+04  5.053E+02 -6.274E+00  1.914E+00  0.
2.090E+00     3.415E+03  2.174E+02  5.815E+01  1.165E+01  2.447E+01     2.079E+04  5.055E+02 -6.350E+00  1.934E+00  0.
2.100E+00     3.414E+03  2.174E+02  5.814E+01  1.166E+01  2.447E+01     2.079E+04  5.057E+02 -6.426E+00  1.955E+00  0.
2.110E+00     3.414E+03  2.174E+02  5.814E+01  1.167E+01  2.447E+01     2.079E+04  5.060E+02 -6.502E+00  1.975E+00  0.
2.120E+00     3.414E+03  2.174E+02  5.813E+01  1.168E+01  2.447E+01     2.079E+04  5.062E+02 -6.578E+00  1.995E+00  0.
2.130E+00     3.414E+03  2.174E+02  5.812E+01  1.168E+01  2.447E+01     2.079E+04  5.064E+02 -6.654E+00  2.015E+00  0.
2.140E+00     3.414E+03  2.174E+02  5.811E+01  1.169E+01  2.447E+01     2.079E+04  5.067E+02 -6.729E+00  2.035E+00  0.
2.150E+00     3.414E+03  2.174E+02  5.810E+01  1.170E+01  2.447E+01     2.079E+04  5.069E+02 -6.805E+00  2.055E+00  0.
2.160E+00     3.414E+03  2.174E+02  5.809E+01  1.171E+01  2.447E+01     2.079E+04  5.071E+02 -6.880E+00  2.075E+00  0.
2.170E+00     3.414E+03  2.175E+02  5.808E+01  1.172E+01  2.447E+01     2.079E+04  5.074E+02 -6.955E+00  2.095E+00  0.
2.180E+00     3.414E+03  2.175E+02  5.807E+01  1.173E+01  2.447E+01     2.079E+04  5.076E+02 -7.030E+00  2.115E+00  0.
2.190E+00     3.414E+03  2.175E+02  5.807E+01  1.174E+01  2.447E+01     2.079E+04  5.078E+02 -7.104E+00  2.134E+00  0.
2.200E+00     3.414E+03  2.175E+02  5.806E+01  1.175E+01  2.447E+01     2.079E+04  5.080E+02 -7.179E+00  2.154E+00  0.
2.210E+00     3.414E+03  2.175E+02  5.805E+01  1.176E+01  2.447E+01     2.079E+04  5.083E+02 -7.253E+00  2.174E+00  0.
2.220E+00     3.414E+03  2.175E+02  5.804E+01  1.177E+01  2.447E+01     2.079E+04  5.085E+02 -7.327E+00  2.193E+00  0.
2.230E+00     3.414E+03  2.175E+02  5.803E+01  1.177E+01  2.447E+01     2.079E+04  5.087E+02 -7.401E+00  2.212E+00  0.
2.240E+00     3.414E+03  2.176E+[illegible]  5.802E+01  1.178E+01  2.447E+01     2.079E+04  5.090E+02 -7.474E+00  2.232E+00  0.
2.250E+00     3.414E+03  2.176E+02  5.801E+01  1.179E+01  2.447E+01     2.079E+04  5.092E+02 -7.547E+00  2.251E+00  0.
```

```
 2.260E+00        3.414E+03  2.176E+02  5.801E+01  1.180E+01  2.447E+01        2.079E+04  5.094E+02 -7.620E+00  2.270E+00  0.
 2.270E+00        3.414E+03  2.176E+02  5.800E+01  1.181E+01  2.447E+01        2.079E+04  5.097E+02 -7.693E+00  2.289E+00  0.
 2.280E+00        3.414E+03  2.176E+02  5.799E+01  1.182E+01  2.447E+01        2.079E+04  5.099E+02 -7.766E+00  2.308E+00  0.
 2.290E+00        3.414E+03  2.176E+02  5.798E+01  1.183E+01  2.447E+01        2.079E+04  5.101E+02 -7.838E+00  2.327E+00  0.
 2.300E+00        3.414E+03  2.176E+02  5.797E+01  1.184E+01  2.447E+01        2.079E+04  5.104E+02 -7.910E+00  2.346E+00  0.
 2.310E+00        3.414E+03  2.176E+02  5.796E+01  1.184E+01  2.447E+01        2.079E+04  5.106E+02 -7.982E+00  2.365E+00  0.
 2.320E+00        3.414E+03  2.177E+02  5.796E+01  1.185E+01  2.447E+01        2.079E+04  5.108E+02 -8.053E+00  2.384E+00  0.

X 4OUTOFLINEARRANGE-XMAT=.00072
 2.330E+00        3.414E+03  2.177E+02  5.795E+01  1.186E+01  2.447E+01        2.079E+04  5.111E+02 -8.125E+00  2.402E+00  0.
```

```
MATEXP CASE  4
NO. OF EQUATIONS  5
SPECIFIED PRECISION     .00000100
TIME INTERVAL           .01000000
PLOT INCREMENT          .00999700

CONTROL FLAGS -
     MATYES   4
     ICSS     1
     JFLAG    5
     ICONTR   0

MAX. TERMS IN EXPONENTIAL APPROX.   32
SINGLE Z ROW   0
MAX. ALLOWABLE A*DT     1.000
MAX. ALLOWABLE OPT ELEMENT     100.000

MAX.CCEFF. MATRIX ELEMENT = A( 5, 5) =     -1.8860E+02
MAX. A*DT =     .94300000  WITH DELTA T =       .00500000

MINIMUM NON-ZERO ELEMENT = A( 4, 3) =      4.5331E-01
RATIO AMAX/AMIN =       4.1605E+02

NO. OF TERMS IN SERIES APPROX. OF MATEXP = 10

TOTAL NO. OF T HALVINGS =  1


TIME
                                SOLUTIONVECTOR                                                                    DISTURBANCEVECTOR
 2.330E+00       3.414E+03  2.177E+02  5.795E+01  1.186E+01  2.447E+01        2.081E+04  4.860E+02  0.          0.          0.
 2.340E+00       3.414E+03  2.177E+02  5.794E+01  1.187E+01  2.447E+01        2.081E+04  4.860E+02  0.          0.          0.
 2.350E+00       3.414E+03  2.177E+02  5.793E+01  1.188E+01  2.447E+01        2.081E+04  4.862E+02 -7.076E-02  1.884E-02  0.
 2.360E+00       3.414E+03  2.177E+02  5.792E+01  1.189E+01  2.447E+01        2.081E+04  4.865E+02 -1.413E-01  3.760E-02  0.
 2.370E+00       3.414E+03  2.177E+02  5.791E+01  1.189E+01  2.447E+01        2.081E+04  4.867E+02 -2.115E-01  5.629E-02  0.
 2.380E+00       3.414E+03  2.177E+02  5.791E+01  1.190E+01  2.447E+01        2.081E+04  4.869E+02 -2.815E-01  7.491E-02  0.
 2.390E+00       3.414E+03  2.178E+02  5.790E+01  1.191E+01  2.447E+01        2.081E+04  4.871E+02 -3.512E-01  9.345E-02  0.
 2.400E+00       3.414E+03  2.178E+02  5.789E+01  1.192E+01  2.447E+01        2.081E+04  4.874E+02 -4.206E-01  1.119E-01  0.
 2.410E+00       3.414E+03  2.178E+02  5.788E+01  1.193E+01  2.447E+01        2.081E+04  4.876E+02 -4.898E-01  1.303E-01  0.
 2.420E+00       3.414E+03  2.178E+02  5.787E+01  1.194E+01  2.447E+01        2.081E+04  4.878E+02 -5.587E-01  1.486E-01  0.
 2.430E+00       3.414E+03  2.178E+02  5.787E+01  1.194E+01  2.447E+01        2.081E+04  4.880E+02 -6.273E-01  1.669E-01  0.
 2.440E+00       3.414E+03  2.178E+02  5.786E+01  1.195E+01  2.447E+01        2.081E+04  4.883E+02 -6.956E-01  1.851E-01  0.
 2.450E+00       3.413E+03  2.178E+02  5.785E+01  1.196E+01  2.447E+01        2.081E+04  4.885E+02 -7.637E-01  2.032E-01  0.
 2.460E+00       3.413E+03  2.178E+02  5.784E+01  1.197E+01  2.448E+01        2.081E+04  4.887E+02 -8.315E-01  2.212E-01  0.
 2.470E+00       3.413E+03  2.179E+02  5.783E+01  1.198E+01  2.448E+01        2.081E+04  4.889E+02 -8.990E-01  2.392E-01  0.
 2.480E+00       3.413E+03  2.179E+02  5.783E+01  1.199E+01  2.448E+01        2.081E+04  4.891E+02 -9.662E-01  2.571E-01  0.
 2.490E+00       3.413E+03  2.179E+02  5.782E+01  1.199E+01  2.448E+01        2.081E+04  4.894E+02 -1.033E+00  2.749E-01  0.
 2.500E+00       3.413E+03  2.179E+02  5.781E+01  1.200E+01  2.448E+01        2.081E+04  4.896E+02 -1.100E+00  2.927E-01  0.
 2.510E+00       3.413E+03  2.179E+02  5.780E+01  1.201E+01  2.448E+01        2.081E+04  4.898E+02 -1.166E+00  3.103E-01  0.
 2.520E+00       3.413E+03  2.179E+02  5.780E+01  1.202E+01  2.448E+01        2.081E+04  4.900E+02 -1.232E+00  3.280E-01  0.
 2.530E+00       3.413E+03  2.179E+02  5.779E+01  1.203E+01  2.448E+01        2.081E+04  4.902E+02 -1.298E+00  3.455E-01  0.
 2.540E+00       3.413E+03  2.179E+02  5.778E+01  1.203E+01  2.448E+01        2.081E+04  4.905E+02 -1.364E+00  3.630E-01  0.
 2.550E+00       3.413E+03  2.180E+02  5.777E+01  1.204E+01  2.448E+01        2.081E+04  4.907E+02 -1.429E+00  3.804E-01  0.
 2.560E+00       3.413E+03  2.180E+02  5.776E+01  1.205E+01  2.448E+01        2.081E+04  4.909E+02 -1.494E+00  3.977E-01  0.
 2.570E+00       3.413E+03  2.180E+02  5.776E+01  1.206E+01  2.448E+01        2.081E+04  4.911E+02 -1.559E+00  4.150E-01  0.
 2.580E+00       3.413E+03  2.180E+02  5.775E+01  1.206E+01  2.448E+01        2.081E+04  4.913E+02 -1.623E+00  4.322E-01  0.
 2.590E+00       3.413E+03  2.180E+02  5.774E+01  1.207E+01  2.448E+01        2.081E+04  4.915E+02 -1.688E+00  4.493E-01  0.
 2.600E+00       3.413E+03  2.180E+02  5.774E+01  1.208E+01  2.448E+01        2.081E+04  4.917E+02 -1.752E+00  4.664E-01  0.
 2.610E+00       3.413E+03  2.180E+02  5.773E+01  1.209E+01  2.448E+01        2.081E+04  4.920E+02 -1.815E+00  4.834E-01  0.
 2.620E+00       3.413E+03  2.180E+02  5.772E+01  1.210E+01  2.448E+01        2.081E+04  4.922E+02 -1.879E+00  5.003E-01  0.
 2.630E+00       3.413E+03  2.181E+02  5.771E+01  1.210E+01  2.448E+01        2.081E+04  4.924E+02 -1.942E+00  5.171E-01  0.
 2.640E+00       3.413E+03  2.181E+02  5.771E+01  1.211E+01  2.448E+01        2.080E+04  4.926E+02 -2.005E+00  5.339E-01  0.
 2.650E+00       3.413E+03  2.181E+02  5.770E+01  1.212E+01  2.448E+01        2.080E+04  4.928E+02 -2.067E+00  5.507E-01  0.
 2.660E+00       3.413E+03  2.181E+02  5.769E+01  1.213E+01  2.448E+01        2.080E+04  4.930E+02 -2.129E+00  5.673E-01  0.
 2.670E+00       3.413E+03  2.181E+02  5.768E+01  1.213E+01  2.448E+01        2.080E+04  4.932E+02 -2.191E+00  5.839E-01  0.
```

```
2.680E+00      3.413E+03  2.181E+02  5.768E+01  1.214E+01  2.448E+01      2.080E+04  4.934E+02 -2.253E+00  6.004E-01  0.
2.69CE+00      3.413E+03  2.181E+02  5.767E+01  1.215E+01  2.448E+01      2.080E+04  4.936E+02 -2.315E+00  6.169E-01  0.
2.700E+00      3.413E+03  2.181E+02  5.766E+01  1.216E+01  2.448E+01      2.080E+04  4.938E+02 -2.376E+00  6.333E-01  0.
2.710E+00      3.413E+03  2.182E+02  5.766E+01  1.216E+01  2.448E+01      2.080E+C4  4.940E+02 -2.437E+00  6.496E-01  0.
2.72CE+00      3.413E+03  2.182E+02  5.765E+01  1.217E+01  2.448E+01      2.080E+04  4.942E+02 -2.498E+00  6.659E-01  0.
2.730E+00      3.413E+03  2.182E+02  5.764E+01  1.218E+01  2.448E+01      2.080E+04  4.944E+02 -2.558E+00  6.821E-01  0.
2.740E+00      3.413E+03  2.182E+02  5.763E+01  1.219E+01  2.448E+01      2.080E+04  4.946E+02 -2.618E+00  6.982E-01  0.
2.750E+00      3.413E+03  2.182E+02  5.763E+01  1.219E+01  2.448E+01      2.080E+04  4.948E+02 -2.678E+00  7.143E-01  0.
2.760E+00      3.413E+03  2.182E+02  5.762E+01  1.220E+01  2.448E+01      2.080E+04  4.950E+02 -2.738E+00  7.303E-01  0.
2.770E+00      3.413E+03  2.182E+02  5.761E+01  1.221E+01  2.448E+01      2.080E+04  4.952E+02 -2.797E+00  7.462E-01  0.
2.780E+00      3.413E+03  2.182E+02  5.761E+01  1.221E+01  2.448E+01      2.080E+04  4.954E+02 -2.857E+00  7.621E-01  0.
2.790E+00      3.413E+03  2.182E+02  5.760E+01  1.222E+01  2.448E+01      2.080E+04  4.956E+02 -2.916E+00  7.779E-01  0.
2.800E+00      3.413E+03  2.183E+02  5.759E+01  1.223E+01  2.448E+01      2.080E+04  4.958E+02 -2.974E+00  7.936E-01  0.
2.810E+00      3.412E+03  2.183E+02  5.759E+01  1.224E+01  2.448E+01      2.080E+04  4.960E+02 -3.033E+00  8.093E-01  0.
2.820E+00      3.412E+03  2.183E+02  5.758E+01  1.224E+01  2.448E+01      2.080E+04  4.962E+02 -3.091E+00  8.250E-01  0.
2.830E+00      3.412E+03  2.183E+02  5.757E+01  1.225E+01  2.448E+01      2.080E+04  4.964E+02 -3.149E+00  8.405E-01  0.
2.840E+00      3.412E+03  2.183E+02  5.757E+01  1.226E+01  2.448E+01      2.080E+04  4.966E+02 -3.206E+00  8.560E-01  0.
2.85CE+00      3.412E+03  2.183E+02  5.756E+01  1.226E+01  2.448E+01      2.080E+04  4.968E+02 -3.264E+00  8.715E-01  0.
2.860E+00      3.412E+03  2.183E+02  5.755E+01  1.227E+01  2.448E+01      2.080E+04  4.970E+02 -3.321E+00  8.868E-01  0.
2.870E+00      3.412E+03  2.183E+02  5.755E+01  1.228E+01  2.448E+01      2.080E+04  4.972E+02 -3.378E+00  9.021E-01  0.
2.880E+00      3.412E+03  2.183E+02  5.754E+01  1.228E+01  2.448E+01      2.080E+04  4.974E+02 -3.435E+00  9.174E-01  0.
2.890E+00      3.412E+03  2.184E+02  5.753E+01  1.229E+01  2.448E+01      2.080E+04  4.975E+02 -3.491E+00  9.326E-01  0.
2.900E+00      3.412E+03  2.184E+02  5.753E+01  1.230E+01  2.448E+01      2.080E+04  4.977E+02 -3.548E+00  9.477E-01  0.
2.910E+00      3.412E+03  2.184E+02  5.752E+01  1.231E+01  2.448E+01      2.080E+04  4.979E+02 -3.604E+00  9.628E-01  0.
2.920E+00      3.412E+03  2.184E+02  5.751E+01  1.231E+01  2.448E+01      2.080E+04  4.981E+02 -3.660E+00  9.778E-01  0.
2.930E+00      3.412E+03  2.184E+02  5.751E+01  1.232E+01  2.448E+01      2.080E+04  4.983E+02 -3.715E+00  9.927E-01  0.
2.940E+00      3.412E+03  2.184E+02  5.750E+01  1.233E+01  2.448E+01      2.080E+04  4.985E+02 -3.770E+00  1.008E+00  0.
2.950E+00      3.412E+03  2.184E+02  5.750E+01  1.233E+01  2.448E+01      2.080E+04  4.987E+02 -3.826E+00  1.022E+00  0.
2.960E+00      3.412E+03  2.184E+02  5.749E+01  1.234E+01  2.448E+01      2.080E+04  4.988E+02 -3.881E+00  1.037E+00  0.
2.970E+00      3.412E+03  2.184E+02  5.748E+01  1.235E+01  2.448E+01      2.080E+04  4.990E+02 -3.935E+00  1.052E+00  0.
2.980E+00      3.412E+03  2.184E+02  5.748E+01  1.235E+01  2.448E+01      2.080E+04  4.992E+02 -3.990E+00  1.067E+00  0.
2.990E+00      3.412E+03  2.185E+02  5.747E+01  1.236E+01  2.448E+01      2.080E+04  4.994E+02 -4.044E+00  1.081E+00  0.
3.000E+00      3.412E+03  2.185E+02  5.746E+01  1.237E+01  2.448E+01      2.080E+04  4.996E+02 -4.098E+00  1.096E+00  0.
3.010E+00      3.412E+03  2.185E+02  5.746E+01  1.237E+01  2.448E+01      2.080E+04  4.997E+02 -4.152E+00  1.110E+00  0.
```

BMD-05V-General Linear Hypothesis

INTRODUCTION

BMD-05V is written in FORTRAN and operates on the CDC 6400 computer with the NCAR compiler. This program performs the calculations required for the general linear hypothesis model. The independent variables are of two general types:

1. Variables used to specify the analysis of variance classifications.
2. Variables used as covariates.

By use of these variables, the program can be used for balanced or unbalanced analysis of variance or covariance designs and missing value problems. The program is described in detail in the BMD manual.

This program is extremely powerful for several reasons.

1. It can handle missing data and unbalanced designs, both in the analysis of variance and the analysis of covariance.
2. The estimates of the coefficients of the linear model (Model I in this case) are given for any specified hypothesis, along with the residual sums of squares under those hypotheses. Thus, for example, the standard non-descript format for an analysis of covariance (assume the slopes are equal between cells and test for coincidental lines) need not be followed. The analysis may be run under any hypothesis and alternative desired.

   Let $H\Omega$ be the null hypothesis and $H\omega$ be the alternative. Suppose that

   $N - k$ = degrees of freedom of the residual sums of squares under $H\Omega$ (i.e., k coefficients of the linear model estimated under $H\Omega$).

   $N - S$ = degrees of freedom of the residual sum of squares under

$H\omega$ (i.e. s (<k) coefficients of the linear model estimated under $H\omega$).

$R\Omega$ = residual sum of squares under Ω

$R\omega$ = residual sum of squares under ω

Then we reject $H\Omega$ at level of significance α ij.

$$\frac{(R\omega - R\Omega)/(k - s)}{R\Omega/(N-k)} > F_{1-\alpha}\ (k-s,\ N-k).$$

3. Since the sums of squares and degrees of freedom are the same for both a Model I and a MOdel II analysis of variance design, the analysis of variance can be performed for either Model I or Model II.

SAMPLE RUN

The following example is a one way analysis of covariance with unequal numbers of observations in each cell. Three groups are compared:

A		B		C	
y	x	y	x	y	x
5.9	0.8	5.2	1.6	7.8	0.6
10.7	3.1	13.4	5.8	12.4	3.4
11.4	4.4	10.0	3.6	10.9	1.5
9.6	1.6	7.5	2.0	9.9	0.7
12.6	4.6	10.1	4.3	16.8	4.5
8.0	2.6	11.9	5.8	13.9	4.1
12.8	5.5	10.7	4.8	11.4	2.3
7.5	1.1	6.8	3.3	8.9	1.3
12.5	3.9	9.0	2.6	13.7	3.1
14.2	4.9			16.0	4.6
8.4	1.4				

The tests can be divided into two categories:

1. The rate of change of y with respect to x (slope) is the same for each cell. Hence we test the following models.

Model 1: $y_{ij} = \mu + \alpha_i + \beta x_{ij} + \varepsilon_{ij}$

$\varepsilon_{ij} \sim N(o, \sigma^2)$

$$\sum_{i=1}^{3} \alpha_i = 0$$

Model 2: $y_{ij} = \mu + \alpha_i + \beta_i x_{ij} + \varepsilon_{ij}$.

$\varepsilon_{ij} \sim N(o, \sigma^2)$

$$\sum_{i=1}^{3} \alpha_i = 0$$

In order to test the hypothesis that the slope is the same for each cell we let:

$R\Omega$ = residual sum of squares under Model 1

= 28.96339 with 26 df.

$R\omega$ = residual sum of squares under Model 2

= 27.11248 with 24 df.

Then

$$F(2,24) = \frac{(28.96 - 27.11)/2}{27.11/24} < 1$$

Hence we accept the hypothesis that the slopes are equal for the three cells.

2. The intercepts are the same for the three cells, and thus the regression lines are coincidental for the three cells. Thus, under Model 1 we test

$H: \alpha_i = \alpha_2 = \alpha_3 = 0$

If we accept H the final linear model is

$Y_{ij} = \mu + \beta x_{ij}$

If we reject H the final linear model is

$$Y_{ij} = \mu + \alpha_i + \beta x_{ij}$$

where $(\mu + \alpha_i)$ is the intercept of the line for each cell. All that we have to do is, under Model 1, find the residual sum of squares under H: $\alpha_1 = \alpha_2 = \alpha_3 = 0$. Call that RH. Then

$$RH = 121.60408 \text{ with } 29 \text{ df.}$$

We test that in the usual manner against RΩ, giving us

$$F(2,24) = \frac{(121.60 - 28.96)/(29 - 26)}{28.96/26} = 41.58$$

Hence we reject the hypothesis for $\alpha = .01$. Thus the regression lines are parallel but not coincidental. The final linear model is (1). The coefficients are:

$$\mu = 5.645$$

$$\alpha_1 = -0.199$$

$$\alpha_2 = -2.193$$

$$\alpha_3 = -\alpha_1 - \alpha_2 = 2.392$$

$$\beta = 1.584$$

Thus we have the following three regression lines:

Group A: $y = 5.446 + 1.584\ x$

Group B: $y = 3.452 + 1.584\ x$

Group C: $y = 8.037 + 1.584\ x.$

The output of the two runs is given on the following two pages.

```
BMD05V - GENERAL LINEAR HYPOTHESIS - VERSION OF JUNE 30, 1964
HEALTH SCIENCES COMPUTING FACILITY, UCLA


PROBLEM NUMBER  01

NUMBER OF DESIGN CARD SETS      3

NUMBER OF INDEPENDENT VARIABLES      4


DESIGN

   1      1  1  0
   2      1  0  1
   3      1 -1 -1


DESIGN   NO. OF     MEAN      SUM OF SQUARES    VARIANCE    STD. DEV.        MEANS OF COVARIATES
          REPS        Y              Y              Y           Y                  1
   1       11     10.32727        69.54182       6.95418     2.63708          3.08182
   2        9      9.40000        52.96000       6.62000     2.57294          3.75556
   3       10     12.17000        78.64100       8.73789     2.95599          2.61000
WITHIN CELLS SUM OF SQUARES = 2.01142818E+02

HYPOTHESES AND SUMS OF SQUARES EXPLAINED BY HYPOTHESES


   1      0000          0.
   2      1111       3621.68661
   3      1110       3449.50719
   4      0111       3442.67801
   5      1001       3529.04592
   6      0001       3251.69200
   7      1000       3411.20633


ESTIMATES OF COEFFICIENTS
                                   H Y P O T H E S I S

                   1               2               3               4               5               6

VARIABLE
    1             0.         5.645097257     10.632424242     0.               6.741391035     0.
    2             0.         -.198557126      -.305151515      .002562286      0.              0.
    3             0.        -2.192840224    -1.232424242    -3.185330571      0.              0.
    4             0.         1.563718541     0.               3.038397142      1.254352547     2.971834429

RESIDUAL
SUM SQS.     3650.65000      28.96339       201.14282       207.97199       121.60408       398.95792

DEGREES OF
FREEDOM OF        30             26             27              27              28              29
RESIDUALS

F TESTS                                     154.56291       160.69336        41.58109       110.71286

DEGREES OF
```

$$MODEL\ 1:\quad y_{ij} = \mu + \alpha_i + \beta X_{ij} + \epsilon_{ij}$$

ESTIMATES OF COEFFICIENTS UNDER MODEL 1.

R_Ω

R_ω

```
FREEDOM OF                                 1   26        1   26        2   26        3   26
F TESTS


                     7
VARIABLF
    1           10.663333333
    2            0.
    3            0.
    4            0.

RESIDUAL
SUM SQS.         239.44967

DEGREES OF
FREEDOM OF          29
RESIDUALS

F TESTS           62.98304

DEGREES OF
FREEDOM OF          3    26
F TFSTS
```

```
BMD05V - GENERAL LINEAR HYPOTHESIS - VERSION OF JUNE 30, 1964
HEALTH SCIENCES COMPUTING FACILITY, UCLA


PROBLEM NUMBER  02

NUMBER OF DESIGN CARD SETS      3

NUMBER OF INDEPENDENT VARIABLES      6


DESIGN

   1     1  1  1
   2     1  0  1
   3     1 -1 -1
```

MODEL 2. $y_{ij} = \mu + \alpha_i + \beta_i X_{ij} + \epsilon_{ij}$

$$\sum_{i=1}^{3} \alpha_i = 0$$

```
DESIGN   NO. OF     MEAN       SUM OF SQUARES    VARIANCE    STD. DEV.       MEANS OF COVARIATES
          REPS       Y              Y               Y           Y              1          2          3
   1       11     10.32727        69.54182       6.95418     2.63708        3.08182    0.         0.
   2        9      9.40000        52.96000       6.62000     2.57294        0.         3.75556    0.
   3       10     12.17000        78.64100       8.73789     2.95599        0.         0.         2.61000

WITHIN CELLS SUM OF SQUARES = 2.0114281HE+02


HYPOTHESES AND SUMS OF SQUARES EXPLAINED BY HYPOTHESES


    1      000000          0.
    2      111111       3623.53752
    3      111000       3449.50718
    4      011111       3453.87950
    5      100111       3612.49342
    6      000111       3395.70754
    7      100000       3411.20033
```

ESTIMATES OF COEFFICIENTS UNDER MODEL 2.

```
ESTIMATES OF COEFFICIENTS
                                 H Y P O T H E S I S

                  1              2              3              4              5              6

VARIABLE
   1          0.         5.657474495    10.632424242    0.             6.060652179    0.
   2          0.          .218203963     -.305151515    1.607030650    0.             0.
   3          0.        -1.971013664    -1.232424242   -4.828124615    0.             0.
   4          0.         1.444469468    0.              2.534707006    1.397226634    2.945151812
   5          0.         1.521356051    0.              3.492811426     .971639652    2.374914372
   6          0.         1.823645136    0.              3.045707472    2.217357529    3.985358221

RESIDUAL
SUM SQS.     3650.65000     27.11248    201.14282      196.77050       38.15658      254.94246

DEGREES OF
FREEDOM OF       30            24           27              25             26             27
RESIDUALS
```

R_ω

```
F TESTS                                             51.35062        150.18149        4.88813        67.22513

DEGREES OF
FREEDOM OF                                           3    24          1    24         2    24         3    24
F TESTS


VARIABLE
    1            10.6633333333
    2            0.
    3            0.
    4            0.
    5            0.
    6            0.

RESIDUAL
SUM SQS.          239.44947

DEGREES OF
FREEDOM OF          29
RESIDUALS

F TESTS            37.59293

DEGREES OF
FREEDOM OF          5    24
F TESTS


 CP SEC=000003 PP SEC=000004 PR=000006 PU=000000 RD=000326 PL=000000
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  709474  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  709474  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  709474  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  709474  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  709474  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

INPUT DECK FOR SAMPLE RUN OF BMD05V (GENERAL LINEAR HYPOTHESIS)

```
PROBLM01  3 3 1 4 0                                                    1
(2F5.0)
DESIGN 11  1  1  0
  0.8  5.9
  3.1 10.7
  4.4 11.4
  1.6  9.6
  4.6 12.6
  2.6  8.0
  5.5 12.8
  1.1  7.5
  3.9 12.5
  4.9 14.2
  1.4  8.4
DESIGN  9  1  0  1
  1.6  5.2
  5.8 13.4
  3.6 10.0
  2.0  7.5
  4.3 10.1
  5.8 11.9
  4.8 10.7
  3.3  6.8
  2.6  9.0
DESIGN 10  1 -1 -1
  0.6  7.8
  3.4 12.4
  1.5 10.9
  0.7  9.9
  4.5 16.8
  4.1 13.9
  2.3 11.4
  1.3  8.9
  3.1 13.7
  4.6 16.0
HYPOTH1110
HYPOTH0111
HYPOTH1001
HYPOTH0001
PROBLM02  3 3 3 4 0                                                    1
(4F5.0)
DESIGN 11  1  1  0
  0.8    0    0  5.9
  3.1    0    0 10.7
  4.4    0    0 11.4
  1.6    0    0  9.6
  4.6    0    0 12.6
  2.6    0    0  8.0
  5.5    0    0 12.8
  1.1    0    0  7.5
  3.9    0    0 12.5
  4.9    0    0 14.2
  1.4    0    0  8.4
DESIGN  9  1  0  1
    0  1.6    0  5.2
    0  5.8    0 13.4
```

INPUT DECK FOR SAMPLE RUN OF BMD05V (GENERAL LINEAR HYPOTHESIS)

```
    0  3.6    0 10.0
    0  2.0    0  7.5
    0  4.3    0 10.1
    0  5.8    0 11.9
    0  4.8    0 10.7
    0  3.3    0  6.8
    0  2.6    0  9.0
DESIGN 10  1 -1 -1
    0    0  0.6  7.8
    0    0  3.4 12.4
    0    0  1.5 10.9
    0    0  0.7  9.9
    0    0  4.5 16.8
    0    0  4.1 13.9
    0    0  2.3 11.4
    0    0  1.3  8.9
    0    0  3.1 13.7
    0    0  4.6 16.0
HYPOTH111000
HYPOTH011111
HYPOTH100111
HYPOTH000111
FINISH
```

MATRIX MANIPULATION ROUTINES

INTRODUCTION

The matrix manipulation routines consist of several simple and easy to use subroutines, each of which performs a specific matrix operation. Although the subroutines are very short and the coding is simple for most of them, the routines should be used, in preference to performing the operations in the main program, as they will increase the simplicity and clarity of the user's program. The operations performed by the subroutines are as follows:

1. TRNSPO calculates the transpose of a matrix and returns the results in another matrix. The transpose is obtained by interchanging the rows and columns of the matrix.
2. ADDSUB accepts two arrays and returns with two additional arrays which contain the sum and difference of the original matrices.
3. SCAMPY accepts a matrix and a scalar, multiplies each element in the matrix by the scalar and returns the result in the original matrix.
4. MATMPY calculates the product of two matrices and returns the result in a third matrix.
5. MATPWR accepts a square matrix as input, raises it to the requested power and returns with the result in a new matrix.
6. INVERT accepts a square matrix as input and uses the Gauss-Jordan Elimination method to replace that matrix with an approximation of its inverse.

7. INVERT2 is the same as invert except that it also returns the value of the determinant of the matrix. (Since the Gauss-Jordan routine is an approximation, its accuracy decreases if the value of the determinant is close to zero) Consequently, the accuracy of the inversion can be easily checked.

INPUT REQUIREMENTS

All of the subroutines, excluding INVERT and INVERT2 use variable dimensioning. This requires that the dimensions for the arrays being sent are the same as the size specified in the parameter lists. For example, if the size of the matrix being sent to TRANSPO is 3 x 4, the dimension statement in the calling program must also specify the array as being 3 x 4.

The inversion routines both require that the array being sent is dimensioned 60 x 60 in the calling program. This, however, may be changed if desired by changing only the dimension statements in the subroutine to agree with those in the calling program.

Variables which must be passed to the subroutines are as follows:

1. TRNSPO (A, B, M, N)

   A is the M by N matrix being sent

   B is an N by M array which will contain the value of the transpose of A.

2. ADDSUB (A, B, C, M, N, Y)

   A & B are the matrices to be passed

   C is to contain the result of A + B

   Y is to contain the result of A - B

   M is the number of rows for all four matrices

   N is the number of columns for all four matrices

3. SCAMPY (A, X, M, N)

    A is the matrix being sent and will contain the the result of the scalar product

    X is the scalar to be used

    M is the number of rows in A

    N is the number of columns in A

4. MATMPY (A, B, C, M, N, L)

    A is an M x N matrix being sent

    B is an N by L matrix being sent

    C is an M by L matrix to contain the result of A*B

    M,N,&L are the dimensions of the matrices as specified above.

5. MATPWR (A, B, DUM, N, I)

    A is the N x N matrix being sent

    B is the N X N matrix to contain the result of A to the Ith power.

    DUM is an N x N dummy matrix and must also be dimensioned in the main program

    N is the dimension and order for all three matrices

    I is the power that the matrix is to be raised to

6. INVERT (A, N)

    A is the 60 x 60 array which contains the matrix to be inverted and is to contain the result

    N is the order of the matrix to be inverted $1 \leq N \leq 60$

7. INVERT2 (A, N, D)

    A is the 60 x 60 array which contains the matrix to be inverted. Upon completion of the routine, this matrix will be replaced by its inverse

N is the number of rows and columns of A that are to be inverted (order) $1 \leq N \leq 60$.

D is to contain the value of the determinant of A.

Sample listing and output:

A brief, self-explanatory program is included which uses all the routines and displays the results.

```
      PROGRAM MATRIX
C-----
C-----*****************************************************************
C-----
C-----   THIS PROGRAM DISPLAYS AN EXAMPLE OF THE USE OF THE MATRIX
C-----MANIPULATION ROUTINES.  IN THE PROGRAM THE FOLLOWING CALCULATIONS
C-----ARE DISPLAYEDO  TRANSPOSE, ADDITION, SUBTRACTION, MULTIPLICATION,
C-----BOTH BY A SCALAR AND ANOTHER MATRIX, THE INVERSE AND DETERMINANT,
C-----AND THE CALCULATION OF A MATRIX RAISED TO A GIVEN POWER.
C-----
C-----*****************************************************************
C-----
      DIMENSION A(4,3), B(3,4), C(4,4), D(3,4), E(3,4), F(3,4), G(60,60)
     1, H(4,4), P(4,4)
C-----
      READ (5,4) ((A(I,J),J=1,3),I=1,4)
      T=10HA
      WRITE (6,6) T,((A(I,J),J=1,3),I=1,4)
C-----
C-----CALCULATE THE TRANSPOSE OF THE 4X3 MATRIX A
C-----AND STORE IT IN THE 3X4 MATRIX, B
C-----
      CALL TRNSPO (A,B,4,3)
      T=10HA(TRANS.)
      WRITE (6,7) T,((B(I,J),J=1,4),I=1,3)
C-----
      READ (5,5) ((B(I,J),J=1,4),I=1,3)
      T=10HB
      WRITE (6,7) T,((B(I,J),J=1,4),I=1,3)
C-----
      READ (5,5) ((D(I,J),J=1,4),I=1,3)
      T=10HD
      WRITE (6,7) T,((D(I,J),J=1,4),I=1,3)
C-----
C-----CALCULATE B+D AND B-D AND STORE THE RESULTS IN
C-----E AND F RESPECTIVELY.  ALL MATRICES ARE DIMENSIONED 3X4.
C-----
      CALL ADDSUB (B,D,E,3,4,F)
      T=10HE=B+D
      WRITE (6,7) T,((E(I,J),J=1,4),I=1,3)
      T=10HF=B-D
      WRITE (6,7) T,((F(I.J),J=1,4),I=1,3)
C-----
C-----MULTIPLY THE 3X4 MATRIX, E, BY THE SCALAR 10.0
C-----
      CALL SCAMPY (E,10.0,3,4)
      T=10HE=10.0*E
      WRITE (6,7) T,((E(I,J),J=1,4),I=1,3)
C-----
C-----CALCULATE THE PRODUCT, A*B, AND STORE THE RESULT IN C.
C-----A IS DIMENSIONED 4X3, B IS DIMENSIONED 3X4 AND C IS
C-----DIMENSIONED 4X4.
C-----
      CALL MATMPY (A,B,C,4,3,4)
      T=10HC=A*B
      WRITE (6,7) T,((C(I,J),J=1,4),I=1,4)
```

```
C-----
C-----READ IN AND PRINT A NEW  C  MATRIX
C-----
      T=10HC (NEW)
      READ (5,5) ((C(I,J),J=1,4),I=1,4)
      WRITE (6,8) T,((C(I,J),J=1,4),I=1,4)
C-----
C-----THE INVERT SUBROUTINE REQUIRES THAT THE MATRIX SENT IS
C-----A 60X60 ARRAY.  THEREFORE, THE ARRAY, C, MUST BE STORED
C-----IN THE ARRAY, G, BEFORE ITS INVERSE CAN BE CALCULATED.
C-----
      DO 1 I=1,4
      DO 1 J=1,4
1     G(I,J)=C(I,J)
C-----
C-----CALCULATE THE INVERSE OF THE FIRST FOUR ROWS AND FOUR COLUMNS
C-----OF THE MATRIX G.
C-----
      CALL INVERT (G,4)
      T=10HC(INVERSE)
      WRITE (6,8) T,((G(I,J),J=1,4),I=1,4)
C-----
C-----STORE THE INVERSE OF C INTO THE ARRAY H
C-----
      DO 2 I=1,4
      DO 2 J=1,4
2     H(I,J)=G(I,J)
C-----
C-----CALCULATE THE PRODUCT OF C TIMES THE INVERSE OF C
C-----(THIS SHOULD GIVE THE IDENTITY MATRIX)
C-----
      CALL MATMPY (C,H,P,4,4,4)
      T=10HIDENTITY
      WRITE (6,7) T,((P(I,J),J=1,4),I=1,4)
C-----
C-----CALCULATE THE INVERSE AND DETERMINANT OF  C  USING INVERT2
C-----IT ALSO REQUIRES THAT THE MATRIX IS DIMENSIONED 60X60.
C-----
      DO 3 I=1,4
      DO 3 J=1,4
3     G(I,J)=C(I,J)
      CALL INVERT2 (G,4,DET)
      T=10HC(INVERSE)
      WRITE (6,8) T,((G(I,J),J=1,4),I=1,4)
      WRITE (6,9) DET
C-----
C-----RAISE  C  TO THE THIRD POWER AND STORE THE RESULT IN H.
C-----P  IS TO BE USED AS A DUMMY MATRIX BY THE SUBROUTINE.
C-----ALL THREE MATRICES ARE DIMENSIONED 4X4.
C-----
      CALL MATPWR (C,H,P,4,3)
      T=10HC**3
      WRITE (6,8) T,((H(I,J),J=1,4),I=1,4)
      STOP
C-----
4     FORMAT (3F5.0)
```

```
5       FORMAT (4F5.0)
6       FORMAT (1H0,A10,4(3F10.3,/,11X))
7       FORMAT (1H0,A10,4(4F10.3,/,11X))
8       FORMAT (1H0,A10,4(4E15.7,/,11X))
9       FORMAT (*0DETERMINANT = *,E15.7)
        END
```

```
        SUBROUTINE MATMPY (A,B,C,M,N,L)
C-----
C-----   SUBROUTINE MATMPY MULTIPLIES TWO MATRICES (A*B) TOGETHER
C-----AND STORES THE RESULT IN C.  THE DIMENSION STATEMENT IN THE
C-----MAIN PROGRAM MUST HAVE A DIMENSIONED AS AN M BY N MATRIX.
C-----B MUST BE DIMENSIONED N BY L AND C MUST BE DIMENSIONED M BY L.
C-----
        DIMENSION A(M,N), B(N,L), C(M,L)
        DO 2 I=1,M
        DO 2 J=1,L
        S=0.
        DO 1 K=1,N
1       S=S+A(I,K)*B(K,J)
2       C(I,J)=S
        RETURN
        END
```

```
        SUBROUTINE TRNSPO (A,B,M,N)
C-----
C-----   THIS SUBROUTINE CALCULATES THE TRANSPOSE OF THE MXN MATRIX A
C-----AND STORES THE RESULT IN B.  A MUST BE DIMENSIONED MXN IN THE
C-----MAIN PROGRAM AND B MUST BE NXM.
C-----
        DIMENSION A(M,N), B(N,M)
        DO 1 I=1,M
        DO 1 J=1,N
1       B(J,I)=A(I,J)
        RETURN
        END
```

```
        SUBROUTINE SCAMPY (A,X,M,N)
C-----
C-----   THIS SUBROUTINE REPLACES THE MATRIX A BY THE PRODUCT OF A
C-----AND THE SCALAR, X.  A MUST BE DIMENSIONED MXN IN THE MAIN PROGRAM
```

```
C-----
      DIMENSION A(M,N)
      MM=M*N
      DO 1 I=1,MM
1     A(I)=X*A(I)
      RETURN
      END
```

```
      SUBROUTINE ADDSUB (A,B,C,M,N,Y)
C-----
C-----   SUBROUTINE ADDSUB STORES THE RESULT OF A+B IN C AND THE
C------RESULT OF A-B IN Y.  A, B, C, AND Y MUST ALL BE DIMENSIONED AS
C------MXN MATRICES IN THE CALLING PROGRAM.
C-----
      DIMENSION A(M,N), B(M,N), C(M,N), Y(M,N)
      DO 1 I=1,M
      DO 1 J=1,N
      C(I,J)=A(I,J)+B(I,J)
1     Y(I,J)=A(I,J)-B(I,J)
      RETURN
      END
```

```
      SUBROUTINE INVERT (A,N)
      DIMENSION A(60,60), B(60), C(60), LZ(60)
      DO 1 J=1,N
1     LZ(J)=J
      DO 11 I=1,N
      K=I
      Y=A(I,I)
      L=I-1
      LP=I+1
      IF (N-LP) 5,2,2
2     DO 4 J=LP,N
      W=A(I,J)
      IF (ABS(W)-ABS(Y)) 4,4,3
3     K=J
      Y=W
4     CONTINUE
5     DO 6 J=1,N
      C(J)=A(J,K)
      A(J,K)=A(J,I)
      A(J,I)=-C(J)/Y
      A(I,J)=A(I,J)/Y
6     B(J)=A(I,J)
      A(I,I)=1.0/Y
      J=LZ(I)
      LZ(I)=LZ(K)
      LZ(K)=J
      DO 10 K=1,N
      IF (I-K) 7,10,7
7     DO 9 J=1,N
      IF (I-J) 8,9,8
8     A(K,J)=A(K,J)-B(J)*C(K)
9     CONTINUE
10    CONTINUE
11    CONTINUE
      DO 16 I=1,N
      IF (I-LZ(I)) 12,16,12
12    K=I+1
      DO 15 J=K,N
      IF (I-LZ(J)) 15,13,15
13    M=LZ(I)
      LZ(I)=LZ(J)
      LZ(J)=M
      DO 14 L=1,N
      C(L)=A(I,L)
      A(I,L)=A(J,L)
14    A(J,L)=C(L)
15    CONTINUE
16    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVERT2 (COVA,N,D)
C-----
C-----   SUBROUTINE INVERT USES THE GAUSS-JORDAN ELIMINATION METHOD
C-----TO REPLACE THE MATRIX, A, WITH ITS INVERSE.  THE DIMENSION
C-----STATEMENT IN THE MAIN PROGRAM MUST HAVE A LISTED AS A 60X60
C-----MATRIX.  HOWEVER, THE ACTUAL NUMBER OF ROWS AND COLUMNS OF A
C-----THAT ARE TO BE INVERTED IS SPECIFIED BY N  (N MUST BE BETWEEN
C-----0 AND 61)
C-----
      DIMENSION COVA(60,60), A(60,60), L(60), M(60)
      DOUBLE PRECISION A,D,BIGA,HOLD
C-----
C-----CONVERT TO DOUBLE PRECISION
C-----
      DO 1 I=1,N
      DO 1 J=1,N
1     A(I,J)=COVA(I,J)
C-----SEARCH FOR LARGEST ELEMENT
      D=1.0
      DO 17 K=1,N
      L(K)=K
      M(K)=K
      BIGA=A(K,K)
      DO 3 I=K,N
      DO 3 J=K,N
      IF (ABSF(BIGA)-ABSF(A(I,J))) 2,3,3
2     BIGA=A(I,J)
      L(K)=I
      M(K)=J
3     CONTINUE
C-----INTERCHANGE ROWS
      J=L(K)
      IF (L(K)-K) 6,6,4
4     DO 5 I=1,N
      HOLD=-A(K,I)
      A(K,I)=A(J,I)
5     A(J,I)=HOLD
C-----INTERCHANGE COLUMNS
6     I=M(K)
      IF (M(K)-K) 9,9,7
7     DO 8 J=1,N
      HOLD=-A(J,K)
      A(J,K)=A(J,I)
8     A(J,I)=HOLD
C-----DIVIDE COLUMN BY MINUS PIVOT
9     DO 11 I=1,N
      IF (I-K) 10,11,10
10    A(I,K)=A(I,K)/(-A(K,K))
11    CONTINUE
C-----REDUCE MATRIX
      DO 14 I=1,N
      DO 14 J=1,N
      IF (I-K) 12,14,12
12    IF (J-K) 13,14,13
13    A(I,J)=A(I,K)*A(K,J)+A(I,J)
14    CONTINUE
C-----DIVIDE ROW BY PIVOT
      DO 16 J=1,N
      IF (J-K) 15,16,15
15    A(K,J)=A(K,J)/A(K,K)
16    CONTINUE
C-----CONTINUED PRODUCT OF PIVOTS
      D=D*A(K,K)
C-----REPLACE PIVOT BY RECIPROCAL
      A(K,K)=1.0/A(K,K)
17    CONTINUE
C-----FINAL ROW AND COLUMN INTERCHANGE
      K=N
18    K=(K-1)
      IF (K) 25,25,19
19    I=L(K)
      IF (I-K) 22,22,20
20    DO 21 J=1,N
      HOLD=A(J,K)
      A(J,K)=-A(J,I)
21    A(J,I)=HOLD
22    J=M(K)
      IF (J-K) 18,18,23
23    DO 24 I=1,N
      HOLD=A(K,I)
      A(K,I)=-A(J,I)
24    A(J,I)=HOLD
      GO TO 18
C-----CONVERT BACK TO SINGLE PRECISION
25    DO 26 I=1,N
      DO 26 J=1,N
26    COVA(I,J)=A(I,J)
      RETURN
      END
```

```
      SUBROUTINE MATPWR (A,B,DUM,N,I)
C-----THE N BY N MATRIX A IS RAISED TO THE POWER I AND PUT BACK IN B.
C-----DUM IS A DUMMY CALCULATION MATRIX.
      DIMENSION A(N,N), B(N,N), DUM(N,N)
      M=I-1
      DO 1 J=1,N
      DO 1 K=1,N
1     DUM(J,K)=A(J,K)
      DO 5 LL=1,M
      DO 3 II=1,N
      DO 3 JJ=1,N
      S=0.
      DO 2 KK=1,N
2     S=S+A(II,KK)*DUM(KK,JJ)
3     B(II,JJ)=S
      DO 4 J=1,N
      DO 4 K=1,N
4     DUM(J,K)=B(J,K)
5     CONTINUE
      RETURN
      END
```

```
          SAMPLE DATA DECK FOR MATRIX MANIPULATION ROUTINES

              -5      8      3
               6      8     -2
              -3     -6      2
               9      5      5
              -3     -2      5      4
              -3      9      5      6
               2      4      1     -0
               6      5      3      8
               8     -6     -3      2
              -6      7     -5      2
               1      0      2      0
               0      4      0      1
               2      0      3      2
               0      1      2      1



A                 -5.000        8.000        3.000
                   6.000        8.000       -2.000
                  -3.000       -6.000        2.000
                   9.000        5.000        5.000


A(TRANS.)         -5.000        6.000       -3.000        9.000
                   8.000        8.000       -6.000        5.000
                   3.000       -2.000        2.000        5.000


B                 -3.000       -2.000        5.000        4.000
                  -3.000        9.000        5.000        6.000
                   2.000        4.000        1.000       -0.


D                  6.000        5.000        3.000        8.000
                   8.000       -6.000       -3.000        2.000
                  -6.000        7.000       -5.000        2.000


E=B+D              3.000        3.000        8.000       12.000
                   5.000        3.000        2.000        8.000
                  -4.000       11.000       -4.000        2.000


F=B-D             -9.000       -7.000        2.000       -4.000
                 -11.000       15.000        8.000        4.000
                   8.000       -3.000        6.000       -2.000


E=10.0*E          30.000       30.000       80.000      120.000
                  50.000       30.000       20.000       80.000
                 -40.000      110.000      -40.000       20.000


C=A*B             -3.000       94.000       18.000       28.000
                 -46.000       52.000       68.000       72.000
                  31.000      -40.000      -43.000      -48.000
                 -32.000       47.000       75.000       66.000
```

```
C (NEW)       1.0000000E+00  0.             2.0000000E+00  0.
              0.             4.0000000E+00  0.             1.0000000E+00
              2.0000000E+00  0.             3.0000000E+00  2.0000000E+00
              0.             1.0000000E+00  2.0000000E+00  1.0000000E+00

C(INVERSE)    3.6842105E-01  2.1052632E-01  3.1578947E-01 -8.4210526E-01
              2.1052632E-01  2.6315789E-01 -1.0526316E-01 -5.2631579E-02
              3.1578947E-01 -1.0526316E-01 -1.5789474E-01  4.2105263E-01
             -8.4210526E-01 -5.2631579E-02  4.2105263E-01  2.1052632E-01

IDENTITY          1.000        0.           0.           0.
                  0.           1.000        0.           0.
                  -.000         .000        1.000        -.000
                  -.000        -.000        0.           1.000

C(INVERSE)    3.6842105E-01  2.1052632E-01  3.1578947E-01 -8.4210526E-01
              2.1052632E-01  2.6315789E-01 -1.0526316E-01 -5.2631579E-02
              3.1578947E-01 -1.0526316E-01 -1.5789474E-01  4.2105263E-01
             -8.4210526E-01 -5.2631579E-02  4.2105263E-01  2.1052632E-01

DETERMINANT =   -1.9000000E+01

C**3          2.1000000E+01  4.0000000E+00  4.2000000E+01  2.0000000E+01
              4.0000000E+00  7.3000000E+01  1.6000000E+01  2.6000000E+01
              4.2000000E+01  1.6000000E+01  8.3000000E+01  4.4000000E+01
              2.0000000E+01  2.6000000E+01  4.4000000E+01  2.7000000E+01
```

Program BMD02R

INTRODUCTION

This program is the step-wise multiple regression of the Biomedical Computer Program series of the School of Medicine, University of California, Los Angeles. It uses a forward step-wise procedure in that is starts with one independent variable and adds another independent variable to the equation at each step.

The step-wise procedure can be valuable in screening large numbers of independent variables for their relationship with the dependent variables. Because the program operator can force individual independent variables into the equation--or conversely, delete them from consideration--the relationships and interactions among independent variables, or groups of independent variables, can be studied. Users of this program should bear in mind that the step-wise procedure does not necessarily produce the best possible regression as it does not consider all possible combinations of independent variables. It is also quite possible with this program to develop regression equations containing more independent variables than are justified by the number of observations. Considerable care should be exercised in interpreting the results. The program is written in FORTRAN and operates on the CDC 6400 computer with the NCAR compiler.

HOW TO USE THE PROGRAM

The Systems Cards are not described here except to point out that tapes 2 and 3 should be identified as scratch tapes. The sequence of input is as follows: (Cards enclosed in parentheses are optional. All other cards must be included in the order shown.)

a. System Cards

b. Problem Card

(c.) Transgeneration Card(s)

(d.) Labels Card(s)

e. F-type Variable Format Cards

(f.) DATA INPUT Cards (Place data input deck here if data input is from cards.)

g. Sub-problem Card(s)

(h.) Control-Delete Card(s)

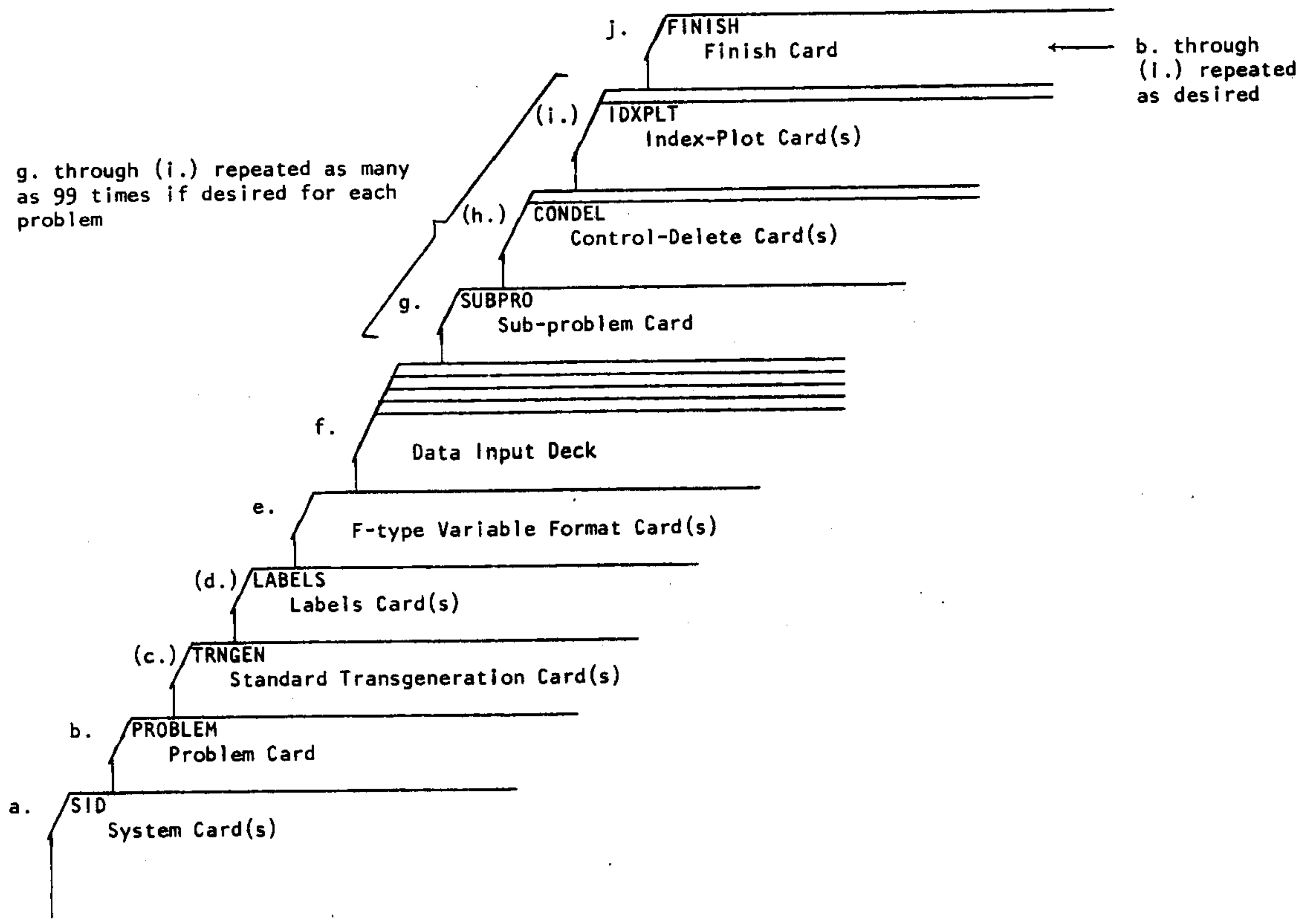
(i.) Index-Plot Card(s)

j. Finish Card

Note: g. through (i.) may be repeated as many as 99 times in each problem;

b. through (i.) may be repeated as often as desired.

Example of Job Deck Set-up:



The cards are prepared as follows:

Problem Card

Card preparation.

Columns 1- 6	PROBLEM	(Mandatory)
Columns 10-15	Alphanumeric problem name	
Columns 17-20	Sample size ($1 \leq n \leq 4000$)	
Columns 24,25	Number of original variables ($2 \leq p \leq 50$)	
Columns 29,30	Number of Transgeneration Cards ($0 \leq m \leq 99$)	
Columns 34,35	Number of variables added by transgeneration ($-9 \leq q \leq 48$)	
Columns 39,40	Tape number if data is on tape ($\neq$ logical 2); otherwise, leave blank.	
Columns 44,45	Number of Sub-problem Cards ($1 \leq s \leq 99$)	
Columns 48,49	Number of variables labeled on Labels Cards. Leave blank if Labels Cards are not used.	
Columns 51-53	YES	If means and standard deviations are to be printed; otherwise, leave blank.
Columns 55-57	YES	If covariance matrix is to be printed; otherwise, leave blank.
Columns 59-61	YES	If correlation matrix is to be printed; otherwise, leave blank.
Columns 63-65	YES	If zero regression intercept is desired; otherwise, leave blank.
Columns 68,69	NO	If tape specified in Columns 39,40 is not to be rewound before this problem; leave blank if Columns 39,40 are blank, or if tape rewind is desired.
Columns 71,72	Number of F-type Variable Format Cards ($1 \leq k \leq 10$)	

Transgeneration Card(s)

The term transgeneration is used to include transformations of input variables and creation of new variables prior to the normal computation performed by the various programs.

The transformations described below are performed on the values of the variables in each case. In these examples, the symbol x_i will denote the ith variable as well as its value.

Examples:

$\log_{10}X_4 \rightarrow X_4$	$\log_{10}X_4$ replaces X_4
$X_5{}^c \rightarrow X_1$	$X_5{}^c$ replaces X_1
$X_2 + X_3 \rightarrow X_2$	$X_2 + X_3$ replaces X_2

The transgenerations available are listed below.

Notation to be used in the following transgeneration list:

i, j, k are variable indices (need not be different)

c is a constant

$a_1, a_2, a_3, \ldots$ are constants

n is the number of cases, or sample size

The mean $\overline{X}_i = \frac{1}{n} \sum_{j=1}^{n} X_{ji}$

The standard deviation $s_i = [\frac{1}{n-1} \sum_{j=1}^{n} (X_{ji} - \overline{X}_i)^2]^{1/2}$

Code	Transgeneration	Restriction
01	$\sqrt{X_i} \rightarrow X_k$	$X_i \geq 0$
02	$\sqrt{X_i} + \sqrt{X_i + 1} \rightarrow X_k$	$X_i \geq 0$
03	$\log_{10}X_i \rightarrow X_k$	$X_i > 0$
04	$e^{X_i} \rightarrow X_k$	--
05	$\arcsin \sqrt{X_i} \rightarrow X_k$	$0 \leq X_i \leq 1$

Code	Transgeneration	Restriction
06	$\arcsin \sqrt{X_i/(n+1)} + \arcsin \sqrt{(X_i+1)/(n+1)} \rightarrow X_k$	$0 \leq (X_i/n) \leq 1$
07	$1/X_i \rightarrow X_k$	$X_i \neq 0$
08	$X_i + c \rightarrow X_k$	--
09	$X_i c \rightarrow X_k$	--
10	$X_i^{\,c} \rightarrow X_k$	$X_i \geq 0$
11	$X_i + X_j \rightarrow X_k$	--
12	$X_i - X_j \rightarrow X_k$	--
13	$X_i X_j \rightarrow X_k$	--
14	$X_i/X_j \rightarrow X_k$	$X_j \neq 0$
15	If $X_i \geq c$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	--
16	If $X_i \geq X_j$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	--
17	$\log_e X_i \rightarrow X_k$	$X_i > 0$
18	$X_i - \overline{X}_i \rightarrow X_k$	--
19	$X_i/s_i \rightarrow X_k$	--
20	$\sin X_i \rightarrow X_k$	--
21	$\cos X_i \rightarrow X_k$	--
22	$\arctan X_i \rightarrow X_k$	--
23	$X_i^{\,X_j} \rightarrow X_k$	$X_i > 0$

Code	Transgeneration	Restriction
24	$c^{X_i} \rightarrow X_k$	$c > 0$
25	$X_i \rightarrow X_k$	--
26	$c \rightarrow X_k$	(Leave code i blank)
27-39	Not defined	
40	If $X_i = a_1$ or a_2 or a_3, . . ., a_7, then $c \rightarrow X_k$; otherwise X_k remains unchanged.	
41	If X_i is blank, then $c \rightarrow X_k$; otherwise X_k remains unchanged. *Note that in reading numeric fields, a blank field and -0 are equivalent.	$(X_i \neq -0)$*
42	If $X_i = a_1$ or a_2 or a_3, . . ., a_7, then $X_j \rightarrow X_k$; otherwise X_k remains unchanged.	
43	If X_i is blank, then $X_j \rightarrow X_k$; otherwise X_k remains unchanged.	$(X_i \neq -0)$

When a violation of a restriction in the right-hand column occurs during transgeneration, the program will print a diagnostic message. Most programs will proceed to the next problem, if any. Some programs will delete the case where the violation occurred and continue the computation. Other programs will screen all the input data for additional restriction violations before proceeding to the next problem, if any.

Card preparation.

Columns 1- 6	TRNGEN	(Mandatory)
Columns 7- 9	Variable index k	
Columns 10,11	Code from transgeneration list (restricted by availability in particular program)	
Columns 12-14	Variable index i	
Columns 15-20	Variable index j or constant c	
Columns 21-25	Blank	
Column 26	Number of a_i's for transformation 40 or 42	
Columns 27-32	a_1 value	
Columns 33-38	a_2 value	
	. . .	
Columns 63-68	a_7 value	

The constants c, a_1, . . ., a_7 are punched with a decimal point if used with variables which have an F-type format and without a decimal point if used with variables which have an I-type format.

Labels Card(s)

Labels Cards allow the user to substitute alphanumeric names for the usual numeric indices (variable numbers or category designations) which appear on the printed output.

Card preparation.

Columns 1- 6	LABELS	(Mandatory)
Columns 7-10	The number of the variable (or category, or index) to be named. This number must be right-justified.	
Columns 11-16	The corresponding alphanumeric name	

Columns 17-20	The number of another variable
Columns 21-26	The corresponding alphanumeric name
.	
.	
.	
Columns 67-70	The number of another variable
Columns 71-76	The corresponding alphanumeric name of that variable (up to 7 per card)

There may be from one to seven pairs of variable numbers and labels on each Labels Card. If desired, only one pair may be specified on each card. However, the total number of labels appearing on all the Labels Cards must equal the number of labels specified on the Problem or Sub-problem Card.

It is not necessary to label all the variables. Those labeled may be listed in any order.

Example:

Suppose the number of variables to be labeled as specified on the Problem Card is 9. Then the Labels Cards might be punched as:

```
LABELS 1OHEIGHT 07WEIGHT 105AGE  003 XI  0051VAR59 0073 X+Y
LABELS 99SEX  0100ANYNAM
LABELS 05STATUS
```

Variable Format Cards

These cards prescribe the format in which the data will be read. Column one contains a left parenthesis followed by the desired format and a right parenthesis. If the entire format specification will not fit into 80 columns, it should be continued on a second card if necessary, starting in Column one. As many as 10 cards may be used for the format.

Sup-problem Card(s)

Card preparation.

Columns 1- 6	SUBPRO	(Mandatory)
Columns 9,10	Number of the dependent variable	
Columns 13-15	Maximum number of steps. This will be 2(p+q) if left blank.	
Columns 20-25	F-level for inclusion. This will be 0.01 if left blank.	
Columns 30-35	F-level for deletion. This will be 0.005 if left blank.	
Columns 40-45	Tolerance level. This will be 0.001 if left blank.	
Columns 49,50	Number of variables on the Index-Plot Card ($0 \leq I \leq 30$)	
Columns 53-55	YES if Control-Delete Cards are included.	
Columns 58-60	YES if list of residuals is to be printed.	
Columns 63-65	YES if summary table is to be printed.	

Control-Delete Card(s)

Card preparation.

Columns 1- 6	CONDEL	(Mandatory)
Column 7	Control value* for first variable	
Column 8	Control value* for second variable	
	. . .	
Column 72	Control value* for 66th variable	

The variable numbers above refer to variables after transgeneration.

*CONTROL VALUES

1 Delete variable (or dependent variable)

2 Free variable

3 Low-level forced variable

. . .

9 High-level forced variable

If no Control-Delete Cards are included, or if a field is left blank on the Control-Delete Cards included in the deck, the value 2 will be assigned if the variable is not the dependent variable and the value 1 assigned if it is the dependent variable.

Index-Plot Card(s)

Variables specified on this card are plotted against the residuals.

Card preparation.

Columns	1- 6	IDXPLT	(Mandatory)
Columns	7, 8	First variable to be plotted	
Columns	9,10	Second variable to be plotted	
		. . .	
Columns	65,66	30th variable to be plotted	

No more than 30 variables may be plotted per sub-problem.

Variables specified refer to the original data after transgeneration.

PROGRAM OPERATION

This program computes a sequence of multiple linear regression equations in a step-wise manner. At each step, one variable is added to the regression equation. The variable added is the one which makes the greatest reduction in the error sum of squares. Equivalently it is the variable which has highest partial correlation with the dependent variable partialed on the variables which have already been added; and equivalently it is the variable which, if it were added, would have the highest F value. In addition, variables can be forced into the regression equation. Non-forced variables are automatically removed when their F values become too low. Regression equations with or without the regression intercept may be selected.

Output from this program includes:

I. At each step:

  A. Multiple R

  B. Standard error of estimate

  C. Analysis-of-variance table

  D. For variables in the equation:

    1. Regression coefficient

    2. Standard error

    3. F to remove

  E. For variables not in the equation:

    1. Tolerance

    2. Partial correlation coefficient

    3. F to enter

II. Optional output prior to performing regression:

  A. Means and standard deviations

  B. Covariance matrix

  C. Correlation matrix

III. Optional output after performing regression:

  A. List of residuals

  B. Plots of residuals vs. input variables

  C. Summary table

IV. Limitations per problem:

  A. p, number of original variables ($2 \le p \le 50$)

  B. q, number of variables added by transgeneration ($-9 \le q \le 48$)

  C. p+q, total number of variables ($2 \le p+q \le 50$)

D. s, number of Sub-problem Cards ($1 \le s \le 99$)

E. k, number of Variable Format Cards ($1 \le k \le 10$)

F. l, number of variables to be plotted ($0 \le l \le 30$)

G. n, number of cases ($1 \le n \le 4000$)

H. m, number of Transgeneration Cards ($0 \le m \le 99$)

SAMPLE RUN

A sample run of the program was made. Three independent variables (water equivalent readings on snow courses) and the dependent variable streamflow were used. No transgenerations were made. Following are the program control cards and data cards used and the resulting output.

```
BMD02R - STEPWISE REGRESSION  - VERSION OF JUNE 2, 1964
HEALTH SCIENCES COMPUTING FACILITY, UCLA

PROBLEM CODE                    SAMPLE
NUMBER OF CASES                   15
NUMBER OF ORIGINAL VARIABLES       4
NUMBER OF VARIABLES ADDED          0
TOTAL NUMBER OF VARIABLES          4
NUMBER OF SUB-PROBLEMS             1



 VARIABLE          MEAN        STANDARD DEVIATION
CPA      1       27.20000           5.75463
CPM      2       28.44667           7.26477
BSA      3        2.34000           1.65680
POUDRE   4      215.93333          82.84277


INPUT DECK FOR SAMPLE RUN OF BMD02R (STEPWISE REGRESSION)


PROBLM    SAMPLE   15    4    0    0        1   4 YES YES YES          1
LABELS    1CPA        2CPM       3BSA       4POUDRE
(F4.1,2(F5.1),1X,F3.0)
20.8 22.1  0.4 147
25.1 17.6  1.6 106
19.3 17.3  0.2 133
30.5 38.9  2.3 202
29.7 36.1  5.0 378
29.4 32.9  4.0 256
30.9 29.7  5.0 228
29.2 26.0  2.8 199
22.3 27.5  2.8 317
38.9 36.5  2.1 274
20.1 19.8  2.4 146
35.4 32.2  1.9 176
27.7 34.3  4.1 325
21.1 21.9  0.0 110
27.6 33.9  0.5 242
SUBPRO   4                                                    YES  YES
FINISH

COVARIANCE MATRIX

VARIABLE       1          2          3          4
 NUMBER

   1        33.116     32.193      4.081     203.550
   2                   52.777      6.056     441.775
   3                               2.745      96.631
   4                                         6862.924
CORRELATION MATRIX

VARIABLE       1          2          3          4
 NUMBER

   1         1.000       .770       .428        .427
   2                    1.000       .503        .734
   3                               1.000        .704
   4                                          1.000
```

```
SUB-PROBLM     1
 DEPENDENT VARIABLE              4
 MAXIMUM NUMBER OF STEPS         6
 F-LEVEL FOR INCLUSION       .010000
 F-LEVEL FOR DELETION        .005000
 TOLERANCE LEVEL             .001000



 STEP NUMBER      1
 VARIABLE ENTERED      2
 MULTIPLE R                  .7340
 STD. ERROR OF EST.        58.3621

 ANALYSIS OF VARIANCE
                          DF     SUM OF SQUARES    MEAN SQUARE    F RATIO
        REGRESSION         1       51770.878        51770.678     15.189
        RESIDUAL          13       44310.055         3408.466
                                                        .
               VARIABLES IN EQUATION                    .              VARIABLES NOT IN EQUATION
                                                        .
  VARIABLE       COEFFICIENT  STD. ERROR  F TO REMOVE   .    VARIABLE   PARTIAL CORR.    TOLERANCE    F TO ENTER
                                                        .
                                                        .
  (CONSTANT      -22.18234 )                            .
 CPM      2        8.37600     2.14780     15.1889      .    CPA      1      -.31917         .4070       1.3611
                                                        .    BSA      3       .57026         .7469       5.7829


 STEP NUMBER      2
 VARIABLE ENTERED      3

 MULTIPLE R                  .8299
 STD. ERROR OF EST.        49.9172

 ANALYSIS OF VARIANCE
                          DF     SUM OF SQUARES    MEAN SQUARE    F RATIO
        REGRESSION         2       66180.218        33090.109     13.280
        RESIDUAL          12       29900.715         2491.726
                                                        .
               VARIABLES IN EQUATION                    .              VARIABLES NOT IN EQUATION
                                                        .
  VARIABLE       COEFFICIENT  STD. ERROR  F TO REMOVE   .    VARIABLE   PARTIAL CORR.    TOLERANCE    F TO ENTER
                                                        .
                                                        .
  (CONSTANT      -1.47907 )                             .
 CPM      2        5.79969     2.12488      7.4503      .    CPA      1      -.44081         .4048       2.6529
 BSA      3       22.40304     9.31445      5.7829      .


 STEP NUMBER      3
 VARIABLE ENTERED      1

 MULTIPLE R                  .8656
 STD. ERROR OF EST.        48.7981
 ANALYSIS OF VARIANCE
                          DF     SUM OF SQUARES    MEAN SQUARE    F RATIO
        REGRESSION         3       71990.276        23996.759     10.957
```

```
LIST OF RESIDUALS

   CASE        RESIDUAL
     1         -1.45753
     2         -5.54663
     3         24.43924
     4        -69.40094
     5         44.14942
     6        -27.03178
     7        -41.16451
     8          5.00542
     9         21.60442
    10         55.72256
    11        -32.39415
    12        -14.04509
    13         17.46157
    14        -25.59238
    15         21.96632

SUMMARY TABLE

  STEP              VARIABLE               MULTIPLE                 INCREASE       F VALUE TO     NUMBER OF INDEPENDENT
 NUMBER          ENTERED  REMOVED         R          RSQ            IN RSQ      ENTER OR REMOVE   VARIABLES INCLUDED

   1              CPM     2             .7340       .5388           .5388          15.1889               1
   2              BSA     3             .8299       .6888           .1500           5.7829               2
   3              CPA     1             .8656       .7493           .0605           2.4529               3


        RESIDUAL        11        24090.657      2190.060

                  VARIABLES IN EQUATION                 .
                                                        .              VARIABLES NOT IN EQUATION
   VARIABLE       COEFFICIENT  STD. ERROR  F TO REMOVE  .   VARIABLE    PARTIAL CORR.    TOLERANCE    F TO ENTER
                                                        .
   (CONSTANT       54.28475 )                           .
   CPA     1       -5.56391     3.41600       2.6529    .
   CPM     2        9.07334     2.82977      10.2809    .
   BSA     3       23.45328     8.75484       7.1712    .

F-LEVEL INSUFFICIENT FOR FURTHER COMPUTATION
```

Program NONLIN

INTRODUCTION

NONLIN is a nonlinear programming program available for use on the FORTRAN Extended compiler of the CDC 6400 SCOPE System.

Many optimization problems cannot be dealt with using linear programming techniques because either the objective function or the restraint equations are not linear functions. Whenever the cost of a resource depends on the amount of resource used, the objective function becomes nonlinear. This is a very common occurrence in all types of problems. Programming problems also become nonlinear when risk and uncertainty are considered. For many problems the risks involved in giving up other decisions must also be considered in the objective function.

The solution of nonlinear problems does not involve one clear-cut method as does linear programming. There are several different methods of solution, and not all methods are equally well suited to a given problem. The task of the user is to discover the logical foundations of a given problem, and to find suitable formulations for the objective functions and the constraint equations. If these formulations can then be expressed as described below, this method should be used.

PROGRAM OPERATION

The program is designed to solve the nonlinear programming problem when stated in the following form:

(A) minimize the function $f(\chi)$ where χ is the nx1 vector:

$$\chi = (x_1, x_2, \ldots, x_n)'$$

subject to the χ obeying the following m (nonlinear) constraining inequalities:

$$R_j(\chi) \geq 0 \quad \text{for } j = 1, 2, \ldots, m$$

The method used depends upon formulation of the problem as one of unconstrained minimization in the following way:

(B) solution of the problem stated in (A) will be the same as the limiting solution to the problem:

for $\quad r_k > 0,\ r_1 > r_2 > \ldots > r_p > 0$

find the minimum with respect to χ and r_k of the function

$$P(\chi, r_k) = f(\chi) + r_k \cdot \sum_{j=1}^{m} [R_j(\chi)]^{-1},$$

if the following conditions on the functions involved hold:

C1 There exists at least one χ such that the inequalities stated in (A) are true (i.e., at least one feasible solution),

C2 $f(\chi)$ and $-R_j(\chi)$ are convex functions and have continuously differentiable first and second derivatives,

C3 For every finite k, the set of χ such that $f(\chi) \leq k$ is bounded, provided also that each χ is a feasible solution,

C4 For every $r > 0$, $P(\chi, r)$ is convex.

Since the function to be minimized in (B) is without constraints, the minimization can be performed by the steepest descent method starting at the known feasible solution. The method consists of starting with some arbitrary positive r_1 and using the steepest descent method (or a variation) to find a vector χ_1 which makes $P(\chi, r_1)$ a minimum. Then select an r_2, say 1/2 of r_1, also positive, and again minimize $P(\chi, r_2)$. In this way a series of feasible solutions, χ_1, χ_2, . . . will be generated. It is possible to show (prove mathematically) that under the conditions stated this sequence of feasible solutions will converge to the optimal feasible solution. (Also, under the conditions stated a single unique optimal feasible solution exists.)

The utility of the program is extended greatly by the fact that, although it finds the optimal solution when the rather stringent conditions above are met, it has also been shown empirically to provide either an optimal solution or a solution which is a great improvement over some initial feasible solution when the conditions are not met. Thus, the program is a nonlinear programming algorithm when the conditions in (B) are met and a nonlinear programming heuristic in other cases.

The program user must provide four FORTRAN coded subroutines as the manner of stating his problem. One program reads in coefficients and other necessary data; a second provides the value of the objective function and the constraining functions $f(\chi)$ and $R_j(\chi)$; a third provides the derivatives of these same functions; and a fourth provides any of m+l nxn matrices of second derivatives of the functions. In addition, a number of parameters, such as initial χ vector, initial r value, decrement for r, etc. must be specified. The exact form for these subroutines and parameter cards follows.

INPUT REQUIREMENTS

User-supplied Information Cards

The user must supply a parameter card which precedes the data deck, an initial starting point or χ vector following the parameter card, and an option card which follows the data deck. These cards give instructions to the computer program and are necessary for execution of the program.

1. Parameter Card

Column	Format	Name	Use
01-12	E12.0	EPSI (ε)	Tolerance used to decide if an unconstrained minimum has been achieved (normally set at 10^{-5}) (see option 9)
13-24	E12.0	RHOIN (r_1)	Possible initial value of r (see option 1)
25-36	E12.0	THETAO (θ_0)	Tolerance used to decide if the solution to problem (A) has been approximated (see option 5)
37-48	E12.0	RATIO (c)	Parameter (> 1) used to compute consecutive values of r; $r_i/-1 = r_i/c$
49-60	E12.0	TMMAX	Maximum amount of time for solving problem (in seconds)
61-64	I4	M	Number (integer) of non-trivial constraints (see option 2); if OPTION 2 = 1, $(M + N) \leq 200$: if OPTION 2 = 2, $M \leq 200$
65-68	I4	N	Number (integer) of variables, $N \leq 100$

2. Initial starting point or χ vector, (χ^0)

The cards designating the initial starting point immediately follow the parameter card.

There are six components per card, requiring (N/6) cards for the vector. Each card format is 6E12.0.

3. Option Card

In general the ith option is designated by a single integer in the (ix7)th column. In the general situation it is recommended that all options be equal to 1. The Fortran variable name is NT1 for Option 1, NT2 for Option 2, etc.

	Value	Meaning
Option 1	= 1	r_1 is given by formula 25, reference 1, p. 47
	= 2	r_1 is given by formula 23, reference 1, p. 47
	= 3	r_1 = RHOIN (see parameter card)
Option 2	= 1	The requirements (trivial constraints) that $x_i \geq 0$ for $i = 1, . . ., n$ are to be automatically included in the problem
	= 2	Only constraints on the problem are those inputed by the user
Option 3	= 1	Standard printout (This includes a call to OUTPUT at the solution of every subproblem, the estimates of the "Lagrange multipliers" and first- and second-order solution estimates.)
	= 2	For additional printout (includes standard printout, every intermediate point, gradient and mapped gradient vectors)
Option 4		Not used
Option 5		Final convergence criterion (see parameter card)
	= 1	quit when $\frac{f[x(r_k)]}{G[x(r_k),u(r_k)]} - 1 < \theta_0$
	= 2	quit when $r \sum_{j=1}^{n} 1/R_j[x(r_k)] < \theta_0$
	= 3	quit when $\frac{\text{first-order estimate of } v_0}{G[x(r_k),u(r_k)]} - 1 < \theta_0$
Option 6		Not used
Option 7		First move after a minimum is achieved
	= 1	No extrapolation
	= 2	Extrapolate through last two minima
	= 3	Extrapolate through last three minima
Option 8	= 1	Matrix of second partials computed every time (Recommended)
	= 2	Repeated use is made of matrix of second partials

	Value	Meaning
Option 9		Subproblem convergence criterion, or when to stop minimizing P-function for fixed value of r (see parameter card)
	= 1	quit when $\left| \nabla_\chi P^T (\chi^i, r)\left[\frac{\partial^2 P(\chi, r)}{\partial \chi_i \partial \chi_j}\right]^{-1} \nabla_\chi P(\chi^i, r)\right| < \varepsilon$
	= 2	quit when $\left| \nabla_\chi P^T (\chi^i, r)\left[\frac{\partial^2 P(\chi, r)}{\partial \chi_i \partial \chi_j}\right]^{-1} \nabla_\chi P(\chi^i, r)\right| < \frac{P(\chi^{i-1}) - P}{5}$
	= 3	quit when $\left| \nabla_\chi P(\chi^i, r)\right| < \varepsilon$
Option 10	= 1	At least one nonlinear constraint
	= 2	Linear constraints
	= 3	Linear constraints and linear objective function (i.e., a linear programming problem)

Note when Option 10 = 3 MATRIX - the user subroutine supplying the second partial derivatives will not be called, when Option 10 = 2 it will be called only to get the second partials of $f(\chi)$.

User-Supplied Subroutines

The first card in each user-supplied subroutine is necessary and connects the essential common region [called the share common region] of the main program with the user routines (see example). Of course, all the common regions may be made accessible to the user by duplication of the common and dimension cards. Blank common is left for the user's use in transferring data between his subroutines. FORTRAN II modifications require one master common card - easily made up from the separate FORTRAN IV regions.

I: Subroutine RESTNT (I,VALU)

When I = 0, this routine must set VAL = $f(\chi)$.

When I ≠ 0, routine must set VAL = $R_i(\chi)$.

χ is found in COMMON.

II. Subroutine GRAD1(I)

When I = 0, GRAD1 must place $\nabla_\chi f(\chi)$ in DEL

When I ≠ 0, GRAD1 must place $\nabla_\chi R_i(\chi)$ in DEL

χ, DEL in common. DEL does *not* have zeros upon entry to GRAD1.

III. Subroutine MATRIX (J)

This subroutine must supply the upper triangle and diagonal portions of the matrix of second partials of f or any R_j on request. The *lower triangle* (the computer variable named array is A(,)) *must not be disturbed.* Note that the upper triangle and diagonal elements of A are all zero upon entry to MATRIX.

When J = 0, MATRIX must place $\frac{\partial^2 f(\chi)}{\partial x_k \partial x_i}$ in A(k,i) for k = 1, . . ., N
i = k, . . ., N

When J ≠ 0, MATRIX must place $\frac{\partial^2 R_j(\chi)}{\partial x_k \partial x_i}$ in A(k,i) for k = 1, . . ., N
i = k, . . ., N

IV. Subroutine READIN

READIN is the first of the user's subroutines to be called and is called only once for each problem. Essentially, the purpose of this routine is to have the user read in the data necessary to evaluate the objective function and constraints and all their first and second partial derivatives.

PROGRAM LISTING AND SAMPLE OUTPUT

The following example was used by G. M. Van Dyne to predict the relationship between botanical and chemical components in fistual forage samples:

Let C_i be the chemical content by weight of a given constituent in the ith mixed sample expressed as a proportion; let P_j be the chemical content of the jth species in the mixed sample expressed as a percent; and let W_{ij} be the percent weight of the jth species in the ith sample. Also let M be the number of species and N be the number of samples.

$$C_i = \sum_j P_j W_{ij} \qquad (0 \leq C_i \leq 1)$$

$$\Sigma W_{ij} = 100 \qquad (0 \leq W_{ij} \leq 100)$$

$$\text{and} \quad 0 \leq P_j \leq 100 \qquad \text{for } i = 1, 2, \ldots, N \text{ and } j = 1, 2, \ldots, M$$

The problem is to solve for the P_j values, given the corresponding values of C_i and W_{ij} for N samples, such that the following function, Q, is minimized:

$$Q = \sum_{i=1}^{N} (C_i - \sum_{j=1}^{M} P_j W_{ij})^2$$

The user-supplied subroutines for this were coded as follows:

I. RESTNT (IN,VAL) evaluates the constraints according to IN.

When IN = 0, VAL will be returned with the present value of the objective function Q.

When IN ≠ 0, VAL is set equal to the value specified by RHOIN on the option card.

II. GRAD1(IN) computes the gradients according to IN.

When IN = 0, the gradients stored in DEL(I) are calculated by $\nabla_X f(X)$. For this problem the formula is expressed as

$$\nabla_X f(X) = -2 \sum_{j=1}^{M} W_{ij} \; (C_j - P_j W_{ij})$$

When IN ≠ 0, the gradient DEL(IN) is set equal to one, and all others are set equal to zero.

III. MATRIX (IN) calculates the second partials.

When IN = 0, the $\frac{\partial^2 f(\chi)}{\partial x_i \partial x_j}$ is calculated by $A_{ij}, = \sum_{k=1}^{M} W_{ik} W_{jk}$ for $i = 1, 2, . . ., N$ and $j = i, i+1, . . ., N$.

When IN $\neq$ 0, the $\frac{\partial^2 R(\chi)}{\partial x_i \partial x_j}$ are returned with zero values.

IV. READIN reads in all necessary data that is specified by the objective function. For this problem READIN supplied the values for M, N, W_{ij}; $i = 1, 2, . . ., N$ and $j = 1, 2, . . ., M$ and C_j, $j = 1, 2, . . ., M$. An identifying header card is also read and printed by READIN.

The output from this problem begins with a printout of the parameter card, and option card F is then listed as the initial value of the objective function. The initial conditions for the P variables are listed as both the current value of χ and the constraint values.

For each feasible solution generated, the values of the problem variables and functions are then printed as follows:

POINT is the number of iterations necessary for convergence.

DOTT is the value of the convergence criterion for each subproblem (a subproblem consists of the calculation of a feasible solution).

RHO is the current value for r at that solution.

MAGNITUDE gives the determinant of the gradient matrix supplied by GRADI.

$|\nabla_\chi P[\chi(r)]|$

F is the current value of the objective function $f[\chi(r)]$.

P is the convergence criterion used to determine if the feasible solution is also an optimal solution.

$f[\chi(r)]$ + RSIGMA

G is the convergence criterion for optimality of the dual solution.

$$f[\chi(r)] - \text{RSIGMA}$$

RSIGMA is the amount of deviation allowed for optimality.

$$\sum_{j=1}^{M} r/R_j[\chi(r)]$$

CURRENT VALUE OF χ is $\chi(r)$ or the values of P_j expressed as a percent.

CONSTRAINT VALUES are $R_1[\chi(r)], \ldots, R_M[\chi(r)]$.

For this problem, these are also the resulting values of P_j.

For each feasible solution, these values are printed and the generation of feasible solutions is continued until the solutions converge to optimality.

```
      PROGRAM NONLIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6)
C-----MAIN    RAC  NON-LINEAR PROGRAMMING ROUTINE
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200),NSOL
      REAL LRJ
C-----PARAMETER CARD
C-----INITIAL  X VERCTOR CARD  FORMAT
C-----OPTION CARD FORMAT
      REWIND 6
      NSOL=0
1     READ (5,5) EPSI,RHOIN,THETAO,RATIO,TMMAX,M,N
      IF(N.EQ.0) GO TO 11
      CALL SIT (TMMAX)
      READ (5,6) (X(I),I=1,N)
      NTCTR=0
      NP1=N+1
      NM1=N-1
C-----SUBROUTINE READIN IS UNDER PROGRAMMER CONTROL
      CALL READIN
C-----OPTION CARD FOLLOWS PROGRAMMERS DATA
      READ (5,7) NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10
      WRITE (6,8)
      WRITE (6,9) N,M,TMMAX,RHOIN,RATIO,EPSI,THETAO
C-----IF NON-NEGS INCLUDED BY LETTING NT2=1 THEN ADD M+N TO GET NO. REST
      MN=M
      GO TO (2,3), NT2
2     MN=MN+N
3     CONTINUE
      WRITE (6,10)
      WRITE (6,7) NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10
      CALL TIMEC
      NPHASE=2
      CALL EVALU
      P0=0.0
      G=0.0
      RSIGMA=0.0
      CALL OUTPUT (2)
      CALL STORE
      CALL FEAS
C-----NPHASE=5 IS USED TO INDICATE NO FEASIBLE POINT EXIST
      GO TO (4,4,4,4,1), NPHASE
4     NPHASE=2
      NTCTR=0
      CALL BODY
      GO TO 1
11     CONTINUE
      ENDFILE 6
      STOP
C-----
5     FORMAT (5E12.0,2I4)
6     FORMAT (6E12.0)
```

```
 7      FORMAT (10I7)
 8      FORMAT (37H1 NONLINEAR PROGRAMMING ROUTINE-RAC  )
 9      FORMAT (1H0,5X,2HN=I3,6X,2HM=I3//8X,10HMAX. TIME=1PE10.3,4X,2HR=1P
       1E14.7,4X,6HRATIO=E10.3,6X,8HEPSILON=E10.3,4X,6HTHETA=E10.3)
10      FORMAT (18H0 OPTIONS SELECTED)
        END
```

```
        SUBROUTINE READIN
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON C,P,NN
        DIMENSION C(150), P(150,20), ID(14)
        DATA (NCASE=0)
C-----VAN DYNE PROBLEM  -  NONNEGATIVE LEAST SQUARES   ONE RHS
 1      NCASE=NCASE+1
        READ (5,3) (ID(I),I=1,14)
        WRITE (6,4) (ID(K),K=1,14)
        READ (5,5) NR,NC
        N=NC
        M=N
        NN=NR
        WRITE (6,8) NR,NC,NCASE
        DO 2 I=1,NR
        READ (5,6) (P(I,J),J=1,NC),C(I)
        WRITE (6,7) I,(P(I,J),J=1,NC),C(I)
 2      CONTINUE
        RETURN
C-----
 3      FORMAT (13A6,1A2)
 4      FORMAT (1H1,13A6,1A2)
 5      FORMAT (2I3)
 6      FORMAT (10F8.3)
 7      FORMAT (1H ,I5,13F8.3,F15.3)
 8      FORMAT (1H ,2I6,5X,8HFILE NO.,I3)
        END
```

```
        SUBROUTINE RESTNT (IN,VAL)
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON C,P,NN
        DIMENSION C(150), P(150,20)
C-----CONSTRAINT EVALUATION    VAN DYNE
        IF (IN) 1,1,4
 1      SUM1=0.0
        DO 3 I=1,NN
        SUM2=0.0
        DO 2 J=1,N
        SUM2=SUM2+X(J)*P(I,J)
 2      CONTINUE
        SUM1=SUM1+(C(I)-SUM2)**2
 3      CONTINUE
        VAL=SUM1
        GO TO 5
 4      VAL=X(IN)
 5      RETURN
        END
```

```
        SUBROUTINE GRAD1 (IN)
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON C,P,NN
        DIMENSION C(150), P(150,20)
C----- GRADIENT COMPUTATION      VAN DYNE
        IF (IN) 1,1,5
 1      DO 4 I=1,N
        SUM1=0.0
        DO 3 J=1,NN
        SUM2=0.0
        DO 2 K=1,N
        SUM2=SUM2+X(K)*P(J,K)
 2      CONTINUE
        PROD=C(J)-SUM2
        SUM1=SUM1+P(J,I)*PROD
 3      CONTINUE
        DEL(I)=-2.0*SUM1
 4      CONTINUE
        GO TO 7
 5      DO 6 I=1,N
        DEL(I)=0.0
 6      CONTINUE
        DEL(IN)=1.0
 7      RETURN
        END
```

```
        SUBROUTINE MATRIX (IN)
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON C,P,NN
        DIMENSION C(150), P(150,20)
C-----MATRIX OF SECOND PARTIALS
C-----VAN DYNE PROBLEM
        IF (IN) 1,1,4
 1      DO 3 I=1,N
        DO 3 J=I,N
        SUM=0.0
        DO 2 K=1,NN
```

```
      SUM=SUM+P(K,I)*P(K,J)
    2 CONTINUE
      A(I,J)=2.0*SUM
    3 CONTINUE
    4 RETURN
      END
```

```
      SUBROUTINE BODY
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      NUMINI=0
C-----OPTION OF GETTING INITIAL RHO
      CALL RHOCOM
    1 CALL PEVALU
    2 IREP=0
      ISME=1
      CALL GRAD (1)
      CALL SECORD (1)
      GO TO 8
    3 IF (NT8.EQ.1) 6,4
C-----IF NT8 IS GREATER THAN ONE THEN REPEATED USE IS MADE OF THE A MATR
    4 IREP=IREP+1
      IF (IREP.LT.NT8) 7,5
    5 IREP=0
    6 ISME=1
      CALL SECORD (2)
      GO TO 8
    7 ISME=2
    8 DO 9 I=1,N
    9 DELX(I)=DELXO(I)
      CALL TIME (XI)
      CALL INVERS (ISME)
      CALL TIME (YS)
      CALL STORE
      CALL TIME (ZO)
      CALL OPT
      CALL TIME (WM)
      CALL MAG
      GO TO (11,10), NT3
   10 WRITE (6,32) XI,YS,ZO,WM
      CALL TIMEC
      CALL OUTPUT (1)
   11 GO TO (27,12,12), NPHASE
   12 CALL CONVRG (N1)
      GO TO (13,3), N1
```

```
C-----MINIMUM  ACHIEVED
13     GO TO (14,15), NT3
14     CALL TIMEC
       CALL OUTPUT (1)
15     NUMINI=NUMINI+1
       GO TO (31,16,16), NPHASE
16     CALL ESTIM
       GO TO (28,24,17), NPHASE
17     CALL FINAL (N2)
       GO TO (18,19), N2
18     RETURN
19     RHO=RHO/RATIO
C-----A VECTOR IS LEFT IN DELX(I) BY ESTIM
       IF (NUMINI-2) 2,20,20
20     GO TO (2,21,21), NT7
21     CALL GRAD (2)
       CALL OPT
       CALL PEVALU
       GO TO (23,22), NT3
22     WRITE (6,33)
       CALL OUTPUT (1)
23     GO TO 5
24     GO TO (25,17), NT10
25     CONTINUE
C-----THISIS UNCODED AT PRESENT
       GO TO 17
C-----AFTER OPTIMUM MOVE GOES HERE IF IN FEASIBILITY PHASE
26     RETURN
       GO TO (30,12,26), NSATIS
C-----TEST FOR NON DUAL FEASIBILITY
27     IF (RJ(NVI)) 1,1,26
28     GO TO (29,17,17), NSATIS
29     CALL STORE
       IF (RJ(NVI)) 1,1,26
30     CALL STORE
       GO TO 1
C-----FORTUITOUSLY SATISFIED CONSTRAINTS ACCEPTED
31     IF (G) 16,16,26
C-----
32     FORMAT(2H0  39HTHE TIME BEFORE THE CALL TO INVERSE WAS  F9.3,
      119H SECONDS, AFTER WAS  F9.3, 8H SECONDS   /   2X, 35HTHE TIME BEF
      2ORE THE CALL TO OPT WAS  F9.3, 19H SECONDS, AFTER WAS  F9.3,
      3 8H SECONDS    )
33     FORMAT (6X,30HMOVED ON EXTRAPOLATION VECTOR )
       END



       SUBROUTINE CONVRG (N1)
       COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
       COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
       COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
       COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETA0,NTCTR,N
```

```
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      COMMON /TSW/ NSWW
C-----LEAVES  A 1 IF  MIN HAS BEEN FOUND, 2 IF NOT
C-----DOT PRODUCT OF  GRAD.AND INVERSE PRODUCT LEFT  (DURING OPT) AT DOT
      N1=2
      NSWW=1
      GO TO (1,2,3), NT9
1     IF (ABS(DOTT)-EPSI) 4,4,5
2     IF (ABS(DOTT)-(P1-P0)/5.) 4,4,5
3     IF (ADELX.GT.EPSI) 5,4
4     N1=1
5     GO TO (6,7), NSWW
6     RETURN
7     CALL EXIT
      END
```

```
      SUBROUTINE ESTIM
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETA0,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
C-----SIX CARDS HAVE BEEN MADE COMMENTS TO ELIMINATE 1ST AND 2ND ORDER
C-----ESTIMATES AND LAGRANGE MULTIPLIER OUTPUT. ***
      CALL STORE
      Z9=SQRT(RATIO)
      Z1=(1./Z9+1./RATIO)
      Z2=Z1+1./Z9**3
      Z3=1./Z9**3
      Z4=RATIO+Z9
      Z5=Z9**3
      Z6=1./((RATIO-1.)*(Z9-1.))
      Z7=1./Z9
      Z8=1./(Z9-1.)
      IF (NUMINI-2) 5,3,1
C-----30 WRITE(6,102)
1     CONTINUE
      P0=(PR2-Z4*PR1+Z5*P1)*Z6
      G=(RATIO*G1-GR1)/(RATIO-1.)
      DO 2 I=1,N
2     X(I)=(XR2(I)-Z4*XR1(I)+Z5*X1(I))*Z6
      CALL EVALU
C-----CALL OUTPUT(2)
      CALL SPECFE
      GO TO (15,3,3), NSATIS
```

```
C-----20 WRITE(6,101)
3      CONTINUE
       P0=(Z9*P1-PR1)*Z8
       G=(RATIO*G1-GR1)/(RATIO-1.)
       DO 4 I=1,N
4      X(I)=(Z9*X1(I)-XR1(I))*Z8
       CALL EVALU
C-----CALL OUTPUT(2)
       CALL SPECFE
       GO TO (15,5,5), NSATIS
C-----105 WRIRE(6,103)
5      CONTINUE
       DO 6 J=1,MN
6      RJ(J)=RJ1(J)**2
       GO TO (7,9), NT2
7      DO 8 I=1,N
8      X(I)=RJ(I)
C-----46 CALL OUTPUT (2)
9      CONTINUE
       CALL REJECT
       CALL PEVALU
       CALL GRAD (2)
       IF (NUMINI-2) 13,16,10
10     GO TO (13,17,11), NT7
C-----SECOND  ORDER MOVE  FOR NEXT MINIMUM
11     DO 12 I=1,N
12     DELX(I)=Z1*X1(I)-Z2*XR1(I)+Z3*XR2(I)
13     PR2=PR1
       GR2=GR1
       PR1=P1
       GR1=G1
       DO 14 I=1,N
       XR2(I)=XR1(I)
14     XR1(I)=X1(I)
15     RETURN
16     GO TO (13,17,17), NT7
17     DO 18 I=1,N
18     DELX(I)=(X1(I)-XR1(I))*Z7
       GO TO 13
C-----
       END
```

```
      SUBROUTINE EVALU
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETA0,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
```

```
C-----DIFFERENT   FOR  PHASE3
      GO TO (1,1,9), NPHASE
1     GO TO (2,4), NT2
2     DO 3 I=1,N
3     RJ(I)=X(I)
      IPN=N
      GO TO 5
4     IPN=0
5     DO 6 I=1,M
      J=IPN+I
      CALL RESTNT (I,RJ(J))
6     CONTINUE
      GO TO (7,8), NPHASE
7     F=-RJ(NVI)
      GO TO 9
8     CALL RESTNT (0,F)
9     RETURN
      END
```

```
      SUBROUTINE FEAS
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETA0,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      NPHASE=1
      NSW=1
      GO TO (1,5), NT2
1     NFIX=1
      DO 3 I=1,N
      IF (X(I)) 2,2,3
2     NFIX=2
      X(I)=1.E-05
3     CONTINUE
      GO TO (5,4), NFIX
4     NPHASE=2
      CALL EVALU
      NPHASE=1
      WRITE (6,11)
      CALL OUTPUT (2)
5     DO 6 I=1,MN
      IF (RJ(I)) 9,9,6
6     CONTINUE
      GO TO (8,7), NSW
7     CALL TIMEC
      WRITE (6,12)
      G=0.0
      P0=0.0
      CALL OUTPUT (2)
8     RETURN
9     NVI=I
      F=-RJ(NVI)
      CALL BODY
      NSW=2
      NVI=0
      IF (F) 5,5,10
10    WRITE (6,13)
C-----TO INDICATE TO MAIN TO START ON NEXT PROBLEM.
      NPHASE=5
      GO TO 8
C-----
11    FORMAT (1H0,2X,48HMADE VIOLATED NON-NEGATIVITIES SLIGHTLY POSITIVE
     1)
12    FORMAT (51H0*****THE FEASIBLE STARTING POINT TO BE USED IS ...)
13    FORMAT (3X,89HTHIS PROBLEM POSSESSES NO FEASIBLE STARTING POINT, W
     1ILL LOOK FOR DATA FOR NEXT PROBLEM.   )
      END
```

```
      SUBROUTINE FINAL (N2)
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETA0,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
C-----FINAL  CONVERGENCE  CRITERION
C-----1 LEFT IF CONVERGENCE MET  2 IF NOT
      GO TO (1,2,3), NT5
1     EPSIL=ABS(F/G-1.)
      IF (EPSIL-THETA0) 5,5,6
2     IF (RSIGMA-THETA0) 5,5,6
3     IF (NUMINI-1) 5,4,4
4     PEST=PR1-(PR1-P0)/(1.-1./SQRT(RATIO))
      EPSIL=ABS(PEST/G-1.)
      IF (EPSIL-THETA0) 5,5,6
5     N2=1
      GO TO 7
6     N2=2
7     RETURN
      END
```

```
      SUBROUTINE GRAD (IS)
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /VALUE/ F,G,PO,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
C-----IF (IS=1) ACCUM. MATRIX OF 2ND PARTIALS IF(IS=2) DONT
      GO TO (1,3), IS
1     DO 2 I=1,N
      DO 2 J=1,I
2     A(I,J)=0.
3     DO 4 I=1,N
4     DELXO(I)=0.
C-----THIS SECTION WORKS CORRECTLY IN FEASIBILITY PHASE AS WELL AS NORMA
      IPN=0
      GO TO (5,9), NT2
5     DO 8 I=1,N
      IF (X1(I)) 8,8,6
6     DELXO(I)=-RHO/X(I)**2
      GO TO (7,8), IS
7     A(I,I)=-2.*DELXO(I)/X(I)
8     CONTINUE
      IPN=N
9     DO 16 J=1,M
      K=IPN+J
      IF (RJ1(K)) 16,16,10
10    CALL GRAD1 (J)
      TT=RHO/RJ(K)**2
      DO 15 I=1,N
      IF (DEL(I)) 11,15,11
C-----IF DEL(I)=0 SKIP ALL THE FOLLOWING COMPUTATION INVOLVING * BY DEL(
11    T=TT*DEL(I)
      DELXO(I)=DELXO(I)-T
      GO TO (12,15), IS
12    T=2.*T/RJ(K)
      DO 14 JJ=1,I
      IF (DEL(JJ)) 13,14,13
13    A(I,JJ)=A(I,JJ)+T*DEL(JJ)
14    CONTINUE
15    CONTINUE
16    CONTINUE
      GO TO (17,19), IS
17    DO 18 I=1,N
18    DIAG(I)=A(I,I)
19    GO TO (20,27,29), NPHASE
20    NJ=NVI
      GO TO (21,25), NT2
21    IF (NVI.GT.N) 24,22
22    DO 23 I=1,N
23    DELXO(I)=-DELXO(I)
      DELXO(NVI)=DELXO(NVI)+1.0
      GO TO 29
24    NJ=NVI-N
25    CALL GRAD1 (NJ)
```

```
      DO 26 I=1,N
26    DELXO(I)=-DELXO(I)+DEL(I)
      GO TO 29
27    CALL GRAD1 (0)
      DO 28 I=1,N
28    DELXO(I)=-DELXO(I)-DEL(I)
C-----LEAVES THE NEG. GRAD OF P IN DELXO
29    RETURN
      END
```

```
      SUBROUTINE INVERS (NSME)
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      EQUIVALENCE (B,DELX)
      DIMENSION B(100)
C-----GETTING THE PRODUCT INVERSE USING THE CROUT METHOD AND TAKING ADVA
C-----AGE OF THE SYMMETRY OF THE A MATRIX
C-----IF A DIVISOR OF ZERO IS GENERATED, THEN DELX IS SET EQUAL TO DELXO
C-----IF NSME=1 HAVE NEW A MATRIX  IF NSME=2  COMPUTE INVERSE PRODUCT US
C-----A AS LEFT FROM LAST TIME.
      GO TO (2,1), NSME
1     GO TO (12,19), KSW
2     KSW=1
      IF (A(1,1)) 23,23,3
3     A(1,1)=1./A(1,1)
      DO 4 I=2,N
4     A(1,I)=A(1,I)*A(1,1)
      DO 11 J=2,N
      JM1=J-1
      T=0.
      DO 6 I=1,JM1
      IF (A(I,J)) 5,6,5
5     T=T+A(J,I)*A(I,J)
6     CONTINUE
      A(J,J)=A(J,J)-T
      IF (A(J,J)) 24,24,7
7     A(J,J)=1./A(J,J)
      IF (J.EQ.N) GO TO 12
      JP1=J+1
      DO 10 L=JP1,N
      T=0.
      DO 9 I=1,JM1
      IF (A(I,J)) 8,9,8
8     T=T+A(L,I)*A(I,J)
9     CONTINUE
      A(L,J)=A(L,J)-T
      A(J,L)=A(L,J)*A(J,J)
10    CONTINUE
11    CONTINUE
12    B(1)=B(1)*A(1,1)
      DO 15 J=2,N
      T=0.
      JM1=J-1
      DO 14 I=1,JM1
      IF (A(J,I)) 13,14,13
13    T=T+A(J,I)*B(I)
14    CONTINUE
      B(J)=(B(J)-T)*A(J,J)
15    CONTINUE
      DO 18 I=1,NM1
      NMK=N-I
      DO 17 J=1,I
      L=NP1-J
      IF (A(NMK,L)) 16,17,16
16    B(NMK)=B(NMK)-A(NMK,L)*B(L)
17    CONTINUE
18    CONTINUE
19    GO TO (22,20), NT3
20    WRITE (6,27)
      WRITE (6,26) (DELXO(I),I=1,N)
      GO TO (21,22), KSW
21    WRITE (6,28)
      WRITE (6,26) (DELX(I),I=1,N)
22    RETURN
23    J=1
24    WRITE (6,29) J,A(J,J)
      KSW=2
      DO 25 I=1,N
25    DELX(2)=DELXO(I)
      GO TO 19
C-----
26    FORMAT (7E17.8)
27    FORMAT (1H0,6X,12HDEL P VECTOR)
28    FORMAT (1H0,6X,24HSECOND ORDER MOVE VECTOR)
29    FORMAT(2H0  51HTHIS MIGHT NOT BE A CONVEX PROGRAMMING PROBLEM, THE
     1 I3,  16HTH DISCRIMINANT=   ,E15.7   )
      END
```

```
      SUBROUTINE MAG
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      ADELX=0.
      DO 1 I=1,N
```

```
1       ADELX=ADELX+DELXO(I)**2
        ADELX=SQRT(ADELX)
        RETURN
        END
```

```
        SUBROUTINE OPT
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
        COMMON /VALUE/ F,G,PO,RSIGMA,RJ(200),RHO
        COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
       1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
       2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
       3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
        REAL LRJ
        KSW=1
        DOTT=0.
        DO 1 J=1,N
1       DOTT=DOTT+DELX(J)*DELXO(J)
        IF (DOTT) 2,2,4
2       DO 3 I=1,N
3       DELX(I)=DELXO(I)
4       CONTINUE
        NTCTR=NTCTR+1
        DO 5 I=1,N
5       X2(I)=X(I)
        PX1=PO
        N401=0
6       N401=N401+1
        DO 7 I=1,N
7       X(I)=X2(I)+DELX(I)
        CALL EVALU
        CALL SPECFE
C-----1 MEANS SATIS.OF CONSTRAINT NT.PREV. 2MEANS NOCHANGE 3MEANS VIOLAT
C-----IF POINT IS NOT FEASIBLE GIVE IT AN ARBRITRARILY HIGH VALUE
        GO TO (46,9,8), NSATIS
8       PX2=10.E35
        PO=10.E35
        GO TO 10
9       CALL PEVALU
        PX2=PO
        IF (PX1-PX2) 10,10,15
10      IF (N401-2) 13,11,11
11      DO 12 I=1,N
12      X1(I)=X(I)
        P1=PX2
        GO TO 37
C-----
C-----ONLY ONE  POINT  SO FAR COMPUTED
13      DO 14 I=1,N
14      X3(I)=X2(I)
        PREV3=PX1
```

```
      GO TO 18
15    DO 16 I=1,N
      X3(I)=X2(I)
      X2(I)=X(I)
16    DELX(I)=1.61803399*DELX(I)
      PREV3=PX1
      PX1=PX2
      GO TO 6
C-----FIBONACCI   METHOD
C-----B VECTOR  GOES TO X1(I)
17    P0=1.E36
18    DO 19 I=1,N
19    X1(I)=X(I)
      P1=P0
      DO 20 I=1,N
      X(I)=.38196601*(X1(I)-X3(I))+X3(I)
20    X2(I)=X(I)
      CALL EVALU
      CALL SPECFE
      GO TO (46,21,17), NSATIS
21    CALL PEVALU
      PX1=P0
      DO 22 I=1,N
22    X(I)=0.38196601*(X1(I)-X2(I))+X2(I)
      CALL EVALU
      CALL SPECFE
      GO TO (46,23,17), NSATIS
23    CALL PEVALU
      PX2=P0
      N401=1
24    N401=N401+1
      IF (N401-25) 28,25,25
25    KSW=2
      IF (N401-40) 26,40,40
26    DO 27 I=1,N
      IF (ABS(X2(I)/X(I)-1.0).GE.1.E-7) 28,27
27    CONTINUE
      GO TO 40
28    IF (ABS(PX1/PX2-1.).LE.1.E-7) 40,29
29    IF (PX1-PX2) 30,40,35
C-----FROM LEFTORIGHT  X3(I)(PREV3)X2(I)(PX1)X(I)PX2 X1(I)P1
30    DO 31 I=1,N
31    X1(I)=X(I)
C-----THROW AWAY RIGHT PART
      P1=PX2
      DO 32 I=1,N
C-----POINTXP1 BECOMES XP2
32    X(I)=.38196601*(X1(I)-X3(I))+X3(I)
C-----TEMPORARILY IN  X STORAGE
      CALL EVALU
      CALL SPECFE
      GO TO (46,33,17), NSATIS
33    CALL PEVALU
      PX2=PX1
C-----SWITCH VECTORS TO PROPER POSITION
      PX1=P0
      DO 34 I=1,N
      XX=X2(I)
      X2(I)=X(I)
34    X(I)=XX
      GO TO 24
C-----LEFT SIDE  TOSSED  AWAY
35    DO 36 I=1,N
      X3(I)=X2(I)
36    X2(I)=X(I)
      PREV3=PX1
      PX1=PX2
37    DO 38 I=1,N
38    X(I)=0.38196601*(X1(I)-X2(I))+X2(I)
      CALL EVALU
      CALL SPECFE
      GO TO (46,39,17), NSATIS
39    CONTINUE
      CALL PEVALU
      PX2=P0
      GO TO 24
C-----FIBONNACCI  POINTS HAVE EQUAL VALUE NOW COMPUTE MIDPOINT
40    DO 41 I=1,N
      DELXO(I)=X(I)
41    X(I)=(DELXO(I)+X2(I))*0.5
      PP1=PX2
      CALL EVALU
      CALL PEVALU
      GO TO (42,43), KSW
42    IF (ABS(P0/PX1-1.).GT.1.E-7) 44,43
C-----NOTE POSSIBLE ACCUM  SECOND PARTIALS HERE ALSO
43    CALL GRAD (2)
      RETURN
44    DO 45 I=1,N
45    X(I)=DELXO(I)
      GO TO 30
46    RETURN
C-----NOT YET CODED FOR PHASE=3
      END
```

```
      SUBROUTINE OUTPUT (K)
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200),NSOL
      REAL LRJ
      GO TO (1,5), K
1     WRITE (6,9)
      WRITE (6,10) NTCTR,DOTT,RHO,ADELX,NPHASE
      IF (NPHASE.NE.1) 5,2
2     NVII=NVI
      GO TO (3,4), NT2
3     NVII=NVI-N
4     WRITE (6,14) NVII
5     WRITE (6,11) F,P0,G,RSIGMA
      WRITE (6,12) (X(I),I=1,N)
      WRITE (6,15)
      GO TO (6,7), NT2
6     WRITE (6,16)
      WRITE (6,13) (RJ(I),I=NP1,MN)
      GO TO 8
7     WRITE (6,13) (RJ(I),I=1,MN)
8     NSOL=NSOL+1
      IF(NSOL.LT.5) RETURN
      NSOL=0
      WRITE(6,17)
      RETURN
C-----
9     FORMAT (50H0*********************************************      )
10    FORMAT (10X,6HPOINT=I4,6X,6H DOTT=1PE15.7,6X,4HRHO=E15.7,6X,10HMAG
     1NITUDE=E15.7,6X,6HPHASE=I2)
11    FORMAT (10X,2HF=1PE15.7,6X,2HP=E15.7,6X,2HG=E15.7,6X,7HRSIGMA=E15.
     17)
12    FORMAT (6X,25HTHE CURRENT VALUE OF X IS/(1P6F20.7))
13    FORMAT (1P6E20.7)
14    FORMAT (1H ,117X,6X,4HNVI=I3)
15    FORMAT (6X,21HTHE CONSTRAINT VALUES)
16    FORMAT (1H ,27X,34HNOT INCLUDING THE NON-NEGATIVITIES)
17    FORMAT(*1NONLINEAR PROGRAMMING ROUTINE-RAC*///)
      END
```

```
      SUBROUTINE PEVALU
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETA0,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      GO TO (5,1,4), NPHASE
1     RSIGMA=0.
      DO 2 I=1,MN
2     RSIGMA=RSIGMA+RHO/RJ(I)
3     P0=RSIGMA+F
      G=F-RSIGMA
4     RETURN
5     RSIGMA=0.
      DO 7 I=1,MN
      IF (RJ1(I)) 7,7,6
```

```
6       RSIGMA=RSIGMA+RHO/RJ(I)
7       CONTINUE
        GO TO 3
        END
```

```
        SUBROUTINE REJECT
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
        COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
        COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
       1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
       2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
       3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
        REAL LRJ
1       DO 2 I=1,N
2       X(I)=X1(I)
        DO 3 J=1,MN
3       RJ(J)=RJ1(J)
        P0=P1
        RSIGMA=RSIG1
        G=G1
        F=F1
        RETURN
        END
```

```
        SUBROUTINE REJCT1
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
        COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
        COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
       1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
       2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
       3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
        REAL LRJ
        DO 1 I=1,N
1       X(I)=X3(I)
        DO 2 J=1,MN
2       RJ(J)=RJ3(J)
        P0=PREV3
        G=G3
        RSIGMA=RSIG3
        F=F3
        RETURN
        END
```

```
      SUBROUTINE SPECFE                                                 SUM
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1          SUM
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO                          SUM
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,NSUM
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(SUM
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3SUM
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)                                 SUM
      REAL LRJ                                                          SUM
C-----3  IF NOT 'FEAS'  . 2 IF NO CHANGE IN VIOLS+SATIS.   1 SATIS  IORMSUM
      NSATIS=2                                                          SUM
      DO 4 J=1,MN                                                       SUM
      IF (RJ1(J)) 2,2,1                                                 SUM
1     IF (RJ(J)) 5,5,4                                                  SUM
2     IF (RJ(J)) 4,4,3                                                  SUM
3     NSATIS=1                                                          SUM
4     CONTINUE                                                          SUM
      RETURN                                                            SUM
5     NSATIS=3                                                          SUM
      RETURN                                                            SUM
      END                                                               SUM
```

```
      SUBROUTINE RHOCOM                                                 SUM
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1          SUM
      COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12 SUM
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO                          SUM
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,NSUM
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(SUM
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3SUM
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)                                 SUM
      REAL LRJ                                                          SUM
C-----SUBROUTINE  TO COMPUTE INITIAL RHO VALUE                          SUM
C----- CONTROLLED BY COL. 7 ON OTION CARD                               SUM
      CALL STORE                                                        SUM
      GO TO (11,5,1,19), NT1                                            SUM
1     RHO=RHOIN                                                         SUM
2     IF (RHO) 3,3,4                                                    SUM
3     RHO=1.                                                            SUM
4     RETURN                                                            SUM
5     NPAR1=1                                                           SUM
6     RHO=1.                                                            SUM
C-----2 MEANS  RHO WHICH  MINIMIZES GRADIENT  MAG.                      SUM
      CALL GRAD (2)                                                     SUM
      CALL GRAD1 (0)                                                    SUM
C-----TO GET -DEL F                                                     SUM
      DO 7 I=1,N                                                        SUM
7     PGRAD(I)=-DEL(I)                                                  SUM
      DO 8 I=1,N                                                        SUM
      DELXO(I)=DELXO(I)-PGRAD(I)                                        SUM
C-----THIS LEAVES DEL SIGMA IN DELXO                                    SUM
8     CONTINUE                                                          SUM
      GO TO (9,13), NPAR1                                               SUM
```

```
9       DOT1=0.
        DOT2=0.
        DO 10 I=1,N
        DOT1=DOT1+DELXO(I)*PGRAD(I)
10      DOT2=DOT2+DELXO(I)**2
        RHO=ABS(DOT1/DOT2)
        GO TO 2
C------3 MEANS COMPUTE RHO SO AS TO MINIMIZE DEL P(/DDP/1.)DEL P
11      NPAR2=1
12      NPAR1=2
C------USE  DF  AND DR  SUBROUTINE
        GO TO 6
13      RHO=1.
C------ASSUME   SIGMA TERM IS CONSID. GRTER THAN  F TERM
        CALL SECORD (2)
        DO 14 I=1,N
14      DELX(I)=PGRAD(I)
        CALL INVERS (1)
        DO 15 I=1,N
        X3(I)=DELX(I)
15      DELX(I)=DELXO(I)
        CALL INVERS (2)
        DO 16 I=1,N
16      XR2(I)=DELX(I)
        GO TO (17,20), NPAR2
17      DOT1=0.
        DOT2=0.
        DO 18 I=1,N
        DOT1=DOT1+PGRAD(I)*X3(I)
18      DOT2=DOT2+DELXO(I)*XR2(I)
        RHO=SQRT(ABS(DOT1/DOT2))
        GO TO 2
19      NPAR2=2
C------RHO MINIMIZES  2ND ORDER MOVE
        GO TO 12
C------USES  INTERNAL  SUB. TO COM  /DDP/-1 DF AND  /DDP/- DR
20      DOT1=0.
        DOT2=0
        DO 21 I=1,N
        DOT1=X3(I)**2+DOT1
21      DOT2=X3(I)*XR2(I)+DOT2
        RHO=ABS(DOT1/DOT2)
        GO TO 2
        END
```

```
        SUBROUTINE SECORD (IS)
        COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
        COMMON /OPTNS/ NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,NT9,NT10,NT11,NT12
        COMMON /VALUE/ F,G,PO,RSIGMA,RJ(200),RHO
        COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
       1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      IPN=0
      GO TO (1,2), NT2
1     IPN=N
2     GO TO (3,5), IS
3     DO 4 I=1,N
      DO 4 J=I,N
4     A(I,J)=0.
      GO TO 18
5     DO 6 I=1,N
      DO 6 J=1,N
6     A(I,J)=0.
C-----GRAD. TERM NOT PREV. COMPUTED
      GO TO (7,10), NT2
7     DO 9 I=1,N
      IF (X1(I)) 9,9,8
8     A(I,I)=2.*RHO/RJ(I)**3
9     CONTINUE
10    DO 16 KK=1,M
      K=IPN+KK
      IF (RJ1(K)) 16,16,11
11    CALL GRADI (KK)
      DO 15 I=1,N
      IF (DEL(I)) 12,15,12
12    T=(2.*RHO/RJ(K)**3)*DEL(I)
      DO 14 J=1,I
      IF (DEL(J)) 13,14,13
13    A(I,J)=A(I,J)+T*DEL(J)
14    CONTINUE
15    CONTINUE
16    CONTINUE
      DO 17 I=1,N
      DIAG(I)=A(I,I)
17    A(I,I)=0.
C-----READY NOW FOR MATRIX OF 2ND PARTIALS OF RESTRAINTS
18    GO TO (23,19,20), NT10
C-----BY PASS COMPUTING SECOND PARITIALS WHEN THEY ARE KNOWN TO BE ZERO
19    GO TO (20,38,38), NPHASE
20    DO 21 I=2,N
      IM1=I-1
      DO 21 J=1,IM1
21    A(J,I)=A(I,J)
      DO 22 I=1,N
22    A(I,I)=DIAG(I)
      GO TO 44
23    DO 28 KK=1,M
      K=IPN+KK
      CALL MATRIX (KK)
      T=-RHO/RJ(K)**2
      DO 25 I=2,N
      IM1=I-1
      DO 25 J=1,IM1
      IF (A(J,I)) 24,25,24
24    A(I,J)=A(I,J)+T*A(J,I)
```

```
      A(J,I)=0.
25    CONTINUE
      DO 27 I=1,N
      IF (A(I,I)) 26,27,26
      A(I,I)=0.
26    DIAG(I)=DIAG(I)+T*A(I,I)
27    CONTINUE
28    CONTINUE
      GO TO (29,38,38), NPHASE
C-----FOR NPHASE=3 ADDITIONS WILL BE ADDED LATER
29    NJ=NVI
      GO TO (30,32), NT2
30    IF (NVI-N) 44,44,31
31    NJ=NVI-N
32    CALL MATRIX (NJ)
      DO 34 I=2,N
      IM1=I-1
      DO 34 J=1,IM1
      IF (A(J,I)) 33,34,33
33    A(I,J)=A(I,J)-A(J,I)
34    A(J,I)=A(I,J)
      DO 37 I=1,N
      IF (A(I,I)) 35,36,35
35    A(I,I)=DIAG(I)-A(I,I)
      GO TO 37
36    A(I,I)=DIAG(I)
37    CONTINUE
      GO TO 44
C-----GET MATRIX OF 2ND PARTIALS OF OBJECTIVE FUNCTION
38    CALL MATRIX (0)
      DO 40 I=2,N
      IM1=I-1
      DO 40 J=1,IM1
      IF (A(J,I)) 39,40,39
39    A(I,J)=A(I,J)+A(J,I)
40    A(J,I)=A(I,J)
      DO 43 I=1,N
      IF (A(I,I)) 41,42,41
41    A(I,I)=DIAG(I)+A(I,I)
      GO TO 43
42    A(I,I)=DIAG(I)
43    CONTINUE
44    RETURN
      END
```

```
      SUBROUTINE STORE
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /VALUE/ F,G,PO,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELXO(100),RHOIN,RATIO,EPSI,THETAO,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),COTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
1     DO 2 I=1,N
2     X1(I)=X(I)
      DO 3 J=1,MN
3     RJ1(J)=RJ(J)
      P1=P0
      F1=F
      G1=G
      RSIG1=RSIGMA
      RETURN
      END
```

```
      SUBROUTINE STORE1
      COMMON /SHARE/ X(100),DEL(100),A(100,100),N,M,MN,NP1,NM1
      COMMON /VALUE/ F,G,P0,RSIGMA,RJ(200),RHO
      COMMON /CRST/ DELX(100),DELX0(100),RHOIN,RATIO,EPSI,THETA0,NTCTR,N
     1UMINI,X1(100),X2(100),X3(100),XR2(100),XR1(100),PR1,PR2,P1,F1,RJ1(
     2200),DOTT,PGRAD(100),DIAG(100),RJ3(200),PREV3,F3,ADELX,RSIG1,G1,G3
     3,RSIG3,NVI,NPHASE,NSATIS,LRJ(200)
      REAL LRJ
      DO 1 I=1,N
1     X3(I)=X(I)
      DO 2 J=1,MN
2     RJ3(J)=RJ(J)
      PREV3=P0
      F3=F
      RSIG3=RSIGMA
      G3=G
      RETURN
      END
```

```
      SUBROUTINE TIMEC
C-----CALLS SUBROUTINE TIME AND WRITES OUT ELAPSED TIME IN SECONDS.
      RETURN
      END
```

```
      SUBROUTINE TIME (X)
C-----THE ELAPSED TIME SINCE CALLING SET IS CONVERTED TO THE NUMBER
C-----OF SECONDS IN FLOATING POINT FORM AND STORED IN LOCATION X.
C-----TERMINATES PRESENT PROBLEM WHEN TMAX IS EXCEEDED.
```

```
      RETURN
      END



      SUBROUTINE SIT (TMAX)
C-----WHERE TMAX IS THE ADRESS OF A FLOATING POINT NUMBER GIVING THE
C-----MAXIMUM NUMBER OF SECONDS THE PROGRAM IS TO BE ALLOWED TO RUN
C-----UNTIL TERMINATION OR ANOTHER CALL TO SET.
      RETURN
      END
```

```
     1.0E-05       2.0E+02       1.0E-06       3.2E+01       2.4E+00    7    7
     3.0E+01       3.0E+01       3.0E+01       3.0E+01       3.0E+01          3.0E+01
     3.0E+01
SEVEN DIETARY BOTANICAL COMPONENTS AND GROSS ENERGY  PERIOD 2 SERIES 2.
 1H   7
   58.50     6.00    23.50     5.00     0.50     0.50    6.004674.189
   68.00     8.50     9.50     6.00     0.50     4.00    3.504732.462
   69.50    13.00     3.50     5.50     0.50     0.50    7.504729.882
   67.50    11.00     7.00     5.50     1.00     1.50    6.504682.4591
   71.50     9.00     5.50     2.50     0.50     0.50   10.504840.8710
   68.50    12.00     3.00     8.00     1.00     2.50    5.004751.6779
   53.00     8.50    14.50     8.50     0.50     1.00   14.004739.6927
   48.00     5.50     6.50     9.00     0.50     0.50   30.004761.5307
   72.50     8.50     1.50     8.00     0.50     3.50    5.504681.1394
   58.30     6.50     7.50    10.50     0.50     5.50   11.204719.1140
   45.00    11.50    16.00     9.50     1.50     1.50   15.004701.8092
   55.00    10.00     6.50    10.50     0.50     4.50   13.004780.6005
   74.00     7.00     5.00     6.50     0.50     1.00    6.004732.0721
   65.50    12.50     5.50     5.00     1.00     1.00    9.504755.1877
   58.00     8.50    14.50     7.50     0.50     1.50    9.504733.2807
   56.00    10.00    15.50     7.50     0.50     2.50    8.004683.1026
   45.70     5.70    12.40    10.50     1.00    16.60    8.104771.3882
   65.50     7.00    12.50     7.50     0.50     3.50    3.504722.6027
     3        1        1        1        2        1        1        1        1        2
```

```
SEVEN DIETARY BOTANICAL COMPONENTS AND GROSS ENERGY  PERIOD 2 SERIES 2.
  18       7      FILE NO.  1
   1  58.500   6.000  23.500   5.000    .500    .500  6.0004674.189
   2  68.000   8.500   9.500   6.000    .500   4.000  3.5004732.462
   3  69.500  13.000   3.500   5.500    .500    .500  7.5004729.882
   4  67.500  11.000   7.000   5.500   1.000   1.500  6.5004682.459
   5  71.500   9.000   5.500   2.500    .500    .500 10.5004840.871
   6  68.500  12.000   3.000   8.000   1.000   2.500  5.0004751.677
   7  53.000   8.500  14.500   8.500    .500   1.000 14.0004739.692
   8  48.000   5.500   6.500   9.000    .500    .500 30.0004761.530
   9  72.500   8.500   1.500   8.000    .500   3.500  5.5004681.139
  10  58.300   6.500   7.500  10.500    .500   5.500 11.2004719.114
  11  45.000  11.500  16.000   9.500   1.500   1.500 15.0004701.809
  12  55.000  10.000   6.500  10.500    .500   4.500 13.0004780.600
  13  74.000   7.000   5.000   6.500    .500   1.000  6.0004732.072
  14  65.500  12.500   5.500   5.000   1.000   1.000  9.5004755.187
  15  59.000   9.500  14.500   7.500    .500   1.500  9.5004733.280
  16  56.000  10.000  15.500   7.500    .500   2.500  8.0004683.102
  17  45.700   5.700  12.400  10.500   1.000  16.600  8.1004771.388
  18  65.500   7.000  12.500   7.500    .500   3.500  3.5004722.602
```

```
NONLINEAR PROGRAMMING ROUTINE-RAC

    N=  7        M=  7

      MAX. TIME= 2.400E+00    R= 2.0000000E+02    RATIO= 3.200E+01      EPSILON= 1.000E-05    THETA= 1.000E-06

OPTIONS SELECTED
    3       1        1        1        2        1        1        1        1        2
        F=  5.4085479E+07      P=  0.                        G=  0.                       RSIGMA=  0.
    THE CURRENT VALUE OF X IS
     3.0000000E+01       3.0000000E+01        3.0000000E+01        3.0000000E+01        3.0000000E+01        3.0000000E+01
     3.0000000E+01
    THE CONSTRAINT VALUES
                          NOT INCLUDING THE NON-NEGATIVITIES
     3.0000000E+01       3.0000000E+01        3.0000000E+01        3.0000000E+01        3.0000000E+01        3.0000000E+01
     3.0000000E+01

********************************************
        POINT=    5       DOTT=  4.1286320E-06      RHO=  2.0000000E+02      MAGNITUDE=  1.0354811E-02      PHASE= 2
        F=  1.5265721E+04       P=  1.5341382E+04       G=  1.5190060E+04       RSIGMA=  7.5661042E+01
    THE CURRENT VALUE OF X IS
     4.7520417E+01       5.1611689E+01        4.5633755E+01        3.6065173E+01        1.6139895E+01        5.5718522E+01
     5.2119295E+01
    THE CONSTRAINT VALUES
                          NOT INCLUDING THE NON-NEGATIVITIES
     4.7520417E+01       5.1611689E+01        4.5633755E+01        3.6065173E+01        1.6139895E+01        5.5718522E+01
     5.2119295E+01

********************************************
        POINT=    9       DOTT=  3.7880604E-06      RHO=  6.2500000E+00      MAGNITUDE=  8.0414981E-04      PHASE= 2
        F=  1.5265002E+04       P=  1.5267412E+04       G=  1.5262591E+04       RSIGMA=  2.4104558E+00
    THE CURRENT VALUE OF X IS
     4.7515238E+01       5.1695085E+01        4.5642378E+01        3.6056356E+01        1.5224960E+01        5.5750905E+01
     5.2126733E+01
    THE CONSTRAINT VALUES
                          NOT INCLUDING THE NON-NEGATIVITIES
     4.7515238E+01       5.1695085E+01        4.5642378E+01        3.6056356E+01        1.5224960E+01        5.5750905E+01
     5.2126733E+01

********************************************
        POINT=   13       DOTT=  1.4993889E-06      RHO=  1.9531250E-01      MAGNITUDE=  5.0124655E-04      PHASE= 2
        F=  1.5265001E+04       P=  1.5265076E+04       G=  1.5264925E+04       RSIGMA=  7.5381102E-02
    THE CURRENT VALUE OF X IS
     4.7515054E+01       5.1698050E+01        4.5642684E+01        3.6056050E+01        1.5192447E+01        5.5752053E+01
     5.2126995E+01
    THE CONSTRAINT VALUES
                          NOT INCLUDING THE NON-NEGATIVITIES
     4.7515054E+01       5.1698050E+01        4.5642684E+01        3.6056050E+01        1.5192447E+01        5.5752053E+01
     5.2126995E+01

********************************************
        POINT=   14       DOTT=  2.7621047E-06      RHO=  6.1035156E-03      MAGNITUDE=  6.8005849E-04      PHASE= 2
        F=  1.5265001E+04       P=  1.5265003E+04       G=  1.5264998E+04       RSIGMA=  2.3557495E-03
    THE CURRENT VALUE OF X IS
     4.7515044E+01       5.1698207E+01        4.5642700E+01        3.6056034E+01        1.5190727E+01        5.5752114E+01
     5.2127009E+01
    THE CONSTRAINT VALUES
                          NOT INCLUDING THE NON-NEGATIVITIES
     4.7515044E+01       5.1698207E+01        4.5642700E+01        3.6056034E+01        1.5190727E+01        5.5752114E+01
     5.2127009E+01
```

```
NONLINEAR PROGRAMMING ROUTINE-RAC


*********************************************
          POINT=  15       DOTT=  2.2415907E-07       RHO=  1.9073486E-04       MAGNITUDE=  1.9371209E-04       PHASE= 2
          F=  1.5265001E+04       P=  1.5265001E+04       G=  1.5265001E+04       RSIGMA=  7.3616370E-05
     THE CURRENT VALUE OF X IS
      4.7515347E+01        5.1698162E+01        4.5642696E+01        3.6056039E+01        1.5191217E+01        5.5752097E+01
      5.2127003E+01
     THE CONSTRAINT VALUES
                         NOT INCLUDING THE NON-NEGATIVITIES
      4.7515347E+01        5.1698162E+01        4.5642696E+01        3.6056039E+01        1.5191217E+01        5.5752097E+01
      5.2127003E+01

*********************************************
          POINT=  16       DOTT=  2.1773704F-08       RHO=  5.9604645E-06       MAGNITUDE=  6.0381074E-05       PHASE= 2
          F=  1.5265001E+04       P=  1.5265001E+04       G=  1.5265001E+04       RSIGMA=  2.3005134E-06
     THE CURRENT VALUE OF X IS
      4.7515046E+01        5.1698176E+01        4.5642697E+01        3.6056037E+01        1.5191064E+01        5.5752102E+01
      5.2127006E+01
     THE CONSTRAINT VALUES
                         NOT INCLUDING THE NON-NEGATIVITIES
      4.7515046E+01        5.1698176E+01        4.5642697E+01        3.6056037E+01        1.5191064E+01        5.5752102E+01
      5.2127006E+01

*********************************************
          POINT=  17       DOTT=  2.0756337E-09       RHO=  1.8626451E-07       MAGNITUDE=  1.8631889E-05       PHASE= 2
          F=  1.5265001E+04       P=  1.5265001E+04       G=  1.5265001E+04       RSIGMA=  7.1891155E-08
     THE CURRENT VALUE OF X IS
      4.7515046E+01        5.1698172E+01        4.5642697E+01        3.6056038E+01        1.5191112E+01        5.5752100E+01
      5.2127006E+01
     THE CONSTRAINT VALUES
                         NOT INCLUDING THE NON-NEGATIVITIES
      4.7515046E+01        5.1698172E+01        4.5642697E+01        3.6056038F+01        1.5191112E+01        5.5752100E+01
      5.2127006E+01
```

and careful thought of each application of the program is necessary if correct results are to be obtained. For an in depth discussion of this program, its development and methods of computation, refer to *Evaluation of a digital computer method for analysis of compartmental models of ecological systems*, L. J. Bledsoe and G. M. Van Dyne, Oak Ridge National Laboratory, Oak Ridge, Tennessee, ORNL-TM-2414, February 1969. This abstract describes a sample problem and illustrates its solution along with a description of the control cards required and the input format of the data.

Program COMSYS2

A FORTRAN program developed at Colorado State University using a CDC 6400 computer using the SCOPE version 2.0 FORTRAN extended compiler.

This program was developed to find the coefficients of a set of linear homogeneous differential equations. The compartment model, characterized by these differential equations, is one of the more general mathematical models available to the systems biologist attempting to explain multivariable data taken from a complex system. A compartment model for a biological system consists of an abstraction in which the system is viewed as a series of discrete compartments between which flow energy or material. Flow between compartments is characterized by first-order differential equations. The net flow associated with each compartment is given as a function of time and as a function of the contents of the other compartments. At least as a first approximation, it is useful to think of these differential equations as being linear and homogeneous. This linear differential equation system is,

$$V_j' = \sum_{i=1}^{n} V_i \cdot f_{ij},$$

where n is the number of compartments in the system,

v_i is a function of time and represents the contents of compartment i,

v_j' is the derivative of the contents of the jth compartment,

f_{ij} is the ith row and jth column entry in a matrix f, representing the constant of proportionality associated with the flow from compartment i to compartment j.

It is the purpose of this program, given some observed values for v_i through time to determine the f_{ij}. The program will handle up to five compartments and as many data points through time as desired. The use of the program is complex

INTRODUCTION

Program LINDIFF is a FORTRAN program developed at Colorado State University using a CDC 6400 computer with the SCOPE version 2.0 FORTRAN extended compiler. The program is also operational using the NCAR compiler on the CDC 6400.

Program LINDIFF solves a set of linear differential equations with constant coefficients of the form:

$$y_i' = \sum_{j=1}^{n} a_{ij} \cdot y_j.$$

The solution of the above set of equations amounts to determining $y_i (i=1,\ldots,n)$ for an interval of interest along the independent variable. The integration scheme is of the Runge-Kutta type and requires initial values of the dependent variable to start the solution. The matrix $[a_{ij}]$ is read in and are the constants in the equations. Although the program is designed for solution of the above type equations, the Runge-Kutta integration routine is somewhat more general and may be easily adapted to other problems. By changing subroutine EQUA which evaluates the derivatives and changing the input scheme appropriately all equations of the form,

$$y' = f(x,y)$$

may be solved. Also since each ordinary differential equation of the second or higher order is equivalent to a system of first order equations, these also admit to solution by Runge-Kutta methods.

The program will solve up to 20 equations in 20 unknowns although this is easily modified to handle longer systems by changing the appropriate dimensions and is subject only to computer memory requirements.

PROGRAM OPERATION

The main program reads in the control cards, and the data, and produces the output. It also sets up and increments the independent variable (usually time) by the amounts to be outputed. For each increment subroutine KUTTA is called which integrates along the independent variable to the new increment point. Subroutine KUTTA in turn calls subroutine EQ1 which evaluates the derivatives for current values of the independent variable by means of matrix multiplication. The coefficients of the differential equations are passed to EQ1 in common, and EQ1 is external in the main program and called by the name EQUA in subroutine KUTTA. In the absence of round-off errors, truncation errors due to a stepwise approximation of continuous integration can become significant. The Runge-Kutta routine will alter the increment size read into the program such that the accuracy which is also input to the program is maintained. Since the Runge-Kutta method employed is of the 4th order, the local truncation errors are of the order of the increment size (within KUTTA) raised to the 4th power. A rough measure of local truncation errors (variable E(I) in KUTTA) is used to adjust if necessary the step size. Also, the starting solution should be computed more accurately than the required solution by at least a factor of ten. Since the program ouputs the integrated values of the dependent variables over a range of the independent variable at fixed intervals some care should be taken to insure that the desired information is output but that the interval size is not so small that the program takes an excessively long time. A specified accuracy of more than one part in 10 to the 12th is futile (unless conversion to double precision is made) due to round off errors in the intermediate calculations, and especially in excessively long intervals when error propagation becomes significant.

Input-Control Card Requirements

Columns	Format	Variable	
I. First Control Card			
1-5	I5	NCOM	No. of equations
6-10	I5	NT	No. of independent variable points.
11-20	F10.3	TI	Initial indirect variable value
21-30	F10.3	TDEL	Indirect variable step value for output
31-40	A4	NSTOP	Flag for halting problem (if "STOP" - no more problems to follow)
II. Second Control Card			
1-10	F10.3	ACC	Desired accuracy of solution
11-20	F10.3	DEL	Estimate of time step required for integration
21-25	I5	ITER	Maximum no. of iterations to be allowed in integration
III. I/O Format Card			
1-80	8A10	FMT	Format statement by which the initial conditions and matrix of coefficient values are read in and outputed.
IV. Data Cards			
--	--	V(I)	Vector of initial conditions (NCOM in number)
--	--	F(I,J)	Matrix of coefficient values (NCOM x NCOM in number)

As many sets of control cards with data may be input to the program in sequence, i.e., I through IV may be repeated as many times as desired as long as a card with the letters STOP in columns 31-34 follows the last set of data. Since V(I) and F(I,J) are read with the same format, allowances must be made such that both variables are read properly. V(I) and F(I,J) are read by two different read statements using the same variable format.

```
      PROGRAM LINDIFF
     -(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     SOLVES A SET OF LINEAR DIFFERENTIAL EQUATIONS
C     NCOM      NO OF EQUATIONS
C     NT        NO OF TIME POINTS
C     T1        INITIAL TIME VALUE
C     TDEL      TIME STEP FOR PRINTOUT
C     ACC       DESIRED ACCURACY IN SOLUTION
C     DEL       ESTIMATE OF TIME STEP  TO BE USED FOR INTEGRATION
C     ITER      NO OF ITERATIONS, MAXIMUM, TO BE ALLOWED IN INTEGRATION
C     FMT       FORMAT STATEMENT TO BE USED TO READ INITIAL CONDITIONS
C               AND MATRIX OF COEFFICIENT VALUES
C     V         VECTOR OF INITIAL CONDITION VALUES
C     F         MATRIX OF COEFFICIENT VALUES
      DIMENSION FMT(8),V(20)
      COMMON/L1/F(20,20),NCOM
      EXTERNAL EQ1
C     READ AND PRINT INPUT DATA
    1 READ(5,100) NCOM,NT,T1,TDEL,NSTOP,ACC,DEL,ITER
      IF(NSTOP.EQ.4HSTOP) GO TO 20
      WRITE(6,200) T1,TDEL,NT,NCOM,ACC,DEL,ITER
      READ(5,300)FMT
      READ(5,FMT) (V(I),I=1,NCOM)
      READ(5,FMT) ((F(I,J),J=1,NCOM),I=1,NCOM)
      WRITE(6,FMT) (V(I),I=1,NCOM)
      WRITE(6,600)
      WRITE(6,FMT) ((F(I,J),J=1,NCOM),I=1,NCOM)
C     SET UP OUTPUT FORMAT
      N=NCOM+1
      ENCODE(15,400,FMT) N
  400 FORMAT(*(F20.4,*I2*F12.4)*)
      WRITE(6,500)
C     ITERATE THROUGH TIME AND FIND EQUATION SOLUTION
      T=T1-TDEL
      DO 10 I=1,NT
      T=T+TDEL
      T2=T+TDEL
      WRITE(6,FMT) T,(V(J),J=1,NCOM)
      CALL KUTTA(T,T2,V,NCOM,DEL,ACC,ITER,EQ1)
   10 CONTINUE
      GO TO 1
   20 STOP
  100 FORMAT(2I5,2F10.3,A4/2F10.3,I5)
  200 FORMAT(*1LINEAR DIFFERENTIAL EQUATION SOLUTION*//* TIME STARTS AT
     1*F10.3*, INCREMENTS BY *F10.3*, FOR  *I6* ITERATIONS*//* *
     2I6* COMPARTMENT SYSTEM*//* ACCURACY OF SOLUTION*E15.4/* TIME STEP
     3SIZE*E15.4/* MAXIMUM ITERATIONS/STEP*I6//* INITIAL VALUES AND MATR
     4IX*//)
  300 FORMAT(8A10)
  500 FORMAT(*1*//*          TIME, COMPARTMENT VALUES, IN ORDER*//)
  600 FORMAT(//)
      END
      SUBROUTINE EQ1(X,Z,ZP)
C     FINDS DERIVATIVES FOR A SET OF LINEAR DIFF. EQS.
C     X -- CURRENT TIME, Z -- CURRENT DEP VAR VALUE, ZP -- CURRENT
C     DERIVATIVE VALUES TO BE FOUND BY MATRIX MULTIPLICATION
      COMMON/L1/LAMBDA(20,20),N
      REAL LAMBDA
      DIMENSION Z(1),ZP(1)
      DO 10 I=1,N
      ZP(I)=0.
      DO 10 J=1,N
   10 ZP(I)=ZP(I)+LAMBDA(J,I)*Z(J)
      RETURN
      END
      SUBROUTINE KUTTA(XL,XU,Y,NE,DEL,ACCURC,IMAX,EQUA)
C PROGRAM AUTHOR F.D.HAMMERLING, CENTRAL DATA PROCESSING, ORGDP.
      DIMENSION Y(1),YI(25),YN(25),K1(25),K2(25),K3(25),
     1          K4(25),K5(25),F(25),E(25),F1(25)
      REAL K1,K2,K3,K4,K5
      LOGICAL QUIT
      ITTER=0
      N=NE
      XN=XL
      H=DEL
      QUIT=.FALSE.
      DO 1 I=1,N
    1 YN(I)=Y(I)
    2 IF(XN+H.LT.XU)GO TO 3
      DEL=H
      H=XU-XN
      QUIT=.TRUE.
    3 CALL EQUA(XN,YN,F1)
    4 DO 5 I=1,N
      K1(I)=H*F1(I)/3.
    5 YI(I)=YN(I)+K1(I)
      CALL EQUA(XN+H/3.,YI,F)
      DO 6 I=1,N
      K2(I)=H*F(I)/3.
    6 YI(I)=YN(I)+K1(I)/2.+K2(I)/2.
      CALL EQUA(XN+H/3.,YI,F)
      DO 7 I=1,N
      K3(I)=H*F(I)/3.
    7 YI(I)=YN(I)+3.*K1(I)/8.+9.*K3(I)/8.
      CALL EQUA(XN+H/2.,YI,F)
      DO 8 I=1,N
      K4(I)=H*F(I)/3.
    8 YI(I)=YN(I)+3.*K1(I)/2.-9.*K3(I)/2.+6.*K4(I)
      CALL EQUA(XN+H,YI,F)
      TEST=0.0
      DO 9 I=1,N
      K5(I)=H*F(I)/3.
      E(I)=(K1(I)-9.*K3(I)/2.+4.*K4(I)-K5(I)/2.)/5.
      TEST=AMAX1(TEST,ABS(E(I)))
    9 CONTINUE
      IF(TEST.LT.ACCURC)GO TO 10
      ITTER=ITTER+1
      IF(ITTER.GE.IMAX)GO TO 13
      H=H/2.
      QUIT=.FALSE.
      GO TO 4
   10 DO 11 I=1,N
   11 YN(I)=YN(I)+(K1(I)+4.*K4(I)+K5(I))/2.
      XN=XN+H
      IF(TEST.LT.ACCURC/32.)H=2.*H
      IF(.NOT.QUIT)GO TO 2
      DO 12 I=1,N
   12 Y(I)=YN(I)
      IMAX=0.
      GO TO 14
   13 DEL=H $ IMAX=ITTER
   14 RETURN
      END
```

INPUT DECK FOR SAMPLE RUN OF LINDIFF

```
    4  201         40.          .1
.00001    .025         50
(4F10.4)
7.0568    4.3436    7.6290    80.972
-.1       .0333333  .0333333  .0333333
0.        -.1       .05       .05
.05       0.        -.1       .05
0.        0.        0.        0.
                              STOP
```

```
LINEAR DIFFERENTIAL EQUATION SOLUTION

TIME STARTS AT     40.000, INCREMENTS BY        .100, FOR     201 ITERATIONS

    4 COMPARTMENT SYSTEM

ACCURACY OF SOLUTION      .1000E-04
TIME STEP SIZE      .2500E-01
MAXIMUM ITERATIONS/STEP    50

INITIAL VALUES AND MATRIX


   7.0568    4.3436    7.6290   80.9720



   -.1000     .0333     .0333     .0333
   0.0000    -.1000     .0500     .0500
    .0500    0.0000    -.1000     .0500
   0.0000    0.0000    0.0000    0.0000
```

```
TIME, COMPARTMENT VALUES, IN ORDER


     40.0000     .7057E+01     .4344E+01     .7629E+01     .8097E+02
     40.1000     .7024E+01     .4324E+01     .7598E+01     .8106E+02
     40.2000     .6992E+01     .4304E+01     .7567E+01     .8114E+02
     40.3000     .6960E+01     .4284E+01     .7536E+01     .8122E+02
     40.4000     .6928E+01     .4265E+01     .7506E+01     .8130E+02
     40.5000     .6897E+01     .4245E+01     .7475E+01     .8138E+02
     40.6000     .6865E+01     .4226E+01     .7445E+01     .8147E+02
     40.7000     .6834E+01     .4206E+01     .7414E+01     .8155E+02
     40.8000     .6803E+01     .4187E+01     .7384E+01     .8163E+02
     40.9000     .6772E+01     .4168E+01     .7354E+01     .8171E+02
     41.0000     .6741E+01     .4149E+01     .7324E+01     .8179E+02
     41.1000     .6710E+01     .4130E+01     .7294E+01     .8187E+02
     41.2000     .6680E+01     .4111E+01     .7264E+01     .8195E+02
     41.3000     .6649E+01     .4092E+01     .7234E+01     .8203E+02
     41.4000     .6619E+01     .4074E+01     .7204E+01     .8210E+02
     41.5000     .6589E+01     .4055E+01     .7175E+01     .8218E+02
     41.6000     .6559E+01     .4036E+01     .7145E+01     .8226E+02
     41.7000     .6529E+01     .4018E+01     .7116E+01     .8234E+02
     41.8000     .6499E+01     .4000E+01     .7087E+01     .8242E+02
     41.9000     .6470E+01     .3981E+01     .7058E+01     .8249E+02
     42.0000     .6441E+01     .3963E+01     .7029E+01     .8257E+02
     42.1000     .6411E+01     .3945E+01     .7000E+01     .8265E+02
     42.2000     .6382E+01     .3927E+01     .6971E+01     .8272E+02
     42.3000     .6354E+01     .3909E+01     .6942E+01     .8280E+02
     42.4000     .6325E+01     .3891E+01     .6913E+01     .8287E+02
     42.5000     .6296E+01     .3873E+01     .6885E+01     .8295E+02
     42.6000     .6268E+01     .3856E+01     .6856E+01     .8302E+02
     42.7000     .6239E+01     .3838E+01     .6828E+01     .8310E+02
     42.8000     .6211E+01     .3820E+01     .6800E+01     .8317E+02
     42.9000     .6183E+01     .3803E+01     .6772E+01     .8324E+02
     43.0000     .6155E+01     .3786E+01     .6744E+01     .8332E+02
     43.1000     .6127E+01     .3768E+01     .6716E+01     .8339E+02
     43.2000     .6100E+01     .3751E+01     .6688E+01     .8346E+02
     43.3000     .6072E+01     .3734E+01     .6660E+01     .8354E+02
     43.4000     .6045E+01     .3717E+01     .6632E+01     .8361E+02
     43.5000     .6018E+01     .3700E+01     .6605E+01     .8368E+02
     43.6000     .5991E+01     .3683E+01     .6577E+01     .8375E+02
     43.7000     .5964E+01     .3666E+01     .6550E+01     .8382E+02
     43.8000     .5937E+01     .3649E+01     .6523E+01     .8389E+02
     43.9000     .5910E+01     .3633E+01     .6496E+01     .8396E+02
     44.0000     .5884E+01     .3616E+01     .6469E+01     .8403E+02
     44.1000     .5857E+01     .3600E+01     .6442E+01     .8410E+02
     44.2000     .5831E+01     .3583E+01     .6415E+01     .8417E+02
     44.3000     .5805E+01     .3567E+01     .6388E+01     .8424E+02
     44.4000     .5779E+01     .3551E+01     .6361E+01     .8431E+02
     44.5000     .5753E+01     .3534E+01     .6335E+01     .8438E+02
     44.6000     .5727E+01     .3518E+01     .6308E+01     .8445E+02
     44.7000     .5701E+01     .3502E+01     .6282E+01     .8452E+02
     44.8000     .5676E+01     .3486E+01     .6256E+01     .8458E+02
     44.9000     .5650E+01     .3470E+01     .6230E+01     .8465E+02
     45.0000     .5625E+01     .3454E+01     .6204E+01     .8472E+02
     45.1000     .5600E+01     .3439E+01     .6178E+01     .8479E+02
     45.2000     .5575E+01     .3423E+01     .6152E+01     .8485E+02
     45.3000     .5550E+01     .3407E+01     .6126E+01     .8492E+02
     45.4000     .5525E+01     .3392E+01     .6100E+01     .8498E+02
     45.5000     .5500E+01     .3376E+01     .6075E+01     .8505E+02
     45.6000     .5476E+01     .3361E+01     .6049E+01     .8512E+02
     45.7000     .5451E+01     .3346E+01     .6024E+01     .8518E+02
     45.8000     .5427E+01     .3330E+01     .5999E+01     .8525E+02
     45.9000     .5403E+01     .3315E+01     .5973E+01     .8531E+02
```

```
46.0000     .5379E+01     .3300E+01     .5948E+01     .8537E+02
46.1000     .5355E+01     .3285E+01     .5923E+01     .8544E+02
46.2000     .5331E+01     .3270E+01     .5898E+01     .8550E+02
46.3000     .5307E+01     .3255E+01     .5874E+01     .8557E+02
46.4000     .5283E+01     .3240E+01     .5849E+01     .8563E+02
46.5000     .5260E+01     .3226E+01     .5824E+01     .8569E+02
46.6000     .5237E+01     .3211E+01     .5800E+01     .8575E+02
46.7000     .5213E+01     .3196E+01     .5775E+01     .8582E+02
46.8000     .5190E+01     .3182E+01     .5751E+01     .8588E+02
46.9000     .5167E+01     .3167E+01     .5727E+01     .8594E+02
47.0000     .5144E+01     .3153E+01     .5703E+01     .8600E+02
47.1000     .5121E+01     .3139E+01     .5678E+01     .8606E+02
47.2000     .5098E+01     .3124E+01     .5654E+01     .8612E+02
47.3000     .5076E+01     .3110E+01     .5631E+01     .8619E+02
47.4000     .5053E+01     .3096E+01     .5607E+01     .8625E+02
47.5000     .5031E+01     .3082E+01     .5583E+01     .8631E+02
47.6000     .5008E+01     .3068E+01     .5559E+01     .8637E+02
47.7000     .4986E+01     .3054E+01     .5536E+01     .8643E+02
47.8000     .4964E+01     .3040E+01     .5513E+01     .8648E+02
47.9000     .4942E+01     .3026E+01     .5489E+01     .8654E+02
48.0000     .4920E+01     .3012E+01     .5466E+01     .8660E+02
48.1000     .4898E+01     .2999E+01     .5443E+01     .8666E+02
48.2000     .4876E+01     .2985E+01     .5420E+01     .8672E+02
48.3000     .4855E+01     .2972E+01     .5397E+01     .8678E+02
48.4000     .4833E+01     .2958E+01     .5374E+01     .8684E+02
48.5000     .4812E+01     .2945E+01     .5351E+01     .8689E+02
48.6000     .4791E+01     .2931E+01     .5328E+01     .8695E+02
48.7000     .4769E+01     .2918E+01     .5306E+01     .8701E+02
48.8000     .4748E+01     .2905E+01     .5283E+01     .8707E+02
48.9000     .4727E+01     .2891E+01     .5261E+01     .8712E+02
49.0000     .4706E+01     .2878E+01     .5238E+01     .8718E+02
49.1000     .4686E+01     .2865E+01     .5216E+01     .8723E+02
49.2000     .4665E+01     .2852E+01     .5194E+01     .8729E+02
49.3000     .4644E+01     .2839E+01     .5172E+01     .8735E+02
49.4000     .4624E+01     .2826E+01     .5150E+01     .8740E+02
49.5000     .4603E+01     .2814E+01     .5128E+01     .8746E+02
49.6000     .4583E+01     .2801E+01     .5106E+01     .8751E+02
49.7000     .4563E+01     .2788E+01     .5084E+01     .8757E+02
49.8000     .4542E+01     .2776E+01     .5063E+01     .8762E+02
49.9000     .4522E+01     .2763E+01     .5041E+01     .8767E+02
50.0000     .4502E+01     .2750E+01     .5020E+01     .8773E+02
50.1000     .4483E+01     .2738E+01     .4998E+01     .8778E+02
50.2000     .4463E+01     .2726E+01     .4977E+01     .8784E+02
50.3000     .4443E+01     .2713E+01     .4956E+01     .8789E+02
50.4000     .4423E+01     .2701E+01     .4935E+01     .8794E+02
50.5000     .4404E+01     .2689E+01     .4914E+01     .8800E+02
50.6000     .4385E+01     .2676E+01     .4893E+01     .8805E+02
50.7000     .4365E+01     .2664E+01     .4872E+01     .8810E+02
50.8000     .4346E+01     .2652E+01     .4851E+01     .8815E+02
50.9000     .4327E+01     .2640E+01     .4830E+01     .8820E+02
51.0000     .4308E+01     .2628E+01     .4810E+01     .8826E+02
51.1000     .4289E+01     .2616E+01     .4789E+01     .8831E+02
51.2000     .4270E+01     .2605E+01     .4769E+01     .8836E+02
51.3000     .4251E+01     .2593E+01     .4748E+01     .8841E+02
51.4000     .4232E+01     .2581E+01     .4728E+01     .8846E+02
51.5000     .4214E+01     .2569E+01     .4708E+01     .8851E+02
51.6000     .4195E+01     .2558E+01     .4688E+01     .8856E+02
51.7000     .4177E+01     .2546E+01     .4667E+01     .8861E+02
51.8000     .4158E+01     .2535E+01     .4648E+01     .8866E+02
51.9000     .4140E+01     .2523E+01     .4628E+01     .8871E+02
52.0000     .4122E+01     .2512E+01     .4608E+01     .8876E+02
52.1000     .4104E+01     .2500E+01     .4588E+01     .8881E+02
52.2000     .4086E+01     .2489E+01     .4568E+01     .8886E+02
52.3000     .4068E+01     .2478E+01     .4549E+01     .8891E+02
52.4000     .4050E+01     .2467E+01     .4529E+01     .8896E+02
52.5000     .4032E+01     .2456E+01     .4510E+01     .8900E+02
```

52.6000	.4014E+01	.2445E+01	.4491E+01	.8905E+02
52.7000	.3996E+01	.2433E+01	.4471E+01	.8910E+02
52.8000	.3979E+01	.2422E+01	.4452E+01	.8915E+02
52.9000	.3961E+01	.2412E+01	.4433E+01	.8920E+02
53.0000	.3944E+01	.2401E+01	.4414E+01	.8924E+02
53.1000	.3927E+01	.2390E+01	.4395E+01	.8929E+02
53.2000	.3909E+01	.2379E+01	.4376E+01	.8934E+02
53.3000	.3892E+01	.2368E+01	.4357E+01	.8938E+02
53.4000	.3875E+01	.2358E+01	.4339E+01	.8943E+02
53.5000	.3858E+01	.2347E+01	.4320E+01	.8948E+02
53.6000	.3841E+01	.2336E+01	.4301E+01	.8952E+02
53.7000	.3824E+01	.2326E+01	.4283E+01	.8957E+02
53.8000	.3808E+01	.2315E+01	.4265E+01	.8961E+02
53.9000	.3791E+01	.2305E+01	.4246E+01	.8966E+02
54.0000	.3774E+01	.2295E+01	.4228E+01	.8970E+02
54.1000	.3758E+01	.2284E+01	.4210E+01	.8975E+02
54.2000	.3741E+01	.2274E+01	.4192E+01	.8979E+02
54.3000	.3725E+01	.2264E+01	.4174E+01	.8984E+02
54.4000	.3708E+01	.2253E+01	.4156E+01	.8988E+02
54.5000	.3692E+01	.2243E+01	.4138E+01	.8993E+02
54.6000	.3676E+01	.2233E+01	.4120E+01	.8997E+02
54.7000	.3660E+01	.2223E+01	.4102E+01	.9002E+02
54.8000	.3644E+01	.2213E+01	.4084E+01	.9006E+02
54.9000	.3628E+01	.2203E+01	.4067E+01	.9010E+02
55.0000	.3612E+01	.2193E+01	.4049E+01	.9015E+02
55.1000	.3596E+01	.2183E+01	.4032E+01	.9019E+02
55.2000	.3580E+01	.2174E+01	.4015E+01	.9023E+02
55.3000	.3565E+01	.2164E+01	.3997E+01	.9028E+02
55.4000	.3549E+01	.2154E+01	.3980E+01	.9032E+02
55.5000	.3533E+01	.2144E+01	.3963E+01	.9036E+02
55.6000	.3518E+01	.2135E+01	.3946E+01	.9040E+02
55.7000	.3502E+01	.2125E+01	.3929E+01	.9045E+02
55.8000	.3487E+01	.2116E+01	.3912E+01	.9049E+02
55.9000	.3472E+01	.2106E+01	.3895E+01	.9053E+02
56.0000	.3457E+01	.2097E+01	.3878E+01	.9057E+02
56.1000	.3441E+01	.2087E+01	.3861E+01	.9061E+02
56.2000	.3426E+01	.2078E+01	.3845E+01	.9065E+02
56.3000	.3411E+01	.2068E+01	.3829E+01	.9069E+02
56.4000	.3396E+01	.2059E+01	.3812E+01	.9073E+02
56.5000	.3382E+01	.2050E+01	.3795E+01	.9077E+02
56.6000	.3367E+01	.2041E+01	.3779E+01	.9082E+02
56.7000	.3352E+01	.2032E+01	.3762E+01	.9086E+02
56.8000	.3337E+01	.2022E+01	.3746E+01	.9090E+02
56.9000	.3323E+01	.2013E+01	.3730E+01	.9094E+02
57.0000	.3308E+01	.2004E+01	.3714E+01	.9098E+02
57.1000	.3294E+01	.1995E+01	.3698E+01	.9101E+02
57.2000	.3279E+01	.1986E+01	.3682E+01	.9105E+02
57.3000	.3265E+01	.1977E+01	.3666E+01	.9109E+02
57.4000	.3251E+01	.1969E+01	.3650E+01	.9113E+02
57.5000	.3236E+01	.1960E+01	.3634E+01	.9117E+02
57.6000	.3222E+01	.1951E+01	.3618E+01	.9121E+02
57.7000	.3208E+01	.1942E+01	.3603E+01	.9125E+02
57.8000	.3194E+01	.1934E+01	.3587E+01	.9129E+02
57.9000	.3180E+01	.1925E+01	.3572E+01	.9132E+02
58.0000	.3166E+01	.1916E+01	.3556E+01	.9136E+02
58.1000	.3152E+01	.1908E+01	.3541E+01	.9140E+02
58.2000	.3139E+01	.1899E+01	.3525E+01	.9144E+02
58.3000	.3125E+01	.1891E+01	.3510E+01	.9148E+02
58.4000	.3111E+01	.1882E+01	.3495E+01	.9151E+02
58.5000	.3098E+01	.1874E+01	.3480E+01	.9155E+02
58.6000	.3084E+01	.1865E+01	.3465E+01	.9159E+02
58.7000	.3071E+01	.1857E+01	.3450E+01	.9162E+02
58.8000	.3057E+01	.1849E+01	.3435E+01	.9166E+02
58.9000	.3044E+01	.1840E+01	.3420E+01	.9170E+02
59.0000	.3030E+01	.1832E+01	.3405E+01	.9173E+02
59.1000	.3017E+01	.1824E+01	.3390E+01	.9177E+02

```
59.2000    .3004E+01    .1816E+01    .3376E+01    .9181E+02
59.3000    .2991E+01    .1808E+01    .3361E+01    .9184E+02
59.4000    .2978E+01    .1800E+01    .3346E+01    .9188E+02
59.5000    .2965E+01    .1791E+01    .3332E+01    .9191E+02
59.6000    .2952E+01    .1783E+01    .3318E+01    .9195E+02
59.7000    .2939E+01    .1775E+01    .3303E+01    .9198E+02
59.8000    .2926E+01    .1768E+01    .3289E+01    .9202E+02
59.9000    .2913E+01    .1760E+01    .3275E+01    .9205E+02
60.0000    .2901E+01    .1752E+01    .3260E+01    .9209E+02
```

Program SNOOP

INTRODUCTION

Program SNOOP allows examination of interactions by extending two-dimensional (2D) graphing procedures to three-dimensional (3D) plotting. A variable hereafter referred to as a plotting character (PC) denotes the third dimension. The PC can assume values from 0 to 9 and is designated by the content of a selected card column in the observation data. The numeric characters that appear on the completed graph are the actual PC values that appear on the cards (a blank results in a PC value of 0).

For example, if volume, diameter, and height are measured on a group of felled trees, the 2D relationship of volume and diameter can be plotted on the same graph and up to 10 values (or classes) of the plotting character (height) can be distinguished. Examination of the graph may then result in selections of appropriate transformations, interaction terms, and/or weights to be applied in regression analysis. This procedure is not necessary for so simple a problem as postulated here, but extension of the possible uses of 3D plotting should be obvious. However, because of its simplicity, a volume-estimation problem is used as an example later.

Two-dimensional plotting is also available to the user of SNOOP. In this case the numeric characters that appear on the completed graph are frequencies of occurrence. The resulting character at any set of coordinates may then be blank (no occurrence), 1 to 9, or C (10 or more).

If overplots are attempted, messages (containing the coordinates and, in the case of 3D plotting, the PC values) are printed after the appropriate graph. An attempted overplot would occur when 2 or more observations have identical coordinates for 3D plotting, or 11 or more observations for 2D plotting.

When several dependent (Y) and independent (X) variables are included in one job, a set of fully labeled graphs is made, including each Y-X combination. And, for 3D, a separate set of graphs is made for each PC specified. The program automatically scales the graphs for each variable used. Other flexibility available to the user, barring certain properties given in the section *Program Limitations*, can be summarized as follows:

2D and 3D jobs can be included in the same run in any order. As many jobs as desired may be included in a single run. Observation values can be negative, zero, and positive. Order of the variables on the observation cards is completely flexible.

PC's can be located anywhere on the observation cards. Multiple cards per observation can be used.

The major use of program SNOOP is in the model-building stage of an estimation problem. A segment of data can be plotted in various ways, and the graphs can be examined for logical trends in dependent-variable values as responses to changes in the values of independent variables. Models can then be formed for regression analyses with the remaining data.

Consequently the value of the program is easily recognized when the researcher is attempting to explain previously unfamiliar functional relationships between many variables. With only a few variables, many observations and expectations that limit the possibilities of functional-relationship alternatives, the researcher can go directly to the use of one of the canned regression routines--for example, Furnival--available at any large computer center.

Program SNOOP was thoroughly tested and is operational on the CDC 6400 computer. The program is written in FORTRAN IV and should run with little or no modification on other computers that accept FORTRAN IV, have a 32K core, and have three tape drives or equivalent input/output devices. The logical tape assignments are:

Unit	Use
5	Control-deck input
6	Program output
9	Intermediate operations (scratch tape)

A special subroutine was necessary for backspacing logical unit 5 at the computer installation where this program was developed. A simple BACKSPACE 5 command or its equivalent is all that is necessary at other installations. The two places in the program where this must be done are marked with comments.

At CSU, the program has been checked out on the NCAR compiler only.

HOW TO USE THE PROGRAM

The program is activated by a control deck supplied by the user. The control deck consists of cards containing the observations and information designating a specific manner in which the observations are to be handled by the program. The following instructions describe the complete control-deck setup (all underlined characters *must* be punched exactly as they appear in the instructions):

Card Group	Number of Cards	Title	Card Columns	Content
1	1	Job description	1- 2	*2D*--For two-dimensional plotting. *3D*--For three-dimensional plotting.
			3- 4	Number of Y's (NYS), right adjusted.
			5- 6	Number of X's (NXS), right adjusted (does not include PC variables).

Card Group	Number of Cards	Title	Card Columns	Content
1 (Continued)				
			7- 8	Number of plotting characters (NPC), right adjusted. Use _00_ (or blanks) for 2D.
			9-11	Number of observations (NOBS), right adjusted.
			12	Number of cards per observation (NCARDS). The use of _0_ (or blank) will be interpreted as _1_.
			13-72	Alphameric job title to be printed above each graph.
2	1	Format	1	_1_--There is no format continuation card. _2_--The format specification is continued on the format continuation card.
			2	_(_--Left parenthesis.
			3-72	Up to 70 alphameric characters, ending with a right parenthesis, and containing a FORTRAN BCD format (F-specifications), which describes the format of the Y's and X's on the observation cards. The right parenthesis is put on the format continuation card if column 1 of the format card = 2.
3	0 or 1	Format continuation	1-72	Continuation of the format specification described for columns 3-72 of the format card.
4	NYS + NXS	Variable labels	1	_Y_--If the variable is a dependent variable. _X_--If the variable is an independent variable.

Card Group	Number of Cards	Title	Card Columns	Content
4 (Continued)				
			2	Blank.
			3-52	Alphameric label (identifying information) associated with the variable. This group of cards must be arranged in the order that the variables appear on the observation cards.
5	0 if 2D, NPC if 3D	3D plotting characters	1-18	*PLOTTING CHARACTER*
			19	Blank.
			20	Number of the card (of the observation set) on which the PC value is found. The use of *0* (or blank) will be interpreted as *1*.
			21-22	Column number (right adjusted) in which the PC value is found.
			23	Blank.
			24-31	Alphameric label (identifying information) associated with the PC.
6	NOBS xNCARDS	Observations	1-80	One or more cards for each observation as designated by the format cards.
7	1	Job control	1- 8	*CONTINUE*--If more than one job is included in the run. Card groups 1-7 are repeated after the job control card. *DONE* (left adjusted)--Designates the end of the last job in the run.

PROGRAM LIMITATIONS

Job restrictions applying both to 2D and 3D plotting are:

$NYS + NXS \le 50$

$NYS \ge 1$

$NXS \ge 1$

$2 \le NOBS \le 500$

$1 \le NCARDS \le 9$

Job restrictions pertaining to 3D plotting only are:

$(NYS + NXS + NPC) \times NOBS \le 10{,}000$

$1 \le NPC \le 10$

A job restriction pertaining to 2D plotting only is:

$(NYS + NXS) \times NOBS \le 10{,}000$

SAMPLE RUN OF PROGRAM

For the researcher who is developing a regression model, the graphs produced by the program can be a valuable aid in selecting transformations, interaction terms, and variance-stabilizing weights, and in ordering of variables.

As an example, let use examine the sample control deck and the output which is attached.

The first page of output consists of summary information pertaining to the entire job. Card groups 1 to 5 of the control deck are printed as well as the values of all the variables in the first observation (the Y's precede the X's in the printout). Because the first job in this example specifies 3D plotting, the values of the plotting characters for the first observation are also included in the summary information.

Now, assume for the moment that expectations of functional relationships involving cubic-foot volume, diameter, and height are vague. The trees plotted in this example could be an initial sample of trees to be measured in the construction of a standard volume table; that is, we want to express cubic volume as a function of both diameter and height, if possible. Consequently, the control deck has been set up to plot a 3D graph of cubic-foot volume as a function of diameter, using 10-ft height class as a plotting character.

The resultant graph shows that volume increases with an increase in diameter, and there is an increase in the slope of the relationship as diameter increases.

The second job is simply a demonstration of 2D graphing. The same observations were used as in the first job. Two graphs were specified by designating volume as a dependent variable.

ACKNOWLEDGMENTS

A FORTRAN listing of SNOOP is contained in U. S. Forest Service Research Paper NE-91, by Warren E. Frayer. It is available from the Northeast Forest Experiment Station, Upper Darby, Pennsylvania. The present information on this program was liberally extracted from that source with the author's kind permission.

```
      PROGRAM SNOOP                                                   SNOOP
     1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,TAPE3)      SNOOP
      DIMENSION FMT(36),LABEL(50,50),ILABEL(50,50),CONLAB(10,2),TITLE(15 SNOOP
     1),IVAR(50),CHECK(5),IPC(10),XMAT(10120),PC(10),V(50),VV(50),KPC(72 SNOOP
     20),R(20),IGRAPH(50,100),XABS(5),IG(12)                         SNOOP
C                                                                     SNOOP
C                                                                     SNOOP
C READ THE CONTROL DECK AND CHECK FOR ERRORS                          SNOOP
C                                                                     SNOOP
C                                                                     SNOOP
      DATA THREED/2H3D/                                               SNOOP
      DATA TWOD/2H2D/                                                 SNOOP
      DATA YY/1HY/                                                    SNOOP
      DATA XX/1HX/                                                    SNOOP
      DATA T1,T2,T3,T4,T5/2HPL,4HOTTI,4HNG C,4HHARA,4HCTER/           SNOOP
      DATA IG/1H ,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H./        SNOOP
      DATA EE,CC/4HDONE,4HCONT/                                       SNOOP
      REWIND 1                                                        SNOOP
      REWIND 3                                                        SNOOP
 5800 READ(5,892)(R(I),I=1,20)                                        SNOOP
      WRITE(1,892)(R(I),I=1,20)                                       SNOOP
      WRITE(3,892)(R(I),I=1,20)                                       SNOOP
      IF(R(1).NE.EE)GO TO 5800                                        SNOOP
      END FILE 1                                                      SNOOP
      END FILE 3                                                      SNOOP
      REWIND 1                                                        SNOOP
      REWIND 3                                                        SNOOP
C READ THE JOB DESCRIPTION CARD                                       SNOOP
      IJOB=0                                                          SNOOP
  799 IJOB=IJOB+1                                                     SNOOP
      READ(1,700)DEM,NYS,NXS,NPC,NOBS,NCARDS,(TITLE(I),I=1,15)        SNOOP
  700 FORMAT(A2,3I2,I3,I1,15A4)                                       SNOOP
      NC=NCARDS                                                       SNOOP
      IF(NCARDS.EQ.0)NC=1                                             SNOOP
      IF(NCARDS.EQ.0)NCARDS=0                                         SNOOP
      WRITE(6,7501)IJOB                                               SNOOP
      WRITE(6,7511)(TITLE(I),I=1,15)                                  SNOOP
      WRITE(6,752)DEM                                                 SNOOP
      WRITE(6,753)NYS                                                 SNOOP
      WRITE(6,754)NXS                                                 SNOOP
      WRITE(6,755)NOBS                                                SNOOP
      WRITE(6,10001)NC                                                SNOOP
10001 FORMAT(34H NUMBER OF CARDS PER OBSERVATION =I1)                 SNOOP
      WRITE(6,756)NPC                                                 SNOOP
      WRITE(6,757)                                                    SNOOP
 7501 FORMAT(11H1JOB NUMBERI3)                                        SNOOP
 7511 FORMAT(12H JOB TITLE =15A4)                                     SNOOP
  752 FORMAT(1H A2,9H PLOTTING)                                       SNOOP
  753 FORMAT(35H0NUMBER OF DEPENDENT(Y) VARIABLES =I2)                SNOOP
  754 FORMAT(37H NUMBER OF INDEPENDENT(X) VARIABLES =I2)              SNOOP
  755 FORMAT(25H NUMBER OF OBSERVATIONS =I3)                          SNOOP
  756 FORMAT(32H NUMBER OF PLOTTING CHARACTERS =I2)                   SNOOP
  757 FORMAT(43H0THE CONTROL DECK FOR THIS JOB IS......      )        SNOOP
      IF(DEM.NE.THREED.AND.DEM.NE.TWOD)GO TO 999                      SNOOP
      IF(NYS.LE.0.OR.NYS.GE.50)GO TO 800                              SNOOP
      IF(NXS.LE.0.OR.NXS.GE.50)GO TO 801                              SNOOP
      IF(NOBS.LT.2.OR.NOBS.GT.500)GO TO 810                           SNOOP
      IF(DEM.EQ.THREED GO TO 702                                      SNOOP
      IF(NPC.NE.0)GO TO 802                                           SNOOP
      IF(((NYS+NXS)*NOBS)-10000)703,703,860                           SNOOP
  702 IF(NPC.LE.0.OR.NPC.GT.10)GO TO 803                                 SNOOP
      IF(((NYS+NXS+NPC)*NOBS)-10000)703,703,861                          SNOOP
  703 WRITE(6,704)DEM,NYS,NXS,NPC,NOBS,NCARDS,(TITLE(I),I=1,15)          SNOOP
  704 FORMAT(1H0A2,3I2,I3,I1,15A4)                                       SNOOP
C READ THE FORMAT CARDS                                                  SNOOP
      READ(1,780)IFORM,(FMT(I),I=1,18)                                   SNOOP
  780 FORMAT(I1,17A4,A3)                                                 SNOOP
      READ(3,12321)ISPACE                                                SNOOP
12321 FORMAT(A1)                                                         SNOOP
      IF(IFORM.LT.1.OR.IFORM.GT.2)GO TO 999                              SNOOP
      WRITE(6,781)IFORM,(FMT(I),I=1,18)                                  SNOOP
  781 FORMAT(1H I1,17A4,A3)                                              SNOOP
      IF(IFORM.EQ.1)GO TO 706                                            SNOOP
      READ(1,705)(FMT(I),I=19,36)                                        SNOOP
  705 FORMAT(18A4)                                                       SNOOP
      READ(3,12321)ISPACE                                                SNOOP
      WRITE(6,782)(FMT(I),I=19,36)                                       SNOOP
  782 FORMAT(1H 18A4)                                                    SNOOP
C READ THE VARIABLE LABELS                                               SNOOP
  706 NVAR=NYS+NXS                                                       SNOOP
      IXX=0                                                              SNOOP
      IYY=0                                                              SNOOP
      DO 790 I=1,50                                                      SNOOP
  790 IVAR(I)=0                                                          SNOOP
      DO 707 I=1,NVAR                                                    SNOOP
      READ(1,708)VAR,(LABEL(I,K),K=1,50)                                 SNOOP
  708 FORMAT(A1,1X,50A1)                                                 SNOOP
      READ(3,12321)ISPACE                                                SNOOP
      IF(VAR.NE.YY.AND.VAR.NE.XX)GO TO 999                               SNOOP
      IF(VAR.EQ.XX)GO TO 709                                             SNOOP
      IYY=IYY+1                                                          SNOOP
      IVAR(I)=IYY                                                        SNOOP
      GO TO 710                                                          SNOOP
  709 IXX=IXX+1                                                          SNOOP
      IVAR(I)=IXX+NYS                                                    SNOOP
  710 WRITE(6,791)VAR,(LABEL(I,K),K=1,50)                                SNOOP
C     ORDER THE VARIABLE LABELS                                          SNOOP
      KKK=IVAR(I)                                                        SNOOP
      DO 711 K=1,50                                                      SNOOP
  711 ILABEL(KKK,K)=LABEL(I,K)                                           SNOOP
  707 CONTINUE                                                           SNOOP
C READ THE PLOTTING CHARACTERS LABELS                                    SNOOP
      IF(DEM.EQ.TWOD)GO TO 751                                           SNOOP
      DO 750 I=1,NPC                                                     SNOOP
      READ(1,712)(CHECK(IJ),IJ=1,5),ICARD,IPC(I),(CONLAB(I,K),K=1,2)     SNOOP
  712 FORMAT(A2,4A4,1X,I1,I2,1X,2A4)                                     SNOOP
      READ(3,12321)ISPACE                                                SNOOP
      IF(CHECK(1).EQ.YY.OR.CHECK(1).EQ.XX)GO TO 830                      SNOOP
      IF(CHECK(1).NE.T1.OR.CHECK(2).NE.T2.OR.CHECK(3).NE.T3.OR.CHECK(4). SNOOP
     1NE.T4.OR.CHECK(5).NE.T5)GO TO 999                                  SNOOP
      IF(IPC(I).LE.0.OR.IPC(I).GT.80)GO TO 999                           SNOOP
      WRITE(6,792)(CHECK(IJ),IJ=1,5),ICARD,IPC(I),(CONLAB(I,K),K=1,2)    SNOOP
  792 FORMAT(1H A2,4A4,2I2,1X,2A4)                                       SNOOP
      IC=ICARD                                                           SNOOP
      IF(IC.EQ.0)IC=1                                                    SNOOP
      IPC(I)=((IC-1)*80)+IPC(I)                                          SNOOP
  750 CONTINUE                                                           SNOOP
  751 CONTINUE                                                           SNOOP
      READ(3,12321)ISPACE                                                SNOOP
C                                                                        SNOOP
```

```
      C                                                                       SNOOP
      C THE OBSERVATIONS ARE NOW PLACED IN A VECTOR CALLED XMAT               SNOOP
      C                                                                       SNOOP
      C                                                                       SNOOP
            WRITE(6,797)                                                      SNOOP
        797 FORMAT(45H0....OBSERVATION CARDS ARE LOCATED HERE....    )        SNOOP
            DO 100 K=1,50                                                     SNOOP
            V(K)=0.0                                                          SNOOP
        100 VV(K)=0.0                                                         SNOOP
            DO 110 K=1,10120                                                  SNOOP
        110 XMAT(K)=0.0                                                       SNOOP
            ICOUNT=0                                                          SNOOP
      C READ THE VARIABLES                                                    SNOOP
        103 READ(1,FMT)(V(K),K=1,NVAR)                                        SNOOP
            ICOUNT=ICOUNT+1                                                   SNOOP
      C READ THE PLOTTING CHARACTERS                                          SNOOP
            KSTART=1                                                          SNOOP
            KEND=80                                                           SNOOP
            DO 9692 IJKL=1,NC                                                 SNOOP
            READ(3,112)(KPC(K),K=KSTART,KEND)                                 SNOOP
            KSTART=KSTART+80                                                  SNOOP
       9692 KEND=KEND+80                                                      SNOOP
            IF(DEM.NE.THREED)GO TO 13                                         SNOOP
        112 FORMAT(80A1)                                                      SNOOP
            DO 113 I=1,NPC                                                    SNOOP
            KKPC=IPC(I)                                                       SNOOP
            IRKPC=KPC(KKPC)                                                   SNOOP
            DO 960 LL=1,11                                                    SNOOP
            IF(IRKPC.NE.IG(LL))GO TO 960                                      SNOOP
            PC(I)=LL-1                                                        SNOOP
            IF(LL.EQ.11)PC(I)=0.                                              SNOOP
            GO TO 113                                                         SNOOP
        960 CONTINUE                                                          SNOOP
            GO TO 835                                                         SNOOP
        113 CONTINUE                                                          SNOOP
            GO TO 13                                                          SNOOP
      10013 WRITE(6,970)                                                      SNOOP
        970 FORMAT(54H0THE PLOTTING CHARACTERS FOR THIS OBSERVATION ARE.... ) SNOOP
            DO 972 I=1,NPC                                                    SNOOP
            L=PC(I)                                                           SNOOP
        972 WRITE(6,971)I,L                                                   SNOOP
        971 FORMAT(4H PC(I2,2H)=I1)                                           SNOOP
            GO TO 795                                                         SNOOP
      C ORDER THE VARIABLES                                                   SNOOP
         13 DO 200 ISKIP=1,NVAR                                               SNOOP
            IP=IVAR(ISKIP)                                                    SNOOP
        200 VV(IP)=V(ISKIP)                                                   SNOOP
            IF(ICOUNT.NE.1)GO TO 795                                          SNOOP
            WRITE(6,7999)                                                     SNOOP
       7999 FORMAT(48H0THE VARIABLES IN THE FIRST OBSERVATION ARE...    )     SNOOP
            DO 650 I=1,NYS                                                    SNOOP
        650 WRITE(6,651)I,VV(I)                                               SNOOP
        651 FORMAT(3H Y(I2,2H)=,E14.8)                                        SNOOP
            DO 652 I=1,NXS                                                    SNOOP
            IXS=I+NYS                                                         SNOOP
        652 WRITE(6,653)I,VV(IXS)                                             SNOOP
        653 FORMAT(3H X(I2,2H)=,E14.8)                                        SNOOP
            IF(DEM.EQ.THREED)GO TO 10013                                      SNOOP
      C STORE THE VARIABLES                                                   SNOOP
        795 KK=0                                                              SNOOP
      DO 410 K=1,NVAR                                                   SNOOP
      KSUB=((K-1)*NOBS)+KK+ICOUNT                                       SNOOP
      KMIN=KSUB+NOBS-ICOUNT+1                                           SNOOP
      KMAX=KMIN+1                                                       SNOOP
      XMAT(KSUB)=VV(K)                                                  SNOOP
      KK=KK+2                                                           SNOOP
C STORE THE MAXIMUM AND MINIMUM VALUES                                  SNOOP
      IF(ICOUNT.EQ.1)GO TO 107                                          SNOOP
      AMAX=XMAT(KMAX)                                                   SNOOP
      AMIN=XMAT(KMIN)                                                   SNOOP
 120  IF((AMAX-VV(K))*(VV(K)-AMIN))106,106,410                          SNOOP
  106 IF(VV(K)-AMAX)108,107,107                                         SNOOP
 107  AMAX=VV(K)                                                        SNOOP
      XMAT(KMAX)=AMAX                                                   SNOOP
      IF(ICOUNT.NE.1)GO TO 410                                          SNOOP
 108  AMIN=VV(K)                                                        SNOOP
      XMAT(KMIN)=AMIN                                                   SNOOP
 410  CONTINUE                                                          SNOOP
C STORE THE PLOTTING CHARACTERS                                         SNOOP
      IF(DEM.NE.THREED)GO TO 500                                        SNOOP
      DO 401 L=1,NPC                                                    SNOOP
      KSLOT=(NOBS+2)*(NVAR+L-1)+ICOUNT                                  SNOOP
 401  XMAT(KSLOT)=PC(L)                                                 SNOOP
 500  IF(ICOUNT.NE.NOBS)GO TO 103                                       SNOOP
C                                                                       SNOOP
C                                                                       SNOOP
C THE VALUES IN XMAT ARE CHANGED TO ROW AND COLUMN NUMBERS FOR THE      SNOOP
C           OUTPUT MATRIX(IGRAPH)                                       SNOOP
C                                                                       SNOOP
C                                                                       SNOOP
      KX=1                                                              SNOOP
      DO 1050 K=1,NVAR                                                  SNOOP
      NL=(K-1)*NOBS+KX                                                  SNOOP
      ML=(NOBS*K)+KX-1                                                  SNOOP
      KX=KX+2                                                           SNOOP
      DO 1050 LK=NL,ML                                                  SNOOP
      IF(K.GT.NYS)GO TO 1010                                            SNOOP
      DIV=50.                                                           SNOOP
      GO TO 1011                                                        SNOOP
 1010 DIV=100.                                                          SNOOP
 1011 XINC=(XMAT(ML+2)-XMAT(ML+1))/DIV                                  SNOOP
      IF(XINC.EQ.0.)XINC=1.0                                            SNOOP
      A=XMAT(ML+1)                                                      SNOOP
      RXMAT=0.                                                          SNOOP
 1007 IF(A.GT.XMAT(LK))GO TO 1006                                       SNOOP
      A=A+XINC                                                          SNOOP
      RXMAT=RXMAT+1.                                                    SNOOP
      GO TO 1007                                                        SNOOP
 1006 XMAT(LK)=RXMAT                                                    SNOOP
      IF(K.GT.NYS)GO TO 1020                                            SNOOP
      IF(XMAT(LK).LE.50.)GO TO 1050                                     SNOOP
      XMAT(LK)=50.                                                      SNOOP
      GO TO 1050                                                        SNOOP
 1020 IF(XMAT(LK).LE.100.)GO TO 1050                                    SNOOP
      XMAT(LK)=100.                                                     SNOOP
 1050 CONTINUE                                                          SNPCO
      GO TO 3145                                                        SNPCO
 800  WRITE(6,990)NYS                                                   SNOOP
      GO TO 1000                                                        SNOOP
 801  WRITE(6,991)NXS                                                   SNOOP
```

```
      GO TO 1000                                                        SNOOP
  802 WRITE(6,992)NPC                                                   SNOOP
      GO TO 1000                                                        SNOOP
  803 WRITE(6,993)NPC                                                   SNOOP
      GO TO 1000                                                        SNOOP
  810 WRITE(6,994)NOBS                                                  SNOOP
      GO TO 1000                                                        SNOOP
  830 WRITE(6,995)                                                      SNOOP
      GO TO 1000                                                        SNOOP
  835 WRITE(6,996)ICOUNT,I                                              SNOOP
      GO TO 12322                                                       SNOOP
  860 NOVER=((NYS+NXS)*NOBS)                                            SNOOP
      WRITE(6,997)NOVER                                                 SNOOP
      GO TO 1000                                                        SNOOP
  861 NOVER=((NYS+NXS+NPC)*NOBS)                                        SNOOP
      WRITE(6,998)NOVER                                                 SNOOP
      GO TO 1000                                                        SNOOP
  999 READ(3,892)(R(I),I=1,20)                                          SNOOP
      WRITE(6,891)                                                      SNOOP
      WRITE(6,893)(R(I),I=1,20)                                         SNOOP
      GO TO 12322                                                       SNOOP
  990 FORMAT(5H0NYS=I2,/44H NYS MUST BE GREATER THAN 0 AND LESS THAN 50) SNOOP
  991 FORMAT(5H0NXS=I2,/44H NXS MUST BE GREATER THAN 0 AND LESS THAN 50) SNOOP
  992 FORMAT(5H0NPC=I2,/44H NPC MUST BE BLANK OR 0                     ) SNOOP
  993 FORMAT(5H0NPC=I2,/44H NPC MUST BE GREATER THAN 0 AND LESS THAN 11) SNOOP
  994 FORMAT(6H0NOBS=I3,/46H NOBS MUST BE GREATER THAN 1 AND LESS THAN 5 SNOOP
     101)                                                               SNOOP
  995 FORMAT(63H0THERE IS A VARIABLE LABEL WITH THE 3D PLOTTING CHARACTE SNOOP
     1R GROUP)                                                          SNOOP
  996 FORMAT(49H0THERE IS A NON-NUMERIC CHARACTER IN OBSERVATION I3,24H  SNOOP
     1FOR PLOTTING CHARACTER I2)                                        SNOOP
  997 FORMAT(18H0(NYS+NXS) X NOBS=I6,/42H THIS MUST BE EQUAL TO OR LESS  SNOOP
     1THAN 10,000)                                                      SNOOP
  998 FORMAT(22H0(NYS+NXS+NPC) X NOBS=I6,/42H THIS MUST BE EQUAL TO OR L SNOOP
     1ESS THAN 10,000)                                                  SNOOP
  891 FORMAT(60H0THE FOLLOWING CONTROL CARD IS MISPUNCHED OR OUT OF ORDE SNOOP
     1R...)                                                             SNOOP
  892 FORMAT(20A4)                                                      SNOOP
  893 FORMAT(1H 20A4)                                                   SNOOP
 1000 READ(3,12321)ISPACE                                               SNOOP
12322 WRITE(6,1001)                                                     SNOOP
 1001 FORMAT(27H0UNABLE TO PROCESS THIS JOB/55H THE PROGRAM IS NOW SEARC SNOOP
     1HING FOR THE NEXT JOB(IF ANY)   )                                 SNOOP
 1002 READ(1,1003)TAL                                                   SNOOP
      READ(3,12321)ISPACE                                               SNOOP
 1003 FORMAT(A4)                                                        SNOOP
      IF(TAL.EQ.EE)GO TO 2000                                           SNOOP
      IF(TAL.EQ.CC)GO TO 799                                            SNOOP
      GO TO 1002                                                        SNOOP
C                                                                       SNOOP
C                                                                       SNOOP
C CONSTRUCT THE GRAPHS (IGRAPH)                                         SNOOP
C                                                                       SNOOP
C                                                                       SNOOP
 3145 IF(DEM.EQ.TWOD)NPC=1                                              SNPC001
      DO 2010 J=1,NPC                                                   SNOOP
      DO 2010 K=1,NYS                                                   SNOOP
      KX=(2*(K-1))+1                                                    SNOOP
      KY=(2*NYS)+1                                                      SNOOP
      LNYS=NYS+1                                                        SNOOP
      DO 2010 L=LNYS,NVAR                                                    SNOOP
      REWIND 2                                                               SNOOP
      DO 2030 I=1,50                                                         SNOOP
      DO 2030 M=1,100                                                        SNOOP
 2030 IGRAPH(I,M)=100                                                        SNOOP
      MT=(K-1)*NOBS+KX                                                       SNOOP
      NK=(L-1)*NOBS+KY                                                       SNOOP
      MK=NK+NOBS-1                                                           SNOOP
      KY=KY+2                                                                SNOOP
      ICNT=0                                                                 SNOOP
      ID=0                                                                   SNOOP
      DO 2011 LK=NK,MK                                                       SNOOP
      IY=XMAT(MT)+.001                                                       SNOOP
      IX=XMAT(LK)+.001                                                       SNOOP
      IF(DEM.EQ.THREED)GO TO 2015                                            SNOOP
      IF(IGRAPH(IY,IX).EQ.110)GO TO 3000                                     SNOOP
      IGRAPH(IY,IX)=IGRAPH(IY,IX)+1                                          SNOOP
      GO TO 2011                                                             SNOOP
 3000 WRITE(2,3002)IY,IX                                                     SNOOP
 791  FORMAT(1H A1,1X,50A1)                                                  SNOOP
 3002 FORMAT(2I3)                                                            SNOOP
      ID=ID+1                                                                SNOOP
      GO TO 2011                                                             SNOOP
 2015 ICNT=ICNT+1                                                            SNOOP
      KSLOT=(NOBS+2)*(NVAR+J-1)+ICNT                                         SNOOP
      KPLOT=XMAT(KSLOT)+.001                                                 SNOOP
      IF(IGRAPH(IY,IX).NE.100)GO TO 3001                                     SNOOP
      IGRAPH(IY,IX)=KPLOT                                                    SNOOP
      GO TO 2011                                                             SNOOP
 3001 WRITE(2,3003)IY,IX,KPLOT                                               SNOOP
 3003 FORMAT(3I3)                                                            SNOOP
      ID=ID+1                                                                SNOOP
 2011 MT=MT+1                                                                SNOOP
      END FILE 2                                                             SNOOP
      REWIND 2                                                               SNOOP
C PRINT A GRAPH                                                               SNOOP
      WRITE(6,3041)IJOB                                                      SNOOP
 3041 FORMAT(1H115X,11HJOB NUMBER I3)                                        SNOOP
      WRITE(6,3040)(TITLE(I),I=1,15)                                         SNOOP
 3040 FORMAT(1H 15X,15A4)                                                    SNOOP
      IF(DEM.EQ.THREED)GO TO 3042                                            SNOOP
      WRITE(6,3043)                                                          SNOOP
 3043 FORMAT(1H 15X,23HTWO-DIMENSIONAL GRAPH      )                          SNOOP
      GO TO 3044                                                             SNOOP
 3042 WRITE(6,3045)J,(CONLAB(J,JC),JC=1,2)                                   SNOOP
 3045 FORMAT(1H 15X,27HTHREE-DIMENSIONAL GRAPH,PC(I2,16H) IS DEFINED AS      SNOOP
     12A4)                                                                   SNOOP
 3044 CONTINUE                                                               SNOOP
      DO 3050 IY=1,50                                                        SNOOP
      DO 3050 IX=1,100                                                       SNOOP
      IF(IGRAPH(IY,IX).NE.100)GO TO 3051                                     SNOOP
      IGRAPH(IY,IX)=IG(1)                                                    SNOOP
      GO TO 3050                                                             SNOOP
 3051 IF(IGRAPH(IY,IX).GT.100)IGRAPH(IY,IX)=IGRAPH(IY,IX)-100                SNOOP
      DO 3053 I=1,10                                                         SNOOP
      IF(IGRAPH(IY,IX).NE.I)GO TO 3053                                       SNOOP
      IGRAPH(IY,IX)=IG(I+1)                                                  SNOOP
      GO TO 3050                                                             SNOOP
 3053 CONTINUE                                                               SNOOP
      IGRAPH(IY,IX)=IG(11)                                                   SNOOP
 3050 CONTINUE                                                            SNOOP
      YINC=XMAT(MT+1)-XMAT(MT)                                            SNOOP
      YINC=YINC/50.                                                       SNOOP
      BEGIN=XMAT(MT+1)-(.5*YINC)                                          SNOOP
      DO 3055 IJK=1,50                                                    SNOOP
      I=50-IJK+1                                                          SNOOP
      IF(I.EQ.50.OR.I.EQ.40.OR.I.EQ.30.OR.I.EQ.20.OR.I.EQ.10)GO TO 3060   SNOOP
      WRITE(6,3061)ILABEL(K,IJK),(IGRAPH(I,II),II=1,100)                  SNOOP
 3061 FORMAT(1H A1,14X,1H.,100A1)                                         SNOOP
      GO TO 3055                                                          SNOOP
 3060 WRITE(6,3063)ILABEL(K,IJK),BEGIN,(IGRAPH(I,II),II=1,100)            SNOOP
 3063 FORMAT(1H A1,2X,E11.5,2H..100A1)                                    SNOOP
 3055 BEGIN=BEGIN-YINC                                                    SNOOP
 3071 WRITE(6,3070)BEGIN                                                  SNOOP
 3070 FORMAT(4H    E11.5,102H.............................................. SNOOP
     1...............................................................  )  SNOOP
      WRITE(6,3080)                                                       SNOOP
 3080 FORMAT(1H 15X,1H.19X,1H.19X,1H.19X,1H.19X,1H.19X,1H.)               SNOOP
      XINC=(XMAT(MK+2)-XMAT(MK+1))/100.                                   SNOOP
      XBEGIN=XMAT(MK+1)-(.5*XINC)                                         SNOOP
      DO 6070 IIJK=20,100,20                                              SNOOP
      IIIJK=IIJK/20                                                       SNOOP
      XIK=IIJK                                                            SNOOP
 6070 XABS(IIIJK)=XBEGIN+(XIK*XINC)                                       SNOOP
      WRITE(6,6071)XBEGIN,(XABS(I),I=1,5)                                 SNOOP
 6071 FORMAT(1H 10X,E11.5,5(9X,E11.5))                                    SNOOP
      WRITE(6,3081)(ILABEL(L,II),II=1,50)                                 SNOOP
 3081 FORMAT(1H015X,50A1)                                                 SNOOP
      WRITE(6,3082)ID                                                     SNOOP
 3082 FORMAT(31H NUMBER OF ATTEMPTED OVERPLOTS=I3)                        SNOOP
      IF(ID.EQ.0)GO TO 2010                                               SNOOP
      WRITE(6,3083)                                                       SNOOP
 3083 FORMAT(37H1COORDINATES OF ATTEMPTED OVERPLOTS      )                SNOOP
C PRINT THE OVERPLOTS                                                    SNOOP
      DO 3085 I=1,ID                                                      SNOOP
      IF(DEM.EQ.THREED)GO TO 3090                                         SNOOP
      READ(2,4040)IY,IX                                                   SNOOP
 4040 FORMAT(2I3)                                                         SNOOP
      YI=IY                                                               SNOOP
      YI=(BEGIN+(YI*YINC))                                                SNOOP
      XI=IX                                                               SNOOP
      XI=(XBEGIN+(XI*XINC))                                               SNOOP
      WRITE(6,3091)YI,XI                                                  SNOOP
 3091 FORMAT(3H Y=E11.5,3X,2HX=E11.5)                                     SNOOP
      GO TO 3085                                                          SNOOP
 3090 READ(2,4041)IY,IX,KPLOT                                             SNOOP
 4041 FORMAT(3I3)                                                         SNOOP
      YI=IY                                                               SNOOP
      YI=(BEGIN+(YI*YINC))                                                SNOOP
      XI=IX                                                               SNOOP
      XI=(XBEGIN+(XI*XINC))                                               SNOOP
      WRITE(5,3092)YI,XI,KPLOT                                            SNOOP
 3092 FORMAT(3H Y=E11.5,3X,2HX=E11.5,3X,3HPC=I2)                          SNOOP
 3085 CONTINUE                                                            SNOOP
 2010 CONTINUE                                                            SNOOP
      READ(1,3020)TAIL,TAIM                                               SNOOP
      IF(TAIL.NE.EE.AND.TAIL.NE.CC)GO TO 999                              SNOOP
      READ(3,12321)ISPACE                                                 SNOOP
 3020 FORMAT(2A4)                                                         SNOOP
      WRITE(6,3021)TAIL,TAIM                                              SNOOP
```

```
3021 FORMAT(1H12A4)                                                    SNOOP
     IF(TAIL.EQ.CC)GO TO 799                                           SNOOP
2000 CONTINUE                                                          SNOOP
     STOP                                                              SNOOP
     END                                                               SNOOP
```

INPUT DECK FOR SAMPLE RUN OF SNOOP

```
3D010101095lRED PINE VOLUME TABLE CONSTRUCTION (PRELIMINARY PLOTTING)
1(5X,F3.1,5X,F4.1)
X               DIAMETER AT BREAST HEIGHT
Y                    CUBIC-FOOT VOLUME
PLOTTING CHARACTER 111 HT CLASS
      143  62  432
      145  72  513
      148  80  597
      150  46  351
      152  45  356
      153  40  317
      154  52  418
      156  68  561
      157  56  470
      159  75  640
      160  60  521
      163  73  657
      165  85  780
      168  80  770
      170  50  480
      173  60  609
      175  56  570
      177  58  617
      178  64  694
      179  72  790
      180  58  630
      181  69  766
      183  62  704
      184  74  840
      186  76  900
      188  64  788
      131  44  250
      132  56  335
      134  48  292
      136  47  290
      137  53  338
      139  56  369
      140  59  390
       50  20   17
       51  22   19
       52  26   25
       54  28   23
       56  24   26
       59  36   42
       60  30   36
       62  33   43
       63  24   33
       65  26   38
       65  36   52
       70  40   66
       72  39   69
       74  42   79
       77  42   82
       79  45   94
       80  46  100
       83  45  102
```

INPUT DECK FOR SAMPLE RUN OF SNOOP

```
      86  50  123
      90  30   84
      92  35  101
      94  36  111
      97  44  140
     111  58  242
     100  51  175
     101  40  138
     103  52  187
     104  48  174
     106  55  209
     189  84 1030
     193  70  880
     198  90 1222
     109  58  234
     110  39  160
     199  86 1141
     200  60  801
      99  38  123
     119  45  216
     219  82 1330
     220  95 1502
     221  98 1634
     223  74 1263
     225  86 1530
     119  65  315
     120  54  260
     126  70  377
     230  60 1036
     233  95 1750
     130  40  235
     130  42  241
     240  97 1894
     261  60 1360
     263 100 2371
     266  87 2104
     268 101 2453
     280 102 2763
     113  40  170
     155  64  283
     202  72 1002
     210  74  983
     218  75 1223
     116  43  199
CONTINUE
2D0102000951RED PINE DATA - EXAMPLES OF TWO-DIMENSIONAL PLOTTING)
1(5X,F3.1,1X,F3.0,1X,F4.1)
X                DIAMETER AT BREAST HEIGHT
X                      TOTAL HEIGHT
Y                     CUBIC-FOOT VOLUME
     143  62  432
     145  72  513
     148  80  597
     150  46  351
     152  45  356
     153  40  317
```

INPUT DECK FOR SAMPLE RUN OF SNOOP

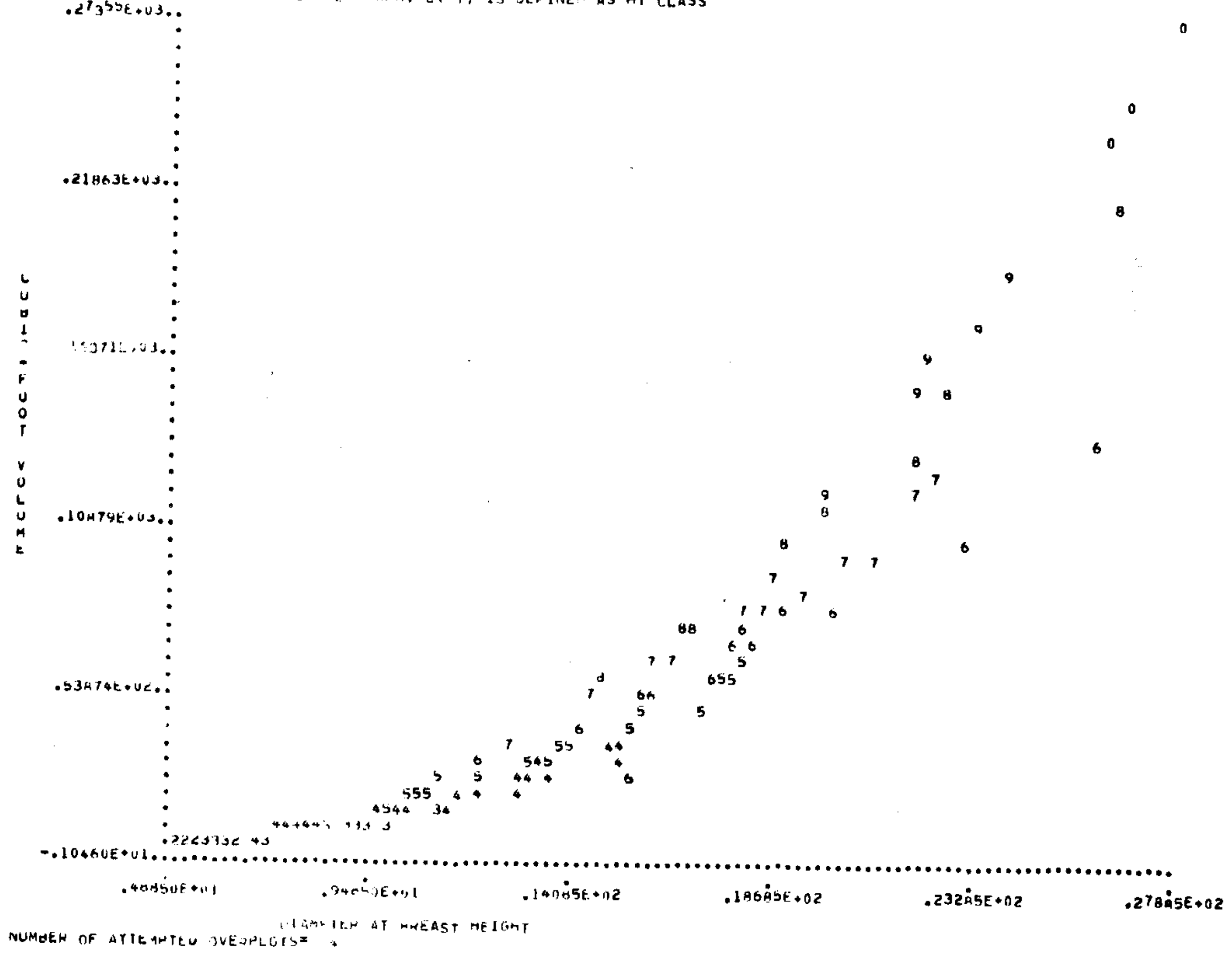
```
154  52  418
156  68  561
157  56  470
159  75  640
160  60  521
163  73  657
165  85  780
168  80  770
170  50  480
173  60  609
175  56  570
177  58  617
178  64  694
179  72  790
180  58  630
181  69  766
183  62  704
184  74  840
186  76  900
188  64  788
131  44  250
132  56  335
134  48  292
136  47  290
137  53  338
139  56  369
140  59  390
 50  20   17
 51  22   19
 52  26   25
 54  28   23
 56  24   26
 59  36   42
 60  30   36
 62  33   43
 63  24   33
 65  26   38
 65  36   52
 70  40   66
 72  39   69
 74  42   79
 77  42   82
 79  45   94
 80  46  100
 83  45  102
 86  50  123
 90  30   84
 92  35  101
 94  36  111
 97  44  140
111  58  242
100  51  175
101  40  138
103  52  187
104  48  174
106  55  209
```

INPUT DECK FOR SAMPLE RUN OF SNOOP

```
     189  84 1030
     193  70  880
     198  90 1222
     109  58  234
     110  39  160
     199  86 1141
     200  60  801
      99  38  123
     119  45  216
     219  82 1330
     220  95 1502
     221  98 1634
     223  74 1263
     225  86 1530
     119  65  315
     120  54  260
     126  70  377
     230  60 1036
     233  95 1750
     130  40  235
     130  42  241
     240  97 1894
     261  60 1360
     263 100 2371
     266  87 2104
     268 101 2453
     280 102 2763
     113  40  170
     155  64  283
     202  72 1002
     210  74  983
     218  75 1223
     116  43  199
DONE
```

```
JOB NUMBER  1
JOB TITLE =RED PINE VOLUME TABLE CONSTRUCTION (PRELIMINARY PLOTTING)
3D PLOTTING

NUMBER OF DEPENDENT(Y) VARIABLES = 1
NUMBER OF INDEPENDENT(X) VARIABLES = 1
NUMBER OF OBSERVATIONS = 95
NUMBER OF CARDS PER OBSERVATION =1
NUMBER OF PLOTTING CHARACTERS = 1

THE CONTROL DECK FOR THIS JOB IS.......
3D 1 1 1 951RED PINE VOLUME TABLE CONSTRUCTION (PRELIMINARY PLOTTING)
1(5X,F3.1,5X,F4.1)
X                DIAMETER AT BREAST HEIGHT
Y                     CUBIC-FOOT VOLUME
PLOTTING CHARACTER 111 HT CLASS

....OBSERVATION CARDS ARE LOCATED HERE....

THE VARIABLES IN THE FIRST OBSERVATION ARE...
Y( 1)= .43200000E+02
X( 1)= .14300000E+02

THE PLOTTING CHARACTERS FOR THIS OBSERVATION ARE....
PC( 1)=6


COORDINATES OF ATTEMPTED OVERPLOTS
Y= .44460E+01   X= .51160E+01   PC= 2
Y= .44460E+01   X= .51150E+01   PC= 2
Y= .44460E+01   X= .62660E+01   PC= 2
Y= .44460E+01   X= .64950E+01   PC= 3
```

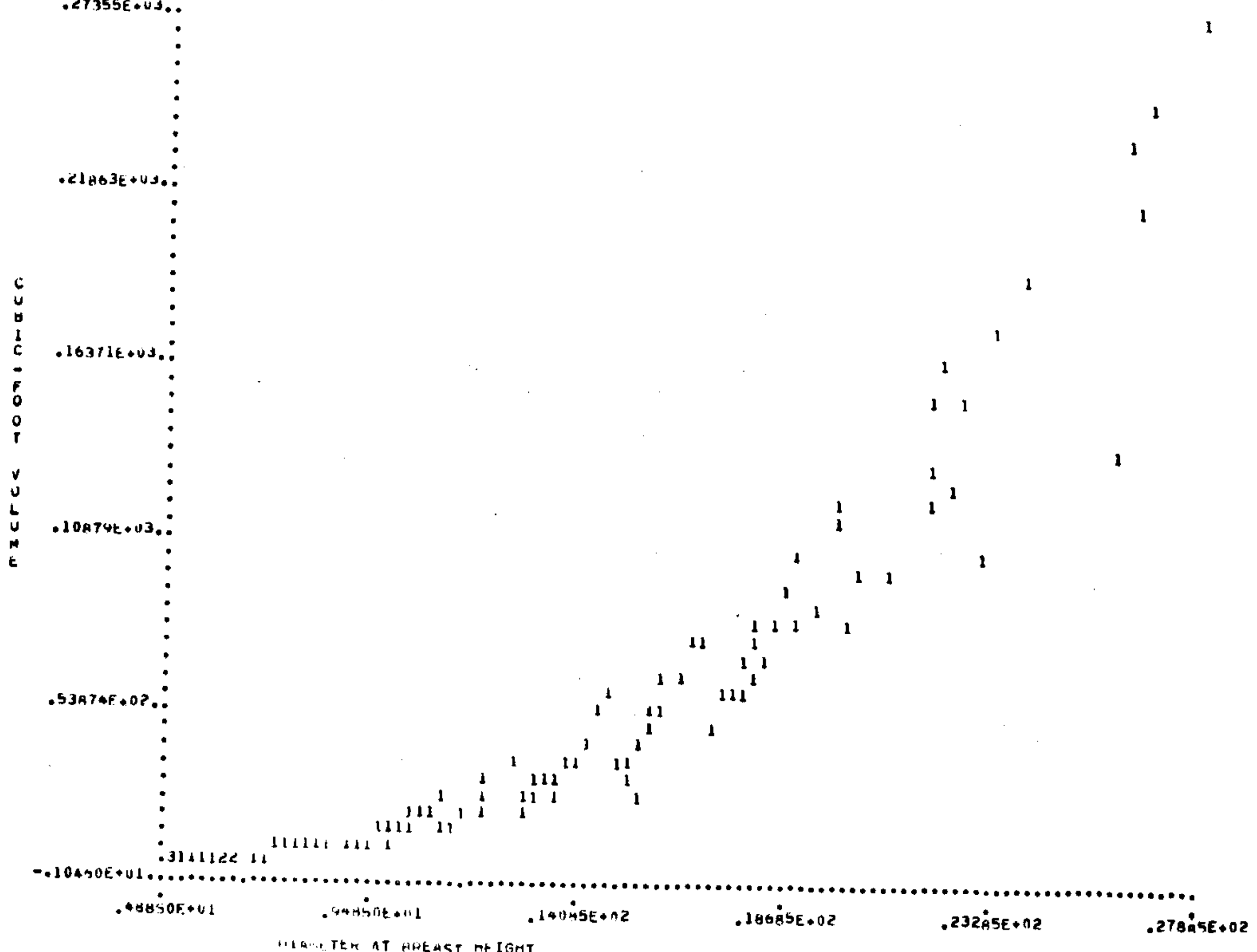
```
JOB NUMBER   1
RED PINE VOLUME ... CONSTRUCTION (PRELIMINARY PLOTTING)
THREE-DIMENSIONAL GRAPH,PC( 1) IS DEFINED AS HT CLASS
```



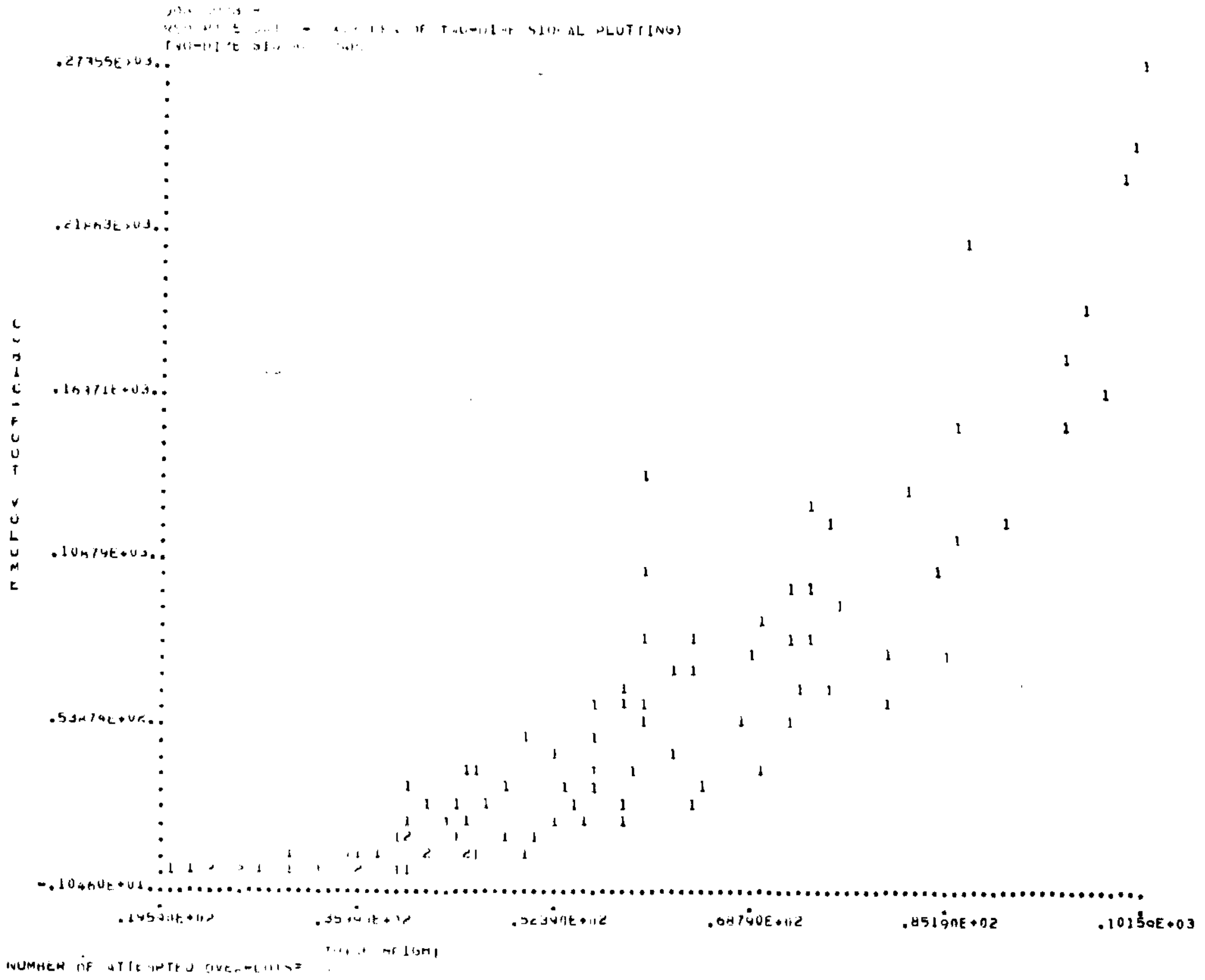
```
NUMBER OF ATTEMPTED OVERPLOTS=  4
```

```
JOB NUMBER  2
JOB TITLE *RED PINE DATA - EXAMPLES OF TWO-DIMENSIONAL PLOTTING)
2D PLOTTING

NUMBER OF DEPENDENT(Y) VARIABLES = 1
NUMBER OF INDEPENDENT(X) VARIABLES = 2
NUMBER OF OBSERVATIONS = 95
NUMBER OF CARDS PER OBSERVATION =1
NUMBER OF PLOTTING CHARACTERS = 0

THE CONTROL DECK FOR THIS JOB IS.......

2D 1 2 0 95)RED PINE DATA - EXAMPLES OF TWO-DIMENSIONAL PLOTTING)
1(5X,F3.1,1X,F3.0,1X,F4.1)
X               DIAMETER AT BREAST HEIGHT
X                     TOTAL HEIGHT
Y                    CUBIC-FOOT VOLUME

....OBSERVATION CARDS ARE LOCATED HERE....

THE VARIABLES IN THE FIRST OBSERVATION ARE...
Y( 1)= .43200000E+02
X( 1)= .14300000E+02
X( 2)= .62000000E+02
```

```
JOB NUMBER  2
RED PINE DATA - (EXAMPLES OF TWO-DIMENSIONAL PLOTTING)
TWO-DIMENSIONAL GRAPH
```



```
NUMBER OF ATTEMPTED OVERPLOTS=  0
```


SIGNAL PLOTTING)
.27355E+03
.21663E+03
.16371E+03
.10479E+03
.53874E+02
-.10460E+01
CUBIC-FOOT VOLUME
.19533E+02
.35763E+02
.52390E+02
.68790E+02
.85190E+02
.10159E+03
TOTAL HEIGHT
NUMBER OF ATTEMPTED OVERWRITES=

LINPROG

INTRODUCTION

LINPROG is a linear programming program originally written by R. J. Classen and converted for use on the FORTRAN Extended compiler of the CDC 6400 SCOPE system.

Linear programming deals with the problem of allocating limited resources among competing activities in an optimal manner. This problem of allocation can arise whenever one must select the level of certain activities which must compete for certain scarce resources necessary to perform those activities. The applications of linear programming extend to a wide variety of situations. For the ecologist the applications extend from the determination of the maximum profit in resource management to the achievement of the greatest cost-effectiveness of one's research procedures. For the systems ecologist, the application of optimization techniques such as linear programming reduces the likelihood of suboptimization. Suboptimization can result from too much concentration on one of the subsystems at the expense of a more important subsystem. Whatever the application, all linear programming problems involve the planning of activities in order to obtain the result, among all the feasible alternatives, which reaches the specified goal best.

PROGRAM OPERATION

The mathematical statement of a general form of the linear programming problem is the following. Find z_1, z_2---z_n which maximizes the linear objective function,

$$X = \sum_{j=1}^{n} c_j \ z_j$$

subject to the r constraints

$$\sum_{j=1}^{n} A_{ij} z_j = B_i \text{ for } i = 1,2,\ldots r$$

and $z_j \geq 0$ for all j.

Where the A_{ij}, B_i, and C_j are given constants and the z_j are the unknown decision variables.

The method used by the program is a variation of the explicit inverse form of the simplex method. All computations are in single precision arithmetic. The limitations of the program allow up to 100 decision variables and as many as 82 constraints. The simplex algorithm used by the program is a subroutine written by R. J. Classen in September, 1961. The subroutine is very versatile and complex and allows for the solution of many different types of linear programming problems. LINPROG was written to make this subroutine easy to use for the basic type of problem. It can easily be altered if a special case is to be solved.

INPUT REQUIREMENTS

Input to the problem is as follows:

Card 1 (Problem identification)

Col. 1-78	XID--alphanumeric title for problem
79-80	NTYPE--< zero to stop program = zero to read in a new problem > zero to read in a new resource constraint for the right hand side and rework the last problem.

Card 2 (if NTYPE > 0)

Col. 1-5	row number of the new constraint
6-15	value of the new constraint

(this is the only card required if NTYPE > 0)

Card 2 (If NTYPE = 0)

Col. 1-5	NR--number of constraint equations
6-10	NC--number of variables including artificial and slack variables
11-15	INVOFF ≠ 0 to print inverse matrix = 0 to skip printing

Card 3 (variable format card of the type "FORMAT I NN(...)")

Col. 1-6	"FORMAT"
8-10	I--"ROW" if the A matrix is to be read in by rows "COL" if it is to be read by columns
12-13	NN--number of coefficients per card
15-80	(...)--format specification, this specification must allow for two leading index numbers on each card which indicate the matrix position of the first coefficient on the card

Card 4 - number required (constraint matrix)

The "A" matrix is input as specified by variable format card or card 3 and is followed by a blank card.

"B" card - (right hand sides of constraint equations)

the B_I are input in an 8E10.0 format

ID card to start a new problem or else stop the program as explained above.

To allow for inequalities and free varaibles the "A" matrix should be set up as follows:

1. Inequalities - to change equation II from an equality to $\leq$, add a column, J1, in which all entries are zero except that

A(I1,J1) = 1. Variable J1 is then a positive slack variable. To change the equation to a $\geq$, do the same as above except A(I1,J1) = -1.

2. free variables - to remove the restriction on variable J1, that B(J1) $\geq$ 0 add a column J2 in which all entries A(I,J2) = -A(I,J1).

LISTING AND SAMPLE OUTPUT

The output shown is the solution of the following problem:

Minimize $4z_1 + 7z_2 + 8z_3 + 6z_4$

while satisfying

$$150z_1 + 140z_2 + 170z_3 + 160z_4 = 150$$

$$0.1z_1 + 0.1z_2 + 0.3z_3 + 0.3z_4 \leq 0.2$$

$$2z_1 + 4z_2 + 5z_3 + 3z_4 \geq 3$$

and $Z_1 \geq 0,\ z_2 \geq 0,\ z_3 \geq 0,\ z_4 \geq 0$

The first output grouping shows the initial tableau or the "A" matrix as it was read in. Row 1 shows the C_j coefficients of the objective function and the remaining rows are the a_{ij} coefficients of the constraint equations. This page also shows the value of the objective function at each iteration.

The second ouput grouping shows the characteristics of the problem. The seven output constants are:

KOUT (1) = K, output condition:

3 - feasible and optimal

4 - no feasible solution

5 - no pivot, infinite solution

6 - iteration limit exceeded

7 - illegal input quantity

KOUT(2) = ITER, number of iterations taken.

KOUT(3) = INVC, number of iterations since last inversion (ignoring final inversions, if done).

KOUT(4) = NUMVR, number of inversions done (including final and initial inversions).

KOUT(5) = NUMPV, number of pivots done

KOUT(6) = INFS, infeasibility flag, 1 = infeasible, 0 = feasible

KOUT(7) = JT, final pivot column selected. (Conditions 5 and 6 only.)

The eight quantities in ERROR are:

ERR(1) Sum of the feasibility errors (AZ-B).

ERR(2) Maximum feasibility error.

ERR(3) Sum of the reduced costs in basis.

ERR(4) Maximum reduced cost (in absolute value) in basis.

In a final inversion is performed, the above errors will be the errors before inversion and ERR (5, 6, 7,8) will be the corresponding errors after inversion.

The eight quantities in INFIX are as follows:

INFIX (1) = INFLAG, the input condition, usually "4"; other values are explained below.

INFIX (2) = N, the number of columns in "A" matrix.

INFIX (3) = ME, the length of one column in "A" matrix, i.e., the first dimension of "A" matrix.

INFIX (4) = M, the row number of the final constraint in the "A" matrix; $MF \leq ME$.

INFIX (5) = MF, the row number of the first constraint in the "A" matrix; $MF \leq M$.

INFIX (6) = MC, the row number of the objective form (costs) in the "A" matrix; MF < MC < 0.

INFIX (7) = NCUT, the maximum number of iterations that will be allowed to solve the problem.

INFIX (8) = NVER, the reinversion frequency; NVER = 0 means don't reinvert.

The four tolerances are:

TOL (1) = TPIV, pivot tolerance

TOL (2) = TZERO, tolerance for setting "X" to zero.

TOL (3) = TCOST $\leq$ 0, reduced cost is considered to be negative only if it is below this quantity.

TOL (4) = TECOL, quantities in the pivot row of the inverse are assumed zero if magnitude below this quantity (used only in inversion 2 of the subroutine).

The third output grouping shows the solution of the problem. The column JJH shows the decision variables which are in the solution at the point of optimality. z(0) is the objective function. BBAR shows the final value of the objective function for z(0) and the final values for the other z variables in solution. PI shows the negatives of the solution of the dual problem. YTEMP(1) shows the negative value of the objective function and the remaining values show the truncation error when the z values are substituted into the constraint equations.

The fourth output grouping shows the inverse matrix if it is requested.

The fifth grouping shows, for each z_j, the amount that the cost c_j would have to be reduced in order to include that variable in the solution. This amount is zero for the variables that are already in the optimal solution.

```
      PROGRAM LINPROG
C-----
C-----****************************************************************
C-----
C-----LINEAR PROGRAMMING SETUP ROUTINE
C-----CARD 1
C-----   COL.  1-78  ALPHANUMERIC PROBLEM IDENTIFICATION
C-----   COL. 79-80  LESS THAN ZERO TO STOP PROGRAM
C-----               EQUAL TO ZERO TO READ IN A NEW PROBLEM
C-----               GREATER THAN ZERO TO READ IN A NEW RESOURCE
C-----               CONSTRAINT FOR THE RIGHT HAND SIDE AND REWORK
C-----               THE LAST PROBLEM.
C-----CARD 2
C-----   COL    1- 5  NUMBER OF CONSTRAINT EQUATIONS
C-----          6-10  NUMBER OF VARIABLES INCLUDING ARTIFICIAL AND
C-----                SLACK VARIABLES
C-----         11-15  NOT EQUAL TO ZERO TO PRINT INVERSE MATRIX.
C-----CARD 3
C-----   COL    1- 6  FORMAT
C-----          8-10  ROW IF CONSTRAINT MATRIX IS TO BE INPUT BY ROWS
C-----                COL IF IT IS INPUT BY COLUMNS.
C-----         12-13  NUMBER OF COEFFICIENTS PER CARD.
C-----         15-80  VARIABLE FORMAT TO TO READ IN TWO INDEX NUMBERS,
C-----                INDICATING THE MATRIX POSITION OF THE FIRST COEF-
C-----                FICIENT ON THAT CARD, AND THE NUMBER OF COEFFICIENT
C-----                SPECIFIED IN COLUMNS 12 AND 13
C-----CARDS 4 THROUGH N
C-----   THE CONSTRAINT MATRIX IS INPUT AS SPECIFIED BY THE
C-----   VARIABLE FORMAT IN CARD 3
C-----CARD N+1
C-----   THE RIGHT HAND SIDE OF THE TABLEAU IS INPUT IN AN 8F10.0 FORMAT
C-----CARD N+2
C-----   THE SAME AS CARD 1
C-----
C-----****************************************************************
C-----
      DIMENSION TAB(82,100), E(82,82), RHS(82), JJH(82), BBAR(82), PI(82
     1), YTEMP(82), KB(100), RC(100), KOUT(8), ERR(8), INFIX(8), TOL(8),
     2 XID(14)
      NRT=82
      REWIND 6
1     READ (5,9) (XID(I),I=1,13),NTYPE
      WRITE (6,10) (XID(I),I=1,13)
      IF (NTYPE) 8,2,3
2     READ (5,11) NR,NC,INVOFF
C-----PLACE CONSTRAINTS IN TABLEAU
      IMAX=NR+1
      JMAX=NC
      CALL MATIN (TAB,NRT,IMAX,NC)
C-----WRITE TABLEAU
      WRITE (6,14)
      CALL FPRINT (TAB,NRT,IMAX,JMAX,3,1,1)
C-----SET UP RIGHT HAND SIDE
      RHS(1)=0.0
      READ (5,12) (RHS(I),I=2,IMAX)
      INFIX(1)=0
      INFIX(2)=JMAX
      INFIX(3)=NRT
      INFIX(4)=IMAX
      INFIX(5)=2
      INFIX(6)=1
      INFIX(7)=200
      INFIX(8)=2*IMAX
      TOL(1)=1.0E-06
      TOL(2)=1.0E-05
      TOL(3)=-1.0E-03
      TOL(4)=1.0E-10
      TOL(5)=0.0
      TOL(6)=0.0
      TOL(7)=0.0
      TOL(8)=0.0
      KOUT(8)=0
      PRM=0.0
      GO TO 4
3     INFIX(1)=5
      READ (5,13) I,RHS(I)
4     CALL SIMPLX (INFIX,TAB,RHS,TOL,PRM,KOUT,ERR,JJH,BBAR,PI,YTEMP,KB,E
     1)
      WRITE (6,10) (XID(I),I=1,13)
      WRITE (6,15)
      WRITE (6,16) (I,KOUT(I),ERR(I),INFIX(I),TOL(I),I=1,8)
      WRITE (6,10) (XID(I),I=1,13)
      WRITE (6,17)
      WRITE (6,18) (I,JJH(I),BBAR(I),PI(I),RHS(I),YTEMP(I),I=1,IMAX)
      IF (INVOFF) 5,6,5
5     WRITE (6,10) (XID(I),I=1,13)
      WRITE (6,19)
      CALL FPRINT (E,IMAX,IMAX,IMAX,3,1,1)
6     DO 7 J=1,JMAX
      CALL DEL (J,RC(J),INFIX(4),TAB,PI,INFIX(3))
7     CONTINUE
      WRITE (6,10) (XID(I),I=1,13)
      WRITE (6,20)
      WRITE (6,21) (I,KB(I),RC(I),I=1,JMAX)
      GO TO 1
8     CONTINUE
      ENDFILE 6
      STOP
C-----
9     FORMAT (13A6,I2)
10    FORMAT (1H-,8X,13A6)
11    FORMAT (5I5)
12    FORMAT (8F10.4)
13    FORMAT (I5,F10.4)
14    FORMAT (1H0,4X,7HTABLEAU)
15    FORMAT (1H0,14X,4HKOUT,7X,5HERROR,9X,5HINFIX,7X,3HTOL)
16    FORMAT (1H0,I5,I12,5X,E14.7,I7,5X,E12.5)
17    FORMAT (1H0,14X,3HJJH,9X,4HBBAR,12X,2HPI,14X,3HRHS,13X,5HYTEMP)
18    FORMAT (1H0,I8,4X,2HX(,I3,3H) =,4E16.7)
19    FORMAT (1H0,4X,14HINVERSE MATRIX)
20    FORMAT (1H0,13X,2HKB,7X,10HREDUCED C.)
21    FORMAT (1H ,I8,I7,2X,E16.7)
```

```
      END
```

```
      SUBROUTINE FPRINT (A,MR,NR,NC,K,L,N)
C-----OUTPUT ROUTINE FOR 1 OR 2 DIMENSIONAL ARRAYS    (FORTRAN IV)
C-----
C-----MR = FIRST DIMENSION NUMBER OF THE A ARRAY
C-----NR = NO. OF ROWS IN THE ARRAY TO BE PRINTED
C-----NC = NO. OF COLUMNS IN THE ARRAY TO BE PRINTED
C-----K AND L ARE NOT PRESENTLY UTILIZED
C-----FOR RETURN WITHOUT ACTION, SET N = 0
C-----
      DIMENSION A(MR,888)
      IF (N) 1,11,1
1     NFULLB=NC/10
      NCLEFT=NC-10*NFULLB
      IF (NCLEFT) 2,3,2
2     NFULLB=NFULLB+1
3     NSKIP=MAX0(1,60/(NR+4))
      DO 10 M=1,NFULLB
      NC1=10*M-9
      IF (M-NFULLB) 5,4,4
4     NC2=NC
      GO TO 6
5     NC2=10*M
6     WRITE (6,12)
      WRITE (6,13) (J,J=NC1,NC2)
      WRITE (6,12)
      DO 7 I=1,NR
      WRITE (6,14) I,(A(I,J),J=NC1,NC2)
7     CONTINUE
      IF (NC2-NC) 8,11,11
8     IF (M-NSKIP*(M/NSKIP)) 10,9,10
9     WRITE (6,15)
10    CONTINUE
11    RETURN
C-----
C-----
12    FORMAT (1X)
13    FORMAT (1H ,1X,10I10)
14    FORMAT (1H ,I5,2X,10E11.3)
15    FORMAT (1H1)
      END
```

```
      SUBROUTINE MATIN (A,MR,NR,NC)
C-----THIS PROGRAM READS DATA INTO AN ARRAY.
C-----A FORMAT CARD MUST PRECEDE THE DATA,
C-----  I.E.,FORMAT I  NN (...)
C-----         WITH  'FORMAT'  IN COL. 1-6,
C-----                 I = 'ROW' FOR ROW FORMAT, I = 'COL' FOR COLUMN FORMAT
C-----                 NN = NO. DATA WORDS PER CARD,
C-----              (...) = FORMAT SPECIFICATION.
C-----EACH DATA CARD MUST HAVE TWO INDEX NUMBERS
C-----  INDICATING THE MATRIX POSITION OF THE FIRST DATA WORD.
C-----THE READING OF CARDS IS COMPLETED BY ONE BLANK CARD.
C-----THIS ROUTINE HAS A PROVISION FOR CHANGING FORMATS IN MIDSTREAM.
C-----A CARD WITH BOTH INDICES NEGATIVE IMPLIES THAT A FORMAT CARD FOLLO
C-----
      DIMENSION A(MR,NC), FMT(17), XLIST(78)
      CALL EQUIV (CHECK1,4HFORM)
      CALL EQUIV (CHECK2,4HAT  )
      CALL EQUIV (ROW,3HROW)
      DO 1 I=1,MR
      DO 1 J=1,NC
1     A(I,J)=0.0
C-----READ FORMAT CARD
2     READ (5,28) WORD1,WORD2,ROC,NN,(FMT(I),I=1,17)
      IF (WORD1-CHECK1) 4,3,4
3     IF (WORD2-CHECK2) 4,5,4
4     WRITE (6,32)
      GO TO 27
C-----READ DATA CARD
5     READ (5,FMT) I,J,(XLIST(K),K=1,NN)
C-----CHECK FOR NEW FORMAT CARD
      IF (I) 6,7,7
6     IF (J) 2,19,10
C-----CHECK FOR BLANK CARD TO END LIST
7     IF (I+J) 8,27,8
8     IF (ROC-ROW) 18,9,18
C-----ROW FORMAT ***
9     IF (I-NR) 11,11,10
10    WRITE (6,29) I,J,MR,NR,NC
      GO TO 5
11    DO 17 JJ=1,NN
      JJJ=J+JJ-1
      IF (XLIST(JJ)) 12,17,12
12    IF (JJJ-NC) 14,14,13
13    WRITE (6,30) I,JJJ,MR,NR,NC
      GO TO 5
14    IF (A(I,JJJ)) 15,16,15
15    WRITE (6,31) I,JJJ,MR,NR,NC
      GO TO 17
16    A(I,JJJ)=XLIST(JJ)
17    CONTINUE
      GO TO 5
C-----COLUMN FORMAT ***
18    IF (J-NC) 20,20,19
19    WRITE (6,30) I,J,MR,NR,NC
      GO TO 5
20    DO 26 II=1,NN
      III=I+II-1
      IF (XLIST(II)) 21,26,21
21    IF (III-NR) 23,23,22
```

```
22     WRITE (6,29) III,J,MR,NR,NC
       GO TO 5
23     IF (A(III,J)) 24,25,24
24     WRITE (6,31) III,J,MR,NR,NC
       GO TO 26
25     A(III,J)=XLIST(II)
26     CONTINUE
       GO TO 5
27     RETURN
C-----
28     FORMAT (1A4,1A2,1X,1A3,1X,I2,1X,16A4,1A2)
29     FORMAT (22H0ILLEGAL 1ST INDEX, I=,I3,2X,2HJ=,I3,2X,3HMR=,I3,2X,3HN
      1R=,I3,2X,3HNC=,I3)
30     FORMAT (22H0ILLEGAL 2ND INDEX, I=,I3,2X,2HJ=,I3,2X,3HMR=,I3,2X,3HN
      1R=,I3,2X,3HNC=,I3)
31     FORMAT (22H0DUPLICATION ERROR, I=,I3,2X,2HJ=,I3,2X,3HMR=,I3,2X,3HN
      1R=,I3,2X,3HNC=,I3)
32     FORMAT (1H0,51H***  PROGRAM EXPECTS FORMAT CARD AT THIS POINT  ***
      1)
       END
```

```
      SUBROUTINE EQUIV (A,B)
      A=B
      RETURN
      END
```

```
      SUBROUTINE SIMPLX (INFIX,A,B,TOL,PRM,KOUT,ERS,JH,X,P,Y,KB,E)
C-----BOSS    MASTER SUBROUTINE OF  RS MSUB,  VERSION 2.
C-----
      DIMENSION INFIX(8), A(1), B(1), TOL(4), KOUT(7), ERS(8), JH(1), X(
     11), P(1), Y(1), KB(1), E(1), ZZ(4), IOFIX(16), TERR(8)
C-----DIMENSION INFIX(8),A(1),B(1),TOL(4),KOUT(7),ERS(8),JH(1),X(1),
C-----1 P(1),Y(1),KB(1),E(1),ZZ(4), IOFIX(16) , TERR(8)
C-----
      EQUIVALENCE (INFLAG,IOFIX(1)), (N,IOFIX(2)), (ME,IOFIX(3)), (M,IOF
     1IX(4)), (MF,IOFIX(5)), (MC,IOFIX(6)), (NCUT,IOFIX(7)), (NVER,IOFIX
     2(8)), (K,IOFIX(9)), (ITER,IOFIX(10)), (INVC,IOFIX(11)), (NUMVR,IOF
     3IX(12)), (NUMPV,IOFIX(13)), (INFS,IOFIX(14)), (JT,IOFIX(15)), (LA,
     4IOFIX(16)), (ZZ(1),TPIV), (ZZ(2),TZERO), (ZZ(3),TCOST), (ZZ(4),TEC
     5OL)
C-----
C-----                    MOVE INPUTS  ...  ZERO OUTPUTS
      DO 1 I=1,8
      TERR(I)=0.0
      IOFIX(I+8)=0
1     IOFIX(I)=INFIX(I)
```

```
      DO 2 I=1,4
2     ZZ(I)=TOL(I)
      PMIX=PRM
      TCOST=-ABS(TCOST)
      M2=M**2
      INFS=1
      LA=0
C-----                CHECK FOR ILLEGAL INPUT
      IF (N) 7,7,3
3     IF (M-MF) 7,7,4
4     IF (MF-MC) 7,7,5
5     IF (MC) 7,7,6
6     IF (ME-M) 7,8,8
7     K=7
      GO TO 28
8     IF (MOD(INFLAG,4)-1) 9,10,11
9     CALL NEW (M,N,JH(1),KB(1),A(1),B(1),MF,ME)
10    CALL VER (A(1),B(1),JH(1),X(1),E(1),KB(1),Y(1),N,ME,M,MF,INVC,NUMV
     1R,NUMPV,INFS,LA,TPIV,TECOL,M2)
C-----                          PERFORM ONE ITERATION
11    CALL XCK (M,MF,JH(1),X(1),TZERO,JIN)
C-----
C-----         CHECK CHANGE OF PHASE.. GO BACK TO INVERT IF GONE INFEAS.
      IF (INFS-JIN) 10,14,12
C-----               BECOME FEASIBLE
12    INFS=0
13    PMIX=0.0
14    CALL GET (M,MC,MF,JH(1),X(1),P(1),E(1),INFS,PMIX)
      CALL MIN (JT,N,M,A(1),P(1),KB(1),ME,TCOST)
      JM=JT
      J=JM
      IF (JM) 15,15,16
C-----         ALL COSTS NON-NEGATIVE...  K = 3 OR 4
15    K=3+INFS
      GO TO 18
C-----                          NORMAL CYCLE
16    CALL JMY (J,A(1),E(1),M,Y(1),ME)
      CALL ROW (IR,M,MF,JH(1),X(1),Y(1),TPIV)
C-----                          TEST PIVOT
      IF (IR) 17,17,19
C-----                          NO PIVOT
17    K=5
18    IF (PMIX) 13,24,13
C-----                      ITERATION LIMIT FOR CUT OFF
19    IF (ITER-NCUT) 20,23,23
C-----                          PIVOT FOUND
20    CALL PIV (IR,Y(1),M,E(1),X(1),NUMPV,TECOL)
      JOLD=JH(IR)
      IF (JOLD) 22,22,21
21    KB(JOLD)=0
22    KB(JM)=IR
      JH(IR)=JM
      LA=0
      ITER=ITER+1
      WRITE (6,31) ITER,X(1)
      INVC=INVC+1
```

```
C-----                        INVERSION FREQUENCY
      IF (INVC-NVER) 11,10,11
C-----                                       CUT OFF ... TOO MANY ITERATIONS
23    K=6
24    CALL ERR (M,A(1),B(1),TERR(1),JH(1),X(1),P(1),Y(1),ME,LA)
      IF (LA) 26,25,26
25    LA=4
      IF (INFLAG-4) 10,26,26
26    IF (K-5) 28,27,28
27    CALL JMY (J,A(1),E(1),M,Y(1),ME)
C-----                 SET EXIT VALUES
28    DO 29 I=1,8
29    ERS(I)=TERR(I)
      DO 30 I=1,7
30    KOUT(I)=IOFIX(I+8)
      RETURN
C-----
C-----
31    FORMAT (1H ,6HITER =,I6,8H  X(1) =,E15.5)
      END
```

```
      SUBROUTINE DEL (JM,DT,M,A,P,ME)
C-----DELS              DELTA-JAY.  PRICES OUT ONE MATRIX COLUMN
      DIMENSION A(1), P(1)
C-----DIMENSION A(1), P(1)
C-----
1     DT=0.0
      KDEL=(JM-1)*ME
C-----
      DO 4 IDEL=1,M
      KDEL=KDEL+1
      IF (A(KDEL)) 2,4,2
2     IF (P(IDEL)) 3,4,3
3     DT=DT+P(IDEL)*A(KDEL)
4     CONTINUE
C-----
      RETURN
      END
```

```
C-----ERRS               ERROR CHECK.  COMPARES AX WITH B, PA WITH ZERO
      SUBROUTINE ERR (M,A,B,TERR,JH,X,P,Y,ME,LA)
      DIMENSION JH(1), A(1), B(1), X(1), P(1), Y(1), TERR(8)
C-----DIMENSION JH(1), A(1), B(1), X(1), P(1), Y(1), TERR(8)
C-----                             STORE AX-B AT Y
      DO 1 I=1,M
1     Y(I)=-B(I)
```

```
      DO 5 I=1,M
      JA=JH(I)
      IF (JA) 2,5,2
2     IA=ME*(JA-1)
      DO 4 IT=1,M
      IA=IA+1
      IF (A(IA)) 3,4,3
3     Y(IT)=Y(IT)+X(I)*A(IA)
4     CONTINUE
5     CONTINUE
C-----                        FIND SUM AND MAXIMUM OF ERRORS
      DO 9 I=1,M
      YI=Y(I)
      IF (JH(I)) 7,6,7
6     YI=YI+X(I)
7     TERR(LA+1)=TERR(LA+1)+ABS(YI)
      IF (ABS(TERR(LA+2))-ABS(YI)) 8,9,9
8     TERR(LA+2)=YI
9     CONTINUE
C-----                        STORE P TIMES BASIS AT DT
      DO 12 I=1,M
      JM=JH(I)
      IF (JM) 10,12,10
10    CALL DEL (JM,DT,M,A(1),P(1),ME)
      TERR(LA+3)=TERR(LA+3)+ABS(DT)
      IF (ABS(TERR(LA+4))-ABS(DT)) 11,12,12
11    TERR(LA+4)=DT
12    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GET (M,MC,MF,JH,X,P,E,INFS,PMIX)
C----- LABEL
C-----GETS       GET PRICES
      DIMENSION JH(1), X(1), P(1), E(1)
C-----DIMENSION JH(1), X(1), P(1), E(1)
C-----
1     MMM=MC
C-----                   PRIMAL PRICES
      DO 2 J=1,M
      P(J)=E(MMM)
2     MMM=MMM+M
      IF (INFS) 3,11,3
C-----                   COMPOSITE PRICES
3     DO 4 J=1,M
4     P(J)=P(J)*PMIX
      DO 10 I=MF,M
      MMM=I
      IF (X(I)) 5,7,7
5     DO 6 J=1,M
      P(J)=P(J)+E(MMM)
```

```
6       MMM=MMM+M
        GO TO 10
7       IF (JH(I)) 10,8,10
8       DO 9 J=1,M
        P(J)=P(J)-E(MMM)
9       MMM=MMM+M
10      CONTINUE
C-----
11      RETURN
        END
```

```
C-----JMYS              J MULTIPLY.  BASIS INVERSE * COLUMN J
      SUBROUTINE JMY (JT,A,E,M,Y,ME)
      DIMENSION A(1), E(1), Y(1)
C-----DIMENSION A(1), E(1), Y(1)
C-----
1     DO 2 I=1,M
2     Y(I)=0.0
      LP=JT*ME-ME
      LL=0
      DO 6 I=1,M
      LP=LP+1
      IF (A(LP)) 3,5,3
3     DO 4 J=1,M
      LL=LL+1
4     Y(J)=Y(J)+A(LP)*E(LL)
      GO TO 6
5     LL=LL+M
6     CONTINUE
      RETURN
      END
```

```
C-----MINS              MIN D-J.  SELECTS COLUMN TO ENTER BASIS
      SUBROUTINE MIN (JT,N,M,A,P,KB,ME,TCOST)
      DIMENSION A(1), P(1), KB(1)
C-----DIMENSION A(1), P(1), KB(1)
C-----
1     JT=0
      DA=TCOST
C-----
      DO 4 JM=1,N
C-----                         SKIP COLUMNS IN BASIS
      IF (KB(JM)) 4,2,4
2     CALL DEL (JM,DT,M,A(1),P(1),ME)
      IF (DT-DA) 3,4,4
3     DA=DT
```

```
      JT=JM
4     CONTINUE
      RETURN
      END
```

```
C-----NEWS              STARTS PHASE ONE
      SUBROUTINE NEW (M,N,JH,KB,A,B,MF,ME)
      DIMENSION JH(1), KB(1), A(1), B(1)
C-----DIMENSION JH(1), KB(1), A(1), B(1)
C-----                            INITIATE
1     DO 2 I=1,M
2     JH(I)=0
C-----                INSTALL SINGLETONS
      KT=0
      DO 8 J=1,N
      KB(J)=0
      KTA=KT+MF
      KTB=KT+M
C-----                            TALLY ENTRIES IN CONSTRAINTS
      KQ=0
      DO 4 L=KTA,KTB
      IF (A(L)) 3,4,3
3     KQ=KQ+1
      LQ=L
4     CONTINUE
C-----                            CHECK WHETHER J IS CANDIDATE
      IF (KQ-1) 8,5,8
5     IQ=LQ-KT
      IF (JH(IQ)) 8,6,8
6     IF (A(LQ)*B(IQ)) 8,7,7
C-----                            J IS CANDIDATE. INSTALL
7     JH(IQ)=J
      KB(J)=IQ
8     KT=KT+ME
      RETURN
      END
```

```
C-----PIVS              PIVOT.  PIVOTS ON GIVEN ROW
      SUBROUTINE PIV (IR,Y,M,E,X,NUMPV,TECOL)
      DIMENSION Y(1), E(1), X(1)
C-----DIMENSION Y(1), E(1), X(1)
C-----                      LEAVE TRANSFORMED COLUMN IN Y(I)
C-----
1     NUMPV=NUMPV+1
C-----
      T2=-Y(IR)
      Y(IR)=-1.0
      LL=0
C-----                               TRANSFORM INVERSE
      DO 5 JP=1,M
      L=LL+IR
      IF (ABS(E(L))-TECOL) 2,2,3
2     LL=LL+M
      GO TO 5
3     T3=E(L)/T2
      E(L)=0.0
      DO 4 I=1,M
      LL=LL+1
4     E(LL)=E(LL)+T3*Y(I)
5     CONTINUE
C-----                               TRANSFORM X
      T3=X(IR)/T2
      X(IR)=0.0
      DO 6 I=1,M
6     X(I)=X(I)+T3*Y(I)
C-----                          RESTORE Y(IR)
      Y(IR)=-T2
C-----
      RETURN
      END
```

```
C-----ROWS        ROW SELECTION--COMPOSITE
      SUBROUTINE ROW (IR,M,MF,JH,X,Y,TPIV)
      DIMENSION JH(1), X(1), Y(1)
C-----DIMENSION JH(1), X(1), Y(1)
C-----
C-----AMONG EQS. WITH X=0, FIND MAX ABS(Y) AMONG ARTIFICIALS, OR, IF NON
C-----GET MAX POSITIVE Y(I) AMONG REALS.
1     IR=0
      AA=0.0
      IA=0
      DO 10 I=MF,M
      IF (X(I)) 10,2,10
2     YI=ABS(Y(I))
      IF (YI-TPIV) 10,10,3
3     IF (JH(I)) 4,6,4
4     IF (IA) 10,5,10
5     IF (Y(I)) 10,10,7
6     IF (IA) 7,8,7
7     IF (YI-AA) 10,10,9
8     IA=1
9     AA=YI
      IR=I
10    CONTINUE
      IF (IR) 21,11,21
11    AA=1.0E+20
C-----             FIND MIN. PIVOT AMONG POSITIVE EQUATIONS
```

```
      DO 16 IT=MF,M
      IF (Y(IT)-TPIV) 16,16,12
12    IF (X(IT)) 16,16,13
13    XY=X(IT)/Y(IT)
      IF (XY-AA) 15,14,16
14    IF (JH(IT)) 16,15,16
15    AA=XY
      IR=IT
16    CONTINUE
C-----FIND PIVOT AMONG NEGATIVE EQUATIONS, IN WHICH X/Y IS LESS THAN THE
C-----MINIMUM X/Y IN THE POSITIVE EQUATIONS, THAT HAS THE LARGEST ABSF(Y
      BB=-TPIV
      DO 20 I=MF,M
      IF (X(I)) 17,20,20
17    IF (Y(I)-BB) 18,20,20
18    IF (Y(I)*AA-X(I)) 19,19,20
19    BB=Y(I)
      IR=I
20    CONTINUE
21    RETURN
      END
```

```
C-----VERS                    FORMS INVERSE FROM  KB
      SUBROUTINE VER (A,B,JH,X,E,KB,Y,N,ME,M,MMF,INVC,NUMVR,NUMPV,INFS,L
     1A,TPIV,TECOL,M2)
      DIMENSION A(1), B(1), JH(1), X(1), E(1), KB(1), Y(1)
C-----DIMENSION A(1), B(1), JH(1), X(1), E(1), KB(1), Y(1)
      MF=MMF
C-----               INITIATE
      IF (LA) 1,1,2
1     INVC=0
2     NUMVR=NUMVR+1
      DO 3 I=1,M2
3     E(I)=0.0
      MM=1
      DO 4 I=1,M
      E(MM)=1.0
      X(I)=B(I)
4     MM=MM+M+1
      DO 6 I=MF,M
      IF (JH(I)) 5,6,5
5     JH(I)=12345
6     CONTINUE
      INFS=1
C-----               FORM INVERSE
      DO 13 J=1,N
      IF (KB(J)) 7,13,7
7     CALL JMY (J,A(1),E(1),M,Y(1),ME)
C-----                         CHOOSE PIVOT
      TY=0.0
      DO 10 I=MF,M
      IF (JH(I)-12345) 10,8,10
8     IF (ABS(Y(I))-TY) 10,10,9
9     IR=I
      TY=ABS(Y(I))
10    CONTINUE
C-----                 TEST PIVOT
      IF (TY-TPIV) 11,12,12
C-----   BAD  PIVOT,  ROW IR,  COLUMN J
11    KB(J)=0
      GO TO 13
C-----                 PIVOT
12    JH(IR)=J
      KB(J)=IR
      CALL PIV (IR,Y(1),M,E(1),X(1),NUMPV,TECOL)
13    CONTINUE
C-----             RESET ARTIFICIALS
      DO 15 I=1,M
      IF (JH(I)-12345) 15,14,15
14    JH(I)=0
15    CONTINUE
      RETURN
      END
```

```
C-----XCKS        X CHECKER
      SUBROUTINE XCK (M,MF,JH,X,TZERO,JIN)
      DIMENSION JH(1), X(1)
C-----DIMENSION JH(1), X(1)
C-----
C-----        RESET X AND CHECK FOR INFEASIBILITIES
1     JIN=0
      DO 6 I=MF,M
      IF (ABS(X(I))-TZERO) 2,3,3
2     X(I)=0.0
      GO TO 6
3     IF (X(I)) 5,6,4
4     IF (JH(I)) 6,5,6
5     JIN=1
6     CONTINUE
      RETURN
      END
```

```
PROGRAM LINPROGO  TEST PROBLEM FROM WRITE-UP
    3    6    1
FORMAT ROW  6 (2I5,6F10.0)
    1    1         4         7         8         6         0         0
    2    1       150       140       170       160         0         0
    3    1       0.1       0.1       0.3       0.3         1         0
    4    1         2         4         5         3         0        -1

       150       0.2         3
```

```
      PROGRAM LINPROG    TEST PROBLEM FROM WRITE-UP

   TABLEAU

          1          2          3          4          5          6

    1   4.000E+00  7.000E+00  8.000E+00  6.000E+00  0.         0.
    2   1.500E+02  1.400E+02  1.700E+02  1.600E+02  0.         0.
    3   1.000E-01  1.000E-01  3.000E-01  3.000E-01  1.000E+00  0.
    4   2.000E+00  4.000E+00  5.000E+00  3.000E+00  0.        -1.000E+00
ITER =     1  X(1) =    -4.80000E-04
ITER =     2  X(1) =    -5.26829E-04

      PROGRAM LINPROG    TEST PROBLEM FROM WRITE-UP

           KOUT        ERROR          INFIX        TOL

     1       3      1.1666015E-16       0       1.00000E-06

     2       2     -6.1149003E-17       6       1.00000E-05

     3       2      4.2632564E-14      82      -1.00000E-03

     4       2      2.8421709E-14       4       1.00000E-10

     5       6      1.1850330E-16       2       0.

     6       0     -6.1257423E-17       1       0.

     7       0      0.                200       0.

     8       0      0.                  8       0.

      PROGRAM LINPROG    TEST PROBLEM FROM WRITE-UP

           JJH          BBAR            PI                RHS              YTEMP

     1    X(  0) =  -5.2682927E-04   1.0000000E+00   0.                5.2682927E-04

     2    X(  1) =   5.8536585E-05  -9.7560976E-03   1.5000000E-02  -5.5511151E-17

     3    X(  5) =   1.9998317E-01   0.               2.0000000E-01  -6.1257423E-17

     4    X(  3) =   3.6585366E-05  -1.2682927E+00   3.0000000E-04  -1.7347235E-18

      PROGRAM LINPROG    TEST PROBLEM FROM WRITE-UP

   INVERSE MATRIX

          1          2          3          4

    1   1.000E+00 -9.756E-03  0.        -1.268E+00
    2   0.         1.220E-02  0.        -4.146E-01
    3   0.         2.439E-04  1.000E+00 -6.829E-02
    4   0.        -4.878E-03  0.         3.659E-01

      PROGRAM LINPROG    TEST PROBLEM FROM WRITE-UP
```

```
          KB        REDUCED C.
    1      2       0.
    2      0       5.6097561E-01
    3      4       0.
    4      0       6.3414634E-01
    5      3       0.
    6      0       1.2682927E+00
```

Subroutine KUTTA

GENERAL PURPOSE

To numerically solve a set of Ordinary Differential Equations (ODE) of the form

$$\dot{Y}_i = f_i\ (t, \underset{\sim}{Y}, C_1(t), C_2(t), \ldots C_m(t)),\ i = 1, \ldots, n$$

Where $\dot{Y}_i$ is the ith dependent variable

$\underset{\sim}{Y}$ is a vector whose n components are the $\dot{Y}_i$'s

t is time

$C_j(t)$, $j = 1, \cdots m$ is a set of extrinsic functions of time, some of which might be constants.

METHOD OF OPERATION

KUTTA has arguments (XL, XU, Y, NE, DEL, ACC, IMAX, EQUA). Y is dimensioned 25, other variables are scalars. When KUTTA is called by a main program, Y should contain the NE values (NE = n) which represent the state of the system at time XL. KUTTA will provide an estimate of Y at time XU. The estimating method is a 5th order Runge-Kutta technique with an error estimation procedure designed to ensure that the error in the estimation of the components of Y will be less than ACC*Y(I) for the Ith component. DEL is a number, less than or equal to (XU - XL), which is the users' estimate of the proper time step to use for obtaining the desired accuracy (as set by ACC) with a 5th order R-K procedure.

KUTTA will use a time step either greater or smaller than DEL, depending upon its error estimation procedure. The step size will start as DEL and be successively halved (or doubled) until the necessary accuracy is reached. If more than IMAX halvings (or doublings) are required for the requested accuracy, KUTTA will return to the main program with the values of Y unmodified. On our return to the main program IMAX will be set to the number of iterations (halvings or doublings of DEL) which were required. An error check in the main program can be made by comparing the actual number of iterations with the initial value of IMAX. If these are the same, it is likely that the numerical method has failed. This is most likely due to discontinuous functions for the derivatives, or functions which change very erratically between XL and XU (assuming that all difficulties due strictly to coding have been solved).

The independent variable, of course, need not be time. KUTTA can be used to solve partial differential equations and integro-differential equations by reformulating these equations as ODE's.

EQUA is the name of a subroutine which must be declared EXTERNAL in the main program. EQUA has arguments (T, Y, YP) where YP and Y is dimensioned 25. EQUA must use T, Y, and the $C_j(t)$ functions, transmitted either through COMMON or as additional subprograms, to evaluate $\dot{Y}_i$, i = 1, NE in array YP.

The device whereby a subroutine name is transmitted through an argument list and declared EXTERNAL in the main program is little used in FORTRAN IV but very convenient in this application. The sample program outline below illustrates the technique whereby the name of the subroutine used by KUTTA to evaluate the derivatives is declared to be EQ1 when KUTTA is called:

```
      PROGRAM ODE
      COMMON/LI/NT, NE, LAMBDA (25, 25)
      EXTERNAL EQ1
......................
      T = 0.
      DO 10 I = 1, NT
      T = T + DT
......................
      CALL KUTTA (T - DT, T, Y, NE, DT*.1, .01, 50, EQ1)
......................
   10 PRINT 100, T, Y
......................
```

```
      END
      SUBROUTINE KUTTA (XL, XU, Y, NE, DEL, ACC, IMAX, EQUA)
......................
      CALL EQUA (T, Y, YP)
......................
      END
      SUBROUTINE EQI (T, Y, YP)
      COMMON/LI/NT, NE, LAMBDA (25, 25)
......................
      DO 10 I = 1, NE
      YP(I) = 0.
      DO 10 J = 1, NE
   10 YP(I) = YP(I) + Y(J) * LAMBDA(I, J)
      RETURN
      END
......................
```

Though KUTTA calls EQUA, transfer is actually made to EQI.

The accompanying FORTRAN IV program (ODE) illustrates the use of KUTTA to solve a set of *linear*, homogeneous, ordinary differential equations. KUTTA can be used to solve other (nonlinear) ODE's by replacing EQI with an appropriate routine. We have tested KUTTA with a variety of nonlinear equations, both partial and ordinary, and with integro-differential equations of the Volterra and Fredholm types and found it to be reliably accurate and rapid. Its speed is dependent upon the complexity of the equation to be solved and thus avoids much of the need for multiple integration techniques. The only restriction is that the derivatives must be continuous functions of time, though the routine works in some cases when this restriction is violated.

KUTTA was written by F. D. Hammerling, ORGDP, Oak Ridge National Laboratory.

```
      PROGRAM LINDIFF
C     SOLVES A SET OF LINEAR DIFFERENTIAL EQUATIONS
C     NCOM      NO OF EQUATIONS
C     NT        NO OF TIME POINTS
C     T1        INITIAL TIME VALUE
C     TDEL      TIME STEP FOR PRINTOUT
C     ACC       DESIRED ACCURACY IN SOLUTION
C     DEL       ESTIMATE OF TIME STEP  TO BE USED FOR INTEGRATION
C     ITER      NO OF ITERATIONS, MAXIMUM, TO BE ALLOWED IN INTEGRATION
C     FMT       FORMAT STATEMENT TO BE USED TO READ INITIAL CONDITIONS
C               AND MATRIX OF COEFFICIENT VALUES
C     V         VECTOR OF INITIAL CONDITION VALUES
C     F         MATRIX OF COEFFICIENT VALUES
      DIMENSION FMT(8),V(20)
      COMMON/L1/F(20,20),NCOM
      EXTERNAL EQ1
C     READ AND PRINT INPUT DATA
    1 READ 100,NCOM,NT,T1,TDEL,ACC,DEL,ITER
      PRINT 200,T1,TDEL,NT,NCOM,ACC,DEL,ITER
      READ 300,FMT
      READ FMT,(V(I),I=1,NCOM)
      READ FMT,((F(I,J),J=1,NCOM),I=1,NCOM)
      PRINTFMT,(V(I),I=1,NCOM)
      PRINT 600
      PRINTFMT,((F(I,J),J=1,NCOM),I=1,NCOM)
C     SET UP OUTPUT FORMAT
      N=NCOM+1
      ENCODE(15,400,FMT) N
  400 FORMAT(*(F20.4,*I2*F12.4)*)
      PRINT 500
C     ITERATE THROUGH TIME AND FIND EQUATION SOLUTION
      T=T1-TDEL
      DO 10 I=1,NT
      T=T+TDEL
      T2=T+TDEL
      PRINT FMT,T,(V(I),I=1,NCOM)
      CALL KUTTA(T,T2,V,NCOM,DEL,ACC,ITER,EQ1)
   10 CONTINUE
      GO TO 1

  100 FORMAT(2I5,2F10.3/2F10.3,I5)
  200 FORMAT(*1LINEAR DIFFERENTIAL EQUATION SOLUTION*//* TIME STARTS AT
     1*F10.3*, INCREMENTS BY *F10.3*, FOR  *I6* ITERATIONS*//* *
     2I6* COMPARTMENT SYSTEM*//* ACCURACY OF SOLUTION*E15.4/* TIME STEP
     3SIZE*E15.4/* MAXIMUM ITERATIONS/STEP*I6//* INITIAL VALUES AND MATR
     4IX*//)
  300 FORMAT(8A10)
  500 FORMAT(*1*//*          TIME, COMPARTMENT VALUES, IN ORDER*//)
  600 FORMAT(//)
      END
      SUBROUTINE EQ1(X,Z,ZP)
C     FINDS DERIVATIVES FOR A SET OF LINEAR DIFF. EQS.
C     X -- CURRENT TIME, Z -- CURRENT DEP VAR VALUE, ZP -- CURRENT
C     DERIVATIVE VALUES TO BE FOUND BY MATRIX MULTIPLICATION
      COMMON/L1/LAMBDA(20,20),N
      REAL LAMBDA
      DIMENSION Z(1),ZP(1)
      DO 10 I=1,N
      ZP(I)=0.
      DO 10 J=1,N
   10 ZP(I)=ZP(I)+LAMBDA(J,I)*Z(J)
      RETURN
      END
      SUBROUTINE KUTTA(XL,XU,Y,NE,DEL,ACCURC,IMAX,EQUA)
C PROGRAM AUTHOR F.D.HAMMERLING, CENTRAL DATA PROCESSING, ORGUP.
      DIMENSION Y(1),YI(25),YN(25),K1(25),K2(25),K3(25),
     1          K4(25),K5(25),F(25),E(25),F1(25)
      REAL K1,K2,K3,K4,K5
      LOGICAL QUIT
      ITTER=0
      N=NE
      XN=XL
      H=DEL
      QUIT=.FALSE.
      DO 1 I=1,N
    1 YN(I)=Y(I)
    2 IF(XN+H.LT.XU)GO TO 3
      DEL=H
      H=XU-XN
      QUIT=.TRUE.
    3 CALL EQUA(XN,YN,F1)
    4 DO 5 I=1,N
      K1(I)=H*F1(I)/3.
    5 YI(I)=YN(I)+K1(I)
      CALL EQUA(XN+H/3.,YI,F)
      DO 6 I=1,N
      K2(I)=H*F(I)/3.
    6 YI(I)=YN(I)+K1(I)/2.+K2(I)/2.
      CALL EQUA(XN+H/3.,YI,F)
      DO 7 I=1,N
      K3(I)=H*F(I)/3.
    7 YI(I)=YN(I)+3.*K1(I)/8.+9.*K3(I)/8.
      CALL EQUA(XN+H/2.,YI,F)
      DO 8 I=1,N
      K4(I)=H*F(I)/3.
    8 YI(I)=YN(I)+3.*K1(I)/2.-9.*K3(I)/2.+6.*K4(I)
      CALL EQUA(XN+H,YI,F)
      TEST=0.0
      DO 9 I=1,N
      K5(I)=H*F(I)/3.
      E(I)=(K1(I)-9.*K3(I)/2.+4.*K4(I)-K5(I)/2.)/5.
      TEST=AMAX1(TEST,ABS(E(I)))
    9 CONTINUE
      IF(TEST.LT.ACCURC)GO TO 10
      ITTER=ITTER+1
      IF(ITTER.GE.IMAX)GO TO 13
      H=H/2.
      QUIT=.FALSE.
      GO TO 4
   10 DO 11 I=1,N
   11 YN(I)=YN(I)+(K1(I)+4.*K4(I)+K5(I))/2.
      XN=XN+H
      IF(TEST.LT.ACCURC/32.)H=2.*H
      IF(.NOT.QUIT)GO TO 2
      DO 12 I=1,N
   12 Y(I)=YN(I)
      IMAX=0.
      GO TO 14
   13 DEL=H $ IMAX=ITTER
   14 RETURN
      END
    4  201          40.           .1
.00001     .025          50
(4F10.4)
7.0568     4.3436     7.6290     80.972
-.1        .0333333   .0333333   .0333333
0.         -.1        .05        .05
.05        0.         -.1        .05
0.         0.         0.         0.

 CP SEC=000003 PP SEC=000013 PR=000008 PU=000000 RD=000176 PL=000000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   119250   %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   119250   %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   119250   %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   119250   %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   119250   %%%%%%%%%%
```

APPENDIX I

OTHER USEFUL PROGRAMS

I. STAT03C - Factorial Analysis of Covariance

This program computes a full factorial analysis of covariance. Output includes:

1. The total covariance matrix and a breakdown into its full factorial design components.
2. The components in 1 adjusted by residuals.
3. Inverses of the covariate parts of the adjusted components.
4. Regression coefficients
5. t-values, an F-statistic, and the residual mean square.
6. An analysis-of-variance table for the factorial components of the design.

Limitations in the program are:

1. v, number of AOV classifications ($1 \leq v \leq 6$)
2. p, number of covariates ($1 \leq p \leq 8$)
3. r, number of replicates ($r \leq 999$)
4. L_i, number of categories or levels of any one classification ($L_i \leq 999$) and ($L_1 \times L_2 \times L_3 \times \ldots \times L_v \leq 1200$)
5. q, number of covariates generated. ($-7 \leq q \leq 7$) and [$0 < (p+q) \leq 8$]

II. BMD07D Description of Strata with Histograms

This program groups the data into a specified number of groups based on the order of entry of the data or into groups whose values for a base variable are w/in intervals established by specified cut points. For these groups, histograms are printed for each variable. The number of classes or categories of the histograms may be specified or they may be computed by the program. Means, standard deviations and correlation coefficients are computed for each group; means and standard deviations are computed also for the combined groups of a variable. Special values may be specified for all variables, except the base variable, to exclude certain values to codes from computations.

Output includes:

1. Input data after transgeneration
2. Input data after ordering from high to low on the specified base variable
3. Histograms for each variable showing the frequencies of distribution of C classes over the g groups.
4. Correlation matrices for each group
5. Means and standard deviations
6. Tabulations of special values.

Limitations in the program are:

1. p, number of variables ($1 \leq p \leq 100$)
2. n, number of observations ($1 \leq n \leq 9999$)
3. g, number of groups ($0 \leq g \leq 10$)
4. c, number of classes ($5 \leq c \leq 30$)
5. m_i, number of special values for variable i ($0 \leq m_i \leq 5$)
6. q, number of variables added in transgeneration ($0 \leq q \leq 99$)
7. T, number of group cut points ($1 \leq T \leq 9$)
8. n(p+q), [$n(p+q) \leq 19,000$]

III. BMD10D Data Patterns for Dichotomies

This program finds frequencies and patterns of any one particular specified code in the input data. Frequent use will be for a code representing missing values. The program prints 0's to designate the specified code, or missing values, and 1's to designate and other values. A data matrix of 0's and 1's is printed. Frequencies of the specified code, or missing values are computed; and the cases having the specified code are identified by item numbers which correspond to the order in which the cards appear in the data input deck. If desired, patterns of data may be obtained also after eliminating each variable in turn. Thus, patterns are available for the p + 1 different choices of p variables.

Output includes.

1. Patterns of data (0's for the specified code or missing values; 1's for all other values). Tabulated by numbers of missing values, and item numbers to identify cases.
2. Data matrix of 0's and 1's

Limitations in the program are:

1. p, number of variable ($1 \leq p \leq 30$)
2. n, number of cases or items ($1 \leq n \leq 700$)

IV. STAT31S Chi Square Analysis

This program computes the Chi Square test for independence in two way contingency tables. In addition, tests for goodness of fit are calculated for one way classifications.

Output includes:

1. Listing of original table with observed and theoretical frequency as well as the contributions to the test statistic.
2. Calculated x^2 value and degrees of freedom
3. Tabled values of x_α^2 for comparison
4. Marginal tests of row and column totals.

The limitation in the program is that it is for two way tables no larger than 10 x 10.

V. STAT02V ANOVA for Balanced Factorial Design

This program computes an analysis of variance for a balanced factorial design.

Output includes:

1. Analysis of variance table and the grand mean.
2. A breakdown of the sums of squares into orthogonal polynomial components for as many as four main effects and all of their first order interactions.
3. Main effects and first order interactions for the variables specified in 2.
4. Cell means and variances, main effect means and plots.
5. Two factor and three factor interaction means if there are more than two factors.

Limitations in the program are:

1. F, number of factors (independent variables) or ways ($F \leq 8$)
2. R, number of observations per cell ($R \leq 999$)
3. L_i, Number of categories or levels of any one factor ($L_i \leq 999$) and ($L_1 \times L_2 \times L_3 \times \ldots \times L_F \leq 4000$).

VI. STAT31V ANOVA for Unequal Subclass Frequencies

This program performs an analysis of variance for data from an unbalanced (unequal subclass frequencies) three (or two) way classification. The method used is one in *Topics in Intermediate Statistical Methods* by T. A. Bancroft (1968). The analysis consists of two steps, the preliminary ANOVA and the final ANOVA.

If interactions are known not to exist in the population or at least one subclass is empty, the method of fitting constants is used to test main effects in the final ANOVA.

If interactions are known to be present in the population, the method of weighted square of means is used to investigate main effects in the final ANOVA.

If it is not known whether interactions are present or not, a test for it is included in the preliminary ANOVA. The method of fitting constants or the method of weighted square of means is used on the outcome of the test of whether interactions are absent or present. In this situation, the tests on main effects will not exactly be at α level because the test for interaction is a preliminary test which precedes the test for main effects. Test for significance of interaction should, in general, be made at .25 significance level to insure th main effect subsequent tests to be made at the .05 level.

Output includes the preliminary and final ANOVA. Subclass means and variances and marginal totals are optional.

Limitations in the program are that a problem may have a maximum of:

1. three factors
2. 20 levels for each factor
3. 100 observations in each subclass.