

A Computer Program for Subset
Selection in Regression Analysis (IBM Version)

by

G. P. McCabe, J. N. Arvesen, and
T. R. Wong

October 10, 1974

I. INTRODUCTION

This is intended as a computer supplement to "A Subset Selection Procedure for Regression Variables," by G. P. McCabe and J. N. Arvesen, appearing in the Journal of Statistical Computation and Simulation, 1974, Vol. 5, pp. 137-146. The original version of their computer program is described in "A Computer Program for Subset Selection in Regression Analysis," by G. P. McCabe, J. N. Arvesen and R. J. Pohl, Purdue University Statistics Department Mimeo Series No. 317, May 1973. The following writeup describes a revised program which was implemented on an IBM 360/91 at Columbia University. The original version was implemented on a CIC 6500.

The major changes to the original program are:

1. The ability to substitute the correlation matrix in lieu of the input data.
2. The ability to specify specific subsets of independent variables.
3. The ability to process different sets of data in the same run.
4. The use of the Box-Muller normal random number generator. The IBM SSP pseudo random number generator RANDU is used. Since there is a plethora of better pseudo random number generators, we leave it to the user to substitute if he desires.

III. SEQUENCE OF CARDS FOR PROCESSING

- A. Data description card
- B. Parameter card
- C. Critical 'A' value output cards
- D. Specific subsets cards (Optional)
- E. Input format card
- F. Data cards (Optional)

Repeat A-F as needed for different sets of data.

III. PREPARATION OF INPUT CARDS

- A. Data description card

<u>COL.</u>	<u>DESCRIPTION</u>
1-8	DATADESC
9-80	Description used for labeling output

- B. Parameter card

<u>COL.</u>	<u>DESCRIPTION</u>
1-5	Number of observations.
6-10	Number of independent variables (no more than 10).
11-15	Desired subset size.
16-20	Device number for input data. If data is on cards, enter 5; otherwise, enter 8.
21-25	Number of 'A' values to be calculated.

- 26-30 Output option:
 0--Input data and correlation matrix are not printed out.
 1--Input variables are printed.
 2--Both input data and correlation matrix are printed.
- 31-39 Positive odd integer used as seed for random number generator.
- 41 If not zero, input data is correlation matrix.
- 43 If not zero, specific subsets of independent variables are read in.

C. Critical 'A' value output cards

<u>COL.</u>	<u>DESCRIPTION</u>
1-5	Number of pairs (no more than 10) of lower and upper limits for printing subsets of the sorted sequence of stochastically generated 'A' values.
6-10	Lower limit for 1st pair.
11-15	Upper limit for 1st pair.
16-20	Lower limit for 2nd pair.
21-25	Upper limit for 2nd pair.
.	
.	
.	
76-80	Lower limit for 9th pair.
	Continue on 2nd card if necessary.
1-5	Upper limit for 9th pair.
6-10	Lower limit for 10th pair.
11-15	Upper limit for 10th pair.

D. Specific subsets cards (Optional)

<u>COL.</u>	<u>DESCRIPTION</u>
1-3	Number of specific subsets of independent variables (no more than 10). These are entered as <u>hexidecimal</u> (i.e. base 16) images.
	e.g. For 10 independent variables, if a specific subset consists of variables 1, 6 and 8, the bit pattern is ...10100001. Hence, the hexidecimal image is 000000A1.
9-18	1st subset
19-27	2nd subset
.	
.	
.	
73-80	9th subset.

Continue on 2nd card if necessary.

1-8 10th subset.

E. Input format card

<u>COL.</u>	<u>DESCRIPTION</u>
1-80	Fortran type format statement used to read input data.

F. Data cards (Optional)

1. If original observations are being used, all values of the independent variables for observation 1 are read first, followed by those for observation 2, and so forth. Each set of observations is punched beginning on a new card.

2. If the correlation matrix is being used, each row is punched beginning on a new card. Because of symmetry, only enter those values from the lower portion of the matrix.
- e.g. To enter a 3x3 correlation matrix, the 3 data cards could appear as follows:

Card 1	XXX.X		
Card 2	XXX.X	XXX.X	
Card 3	XXX.X	XXX.X	XXX.X

Note: The observations or correlations, if necessary, can reside on a magnetic tape or a direct access storage device. Just be sure to denote the device number as 8.

IV. SAMPLE INPUT/OUTPUT AND PROGRAM LISTING

Input Description:

1. Card 1 describes the problem data set.
2. Card 2 tells the program:

.there are 16 observations
.there are 6 independent variables
.the subset size to be considered is 3
.the data is read from cards (i.e. device number 5)
.the number of 'A' values to be calculated is 100
.2 means that input matrix 'X' and the correlation matrix is to be printed
.the seed for generating random numbers is 123456789
.the original observations are to be read in as the input data
.the program is to find the 'best' subset of size 3

3. Card 3 asks for the printing of 3 groups of 'A' variables: 49-51, 89-91, 94-96.
4. Card 4 is the variable format card.
5. Card 5-Card 20 contain the input data. Note that the first number on these data cards is ignored.

Output Description:

1. The 6 independent variables are listed.
2. The standardized correlation matrix is printed out.
Since the matrix is symmetric, only the lower half is printed out.
3. The best subset is displayed.
4. The 'A' values are listed.

V. AUXILIARY PROGRAMS

The following subroutines are from the IMSL Library:

.VSORTA	.LUDECP
.LINVIP	.LUELMP
.LETZLP	.UERTST

For further information contact

International Mathematical and Statistical Libraries, Inc.
GNB Bldg.-Sixth Floor
7500 Bellaire
Houston, Texas 77036

CARD	1-->	DATADESC	LONGLEV	DATA	SET			
CARD	2-->	16	6	3	5	100	2123456789	
CARD	3-->		3	49	51	89	91	94 96
CARD	4-->	(10X.6F10+3)						
CARD	5-->	60323.	83.	234289.	2356.	1590.	107608.	1947.
CARD	6-->	61122.	88.5	259426.	2325.	1456.	108632.	1948.
CARD	7-->	60171.	88.7	258054.	3642.	1615.	109773.	1949.
CARD	8-->	61187.	89.1	284599.	3351.	1650.	110929.	1950.
CARD	9-->	63221.	96.2	328975.	2099.	3099.	112075.	1951.
CARD	10-->	61639.	98.1	346099.	1932.	3594.	113270.	1952.
CARD	11-->	64989.	99.	365285.	1070.	3547.	115094.	1953.
CARD	12-->	63761.	100.	363112.	3578.	3350.	116219.	1954.
CARD	13-->	66019.	101.2	397469.	2904.	3048.	117388.	1955.
CARD	14-->	67857.	104.6	419180.	2822.	2857.	118734.	1956.
CARD	15-->	68169.	108.4	442769.	2936.	2798.	120445.	1957.
CARD	16-->	66513.	110.8	444546.	4681.	2637.	121950.	1958.
CARD	17-->	68655.	112.6	482704.	3813.	2552.	123336.	1959.
CARD	18-->	69964.	114.2	502601.	3931.	2514.	125360.	1960.
CARD	19-->	69331.	115.7	518173.	4806.	2572.	127052.	1961.
CARD	20-->	70551.	116.9	554894.	4007.	2827.	130081.	1962.

DATA DESCRIPTION : LONGLEY DATA SET

NO. OF OBSERVATIONS	16
NO. OF INDEPENDENT VARIABLES (EXCLUDES INTERCEPT)	6
SUBSET SIZE	3
NO. OF AT VARIABLES DESIRED	100

INPUT DATA MATRIX

0.8300000000 02	0.2342890000 05	0.2356000003 04	0.1590003003 04	0.1076020000 05	0.1947002002 04
0.8850000000 02	0.2594260009 05	0.2325000003 04	0.1456003000 04	0.1086320000 06	0.1949003000 04
0.8820000000 02	0.2580540000 05	0.3682000003 04	0.1616003003 04	0.1097730000 06	0.1949003000 04
0.8950000000 02	0.2845990000 04	0.3451000001 04	0.1650000000 04	0.1109290000 06	0.1950000000 04
0.9620000000 02	0.3289750000 06	0.2099000003 04	0.3099000002 04	0.1120750000 06	0.1951000000 04
0.9410000000 02	0.3469990000 06	0.1932000007 04	0.3594000007 04	0.1132700000 06	0.1952000000 04
0.9900000000 02	0.3653950000 06	0.1870000003 04	0.3547000003 04	0.1150940000 06	0.1953000000 04
0.1000000000 03	0.3631120003 04	0.3578000003 04	0.3350000009 04	0.1162190000 06	0.1954000000 04
0.1012000000 03	0.3974690002 06	0.2904000003 04	0.3048000009 04	0.1173890000 06	0.1955000000 04
0.1046000000 03	0.4191800003 05	0.2822300003 04	0.2857000007 04	0.1187340000 06	0.1956000000 04
0.1044000000 03	0.4427690003 06	0.2931000002 04	0.2798000007 04	0.1204450000 06	0.1957000000 04
0.1108000000 03	0.4445460007 04	0.4681000003 04	0.2637000007 04	0.1219500000 06	0.1958000000 04
0.1126000000 03	0.4827040003 06	0.3813000003 04	0.2552000003 04	0.1233360000 06	0.1959000000 04
0.1142000000 01	0.5026010002 06	0.3931000003 04	0.2514000007 04	0.1257680000 06	0.1960000000 04
0.1157000000 03	0.5181730000 06	0.4806000003 04	0.2572000009 04	0.1278520000 06	0.1961000000 04
0.1169000000 03	0.5548940000 06	0.4007000003 04	0.2827000003 04	0.1300810000 06	0.1962000000 04

CORRELATION MATRIX OF STANDARDIZED DATA

OPTIMUM SUBSET SELECTED FROM THE 6 INDEPENDENT VARIABLES

(INTERCEPT ALREADY INCLUDED) 1 2 3 4

E POINT A VARIABLE

49	0.1125516344D 01
50	0.1126922761D 01
51	0.1130968931D 01
89	0.1438269193D 01
90	0.1492157517D 01
91	0.1495323862D 01
94	0.1569926047D 01
95	0.1648867596D 01
96	0.1966478542D 01

NORMAL TERMINATION: NO MORE DATA SETS

```

C ***** R2SSP001
C ***** R2SSP002
C ***** R2SSP003
C ***** R2SSP004
C ***** R2SSP005
C ***** R2SSP006
C ***** R2SSP007
C ***** R2SSP008
C ***** R2SSP009
C ***** R2SSP010
C ***** R2SSP011
C ***** R2SSP012
C ***** R2SSP013
C ***** R2SSP014
C ***** R2SSP015
C ***** R2SSP016
C ***** R2SSP017
C ***** R2SSP018
C ***** R2SSP019
C ***** R2SSP020
C ***** R2SSP021
C ***** R2SSP022
C ***** R2SSP023
C ***** R2SSP024
C ***** R2SSP025
C ***** R2SSP026
C ***** R2SSP027
C ***** R2SSP028
C ***** R2SSP029
C ***** R2SSP030
C ***** R2SSP031
C ***** R2SSP032
C ***** R2SSP033
C ***** R2SSP034
C ***** R2SSP035
C ***** R2SSP036
C ***** R2SSP037
C ***** R2SSP038
C ***** R2SSP039
C ***** R2SSP040
C ***** R2SSP041
C ***** R2SSP042
C ***** R2SSP043
C ***** R2SSP044
C ***** R2SSP045
C ***** R2SSP046
C ***** R2SSP047
C ***** R2SSP048
C ***** R2SSP049
C ***** R2SSP050

C THIS PROGRAM IS AN ALGORITHM FOR THE SUBSET SELECTION
C PROCEDURE DESCRIBED IN THE PAPER
C * A SUBSET SELECTION PROCEDURE FOR REGRESSION VARIABLES *
C BY GEORGE MCCABE AND JAMES ARVESEN
C DEPARTMENT OF STATISTICS
C PURDUE UNIVERSITY
C WEST LAFAYETTE, IN 47907
C FEBRUARY 1973
C Mimeo Series 316

C REFERENCES ARE MADE IN THE DOCUMENTATION OF THIS PROGRAM TO
C THE CORRESPONDING SECTIONS OF THE PAPER.

C PROGRAMER -- RICHARD POHL
C DEPARTMENT OF STATISTICS
C PURDUE UNIVERSITY
C FEBRUARY 1973

C ***** INPUT INFORMATION *****
C
C THE INFORMATIONAL INPUT IS FROM DEVICE 5. THE DATA IS
C READ IN VIA DEVICE 5 OR DEVICE 8. ALL OUTPUT IS ONTO DEVICE 6.

C N IS THE NUMBER OF OBSERVATIONS.
C NP IS THE NUMBER OF INDEPENDENT VARIABLES (MUST BE LESS
C THAN OR EQUAL TO TEN).
C IT IS THE DESIRED SUBSET SIZE.
C IN SPECIFIES THE TAPE FROM WHICH THE DATA IS TO BE READ
C IF 'IN' IS NOT 8 THEN IT IS SET TO 5.
C M IS THE NUMBER OF 'A' VALUES TO BE CALCULATED.
C [CURRENTLY M MUST BE LESS THAN OR EQUAL TO 1000. IF
C A GREATER NUMBER IS DESIRED THEN CHANGE THE
C DIMENSION OF 'A']
C IDPT IS AN OUTPUT OPTION. IF IDPT=
C   0 THEN THE INPUT DATA AND THE XTX MATRIX ARE NOT
C     PRINTED OUT.
C   1 THEN THE INPUT VARIABLES ARE PRINTED.
C   2 THEN BOTH THE INPUT DATA AND THE XTX MATRIX ARE
C     PRINTED.
C ICOR IF NON-ZERO, INPUT DATA IS CORRELATION MATRIX.
C IXNUMB IF NON-ZERO, USER SPECIFIES REGRESSION SUBSETS.
C FMT IS THE FORMAT OF THE INDEPENDENT VARIABLES TO BE READ
C IN.
C NLIST IS THE NUMBER OF BLOCKS OF THE 'A' VARIABLES TO BE
C PRINTED. CURRENTLY NLIST MUST BE LESS THAN OR

```

```

C EQUAL TO TEN. *R2SSP051
C ILOW AND IUP ARE THE LOWER AND UPPER LIMITS OF THE NLIST *P2SSP052
C BLOCKS OF 'A' TO BE PRINTED. *R2SSP053
C *R2SSP054
C ***** *R2SSP055
C THE FOLLOWING SUBROUTINES ARE CALLED BY THE PROGRAM-- *P2SSP056
C
C (A) PROGRAMS WRITTEN OR REVISED FOR THIS PROGRAM. *P2SSP057
C NUMB IS A FUNCTION SUBPROGRAM WHICH RETURNS THE REFERENCE *P2SSP058
C TO AN R-SQUARED VALUE TO BE OBTAINED FROM SCNA. *R2SSP059
C PRMUT IS THE SUBROUTINE WHICH RETURNS THE PERMUTATIONS ON A *D2SSP060
C SET OF NUMBERS. *D2SSP061
C SCNA IS A SUBROUTINE WHICH CALCULATES ALL R-SQUARES FOR A *R2SSP062
C REGRESSION PROBLEM. *R2SSP063
C INDX IS A FUNCTION WHICH RETURNS THE SYMMETRIC STORAGE *D2SSP064
C MODE POSITION OF AN ELEMENT IN A MATRIX. *P2SSP065
C SUBSET PRINTS HEXADECIMAL IMAGE OF A SPECIFIC SUBSET OF *P2SSP066
C INDEPENDENT VARIABLES. *R2SSP067
C
C GAUSS IS A SUBROUTINE WHICH GENERATES NORMAL(0,1) *R2SSP068
C RANDOM NUMBERS VIA THE METHOD OF BOX-MULLER. *P2SSP069
C RANDU IS A SUBROUTINE WHICH GENERATES UNIFORM(0,1) NUMBERS *R2SSP070
C AND IS CALLED BY GAUSS. *R2SSP071
C **NOTE** SINCE RANDOM NUMBER GENERATORS ARE MACHINE DEPEND- *P2SSP072
C ENT, MODIFICATION WILL BE NEEDED TO RUN THIS PRO- *R2SSP073
C GRAM ON ANOTHER SYSTEM. *R2SSP074
C THE USER MAY WISH TO REPLACE THE CALL TO RANDU IN *P2SSP075
C THE GAUSS SUBROUTINE BY ANOTHER UNIFORM RANDOM *R2SSP076
C NUMBER GENERATOR OR THE USER MAY REPLACE GAUSS BY *P2SSP077
C ANOTHER NORMAL(0,1) RANDOM NUMBER GENERATOR. *R2SSP078
C
C (B) PUCC SUBPROGRAMS-- *R2SSP079
C LINEQ1 IS A SUBROUTINE WHICH SOLVES THE SYSTEM OF LINEAR *P2SSP080
C EQUATIONS A*X=B FOR X, WHERE A,X,AND B ARE ARRAYS. *R2SSP081
C LINEQ1 CALLS A FUNCION ARITH1. *P2SSP082
C
C (C) SUBPROGRAMS FROM THE IMSL LIBRARY. *R2SSP083
C VSORTA IS A SUBROUTINE WHICH SORTS THE ELEMENTS OF A *P2SSP084
C VECTOR FROM LOW TO HIGH VALUE. *R2SSP085
C LINVIP IS A SUBROUTINE WHICH CALCULATES THE INVERSE OF *R2SSP086
C A MATRIX IN SYMMETRIC STORAGE MODE. *R2SSP087
C THE FOLLOWING SUBROUTINES ARE CALLED BY LINVIP-- *P2SSP088
C
C LETQ1P *R2SSP089
C LUDECP *R2SSP090
C LUELMP *R2SSP091
C UERTST *R2SSP092
C ***** *R2SSP093
C
C MATRIX XTX IS STORED IN SYMMETRIC STORAGE MODE TO SAVE *P2SSP094
C SPACE. THE ELEMENTS ARE ORDERED AS (A11,A21,A22,A31,A32, *R2SSP095
C *R2SSP096
C ***** *R2SSP097
C *R2SSP098
C *R2SSP099
C *R2SSP100
C *R2SSP101

```

```

C      A33,...} THE ELEMENT I,J OF XTX IS IN THE POSITION          *R2SSP102
C      I*(I-1)/2+J WHERE I IS LESS THAN OR EQUAL TO J.          *R2SSP103
C      SINCE MATRIX L IS LOWER TRIANGULAR IT IS ALSO STORED IN    *P2SSP104
C      SYMMETRIC STORAGE MODE.                                     *P2SSP105
C
C*****P2SSP106
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C IDUMMY FORCES PROPER ALIGNMENT IN COMMON AREA
COMMON /BLOCK/ IDUMMY, IJK1(1023), RL1(1035), RL2(1331)          P2SSP108
COMMON /IWX/ IX
REAL*8 L(78),LABEL(9)                                              R2SSP109
INTEGER XNUMBS(100)
LOGICAL CORREL, XNUMB
DIMENSION X(11),XTX(78),A(1000),XXTX(78)                         R2SSP110
COMMON /SCRBLK/ JDUMMY,NP,XTX,IT,NUMBR,RSQ1,RSQ2                  R2SSP111
DIMENSION FMT(10),ILOW(10),IUP(10)                                    P2SSP112
SORTID)=DSORTID(D)                                                 R2SSP113
C
C      601 FORMAT(1H1)                                              P2SSP114
12345 CONTINUE
READ(5,499,END=54321) LABEL
499 FORMAT(8X,9A8)
READ(5,500) N,NP,IT,IN,M,IOPT,IX,ICOR,IXNUMB
500 FORMAT(6I5,I9,1X,I1,1X,I1)                                         P2SSP123
CORREL=.FALSE.
IFI( ICOR.NE.0 ) CORREL=.TRUE.
READ(5,501) NLIST,(ILOW(I),IUP(I),I=1,NLIST)                      P2SSP124
501 FORMAT(16I5)
XNUMB=.FALSE.
IFI( IXNUMB.NE.0 ) XNUMB=.TRUE.
IFI( .NOT.XNUMB ) GO TO 55055
READ(5,503) NXNUMB,(XNUMBS(I),I=1,NXNUMB)                         P2SSP125
503 FORMAT(13,5X,9Z8/(1028))
55055 CONTINUE
READ(5,502) FMT
502 FORMAT(10A8)
WRITE(6,50505) LABEL,N,NP,IT,M
50505 FORMAT('1DATA DESCRIPTION : ',9A8/'-',6X,
1 'NO. OF OBSERVATIONS .....',15/'0',5X,                                R2SSP131
2 'NO. OF INDEPENDENT VARIABLES/1IX,'(EXCLUDES INTERCEPT) .....',15/'0',5X,   R2SSP132
3 ',',15/'0',5X,'SUBSET SIZE .',7('...'),15/'0',5X,                         R2SSP133
4 'NO. OF "A" VARIABLES DESIRED ....',15)                                 R2SSP134
IFI(IN.NE.8) IN=5
IFI(NP.GT.10) STOP
C
C*****P2SSP135
C
C      INITIALIZE VALUES
C
C*****P2SSP136
C
C*****P2SSP137
C
C*****P2SSP138
C
C*****P2SSP139
C
C*****P2SSP140
C
C*****P2SSP141
C
C*****P2SSP142
C
C*****P2SSP143
C
C*****P2SSP144
C
C*****P2SSP145
C
C*****P2SSP146
C
C*****P2SSP147
C
C*****P2SSP148
C
C*****P2SSP149
C
C*****P2SSP150
C
C*****P2SSP151
C
C*****P2SSP152

```

```

C
  IP=NP+1          P2SSP153
  NB=NP+2          R2SSP154
  N8A=N8*(NB+1)/2 R2SSP155
  N8B=N8*(NB-1)/2 R2SSP156
  ANIP=FLOAT(N-IP) R2SSP157
  AA=DSQRT(2./9./ANIP)
  AB=2./9./ANIP+1  R2SSP158
  XTX(1)=N          R2SSP159
  DO 1 I=2,78      R2SSP160
  1 XTX(I)=0.0      R2SSP161
C                                         R2SSP162
C                                         R2SSP163
C                                         R2SSP164
C***** R2SSP165
C
C   READ DATA AND FORM XTX          *R2SSP166
C
C***** R2SSP168
C
C IS INPUT CORRELATION MATRIX ?    P2SSP169
  IF( .NOT.CORREL ) GO TO 99099  P2SSP170
  WRITE(6,99001)                 R2SSP171
  99001 FORMAT('1INPUT CORRELATION MATRIX//')
  L1=2                         R2SSP172
  DO 99000 I=1,NP               R2SSP173
  L1=L1 + 1                     R2SSP174
  L2=L1 + I - 1                R2SSP175
  READ(IN,FMT) (XTX(J),J=L1,L2) R2SSP176
  WRITE(6,99002) (XTX(J),J=L1,L2) R2SSP177
  99002 FORMAT(1X,7E18.9)        R2SSP178
  99000 CONTINUE
  GO TO 99990
  99099 WRITE(6,99199)
  99199 FORMAT('1INPUT DATA MATRIX//')
  DO 10 J=1,N                  R2SSP180
  READ(IN,FMT) (X(J),J=2,IP)    R2SSP181
  DO 5 J=2,IP                  R2SSP182
  JSUB=J*(J-1)/2                R2SSP183
  XTX(JSUB+1)=XTX(JSUB+1)+X(J) R2SSP184
  DO 5 JJ=J,IP                 R2SSP185
  JSUB=(JJ-1)*JJ/2              R2SSP186
  5 XTX(JSUB+JJ)=XTX(JSUB+JJ)+X(J)*X(JJ) R2SSP187
  IF(IOPT.GE.1) WRITE(6,600) (X(J),J=2,IP) R2SSP188
  10 CONTINUE
  600 FORMAT(1X,7E18.9)
  DO 105 J=2,IP
  JSUB=J*(J-1)/2
  DO 105 JJ=J,IP
  JSUB=JJ*(JJ-1)/2
  JSUB1=JSUB+J
  JSUB1=JSUB+1
  JSUB2=JSUB+J
  JSUB2=JSUB+1
  JSUB3=JSUB+J
  JSUB3=JSUB+1
  JSUB4=JSUB+J
  JSUB4=JSUB+1
  JSUB5=JSUB+J
  JSUB5=JSUB+1
  JSUB6=JSUB+J
  JSUB6=JSUB+1
  JSUB7=JSUB+J
  JSUB7=JSUB+1
  JSUB8=JSUB+J
  JSUB8=JSUB+1
  JSUB9=JSUB+J
  JSUB9=JSUB+1
  JSUB10=JSUB+J
  JSUB10=JSUB+1
  JSUB11=JSUB+J
  JSUB11=JSUB+1
  JSUB12=JSUB+J
  JSUB12=JSUB+1
  JSUB13=JSUB+J
  JSUB13=JSUB+1
  JSUB14=JSUB+J
  JSUB14=JSUB+1
  JSUB15=JSUB+J
  JSUB15=JSUB+1
  JSUB16=JSUB+J
  JSUB16=JSUB+1
  JSUB17=JSUB+J
  JSUB17=JSUB+1
  JSUB18=JSUB+J
  JSUB18=JSUB+1
  JSUB19=JSUB+J
  JSUB19=JSUB+1
  JSUB20=JSUB+J
  JSUB20=JSUB+1
  JSUB21=JSUB+J
  JSUB21=JSUB+1
  JSUB22=JSUB+J
  JSUB22=JSUB+1
  JSUB23=JSUB+J
  JSUB23=JSUB+1
  JSUB24=JSUB+J
  JSUB24=JSUB+1
  JSUB25=JSUB+J
  JSUB25=JSUB+1
  JSUB26=JSUB+J
  JSUB26=JSUB+1
  JSUB27=JSUB+J
  JSUB27=JSUB+1
  JSUB28=JSUB+J
  JSUB28=JSUB+1
  JSUB29=JSUB+J
  JSUB29=JSUB+1
  JSUB30=JSUB+J
  JSUB30=JSUB+1
  JSUB31=JSUB+J
  JSUB31=JSUB+1
  JSUB32=JSUB+J
  JSUB32=JSUB+1
  JSUB33=JSUB+J
  JSUB33=JSUB+1
  JSUB34=JSUB+J
  JSUB34=JSUB+1
  JSUB35=JSUB+J
  JSUB35=JSUB+1
  JSUB36=JSUB+J
  JSUB36=JSUB+1
  JSUB37=JSUB+J
  JSUB37=JSUB+1
  JSUB38=JSUB+J
  JSUB38=JSUB+1
  JSUB39=JSUB+J
  JSUB39=JSUB+1
  JSUB40=JSUB+J
  JSUB40=JSUB+1
  JSUB41=JSUB+J
  JSUB41=JSUB+1
  JSUB42=JSUB+J
  JSUB42=JSUB+1
  JSUB43=JSUB+J
  JSUB43=JSUB+1
  JSUB44=JSUB+J
  JSUB44=JSUB+1
  JSUB45=JSUB+J
  JSUB45=JSUB+1
  JSUB46=JSUB+J
  JSUB46=JSUB+1
  JSUB47=JSUB+J
  JSUB47=JSUB+1
  JSUB48=JSUB+J
  JSUB48=JSUB+1
  JSUB49=JSUB+J
  JSUB49=JSUB+1
  JSUB50=JSUB+J
  JSUB50=JSUB+1
  JSUB51=JSUB+J
  JSUB51=JSUB+1
  JSUB52=JSUB+J
  JSUB52=JSUB+1
  JSUB53=JSUB+J
  JSUB53=JSUB+1
  JSUB54=JSUB+J
  JSUB54=JSUB+1
  JSUB55=JSUB+J
  JSUB55=JSUB+1
  JSUB56=JSUB+J
  JSUB56=JSUB+1
  JSUB57=JSUB+J
  JSUB57=JSUB+1
  JSUB58=JSUB+J
  JSUB58=JSUB+1
  JSUB59=JSUB+J
  JSUB59=JSUB+1
  JSUB60=JSUB+J
  JSUB60=JSUB+1
  JSUB61=JSUB+J
  JSUB61=JSUB+1
  JSUB62=JSUB+J
  JSUB62=JSUB+1
  JSUB63=JSUB+J
  JSUB63=JSUB+1
  JSUB64=JSUB+J
  JSUB64=JSUB+1
  JSUB65=JSUB+J
  JSUB65=JSUB+1
  JSUB66=JSUB+J
  JSUB66=JSUB+1
  JSUB67=JSUB+J
  JSUB67=JSUB+1
  JSUB68=JSUB+J
  JSUB68=JSUB+1
  JSUB69=JSUB+J
  JSUB69=JSUB+1
  JSUB70=JSUB+J
  JSUB70=JSUB+1
  JSUB71=JSUB+J
  JSUB71=JSUB+1
  JSUB72=JSUB+J
  JSUB72=JSUB+1
  JSUB73=JSUB+J
  JSUB73=JSUB+1
  JSUB74=JSUB+J
  JSUB74=JSUB+1
  JSUB75=JSUB+J
  JSUB75=JSUB+1
  JSUB76=JSUB+J
  JSUB76=JSUB+1
  JSUB77=JSUB+J
  JSUB77=JSUB+1
  JSUB78=JSUB+J
  JSUB78=JSUB+1
  JSUB79=JSUB+J
  JSUB79=JSUB+1
  JSUB80=JSUB+J
  JSUB80=JSUB+1
  JSUB81=JSUB+J
  JSUB81=JSUB+1
  JSUB82=JSUB+J
  JSUB82=JSUB+1
  JSUB83=JSUB+J
  JSUB83=JSUB+1
  JSUB84=JSUB+J
  JSUB84=JSUB+1
  JSUB85=JSUB+J
  JSUB85=JSUB+1
  JSUB86=JSUB+J
  JSUB86=JSUB+1
  JSUB87=JSUB+J
  JSUB87=JSUB+1
  JSUB88=JSUB+J
  JSUB88=JSUB+1
  JSUB89=JSUB+J
  JSUB89=JSUB+1
  JSUB90=JSUB+J
  JSUB90=JSUB+1
  JSUB91=JSUB+J
  JSUB91=JSUB+1
  JSUB92=JSUB+J
  JSUB92=JSUB+1
  JSUB93=JSUB+J
  JSUB93=JSUB+1
  JSUB94=JSUB+J
  JSUB94=JSUB+1
  JSUB95=JSUB+J
  JSUB95=JSUB+1
  JSUB96=JSUB+J
  JSUB96=JSUB+1
  JSUB97=JSUB+J
  JSUB97=JSUB+1
  JSUB98=JSUB+J
  JSUB98=JSUB+1
  JSUB99=JSUB+J
  JSUB99=JSUB+1
  JSUB100=JSUB+J
  JSUB100=JSUB+1
  JSUB101=JSUB+J
  JSUB101=JSUB+1
  JSUB102=JSUB+J
  JSUB102=JSUB+1
  JSUB103=JSUB+J
  JSUB103=JSUB+1
  JSUB104=JSUB+J
  JSUB104=JSUB+1
  JSUB105=JSUB+J
  JSUB105=JSUB+1
  JSUB106=JSUB+J
  JSUB106=JSUB+1
  JSUB107=JSUB+J
  JSUB107=JSUB+1
  JSUB108=JSUB+J
  JSUB108=JSUB+1
  JSUB109=JSUB+J
  JSUB109=JSUB+1
  JSUB110=JSUB+J
  JSUB110=JSUB+1
  JSUB111=JSUB+J
  JSUB111=JSUB+1
  JSUB112=JSUB+J
  JSUB112=JSUB+1
  JSUB113=JSUB+J
  JSUB113=JSUB+1
  JSUB114=JSUB+J
  JSUB114=JSUB+1
  JSUB115=JSUB+J
  JSUB115=JSUB+1
  JSUB116=JSUB+J
  JSUB116=JSUB+1
  JSUB117=JSUB+J
  JSUB117=JSUB+1
  JSUB118=JSUB+J
  JSUB118=JSUB+1
  JSUB119=JSUB+J
  JSUB119=JSUB+1
  JSUB120=JSUB+J
  JSUB120=JSUB+1
  JSUB121=JSUB+J
  JSUB121=JSUB+1
  JSUB122=JSUB+J
  JSUB122=JSUB+1
  JSUB123=JSUB+J
  JSUB123=JSUB+1
  JSUB124=JSUB+J
  JSUB124=JSUB+1
  JSUB125=JSUB+J
  JSUB125=JSUB+1
  JSUB126=JSUB+J
  JSUB126=JSUB+1
  JSUB127=JSUB+J
  JSUB127=JSUB+1
  JSUB128=JSUB+J
  JSUB128=JSUB+1
  JSUB129=JSUB+J
  JSUB129=JSUB+1
  JSUB130=JSUB+J
  JSUB130=JSUB+1
  JSUB131=JSUB+J
  JSUB131=JSUB+1
  JSUB132=JSUB+J
  JSUB132=JSUB+1
  JSUB133=JSUB+J
  JSUB133=JSUB+1
  JSUB134=JSUB+J
  JSUB134=JSUB+1
  JSUB135=JSUB+J
  JSUB135=JSUB+1
  JSUB136=JSUB+J
  JSUB136=JSUB+1
  JSUB137=JSUB+J
  JSUB137=JSUB+1
  JSUB138=JSUB+J
  JSUB138=JSUB+1
  JSUB139=JSUB+J
  JSUB139=JSUB+1
  JSUB140=JSUB+J
  JSUB140=JSUB+1
  JSUB141=JSUB+J
  JSUB141=JSUB+1
  JSUB142=JSUB+J
  JSUB142=JSUB+1
  JSUB143=JSUB+J
  JSUB143=JSUB+1
  JSUB144=JSUB+J
  JSUB144=JSUB+1
  JSUB145=JSUB+J
  JSUB145=JSUB+1
  JSUB146=JSUB+J
  JSUB146=JSUB+1
  JSUB147=JSUB+J
  JSUB147=JSUB+1
  JSUB148=JSUB+J
  JSUB148=JSUB+1
  JSUB149=JSUB+J
  JSUB149=JSUB+1
  JSUB150=JSUB+J
  JSUB150=JSUB+1
  JSUB151=JSUB+J
  JSUB151=JSUB+1
  JSUB152=JSUB+J
  JSUB152=JSUB+1
  JSUB153=JSUB+J
  JSUB153=JSUB+1
  JSUB154=JSUB+J
  JSUB154=JSUB+1
  JSUB155=JSUB+J
  JSUB155=JSUB+1
  JSUB156=JSUB+J
  JSUB156=JSUB+1
  JSUB157=JSUB+J
  JSUB157=JSUB+1
  JSUB158=JSUB+J
  JSUB158=JSUB+1
  JSUB159=JSUB+J
  JSUB159=JSUB+1
  JSUB160=JSUB+J
  JSUB160=JSUB+1
  JSUB161=JSUB+J
  JSUB161=JSUB+1
  JSUB162=JSUB+J
  JSUB162=JSUB+1
  JSUB163=JSUB+J
  JSUB163=JSUB+1
  JSUB164=JSUB+J
  JSUB164=JSUB+1
  JSUB165=JSUB+J
  JSUB165=JSUB+1
  JSUB166=JSUB+J
  JSUB166=JSUB+1
  JSUB167=JSUB+J
  JSUB167=JSUB+1
  JSUB168=JSUB+J
  JSUB168=JSUB+1
  JSUB169=JSUB+J
  JSUB169=JSUB+1
  JSUB170=JSUB+J
  JSUB170=JSUB+1
  JSUB171=JSUB+J
  JSUB171=JSUB+1
  JSUB172=JSUB+J
  JSUB172=JSUB+1
  JSUB173=JSUB+J
  JSUB173=JSUB+1
  JSUB174=JSUB+J
  JSUB174=JSUB+1
  JSUB175=JSUB+J
  JSUB175=JSUB+1
  JSUB176=JSUB+J
  JSUB176=JSUB+1
  JSUB177=JSUB+J
  JSUB177=JSUB+1
  JSUB178=JSUB+J
  JSUB178=JSUB+1
  JSUB179=JSUB+J
  JSUB179=JSUB+1
  JSUB180=JSUB+J
  JSUB180=JSUB+1
  JSUB181=JSUB+J
  JSUB181=JSUB+1
  JSUB182=JSUB+J
  JSUB182=JSUB+1
  JSUB183=JSUB+J
  JSUB183=JSUB+1
  JSUB184=JSUB+J
  JSUB184=JSUB+1
  JSUB185=JSUB+J
  JSUB185=JSUB+1
  JSUB186=JSUB+J
  JSUB186=JSUB+1
  JSUB187=JSUB+J
  JSUB187=JSUB+1
  JSUB188=JSUB+J
  JSUB188=JSUB+1
  JSUB189=JSUB+J
  JSUB189=JSUB+1
  JSUB190=JSUB+J
  JSUB190=JSUB+1
  JSUB191=JSUB+J
  JSUB191=JSUB+1
  JSUB192=JSUB+J
  JSUB192=JSUB+1
  JSUB193=JSUB+J
  JSUB193=JSUB+1
  JSUB194=JSUB+J
  JSUB194=JSUB+1
  JSUB195=JSUB+J
  JSUB195=JSUB+1
  JSUB196=JSUB+J
  JSUB196=JSUB+1
  JSUB197=JSUB+J
  JSUB197=JSUB+1
  JSUB198=JSUB+J
  JSUB198=JSUB+1
  JSUB199=JSUB+J
  JSUB199=JSUB+1
  JSUB200=JSUB+J
  JSUB200=JSUB+1
  JSUB201=JSUB+J
  JSUB201=JSUB+1
  JSUB202=JSUB+J
  JSUB202=JSUB+1
  JSUB203=JSUB+J
  JSUB203=JSUB+1

```

```

JSUB1=JSUB+1 R 2SSP204
JJSUBJJ=JJSUB+JJ R 2SSP205
XXN=XXN R 2SSP206
XNUM=XTX(JJSUBJJ)-XTX(JJSUB1)*XTX(JSUB1)/XXN R 2SSP207
DENOM=(XTX(JJSUBJJ)-XTX(JJSUB1)**2/XXN)*(XTX(JSUBJJ)-XTX( R 2SSP208
1JSUB1)**2/XXN) R 2SSP209
105 XXTX(JJSUB+JJ)=XNUM/(DSORT(DENOM)) P 2SSP210
DO 205 J=2,IP R 2SSP211
JSUB=J*(J-1)/2 R 2SSP212
205 XTX(JJSUB+1)=0* R 2SSP213
DO 251 J=2,IP R 2SSP214
DO 251 I=J,IP R 2SSP215
II=I*(I-1)/2 R 2SSP216
251 XTX(II+J)=XXTX(II+J) R 2SSP217
C R 2SSP218
C ***** P 2SSP219
C
C PRINT XTX IF DESIRED. *R 2SSP220
C *R 2SSP221
C *R 2SSP222
C ***** R 2SSP223
C
C 99990 CONTINUE R 2SSP224
IF(IOPT.NE.2) GO TO 50 P 2SSP225
IF(CORREL) GO TO 50 R 2SSP226
WRITE(6,9988)
99888 FORMAT('ICORRELATION MATRIX OF STANDARDIZED DATA')// P 2SSP227
DO 60 I=1,IP R 2SSP228
K=I*(I-1)/2 R 2SSP229
60 WRITE(6,640) (XTX(K+J),J=1,I) R 2SSP230
640 FORMAT(1X,7E10.9) R 2SSP231
C R 2SSP232
C ***** R 2SSP233
C ***** R 2SSP234
C ***** R 2SSP235
C
C FIND THE BIT PATTERN CORRESPONDING TO THE R-SQUARED FOR *R 2SSP236
C THE NUMERATOR OF A *R 2SSP237
C *R 2SSP238
C *R 2SSP239
C ***** R 2SSP240
C
C 50 CONTINUE R 2SSP241
IF(XNUMB) GO TO 60000 R 2SSP242
NUMBR=NUMB1XTX,NP,IT,N) R 2SSP243
L1=1 R 2SSP244
L2=1 R 2SSP245
XNUMBS(1)=NUMBR R 2SSP246
GO TO 66666 R 2SSP247
C R 2SSP248
C ***** R 2SSP249
C
C FOR A PROCEDURE DESCRIPTION SEE *R 2SSP250
C COMPUTATIONAL PROCEDURES *R 2SSP251
C (B) ESTIMATION OF C INVERSE *R 2SSP252
C *R 2SSP253
C *R 2SSP254

```

```

C      FACTOR XTX INTO LLT SUCH THAT L IS LOWER TRIANGULAR.          *P 2SSP255
C      *****                                                       *R 2SSP256
C      *****                                                       *R 2SSP257
C      *****                                                       *P 2SSP258
C
60000 CONTINUE                                         R 2SSP259
      WRITE(6,61116) NXNUMB                         R 2SSP260
61116 FORMAT(//'*-WILL PROCESS',I5,' SPECIFIED REGRESSION SUBSET(S)*') P 2SSP261
      L1=1                                         R 2SSP262
      L2=NXNUMB                                     R 2SSP263
C SAVE XTX FOR PROCESSING EXTRA NUMBRs
      NXXTX=IP*(IP+1)/2                           R 2SSP264
      DO 16661 I=1,NXXTX                          R 2SSP265
16661 XXTX(I)=XTX(I)                           R 2SSP266
16666 DO 77777 KX=L1,L2                      R 2SSP267
      NUMBR=XNUMBS(KX)                            R 2SSP268
      IF( XNUMB ) CALL SUBSET(NUMBR,NP,IT)        R 2SSP269
C
      IF( KX.GT.1 ) GO TO 60066                    R 2SSP270
      CALL LUDECP(XTX,L,IP,DET1,DET2,IERROR)    R 2SSP271
      IF(IERROR.NE.-129) GO TO 20                  R 2SSP272
      WRITE(6,6101)
610 FORMAT(' XTX IS SINGULAR')
      STOP                                         R 2SSP273
20 CONTINUE                                         R 2SSP274
      DO 49 I=1,IP                                R 2SSP275
      K=I*(I+1)/2                                 R 2SSP276
      49 L(IK)=1./L(IK)                           R 2SSP277
C
C      *****                                                       *P 2SSP278
C      *****                                                       *R 2SSP279
C      *****                                                       *R 2SSP280
C      *****                                                       *P 2SSP281
C      *****                                                       *R 2SSP282
C      *****                                                       *P 2SSP283
C
C      CALCULATE THE 'A' VARIABLS.                 *P 2SSP284
C
C      GENERATE THE INDEPENDENT VARIABLE COLUMNS FROM THE NORMAL
C      RANDOM NUMBER GENERATOR.                   *P 2SSP285
C
C      *****                                                       *R 2SSP286
C      *****                                                       *P 2SSP287
C      *****                                                       *R 2SSP288
C      *****                                                       *R 2SSP289
C
C      *****                                                       *R 2SSP290
C
60066 CONTINUE                                         R 2SSP291
      DO 100 LL=1,M
      ZAP=GAUSS(1.0,0.0)                         R 2SSP292
      XTX(NBA)=ANIP*(ZAP*AA+AB)**3             R 2SSP293
      DO 70 I=1,IP
      X(I)=GAUSS(1.0,0.0)                         R 2SSP294
      XTX(NBA)=XTX(NBA)+X(I)**2                R 2SSP295
      XTX(NBB+I)=0.0                               R 2SSP296
      K= I*(I-1)/2                                R 2SSP297
      DO 70 J=1,I
      70 XTX(NBB+I)=XTX(NBB+I)+L(K+J)*X(J)     R 2SSP298
      CALL SCRVA
      IF( NUMBR.EQ.0.45014520 ) GO TO 77177    R 2SSP299
      R 2SSP300
      R 2SSP301
      R 2SSP302
      R 2SSP303
      R 2SSP304
      R 2SSP305

```

```

A(LL)=1.0-RSQ2)/(1.0-RSQ1)          P2SSP306
100 CONTINUE                         P2SSP307
C*****                                         P2SSP308
C*****                                         P2SSP309
C*****                                         P2SSP310
C      SORT THE A VARIABLES AND OUTPUT THEM AS SPECIFIED.   *P2SSP311
C*****                                         P2SSP312
C*****                                         P2SSP313
C
CALL VSORTA(A,M)                     P2SSP314
WRITE(6,20202)                        P2SSP315
20202 FORMAT(1X-POINT*,I1X,'A VARIABLE') P2SSP316
DO 200 I=1,NLIST                     P2SSP317
IA=ILOM(I)
IB=IUP(I)
IFI ((IA.LE.0).OR.(IA.GT.M).OR.(IB.LE.0).OR.(IB.GT.M)) GO TO 210
200 WRITE(6,6201) (J,A(J),J=IA,IB)    P2SSP318
620 FORMAT(1X,I5.5X,E20.10)           P2SSP319
C RESTORE XTX                         P2SSP320
77177 IF(.NOT.XNUMB) GO TO 77777     P2SSP321
DO 71117 I=1,NXXTX                   P2SSP322
71117 XTX(I)=XXTX(I)                 P2SSP323
C                                         P2SSP324
77777 CONTINUE                         P2SSP325
C                                         P2SSP326
GO TO 12345                           P2SSP327
120 WRITE(6,6301) IA,IB                P2SSP328
630 FORMAT(//1X,2I5,' INVALID RANGE FOR OUTPUT OF A VARTABLES') P2SSP329
GO TO 12345                           P2SSP330
54321 CONTINUE                         P2SSP331
WRITE(6,55546)                         P2SSP332
55546 FORMAT('INORMAL TERMINATION; NO MORE DATA SETS') P2SSP333
STOP                                     P2SSP334
END                                      P2SSP335

```

```

      SUBROUTINE SUBSET(NUMBR,P,T)
C PRINTS OUT THE SUBSET CORRESPONDING TO THE BIT PATTERN OF 'NUMBR'.
C P = NUMBER OF INDEPENDENT VARIABLES
C T = SUBSET SIZE
C
      INTEGER P,T,SET(11),POW(11)
      DATA POW/1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024/
C
      N=NUMBR
      K=P
      KT=T+1
C
      100 IF( N.LT.POW(K) ) GO TO 200
      N=N-POW(K)
      KT=KT-1
      SET(KT)=K
      IF( N.LE.0 ) GO TO 300
      200 K=K-1
      GO TO 100
C
      300 WRITE(6,1) P,(SET(I),I=1,T)
      1 FORMAT('SPECIFIED SUBSET FROM THE ',I2,' INDEPENDENT VARIABLES'
      1 //6X,'(INTERCEPT ALREADY INCLUDED) .....',11I6)
C
      RETURN
      END
      SUBST001
      SUBST002
      SUBST003
      SUBST004
      SUBST005
      SUBST006
      SUBST007
      SUBST008
      SUBST009
      SUBST010
      SUBST011
      SUBST012
      SUBST013
      SUBST014
      SUBST015
      SUBST016
      SUBST017
      SUBST018
      SUBST019
      SUBST020
      SUBST021
      SUBST022
      SUBST023
      SUBST024
      SUBST025
      SUBST026
      SUBST027

```

```

FUNCTION NUMB(XTX,P,T,N)          NUMB0001
C*****                                         NUMB0002
C                                         *NUMB0003
C   FUNCTION NUMB FINDS THE INDEPENDENT VARIABLES OF THE
C   GIVEN SUBSET SIZE T TO BE USED IN FINDING
C   A= [(1.-RSQ(K))/(1.-RSQ(BEST))] IN THE MAINLINE
C   PROGRAM.                                     *NUMB0004
C                                         *NUMB0005
C                                         *NUMB0006
C                                         *NUMB0007
C                                         *NUMB0008
C                                         *NUMB0009
C                                         *NUMB0010
C                                         *NUMB0011
C                                         *NUMB0012
C   SEE COMPUTATIONAL PROCEDURES           *NUMB0013
C   (A) DETERMINATION OF K (PAGE 6-7)        *NUMB0014
C                                         *NUMB0015
C   XTX   IS X TRANSPOSE X WHERE X IS THE INPUT DATA.    *NUMB0016
C   XTX IS IN SYMMETRIC STORAGE MODE.          *NUMB0017
C   P    IS THE NUMBER OF INDEPENDENT VARIABLES.        *NUMB0018
C   T    IS THE GIVEN SUBSET SIZE.                 *NUMB0019
C   N    IS THE NUMBER OF OBSERVATIONS.          *NUMB0020
C                                         *NUMB0021
C                                         *NUMB0022
C*****                                         NUMB0023
C                                         NUMB0024
C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION XTX(1),X(11,11),Y(11,11),XY(11,11)      NUMB0025
C DIMENSION ZONE(11,11), ZTWO(11,11)                  NUMB0026
C DIMENSION C(11,11), R(11), E(11)                   NUMB0027
C INTEGER PERM(93,11),P,T,PA,TA,SELECT(11)           NUMB0028
C DUMMY FORCES ALIGNMENT                         NUMB0029
C COMMON /BLOCK/ IDUMMY, PERM, GAMMA(1035), GINV(1331)  NUMB0030
C EQUIVALENCE (Y(1),GINV(1)),(XY(1),GINV(122))       NUMB0031
C EQUIVALENCE (ZONE(1),GINV(364)),(ZTWO(1),GINV(485))  NUMB0032
C EQUIVALENCE (X(1),GINV(1210))                    NUMB0033
C EQUIVALENCE (C(1),GINV(606)),(R(1),GINV(727)),(E(1),GINV(738))  NUMB0034
C PA=P+1                                           NUMB0035
C TA=T+1                                           NUMB0036
C AA=N-2*T-2                                       NUMB0037
C AB=N-T-1                                         NUMB0038
C CALL PRMUT(PA,TA,K)                           NUMB0039
C GAMMA(1)=1.0                                      NUMB0040
C                                         NUMB0041
C*****                                         NUMB0042
C                                         *NUMB0043
C   FOR EACH I=1,...,K AND J=1,...,K THE PROGRAM
C   DETERMINES X,Y,AND XY WHERE X REFERS TO THOSE ELEMENTS
C   OF XTX WHICH HAVE THE ROW COORDINATES PERM(I,1),...,  *NUMB0044
C   PERM(I,TA) AND THE COLUMN COORDINATES PERM(I,1),...,  *NUMB0045
C   PERM(I,TA). Y IS FORMED THE SAME WAY EXCEPT THAT I IS  *NUMB0046
C   REPLACED BY J.                                     *NUMB0047
C                                         *NUMB0048
C                                         *NUMB0049
C                                         *NUMB0050

```

```

C      XY IS FORMED BY THE (TA BY TA) SUBMATRIX OF XTX WITH THE      *NUMB0051
C      ROW COORDINATES PERM(I,1),...,PERM(I,TA) AND THE COLUMN      *NUMB0052
C      COORDINATES PERM(J,1),...,PERM(J,TA).      *NUMB0053
C      *****      *****      *****      *****      *****      *****      *NUMB0054
C
C      DO 200 I=2,K      *NUMB0055
C      IM=I-1      *NUMB0056
C      INDEX=I*(I+1)/2      *NUMB0057
C      GAMMA(INDEX)=1.      *NUMB0058
C      *****      *****      *****      *****      *****      *****      *NUMB0059
C
C      DETERMINE X      *NUMB0060
C      *****      *****      *****      *****      *****      *****      *NUMB0061
C
C      DETERMINE Y AND XY      *NUMB0062
C      *****      *****      *****      *****      *****      *****      *NUMB0063
C      *****      *****      *****      *****      *****      *****      *NUMB0064
C      *****      *****      *****      *****      *****      *****      *NUMB0065
C      *****      *****      *****      *****      *****      *****      *NUMB0066
C
C      DO 50 LOOP=1,TA      *NUMB0067
C      INDEX=PERM(I,LOOP)      *NUMB0068
C      DO 50 LOOPA=LOOP,TA      *NUMB0069
C      INDEXA=PERM(I,LOOPA)      *NUMB0070
C      ISUB=INOX(INDEX,INDEXA)      *NUMB0071
C      X(LOOP,LOOPA)=XTX(ISUB)      *NUMB0072
C      X(LOOPA,LOOP)=XTX(ISUB)      *NUMB0073
C      50 CONTINUE      *NUMB0074
C      DO 150 J=1,IM      *NUMB0075
C      *****      *****      *****      *****      *****      *****      *NUMB0076
C
C      DETERMINE Y AND XY      *NUMB0077
C      *****      *****      *****      *****      *****      *****      *NUMB0078
C      *****      *****      *****      *****      *****      *****      *NUMB0079
C      *****      *****      *****      *****      *****      *****      *NUMB0080
C      *****      *****      *****      *****      *****      *****      *NUMB0081
C      *****      *****      *****      *****      *****      *****      *NUMB0082
C
C      DO 55 LOOPA=1,TA      *NUMB0083
C      INDEXA=PERM(J,LOOPA)      *NUMB0084
C      DO 54 LOOPB=LOOPA,TA      *NUMB0085
C      INDEXB=PERM(J,LOOPB)      *NUMB0086
C      ISUB=INOX(INDEXA,INDEXB)      *NUMB0087
C      Y(LOOPA,LOOPB)=XTX(ISUB)      *NUMB0088
C      Y(LOOPB,LOOPA)=XTX(ISUB)      *NUMB0089
C      54 CONTINUE      *NUMB0090
C      DO 55 LOOP=1,TA      *NUMB0091
C      INDEX=PERM(I,LOOP)      *NUMB0092
C      ISUB=INOX(INDEX,INDEXA)      *NUMB0093
C      55 XY(LOOP,LOOPA)=XTX(ISUB)      *NUMB0094
C      *****      *****      *****      *****      *****      *****      *NUMB0095
C
C      FIND ZONE SUCH THAT X*ZONE=XY      *NUMB0096
C      *****      *****      *****      *****      *****      *****      *NUMB0097
C
C      *****      *****      *****      *****      *****      *****      *NUMB0098
C      *****      *****      *****      *****      *****      *****      *NUMB0099
C      *****      *****      *****      *****      *****      *****      *NUMB0100
C
C      *****      *****      *****      *****      *****      *****      *NUMB0101

```

```

CALL LINEQ1(X,XY,ZONE,11,TA,TA,ISING,C,R,E)
IF(ISING.EQ.0) GO TO 60
WRITE(6,610) I,J
610 FORMAT(1X,'MATRIX X IS SINGULAR',5X,215)
STOP
C
C***** *****
C      TAKE TRANSPOSE OF XY MATRIX.
C
C***** *****
C
60 DO 65 LOOP=1,T
LOOPB=LOOP+1
DO 65 LOOPA=LOOPB,TA
SAVE=XY(LOOP,LOOPA)
XY(LOOP,LOOPA)=XY(LOOPA,LOOP)
65 XY(LOOPA,LOOP)=SAVE
C
C***** *****
C      FIND XTWO SUCH THAT Y*ZTWO=XY.
C
C***** *****
C
CALL LINEQ1(Y,XY,ZTWO,11,TA,TA,ISING,C,R,E)
IF(ISING.EQ.0) GO TO 70
WRITE(6,620) I,J
620 FORMAT(1X,'MATRIX Y IS SINGULAR',5X,215)
STOP
70 CONTINUE
C
C***** *****
C      FIND TRACE OF ZONE * ZTWO.
C      PERFORM OPERATION ON IT AND STORE IN GAMMA.
C
C***** *****
C
SUM=0.0
DO 75 LOOP=1,TA
DO 75 LOOPB=1,TA
75 SUM=SUM+ZONE(LOOP,LOOPB)*ZTWO(LOOPB,LOOP)
INDEX=I*(I-1)/2+J
GAMMA(INDEX)=(AA+SUM)/AB
150 CONTINUE
200 CONTINUE
C
C***** *****
C

```

```

C      TAKE INVERSE OF GAMMA          *NUMB0153
C
C*****CALL LINV1P(GAMMA,K,GINV,T,X,IER)          *NUMB0154
C      IF(IER.NE.-129) GO TO 210          *****NUMB0155
C      WRITE(6,630)          NUMB0156
C      630 FORMAT(' GAMMA DID NOT HAVE AN INVERSE')
C      STOP          NUMB0157
C
C*****FIND THE SMALLEST ELEMENT ON THE DIAGONAL OF GAMMA
C      INVERSE.          NUMB0158
C
C*****          NUMB0159
C      210 INDEX=1          NUMB0160
C      VALUE=GINV(1)          NUMB0161
C      DO 220 LOOP=2,K          NUMB0162
C      LOOPA=LOOP*(LOOP-1)/2+LOOP          *****NUMB0163
C      IF(GINV(LOOPA).GE.VALUE) GO TO 220          *NUMB0164
C      VALUE=GINV(LOOPA)
C      INDEX=LOOP          *NUMB0165
C      220 CONTINUE          *NUMB0166
C
C*****          *NUMB0167
C      THE ELEMENTS(MINUS 1) OF THE (INDEX)TH PERMUTATION IN
C      PERM (2,...,TA) REFER TO THE INDEPENDENT VARIABLES IN
C      THE DATA SET TO BE SEARCHED FOR IN CALCULATING THE
C      REGRESSION.
C      THE INDEPENDENT VARIABLES ARE STORED IN NUMB ACCORDING
C      TO THEIR BIT PATTERN.
C      (I.E., 101110 REFERS TO THE INDEPENDENT VARIABLES
C      6,4,3,2,1)
C
C*****          *NUMB0180
C      NUMB=0          *NUMB0181
C      DO 230 I=2,TA          *NUMB0182
C      J=PERM(INDEX,I)-2          *NUMB0183
C      SELECT(I-1)=J+1          *NUMB0184
C      230 NUMB=NUMB+2**J          *NUMB0185
C
C      WRITE(6,90000) P,(SELECT(I),I=1,T)          *NUMB0186
C      90000 FORMAT(I-OPTIMUM SUBSET SELECTED FROM THE ',I2,
C      1 ' INDEPENDENT VARIABLES',//6X,', (INTERCEPT ALREADY INCLUDED) ..... *NUMB0187
C      2-,11I6)          *NUMB0188
C      RETURN          *NUMB0189
C      END          *NUMB0190

```

```
FUNCTION INDX(I,J)          INDEX0001
C*****INDEX0002
C*****INDEX0003
C      FUNCTION INDX TAKES THE TWO INDICES (I,J) OF A FULL
C      STORAGE MODE ARRAY AND COMPUTES THE CORRESPONDING
C      POSITION OF THAT ELEMENT IN AN ARRAY IN SYMMETRIC
C      STORAGE MODE.
C*****INDEX0004
C*****INDEX0005
C*****INDEX0006
C*****INDEX0007
C*****INDEX0008
C*****INDEX0009
C*****INDEX0010
C      INDEX=(I+J+IABS(I-J))/2    INDEX0011
C      INDEX=INDEX*(INDEX-1)/2    INDEX0012
C      INDEX=INDEX+(I+J-IABS(I-J))/2  INDEX0013
C      RETURN                    INDEX0014
C      END                       INDEX0015
C                                INDEX0016
```

```

SUBROUTINE SCRNA
IMPLICIT REAL*8 (A-H,O-Z)

*****SUBROUTINE SCRNA IS A REGRESSION ANALYSIS PROGRAM WRITTEN
BY GEORGE M. FURNIVAL AT YALE UNIVERSITY, WHICH COMPUTES
THE R-SQUARED VALUES FOR ALL PERMUTATIONS OF THE
INDEPENDENT VARIABLES.

THIS PROGRAM HAS BEEN REVISED FROM THE ORIGINAL IN SEVERAL
WAYS---
FIRST THE DIMENSIONS HAVE BEEN CHANGED SO THAT ONLY TEN
OR LESS INDEPENDENT VARIABLES MAY BE USED
SECOND THE XTX MATPIX IS NOW IN SYMMETRIC STORAGE MODE.
THIRD INSTEAD OF OUTPUTTING THE R-SQUARES THE PROGRAM
STORES THEM ALONG WITH THE BIT PATTERN CORRESPOND-
ING TO THE INDEPENDENT VARIABLES USED.

11111 CONTINUE

**NOTE** THE PROGRAM CALCULATES THE REGRESSIONS IN THE
FOLLOWING SEQUENCE (FOR N=4) 1,2,3,4,5,6,7,8,9,A,
B,C,D,E,F WHERE THE NUMBERS ARE IN HEXADECIMAL.
BY BREAKING THESE NUMBERS ONE GETS A BIT PATTERN
OF ZEROS AND ONES SUCH THAT THE BIT IS ONE IF THE
CORRESPONDING INDEPENDENT VARIABLE IS IN THE
REGRESSION AND IS ZERO IF IT IS NOT.
THUS THE VARIABLE JNUM IS INCREMENTED EACH TIME A
REGRESSION IS CALCULATED AND THEREFOR THE BIT
PATTERN OF JNUM GIVES THE INDEPENDENT VARIABLES IN
THE REGRESSION.

THE BEST R-SQUARE FOR SUBSET SIZE 'IT' IS RETURNED VIA RSQ1
THE R-SQUARE WITH THE BIT PATTERN THE SAME AS 'NUMB' IS
RETURNED VIA RSQ2

*****REAL*8 JSAVE(1035),KSAVE
DIMENSION MN(10),MM(10),ISAVE(1023)
DIMENSION NT(11),SS(78), A(11,11,11)
COMMON /SCRBLK/ JDUMMY,N,SS,IT,NUMB,RSQ1,RSQ2
COMMON /BLOCK/ IDUMMY, ISAVE, JSAVE, A
DATA MN/1,11,56,176,386,638,848,968,1013,1023/
IFI N.GT.10 } STOP
DO 100 I=1,10
100 MM(I)=MN(I)-1
JNUM=0

```

SCRNA001
SCRNA002
SCRNA003
SCRNA004
SCRNA005
SCRNA006
SCRNA007
SCRNA008
SCRNA009
SCRNA010
SCRNA011
SCRNA012
SCRNA013
SCRNA014
SCRNA015
SCRNA016
SCRNA017
SCRNA018
SCRNA019
SCRNA020
SCRNA021
SCRNA022
SCRNA023
SCRNA024
SCRNA025
SCRNA026
SCRNA027
SCRNA028
SCRNA029
SCRNA030
SCRNA031
SCRNA032
SCRNA033
SCRNA034
SCRNA035
SCRNA036
SCRNA037
SCRNA038
SCRNA039
SCRNA040
SCRNA041
SCRNA042
SCRNA043
SCRNA044
SCRNA045
SCRNA046
SCRNA047
SCRNA048
SCRNA049
SCRNA050

```

NA=N+1          SCRPA051
NB=N+2          SCRPA052
NBA=NB*(NB-1)/2 SCRPA053
SQ=10./DSQRT(SS(NBA+NB)-SS(NBA+1)*SS(NBA+1)/SS(1))
DO 1 I=2,NA    SCRPA054
ISUB=I*(I-1)/2 SCRPA055
NT(I-1)=0       SCRPA056
B=SS(ISUB+1)/SS(1) SCRPA057
A(NA,1,I)=SQ*(B*SS(NBA+1)-SS(NBA+I)) SCRPA058
DO 1 J=I,NA    SCRPA059
JSUB=J*(J-1)/2 SCRPA060
A(NA,1,J)=B*SS(JSUB+1)-SS(JSUB+I) SCRPA061
1 CONTINUE      SCRPA062
A(NA,1,1)=0.    SCRPA063
NT(NA)=1        SCRPA064
INUM=0          SCRPA065
11 DO 20 I=1,N SCRPA066
IFI NT(I).EQ.0 ) GO TO 21 SCRPA067
INUM=INUM-1    SCRPA068
NT(I)=0          SCRPA069
20 CONTINUE      SCRPA070
10 LIMA=MN(IT)  SCRPA071
LIMD=MH(IT)    SCRPA072
RSQ1=JSAVE(LIMA) SCRPA073
DO 60 LOOP=LIMA,LIMB SCRPA074
IFI NUMB.EQ.JSAVE(LOOP) ) GO TO 61 SCRPA075
60 CONTINUE      SCRPA076
WRITE(6,600) NUMB SCRPA077
600 FORMAT('0***** THE CORRECT BIT PATTERN COULD NOT BE FOUND : ',Z8) SCRPA078
NUMB = 845014520 SCRPA079
RETURN          SCRPA080
61 RSQ2=JSAVE(LOOP) SCRPA081
RETURN          SCRPA092
21 NT(I)=1        SCRPA093
JNUM=JNUM+1    SCRPA094
INUM=INUM+1    SCRPA095
K=I+1          SCRPA096
DO 22 JK,NA    SCRPA097
IFI NT(J).EQ.1 ) GO TO 30 SCRPA098
22 CONTINUE      SCRPA099
30 DO 31 L=1,1  SCRPA090
B=A(J,L,K)/A(J,K,K) SCRPA091
DO 31 M=L,1  SCRPA092
A(I,L,M)=A(J,L,M)-B*A(J,M,K) SCRPA093
31 CONTINUE      SCRPA094
KSAVE=A(I,1,1)*.01 SCRPA095
MM(INUM)=MM(INUM)+1 SCRPA096
LIMA=MN(INUM) SCRPA097
LIMB=MH(INUM) SCRPA098
IFI LIMA.EQ.LIMB ) GO TO 50 SCRPA099
LIMA=LIMA+1    SCRPA100
                                         SCRPA101

```

```
DO 45 LOOPA=LIMA,LIMB           SCRPA102
LOOP=LINB+LIMA-LOOPA           SCRPA103
IF( KSAVE.LE.JSAVE(LOOP-1) ) GO TO 55
JSAVE(LOOP)=JSAVE(LOOP-1)       SCRPA104
45 ISAVE(LOOP)=ISAVE(LOOP-1)    SCRPA105
50 LOOP=MN(INUM)               SCRPA106
55 JSAVE(LOOP)=KSAVE           SCRPA107
ISAVE(LOOP)=JNUM               SCRPA108
GO TO 11                       SCRPA109
END                           SCRPA110
                                SCRPA111
```

```

SUBROUTINE PRMUT(P,T,K)
C*****PRMUT001
C*****PRMUT002
C*****PRMUT003
C*****PRMUT004
C*****PRMUT005
C*****PRMUT006
C*****PRMUT007
C*****PRMUT008
C*****PRMUT009
C*****PRMUT010
C*****PRMUT011
C*****PRMUT012
C*****PRMUT013
C*****PRMUT014
C*****PRMUT015
C*****PRMUT016
C*****PRMUT017
C*****PRMUT018
C*****PRMUT019
C*****PRMUT020
C*****PRMUT021
C*****PRMUT022
C*****PRMUT023
C*****PRMUT024
C*****PRMUT025
C*****PRMUT026
C*****PRMUT027
C*****PRMUT028
C*****PRMUT029
C*****PRMUT030
C*****PRMUT031
C*****PRMUT032
C*****PRMUT033
C*****PRMUT034
C*****PRMUT035
C*****PRMUT036
C*****PRMUT037
C*****PRMUT038
C*****PRMUT039
C*****PRMUT040
C*****PRMUT041
C*****PRMUT042
C*****PRMUT043
C*****PRMUT044
C*****PRMUT045
C*****PRMUT046
C*****PRMUT047
C*****PRMUT048
C*****PRMUT049
C*****PRMUT050

C PRMUT IS A SUBROUTINE WHICH DETERMINES THE PERMUTATIONS ON A SEQUENCE OF NUMBERS.
C THE PERMUTATION IS TAKEN ON (P THINGS TAKEN T AT A TIME).
C BECAUSE THE REGRESSION ANALYSIS PROGRAM SCRNA ALWAYS INCLUDES AN INTERCEPT THE FIRST NUMBER IN EACH PERMUTATION WILL BE 1. THUS THE PERMUTATIONS THAT THE PROGRAM DETERMINES IS THE PERMUTATION (P-1) TAKEN (T-1) AT A TIME DONE ON THE INTEGERS 2,3,...,P.
C THE USAGE OF THE SUBROUTINE IS
C     CALL PRMUT(P,T,K,PERM)
C P IS THE NUMBER OF INDEPENDENT VARIABLES PLUS 1.
C T IS THE SUBSET SIZE BEING CONSIDERED PLUS 1.
C K IS THE RESULTING NUMBER OF PERMUTATIONS.
C PERM IS THE ARRAY WHICH STORES THE DETERMINED PERMUTATIONS.
C ALL ARGUMENTS ARE INTEGERS.

C IDUMMY FORCES ALIGNMENT
COMMON /BLOCK/ IDUMMY, PERM
INTEGER P,T,PERM(93,11),PM,TH
PM=P-1

C CALCULATE K

C K=1
TH=T-1
L=PM-TH+1
DO 10 I=L,PM
10 K=K*I
DO 11 I=1,TH
11 K=K/I

C FILL OUT THE ARRAY PERM.

C I=1

```

```

      DO 15 LOOP=1,T
15 PERM(1,LOOP)=LOOP
C
20 I=I+1
IF(I.EQ.K+1) GO TO 100
DO 50 LOOP=1,T
50 PERM(I,LOOP)=PEPM(I-1,LOOP)
IF(PERM(I,I).EQ.P) GO TO 55
PERM(I,I)=PERM(I,I)+1
GO TO 20
C
55 DO 60 J=2,T
L=T-J+2
IF(PERM(I,L).LT.P-J+2) GO TO 70
60 CONTINUE
GO TO 100
C
70 PERM(I,L)=PERM(I,L)+1
LA=L+1
DO 75 LOOP=LA,P
75 PERM(I,LOOP)=PEPM(I,LOOP-1)+1
GO TO 20
C
100 CONTINUE
RETURN
END
      PRMUT051
      PRMUT052
      PRMUT053
      PRMUT054
      PRMUT055
      PRMUT056
      PRMUT057
      PRMUT058
      PRMUT059
      PRMUT060
      PRMUT061
      PRMUT062
      PRMUT063
      PRMUT064
      PRMUT065
      PRMUT066
      PRMUT067
      PRMUT068
      PRMUT069
      PRMUT070
      PRMUT071
      PRMUT072
      PRMUT073
      PRMUT074
      PRMUT075
      PRMUT076

```

```

FUNCTION GAUSS(SD,AM)
C COMPUTES A NORMALLY DISTRIBUTED RANDOM NUMBER WITH A GIVEN
C MEAN (AM) AND STANDARD DEVIATION (SD)
C METHOD USED IS BOX-MULLER
C REF: NEWHAV AND DELL, THE GENERATION OF RANDOM VARIATES.
C
      REAL*8 GAUSS,SD,AM
      LOGICAL FIRST
      COMMON /WXX/ IX
      DATA FIRST/.TRUE./
C
      IFI .NOT.FIRST I GO TO 100
C
      CALL RANDU(IX,ITY,U)
      IX=ITY
      CALL RANDU(IX,ITY,V)
      IX=ITY
C
      S=SQRT( -2.* ALOG(U) )
      P=6.283185*V
      RN1=S*COS(P)
      RN2=S*SIN(P)
C
      GAUSS = RN1*SD + AM
      FIRST=.FALSE.
      RETURN
C
100  GAUSS = RN2*SD + AM
      FIRST=.TRUE.
      RETURN
C
      END
      GAUSS001
      GAUSS002
      GAUSS003
      GAUSS004
      GAUSS005
      GAUSS006
      GAUSS007
      GAUSS008
      GAUSS009
      GAUSS010
      GAUSS011
      GAUSS012
      GAUSS013
      GAUSS014
      GAUSS015
      GAUSS016
      GAUSS017
      GAUSS018
      GAUSS019
      GAUSS020
      GAUSS021
      GAUSS022
      GAUSS023
      GAUSS024
      GAUSS025
      GAUSS026
      GAUSS027
      GAUSS028
      GAUSS029
      GAUSS030
      GAUSS031
      GAUSS032
      GAUSS033
      GAUSS034

```

```

SUBROUTINE LINEQ1(A,B,X,ND,N,NR,S,C,R,E)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(ND,ND),B(ND,NR),X(ND,NR),C(N,N),R(N),E(N)
INTEGER S
ABS(D)=DABS(D)
AMAX1(F,G)=DMAX1(F,G)
.....LINEQ001
.....LINEQ002
.....LINEQ003
.....LINEQ004
.....LINEQ005
.....LINEQ006
.....LINEQ007
.....LINEQ008
.....LINEQ009
.....LINEQ010
.....LINEQ011
.....LINEQ012
.....LINEQ013
.....LINEQ014
.....LINEQ015
.....LINEQ016
.....LINEQ017
.....LINEQ018
.....LINEQ019
.....LINEQ020
.....LINEQ021
.....LINEQ022
.....LINEQ023
.....LINEQ024
.....LINEQ025
.....LINEQ026
.....LINEQ027
.....LINEQ028
.....LINEQ029
.....LINEQ030
.....LINEQ031
.....LINEQ032
.....LINEQ033
.....LINEQ034
.....LINEQ035
.....LINEQ036
.....LINEQ037
.....LINEQ038
.....LINEQ039
.....LINEQ040
.....LINEQ041
.....LINEQ042
.....LINEQ043
.....LINEQ044
.....LINEQ045
.....LINEQ046
.....LINEQ047
.....LINEQ048
.....LINEQ049
.....LINEQ050

SUBROUTINE LINEQ1
DECKNAME - LINEQ1
PURPOSE
  SOLVES THE REAL MATRIX EQUATION AX=B WITH NR RIGHT-HAND
  SIDES.
USAGE
  CALL LINEQ1(A,B,X,ND,N,NR,S)
DESCRIPTION OF PARAMETERS
  A   - (N X N) REAL COEFFICIENT MATRIX.          LINEQ023
  B   - (N X NR) REAL RIGHT-HAND SIDE ARRAY.      LINEQ024
  X   - (N X NR) REAL ARRAY FOR RETURN OF SOLUTION
        VECTORS.                                     LINEQ025
  ND   - THE NUMBER OF ROWS FOR THE ARRAYS A, B, AND X
        IN THE DIMENSION STATEMENT IN THE USER PROGRAM. LINEQ027
  N    - THE NUMBER OF EQUATIONS TO BE SOLVED.      LINEQ028
  NR   - THE NUMBER OF RIGHT-HAND SIDES TO BE SOLVED. LINEQ029
  S    - INTEGER VARIABLE RETURNED NON-ZERO ONLY IF
        MATRIX A IS SINGULAR TO MACHINE ACCURACY.     LINEQ030
90 CONTINUE
REMARKS
  ARRAYS A AND B ARE NOT DESTROYED.
EXTRA PARAMETERS IN THE ACTUAL FORMAL PARAMETER LIST
  ARE USED IN CONJUNCTION WITH THE SUBROUTINE DYNAMIC.
METHOD
  THE MATRIX A IS FACTORED INTO LOWER AND UPPER TRIANGULAR
  MATRICES L AND U AND THEN THE EQUATIONS LZ=B AND UX=Z ARE
  SOLVED IN TURN. DOUBLE PRECISION ACCUMULATION OF INNER
  PRODUCTS AND ITERATIVE REFINEMENT ARE USED SO SOLUTIONS
  ARE VERY ACCURATE WHENEVER S IS RETURNED EQUAL TO ZERO.
RALSTON AND WILF  MATHEMATICAL METHODS FOR DIGITAL
  COMPUTERS  VOLUME 2  WILEY 1967.

```

```

C      WRITTEN BY DAVID S. DODSON, 06/01/70           LINEQ051
C      COMPUTER SCIENCES DEPARTMENT                 LINEQ052
C      PURDUE UNIVERSITY                           LINEQ053
C
C      *****                                         LINEQ054
C
C      FORM EQUILIBRATION FACTORS IN VECTOR E.       LINEQ055
C
C      DO 12 I=1,N                                  LINEQ056
C      P=0.0                                         LINEQ057
C      DO 10 J=1,N
C      C(I,J)=A(I,J)
C      10 P=AMAX1(ABS(C(I,J)),P)
C      IF(P.EQ.0.0)GO TO 73
C      12 E(I)=1.0/P                               LINEQ058
C
C      FACTOR COEFFICIENT MATRIX WITH PARTIAL PIVOTTING.   LINEQ059
C
C      M=1                                         LINEQ060
C      14 MM=M-1                                 LINEQ061
C      P=0.0                                         LINEQ062
C      DO 22 I=M,N
C      C(I,M)=ARITH1(C(I,M),MM,C(I,1),N,C(1,M),1)
C      Q=E(I)*ABS(C(I,M))
C      IF(P.GE.Q)GO TO 22
C      P=Q
C      K=I
C      22 CONTINUE
C      IF (NP.EQ.N) GO TO 73
C      IF(M.EQ.K)GO TO 30
C      DO 28 J=1,N
C      P=C(M,J)
C      C(M,J)=C(K,J)
C      28 C(K,J)=P
C      E(K)=E(M)
C      30 E(M)=K
C      IF(M.EQ.N)GO TO 37
C      MP=M+1
C      DO 34 J=MP,N
C      34 C(M,J)=ARITH1(C(M,J),MN,C(M,1),N,C(1,J),1)/C(M,M)
C      M=MP
C      GO TO 14
C
C      BACK SUBSTITUTE RIGHT-HAND SIDES WITH ITERATIVE REFINEMENT.   LINEQ063
C
C      37 IF(NR.LE.0)GO TO 71
C      DO 68 M=1,NR
C      P=0.0
C      Q=0.0
C      DO 43 I=1,N

```

LINEQ064
LINEQ065
LINEQ066
LINEQ067
LINEQ068
LINEQ069
LINEQ070
LINEQ071
LINEQ072
LINEQ073
LINEQ074
LINEQ075
LINEQ076
LINEQ077
LINEQ078
LINEQ079
LINEQ080
LINEQ081
LINEQ082
LINEQ083
LINEQ084
LINEQ085
LINEQ086
LINEQ087
LINEQ088
LINEQ089
LINEQ090
LINEQ091
LINEQ092
LINEQ093
LINEQ094
LINEQ095
LINEQ096
LINEQ097
LINEQ098
LINEQ099
LINEQ100
LINEQ101

```

R(I)=B(I,M)
43 X(I,M)=0.0
44 DO 48 I=1,N
  K=E(I)
  T=R(K)
  R(K)=R(I)
48 R(I)=ARITH1(T,I-1,C(I,1),N,R(1),1)/C(I,1)
  I=N
  GO TO 52
51 R(I)=ARITH1(R(I),N-I,C(I,IP),N,R(IP),1)
52 IP=I
  I=I-1
  IF(I.GT.0)GO TO 51
  T=P
  P=0.0
  DO 59 I=1,N
    P=AMAX1(ABS(R(I)),P)
59 X(I,M)=X(I,M)+R(I)

C      TEST FOR CONVERGENCE OF ITERATIVE REFINEMENT.
C
  IF(P.EQ.0.0)GO TO 68
  IF(Q.EQ.0.0)Q=P
  IF ( SNGL(Q+P).EQ. SNGL(Q)) GO TO 68
  IF(T.NE.0.0.AND.P+P.GT.T)GO TO 67
  DO 65 I=1,N
65 R(I)=ARITH1(B(I,M),N,A(I,1),ND,X(I,M),1)
  GO TO 44
67 IF ( SNGL(Q+P/N) .NE. SNGL(Q) ) GO TO 73
68 CONTINUE

C      SET NON-SINGULAR/SINGULAR FLAG AND RETURN.
C
  71 S=0
    RETURN
C
  73 S=1
    WRITE(6,700) Q,P,N
700 FORMAT('MATRIX SINGULAR IN LINEQ1 : Q, P, N ',5X,
     1 2E20.8,15)
    RETURN
C
  END

```

LINEQ102
LINEQ103
LINEQ104
LINEQ105
LINEQ106
LINEQ107
LINEQ108
LINEQ109
LINEQ110
LINEQ111
LINEQ112
LINEQ113
LINEQ114
LINEQ115
LINEQ116
LINEQ117
LINEQ118
LINEQ119
LINEQ120
LINEQ121
LINEQ122
LINEQ123
LINEQ124
LINEQ125
LINEQ126
LINEQ127
LINEQ128
LINEQ129
LINEQ130
LINEQ131
LINEQ132
LINEQ133
LINEQ134
LINEQ135
LINEQ136
LINEQ137
LINEQ138
LINEQ139
LINEQ140
LINEQ141
LINEQ142
LINEQ143
LINEQ144

```
C FUNCTION ARITH1(C,N,A,KA,B,KB)
C CALLED FROM LINEQ1
C
IMPLICIT REAL*8 (A-H,D-Z)
DIMENSION A(KA,N),B(KB,N)
T=C
IF(N.EQ.0) GO TO 5
DO 4 I=1,N
 4 T=T-A(I,I)*B(I,I)
5 ARITH1=T
RETURN
END
```

```
ARITH001
ARITH002
ARITH003
ARITH004
ARITH005
ARITH006
ARITH007
ARITH008
ARITH009
ARITH010
ARITH011
ARITH012
ARITH013
```

```

SUBROUTINE VSORTM (A,LA)                                VSCRT001
C.VSORTM.....S.....VSORT002
C.VSORTA.....VSORT003
C.....VSORT004
C FUNCTION      VSORTM - SORT ARRAYS BY ABSOLUTE VALUE   VSORT005
C VSORTA - SORT ARRAYS BY ALGEBRAIC VALUE
C USAGE          - CALL VSORTM (A,LA)
C                 - CALL VSORTA (A,LA)                         VSORT006
C PARAMETERS     A   - ON INPUT, CONTAINS THE ARRAY TO BE SORTED  VSORT007
C                 LA  - ON OUTPUT, A CONTAINS THE SORTED ARRAY    VSORT008
C                 - INPUT VARIABLE CONTAINING THE NUMBER OF   VSORT009
C                 ELEMENTS IN THE ARRAY TO BE SORTED        VSORT010
C PRECISION       - SINGLE                               VSORT011
C AUTHOR/IMPLEMENTER - N.E. BOSTEN                      VSORT012
C LANGUAGE        - FORTRAN                            VSORT013
C.....VSORT014
C.....VSORT015
C LATEST REVISION - DECEMBER 7,1970                     VSORT016
C.....VSORT017
C.....VSORT018
C.....VSORT019
C.....VSORT020
C.....VSORT021
C.....VSORT022
C.....VSORT023
C.....VSORT024
C.....VSORT025
C.....VSORT026
C.....VSORT027
C.....VSORT028
C.....VSORT029
C.....VSORT030
C.....VSORT031
C.....VSORT032
C.....VSORT033
C.....VSORT034
C.....VSORT035
C.....VSORT036
C.....VSORT037
C.....VSORT038
C.....VSORT039
C.....VSORT040
C.....VSORT041
C.....VSORT042
C.....VSORT043
C.....VSORT044
C.....VSORT045
C.....VSORT046
C.....VSORT047
C.....VSORT048
C.....VSORT049
C.....VSORT050

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION        A(1),IU(21),IL(21)
C               FIND ABSOLUTE VALUES OF ARRAY A
DO 5 I=1,LA
  IF (A(I) .LT. 0.0) A(I)=-A(I)
5  CONTINUE
C               ENTRY VSORTA (A,LA)
C
N=1
I=1
J=LA
R=.375
10 IF (I .EQ. J) GO TO 55
15 IF (R .GT. .5898437) GO TO 20
  R=R+3.90625E-2
  GO TO 25
20 R=R-.21875
25 K=I
C               SELECT A CENTRAL ELEMENT OF THE
C               ARRAY AND SAVE IT IN LOCATION T
IJ=I+(J-I)*R
T=A(IJ)
C               IF FIRST ELEMENT OF ARRAY IS GREATER
C               THAN T, INTERCHANGE WITH T
IF (A(I) .LE. T) GO TO 30
A(IJ)=A(I)
A(I)=T
T=A(IJ)
30 L=J
C               IF LAST ELEMENT OF ARRAY IS LESS THAN VSORT050

```

```

C      IF (A(IJ) .GE. T) GO TO 40          T, INTERCHANGE WITH T           VSORT051
      A(IJ)=A(J)
      A(J)=T
      T=A(IJ)
C      IF (A(I) .LE. T) GO TO 40          IF FIRST ELEMENT OF ARRAY IS GREATER   VSORT052
      A(IJ)=A(I)
      A(I)=T
      T=A(IJ)
      GO TO 40
35    TT=A(L)
      A(L)=A(K)
      A(K)=TT
C      40 L=L-1                         FIND AN ELEMENT IN THE SECOND HALF OF   VSORT053
      IF (A(L) .GT. T) GO TO 40          THE ARRAY WHICH IS SMALLER THAN T     VSORT054
C      45 K=K+1                         FIND AN ELEMENT IN THE FIRST HALF OF   VSORT055
      IF (A(K) .LT. T) GO TO 45          THE ARRAY WHICH IS GREATER THAN T     VSORT056
C      IF (K .LE. L) GO TO 35          INTERCHANGE THESE ELEMENTS           VSORT057
C      IF (L-I .LE. J-K) GO TO 50          SAVE UPPER AND LOWER SUBSCRIPTS OF   VSORT058
      IL(M)=I                           THE ARRAY YET TO BE SORTED            VSORT059
      IU(M)=L                           VSORT060
      I=K
      M=M+1
      GO TO 60
50    IL(M)=K
      IU(M)=J
      J=L
      M=M+1
      GO TO 60
C      BEGIN AGAIN ON ANOTHER PORTION OF   VSORT061
C      THE UNSORTED ARRAY               VSORT062
      55 M=M-1                         VSORT063
      IF (M .EQ. 0) RETURN             VSORT064
      I=IL(M)
      J=IU(M)
      60 IF (J-I .GE. 1) GO TO 25
      IF (I .EQ. 1) GO TO 10
      I=I-1
      65 I=I+1
      IF (I .EQ. J) GO TO 55
      T=A(I+1)
      IF (A(I) .LE. T) GO TO 65
      K=I
      TO A(K+1)=A(K)
      K=K-1
      IF (T .LT. A(K)) GO TO 70
      A(K+1)=T
      GO TO 65
      END

```

VSORT065
VSORT066
VSORT067
VSORT068
VSORT069
VSORT070
VSORT071
VSORT072
VSORT073
VSORT074
VSORT075
VSORT076
VSORT077
VSORT078
VSORT079
VSORT080
VSORT081
VSORT082
VSORT083
VSORT084
VSORT085
VSORT086
VSORT087
VSORT088
VSORT089
VSORT090
VSORT091
VSORT092
VSORT093
VSORT094
VSORT095
VSORT096
VSORT097
VSORT098
VSORT099
VSORT100
VSORT101
VSORT102
VSORT103
VSORT104
VSORT105
VSORT106
VSORT107
VSORT108

```

SUBROUTINE LUDECP (A,UL,N,D1,D2,IER)
C.LUDECP.....S.....LUDEC001
C   FUNCTION      - CHOLESKY DECOMPOSITION OF A MATRIX - LUDEC002
C   USAGE          SYMMETRIC STORAGE MODE LUDEC003
C   PARAMETERS     A - CALL LUDECP (A,UL,N,D1,D2,IER) LUDEC004
C                   A IS N X N POSITIVE DEFINITE MATRIX AND IS LUDEC005
C                   IN SYMMETRIC STORAGE MODE. LUDEC006
C                   UL - THE RESULT L OF THIS ROUTINE IS STORED IN THE LUDEC007
C                   MATRIX UL. UL MAY OCCUPY THE SAME STORAGE AS LUDEC008
C                   A. THE MAIN DIAGONAL VALUES OF UL ARE STORED LUDEC009
C                   IN RECIPROCAL FORM. LUDEC010
C                   N - ORDER OF A LUDEC011
C                   D1 - A NUMBER COMPUTED IN THE SUBROUTINE SUCH THAT LUDEC012
C                         THE DETERMINANT OF A = D1*2.**D2 LUDEC013
C                   D2 - SEE D1 LUDEC014
C                   IER - ERROR PARAMETER LUDEC015
C                         TERMINAL ERROR = 128 + II LUDEC016
C                         N = 1 INDICATES THAT THE MATRIX IS LUDEC017
C                         SINGULAR LUDEC018
C   PRECISION      - SINGLE LUDEC019
C   REQD. IMSL ROUTINES - UERTST LUDEC020
C   AUTHOR/IMPLEMENTER - O.G. JOHNSON/E.W.CHOU LUDEC021
C   LANGUAGE        - FORTRAN LUDEC022
C   LATEST REVISION - MAY 10, 1972 LUDEC023
C   .....LUDEC024
C   IMPLICIT REAL*8 (A-H,D-Z)
C   DIMENSION       A(1),UL(1) LUDEC025
C   DATA            ZERO,ONE,FOUR,SIXTN,SIXTH/0.000,1.00,4.00,16.00LUDEC030
C   1,.062500/
C   ABS(D)=DABS(D) LUDEC031
C   D1=ONE LUDEC032
C   D2=ZERO LUDEC033
C   IP=0 LUDEC034
C   IER=0 LUDEC035
C   DO 40 I=1,N LUDEC036
C     IQ = IP+1 LUDEC037
C     IR = 0 LUDEC038
C   DO 35 J=1,I LUDEC039
C     X=A(IP+1) LUDEC040
C     IF (IP.LT.IQ) GO TO 10 LUDEC041
C     DO 5 K=IQ,IP LUDEC042
C       IR = IR+1 LUDEC043
C     X = X - UL(K)*UL(IR) LUDEC044
C   5  CONTINUE LUDEC045
C   10  IR = IR+1 LUDEC046
C       IP = IP+1 LUDEC047
C     IF (I.NE.J) GO TO 30 LUDEC048
C   LUDEC049
C   LUDEC050

```

D1 = D1*X	LUDEC051
IF (X.NE.-0.0D0) GO TO 15	LUDEC052
D2=ZERO	LUDEC053
GO TO 45	LUDEC054
15 IF (ABS(D1).LT.DNE) GO TO 20	LUDEC055
D1 = D1 * SIXTH	LUDEC056
D2 = D2 + FOUR	LUDEC057
GO TO 15	LUDEC058
20 IF (ABS(D1) .GE. SIXTH) GJ TO 25	LUDEC059
D1 = D1 * SIXTN	LUDEC060
D2 = D2 - FOUR	LUDEC061
GO TO 20	LUDEC062
25 IF (X.LT.0.000) GO TO 45	LUDEC063
UL(IP) = 1.00/DSQRT(X)	LUDEC064
GO TO 35	LUDEC065
30 UL(IP) = X * UL(IIR)	LUDEC066
35 CONTINUE	LUDEC067
40 CONTINUE	LUDEC068
GO TO 9005	LUDEC069
45 IER = 129	LUDEC070
9000 CONTINUE	LUDEC071
CALL VERTST(IER,6HLUDEC,P)	LUDEC072
9005 RETURN	LUDEC073
END	LUDEC074

```

SUBROUTINE UERTST(IER,NAME)
C.UERTST.....-----+-----+-----+-----+-----+-----+-----+
C   FUNCTION      - ERROR MESSAGE GENERATION          UERTS001
C   USAGE         - CALL UERTST(IER,NAME)             UERTS002
C   PARAMETERS    IER      - ERROR PARAMETER. TYPE = N WHERE
C                      TYPE= 128 IMPLIES TERMINAL ERROR     UERTS003
C                      64 IMPLIES WARNING WITH FIX       UERTS004
C                      32 IMPLIES WARNING             UERTS005
C                      N = ERROR CODE RELEVANT TO CALLING ROUTINEUERTS006
C   NAME          - THE LITERAL STRING IDENTIFYING THE NAME OF     UERTS007
C                      THE CALLING ROUTINE(IFROM 4 TO 6 ALPHANUMERIC)UERTS008
C   AUTHOR/IMPLEMENTER - PEDER SVENDSEN           UERTS009
C   LANGUAGE      - FCRTRAN                         UERTS010
C   LATEST REVISION - JANUARY 19, 1971            UERTS011
C
C
REAL*8 ITYP,NAME
DIMENSION ITYP(2,4),IBIT(4)
INTEGER IAPN,WARF,TERM,PRINTR
EQUIVALENCE (IBIT(1)),IAPN,(IBIT(2),WARF),(IBIT(3),TERM)
DATA ITYP /0WHARNING,0H
*        0WHARNING,0HWITH FIX,
*        0HTERMINAL,0H
*        0HNON-DEF,0HNEW  /
IBIT / 32,64,128,0/
IER2=IER
IF (IER2 .GE. 15) GO TO 5
      NON-DEFINED
IER1=4
GO TO 20
5 IF (IER2 .LT. TERM) GO TO 10
      TERMINAL
IER1=3
GO TO 20
10 IF (IER2 .LT. WARF) GO TO 15
      WARNING(WITH FIX)
IER1=2
GO TO 20
C 15 IER1=1
      WARNING
C 20 IER2=IER2-IBIT(IER1)
      EXTRACT *N*
      PRINT ERROR MESSAGE
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,2),NAME,IER2
25 FORMAT(26H *** I N S UERTST! *** ,2AB,4A,A6,4K,12)
      RETURN
      UERTS012
      UERTS013
      UERTS014
      UERTS015
      UERTS016
      UERTS017
      UERTS018
      UERTS019
      UERTS020
      UERTS021
      UERTS022
      UERTS023
      UERTS024
      UERTS025
      UERTS026
      UERTS027
      UERTS028
      UERTS029
      UERTS030
      UERTS031
      UERTS032
      UERTS033
      UERTS034
      UERTS035
      UERTS036
      UERTS037
      UERTS038
      UERTS039
      UERTS040
      UERTS041
      UERTS042
      UERTS043
      UERTS044
      UERTS045
      UERTS046
      UERTS047
      UERTS048
      UERTS049
      UERTS050
      UERTS051
END

```

```

SUBROUTINE LEQTIP (A,M,N,IB,B,IDGT,WKAREA,IER)           LEQT1001
IMPLICIT REAL*8 (A-H,O-Z)                               LEQT1002
C.LEQTIP.....S.....LEQT1003
C FUNCTION      - LINEAR EQUATION SOLUTION - SYMMETRIC STORAGE LEQT1004
C               MODE - SPACE ECONOMIZER SOLUTION             LEQT1005
C USAGE          - CALL LEQTIP (A,M,N,IB,B,IDGT,WKAREA,IEP) LEQT1006
C PARAMETERS     A      - THE COEFFICIENT MATRIX OF THE EQUATION LEQT1007
C                   AX=B, WHERE A IS ASSUMED TO BE POSITIVE LEQT1008
C                   DEFINITE OF SIZE N X N AND IS IN SYMMETRIC LEQT1009
C                   STORAGE MODE. A IS REPLACED BY THE N X N LEQT1010
C LATEST REVISION - MAY 10, 1972                         LEQT1011
C                                     LEQT1012
C                                     LEQT1013
C                                     LEQT1014
C                                     LEQT1015
C                                     LEQT1016
C                                     LEQT1017
C                                     LEQT1018
C                                     LEQT1019
C                                     LEQT1020
C                                     LEQT1021
C                                     LEQT1022
C                                     LEQT1023
C                                     LEQT1024
C                                     LEQT1025
C                                     LEQT1026
C                                     LEQT1027
C                                     LEQT1028
C                                     LEQT1029
C                                     LEQT1030
C                                     LEQT1031
C                                     LEQT1032
C                                     LEQT1033
C                                     LEQT1034
C                                     LEQT1035
C                                     LEQT1036
C                                     LEQT1037
C                                     LEQT1038
C                                     LEQT1039
C                                     LEQT1040
C                                     LEQT1041
C                                     LEQT1042
C                                     LEQT1043
C                                     LEQT1044
C                                     LEQT1045
C                                     LEQT1046
C                                     LEQT1047
C                                     LEQT1048
C                                     LEQT1049
C                                     LEQT1050
C                                     LEQT1051
C                                     LEQT1052
C                                     LEQT1053
C
10101 CONTINUE
C
C M      LU DECOMPOSITION MATRIX                         LEQT1015
C N      - NUMBER OF COLUMNS IN MATRIX B                LEQT1016
C IB     - ORDER OF A AND NUMBER OF ROWS IN B            LEQT1017
C B      - ROW DIMENSION OF B IN THE CALLING PROGRAM   LEQT1018
C       - MATRIX OF THE RIGHT HAND SIDE OF THE EQUATION LEQT1019
C       AX=B, WHERE B IS NXM. THE NXM SOLUTION X          LEQT1020
C       OVERWRITES B.                                     LEQT1021
C IDGT   - THE ELEMENTS OF A ARE ASSUMED TO BE CORRECT LEQT1022
C       TO IDGT DECIMAL DIGITS(CURRENTLY NOT USED)       LEQT1023
C WKAREA - WORK AREA OF DIMENSION GREATER THAN OR      LEQT1024
C       EQUAL TO N.                                     LEQT1025
C IER    - ERROR PARAMETER                                LEQT1026
C       TERMINAL ERROR = 128 + N                          LEQT1027
C       N = I INDICATES THAT LEQTIP FAILED TO            LEQT1028
C       FIND A SOLUTION. A IS SINGULAR                   LEQT1029
C PRECISION - SINGLE                                     LEQT1030
C REQD. IMSL ROUTINES - LUDECP,LUELMP,UERTST          LEQT1031
C AUTHOR/IMPLEMENTER - C.G. JOHNSON/E.W.CHOU          LEQT1032
C LANGUAGE  - FORTRAN                                    LEQT1033
C
C DIMENSION      A(1),B(1B,1),WKAREA(1)                 LEQT1034
C
C IER=0          INITIALIZE IER                         LEQT1035
C
C CALL LUDECP (A,A,N,D1,D2,IEP)                      LEQT1036
C IF (IER.NE.0) GO TO 15                               LEQT1037
C
C          PERFORM ELIMINATION                         LEQT1038
C DO 10  I=1,M                                         LEQT1039
C   CALL LUELMP (A,B(1,I),N,WKAREA(1))                LEQT1040
C   DO 5  I1=1,N
C     B(I1,I)=WKAREA(I1)                            LEQT1041
C
C 5  CONTINUE
C 10 CONTINUE
C GO TO 9005
C 15 IER=129
C 9000 CONTINUE
C CALL UERTST(IER,6HLEQTIP)
C 9005 RETURN
C END

```

```

SUBROUTINE LINVIP (A,N,AINV, IDGT, WKAREA, IER)           LINV1001
C.LINVIP.....S.....                                         LINV1002
C   FUNCTION      - INVERSION OF MATRIX - SYMMETRIC STORAGE MODE -LINV1003
C   USAGE          SPACE ECONOMIZER SOLUTION                LINV1004
C   PARAMETERS     A           - A GIVEN POSITIVE DEFINITE MATRIX OF SIZE    LINV1005
C                   N X N. A IS DESTROYED, REPLACED BY THE LU    LINV1006
C                   DECOMPOSITION MATRIX.                         LINV1007
C                   N           - ORDER OF A                      LINV1008
C                   AINV         - THE INVERSE OF A IS STORED IN THE N X N MATRIX LINV1009
C                           AINV (IN SYMMETRIC STORAGE MODE)        LINV1010
C                   IDGT         - THE ELEMENTS OF A ARE ASSUMED TO BE CORRECT    LINV1011
C                           TO IDGT DECIMAL PLACES. (CURRENTLY NOT USED) LINV1012
C                   WKAREA        - WORK AREA OF DIMENSION GREATER THAN OR EQUAL LINV1013
C                           TO 2*N                                LINV1014
C                   IER          - ERROR PARAMETER                  LINV1015
C                           TERMINAL ERROR = 128 + N                 LINV1016
C                           N=1, INDICATES THAT LINVIP FAILED TO   LINV1017
C                           FIND THE INVERSE                     LINV1018
C   PRECISION       - SINGLE                               LINV1019
C   REQD. TMSL ROUTINES - LEQTIP,LUELMP,UERTST,LUDECP    LINV1020
C   AUTHOR/IMPLEMENTER - D.G. JOHNSON/E.W.CHOU            LINV1021
C   LANGUAGE         - FORTRAN                            LINV1022
C   LATEST REVISION - MAY 10, 1972                         LINV1023
C   .....                                         LINV1024
C   IMPLICIT REAL*8 (A-H,O-Z)                         LINV1025
DIMENSION          A(1),AINV(1),WKAREA(1)               LINV1026
DATA              ZERO,ONE/0.000,1.000/                 LINV1027
IER=0              LINV1028
K=N               LINV1029
DO 25  I=1,N      LINV1030
  DO 5  J=1,N      LINV1031
    L=K+J          LINV1032
    WKAREA(L)=ZERO  LINV1033
    IF (J.EQ.I) WKAREA(L)=ONE  LINV1034
5 CONTINUE          LINV1035
IF (I.EQ.1) GO TO 10  LINV1036
C             DECOMPOSE A
CALL LEQTIP (A,I,N,N,WKAREA(K+1),IDGT,WKAREA,IER)  LINV1037
IF (IER.EQ.0) GO TO 15  LINV1038
GO TO 30          LINV1039
C             COMPUTE THE INVERSE AND MOVE INTO
10 CALL LUELMP (A,WKAREA(K+1),N,WKAREA(K+1))  LINV1040
15 K1 = I*(I-1)/2 + 1  LINV1041
  DO 20  J=1,N  LINV1042
    K1 = K1+J-1  LINV1043
    K2 = K+J  LINV1044
    AINV(K1)=WKAREA(K2)  LINV1045
20 CONTINUE          LINV1046
25 CONTINUE          LINV1047
  GO TO 9005  LINV1048
30 IER=129          LINV1049
9000 CONTINUE        LINV1050
  CALL UERTST(IER,6HLINV1P)  LINV1051
9005 RETURN          LINV1052
END               LINV1053

```

