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


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.
II. 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

<table>
<thead>
<tr>
<th>COL.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>DATADESC</td>
</tr>
<tr>
<td>9-80</td>
<td>Description used for labeling output</td>
</tr>
</tbody>
</table>

B. Parameter card

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

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

Continue on 2nd card if necessary.

<table>
<thead>
<tr>
<th>COL.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>Upper limit for 9th pair.</td>
</tr>
<tr>
<td>6-10</td>
<td>Lower limit for 10th pair.</td>
</tr>
<tr>
<td>11-15</td>
<td>Upper limit for 10th pair.</td>
</tr>
</tbody>
</table>
D. Specific subsets cards (Optional)

<table>
<thead>
<tr>
<th>COL.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>
| 1-3  | Number of specific subsets of independent variables (no more than 10). These are entered as hexadecimal (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 hexadecimal image is 0000000A1. |
| 9-18 | 1st subset |
| 19-27| 2nd subset |

Continue on 2nd card if necessary.

| 73-80| 9th subset. |

1-8 10th subset.

E. Input format card

<table>
<thead>
<tr>
<th>COL.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>Fortran type format statement used to read input data.</td>
</tr>
</tbody>
</table>

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     .LUDECF
    .LINV1P     .LUELMF
    .LETZ1P     .UERTST

For further information contact

International Mathematical and Statistical Libraries, Inc.
GNB Bldg.-Sixth Floor
7500 Bellaire
Houston, Texas 77036
<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>DATASET: LONLEY DATA SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARD</td>
<td>2</td>
<td>16 6 3 5 100 2123456789</td>
</tr>
<tr>
<td>CARD</td>
<td>3</td>
<td>3 4 9 5 89 91 94 96</td>
</tr>
<tr>
<td>CARD</td>
<td>4</td>
<td>(10,6,4,10,3)</td>
</tr>
<tr>
<td>CARD</td>
<td>5</td>
<td>60323, 831</td>
</tr>
<tr>
<td>CARD</td>
<td>6</td>
<td>61122, 84</td>
</tr>
<tr>
<td>CARD</td>
<td>7</td>
<td>60171, 80.3</td>
</tr>
<tr>
<td>CARD</td>
<td>8</td>
<td>61187, 89.2</td>
</tr>
<tr>
<td>CARD</td>
<td>9</td>
<td>63221, 98.2</td>
</tr>
<tr>
<td>CARD</td>
<td>10</td>
<td>63639, 98.1</td>
</tr>
<tr>
<td>CARD</td>
<td>11</td>
<td>64090, 90.9</td>
</tr>
<tr>
<td>CARD</td>
<td>12</td>
<td>63761, 100</td>
</tr>
<tr>
<td>CARD</td>
<td>13</td>
<td>64019, 101.2</td>
</tr>
<tr>
<td>CARD</td>
<td>14</td>
<td>67537, 104.6</td>
</tr>
<tr>
<td>CARD</td>
<td>15</td>
<td>61869, 108.6</td>
</tr>
<tr>
<td>CARD</td>
<td>16</td>
<td>64513, 110.8</td>
</tr>
<tr>
<td>CARD</td>
<td>17</td>
<td>68655, 112.6</td>
</tr>
<tr>
<td>CARD</td>
<td>18</td>
<td>69961, 114.2</td>
</tr>
<tr>
<td>CARD</td>
<td>19</td>
<td>60931, 115.7</td>
</tr>
<tr>
<td>CARD</td>
<td>20</td>
<td>70591, 116.9</td>
</tr>
</tbody>
</table>
DATA DESCRIPTION : LONGLEY DATA SET

NO. OF OBSERVATIONS ................. 16
NO. OF INDEPENDENT VARIABLES (EXCLUDES INTERCEPT) ............... 6
SURSET SIZE ......................... 3
NO. OF 'A' VARIABLES DESIRED ........ 100
| 0.9100000000 02 | 0.2542890000 04 | 0.2354600000 04 | 0.1590000000 04 | 0.1075000000 04 | 0.1957000000 04 |
| 0.8850000000 02 | 0.2594260000 04 | 0.2325000000 04 | 0.1656000000 04 | 0.1085000000 04 | 0.1954000000 04 |
| 0.8320000000 02 | 0.2508030000 04 | 0.3082000000 04 | 0.1510000000 04 | 0.1077000000 04 | 0.1957000000 04 |
| 0.8940000000 07 | 0.2508030000 04 | 0.3452000000 04 | 0.1500000000 04 | 0.1109000000 04 | 0.1950000000 04 |
| 0.9670000000 02 | 0.2389750000 06 | 0.2099000000 04 | 0.3090000000 04 | 0.1125000000 04 | 0.1931000000 04 |
| 0.9610000000 02 | 0.3469990000 06 | 0.1932000000 04 | 0.3594000000 04 | 0.1132000000 04 | 0.1952000000 04 |
| 0.9200000000 02 | 0.3653950000 06 | 0.1870000000 04 | 0.3547000000 04 | 0.1159000000 04 | 0.1931000000 04 |
| 0.1000000000 01 | 0.3631120000 06 | 0.3950000000 04 | 0.3590000000 04 | 0.1152000000 04 | 0.1940000000 04 |
| 0.1012000000 03 | 0.3974840000 06 | 0.2994000000 04 | 0.3048000000 04 | 0.1173000000 04 | 0.1955000000 04 |
| 0.1056000000 03 | 0.4191830000 06 | 0.2992000000 04 | 0.2837000000 04 | 0.1187000000 04 | 0.1964000000 04 |
| 0.1084000000 03 | 0.4477690000 06 | 0.2935000000 04 | 0.2790000000 04 | 0.1204000000 04 | 0.1965000000 04 |
| 0.1108000000 03 | 0.4445640000 06 | 0.4641000000 04 | 0.2470000000 04 | 0.1219000000 04 | 0.1963000000 04 |
| 0.1124000000 03 | 0.4270130000 06 | 0.3813000000 04 | 0.2552000000 04 | 0.1233000000 04 | 0.1945000000 04 |
| 0.1142000000 03 | 0.5026010000 06 | 0.3931000000 04 | 0.2514000000 04 | 0.1256000000 04 | 0.1960000000 04 |
| 0.1157000000 03 | 0.5181730000 06 | 0.4806000000 04 | 0.2572000000 04 | 0.1278000000 04 | 0.1961000000 04 |
| 0.1166000000 03 | 0.3549440000 06 | 0.4007000000 04 | 0.2827000000 04 | 0.1300100000 04 | 0.1917000000 04 |
CORRELATION MATRIX OF STANDARDIZED DATA

0.10000000  0  0.10000000  0  0.10000000  0  0.10000000  0
0.0  0.9915891780  0.0  0.10000000  0  0.10000000  0  0.10000000  0
0.0  0.0208339330  0.0  0.0047090400  0  0.10000000  0  0.10000000  0
0.0  0.9407541080  0.0  0.4494678780  0  0.1774286100  0  0.10000000  0
0.0  0.9701245290  0.0  0.901031050  0  0.6885790890  0  0.3045278180  0
0.0  0.991191900  0.0  0.9927344430  0  0.6687566093  0  0.4172451500  0
0.0  0.0937266490  0.0  0.10000000  0  0.10000000  0  0.10000000  0

OPTIMUM SUBSET SELECTED FROM THE 6 INDEPENDENT VARIABLES

(INTERCEPT ALREADY INCLUDED) ******* 1  3  4

1 POINT  A VARIABLE
39 0.11255163440  0.1
50 0.1126922710  0.1
51 0.1130346830  0.1
90 0.14382691930  0.1
90 0.14552157510  0.1
91 0.14553233820  0.1
94 0.1587400470  0.1
95 0.16486785940  0.1
96 0.16486785940  0.1

NORMAL TERMINATION: NO MORE DATA SETS
C***************************************************************
C
C THIS PROGRAM IS AN ALGORITHM FOR THE SUBSET SELECTION
C PROCEDURE DESCRIBED IN THE PAPER
C
C A SUBSET SELECTION PROCEDURE FOR REGRESSION VARIABLES *
C
BY GEORGE MCCARF AND JAMES ARVESSEN
C
DEPARTMENT OF STATISTICS
C
PURDUE UNIVERSITY
C
WEST LAFAYETTE, IN 47907
C
FEBRUARY 1973
C
MINED SERIES 316
C
C
C REFERENCES ARE MADE IN THE DOCUMENTATION OF THIS PROGRAM TO
C THE CORRESPONDING SECTIONS OF THE PAPER.
C
C PROGRAMR - RICHARD POHL
C
DEPARTMENT OF STATISTICS
C
PURDUE UNIVERSITY
C
FEBRUARY 1973
C
C***************************************************************

***** INPUT INFORMATION*****

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

N IS THE NUMBER OF OBSERVATIONS.
NP IS THE NUMBER OF INDEPENDENT VARIABLES (MUST BE LESS
THAN OR EQUAL TO TEN).
IT IS THE DESIRED SUBSET SIZE.
IN SPECIFIES THE TAPE FROM WHICH THE DATA IS TO BE READ
IF 'IN' IS NOT 0 THEN IT IS SET TO 3.
M IS THE NUMBER OF 'A' VALUES TO BE CALCULATED.
I.E., CURRENTLY M MUST BE LESS THAN OR EQUAL TO 1000. IF
A GREATER NUMBER IS DESIRED THEN CHANGE THE
DIMENSION OF 'A'.

IOPT IS AN OUTPUT OPTION. IF IOPT =
0 THEN THE INPUT DATA AND THE XTX MATRIX ARE NOT
PRINTED.
1 THEN THE INPUT VARIABLES ARE PRINTED.
2 THEN BOTH THE INPUT DATA AND THE XTX MATRIX ARE
PRINTED.

ICOR IF NON-ZERO, INPUT DATA IS CORRELATION MATRIX.
IXNUMB IF NON-ZERO USER SPECIFIES REGRESSION SUBSETS.
IPRT IS THE FORMAT OF THE INDEPENDENT VARIABLES TO BE READ
IN.
NLIST IS THE NUMBER OF BLOCKS OF THE 'A' VARIABLES TO BE
PRINTED. CURRENTLY NLIST MUST BE LESS THAN OR
EQUAL TO TEN.

LOW AND UPR ARE THE LOWER AND UPPER LIMITS OF THE NLIST
BLOCKS OF *.A* TO BE PRINTED.

THE FOLLOWING SUBROUTINES ARE CALLED BY THE PROGRAM:

(A) PROGRAMS WRITTEN OR REVISED FOR THIS PROGRAM.
NUMB IS A FUNCTION SUBPROGRAM WHICH RETURNS THE REFERENCE
TO AN R-SQUARED VALUE TO BE OBTAINED FROM SCANA.
PRMUT IS THE SUBROUTINE WHICH RETURNS THE PERMUTATIONS ON A
SET OF NUMBERS.
SCANA IS A SUBROUTINE WHICH CALCULATES ALL R-SQUARES FOR A
REGRESSION PROBLEM.
INDX IS A FUNCTION WHICH RETURNS THE SYMMETRIC STORAGE
MODE POSITION OF AN ELEMENT IN A MATRIX.
SUMSET PRINTS HEXADECIMAL IMAGE OF A SPECIFIC SUBSET OF
INDEPENDENT VARIABLES.

GAUSS IS A SUBROUTINE WHICH GENERATES NORMAL(0,1)
RANDOM NUMBERS VIA THE METHOD OF BOX-MULLER.
RANDU IS A SUBROUTINE WHICH GENERATES UNIFORM(0,1) NUMBERS
AND IS CALLED BY GAUSS.

**NOTE** SINCE RANDOM NUMBER GENERATORS ARE MACHINE DEPEND-
ENT, MODIFICATION WILL BE NEEDED TO RUN THIS PRO-
GRAM ON ANOTHER SYSTEM.

THE USER MAY WISH TO REPLACE THE CALL TO RANDU IN
THE GAUSS SUBROUTINE BY ANOTHER UNIFORM RANDOM
NUMBER GENERATOR OR THE USER MAY REPLACE GAUSS BY
ANOTHER NORMAL(0,1) RANDOM NUMBER GENERATOR.

(B) PUCC SUBPROGRAMS--
LINEQ1 IS A SUBROUTINE WHICH SOLVES THE SYSTEM OF LINEAR
EQUATIONS #AX FOR X WHERE A,X,ANDB ARE ARRAYS.
LINEQ1 CALLS A FUNCTION ARITH.

(C) SUBPROGRAMS FROM THE IMSL LIBRARY.
VSORTA IS A SUBROUTINE WHICH Sorts THE ELEMENTS OF A
VECTOR FROM LOW TO HIGH VALUE.
LINVIP IS A SUBROUTINE WHICH Calculates THE INVERSE OF
A MATRIX IN SYMMETRIC STORAGE MODE.

THE FOLLOWING SUBROUTINES ARE CALLED BY LINVIP--
LETP
LUDCIP
LUELP
VERTST

MATRIX XTP IS STORED IN SYMMETRIC STORAGE MODE TO SAVE

#25SP051
#25SP052
#25SP053
#25SP054
#25SP055
#25SP056
#25SP057
#25SP058
#25SP059
#25SP060
#25SP061
#25SP062
#25SP063
#25SP064
#25SP065
#25SP066
#25SP067
#25SP068
#25SP069
#25SP070
#25SP071
#25SP072
#25SP073
#25SP074
#25SP075
#25SP076
#25SP077
#25SP078
#25SP079
#25SP080
#25SP081
#25SP082
#25SP083
#25SP084
#25SP085
#25SP086
#25SP087
#25SP088
#25SP089
#25SP090
#25SP091
#25SP092
#25SP093
#25SP094
#25SP095
#25SP096
#25SP097
#25SP098
#25SP099
#25SP100
#25SP101
A35:.... THE ELEMENT I, J OF X** IS IN THE POSITION
I(K(1+I)/J) WHERE I IS LESS THAN OR EQUAL TO J.
SYMBOLIC MATRX L IS LOWER TRIANGULAR IT IS ALSO STORED IN
SYMTRIC STORAGE MODE.

*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

C

*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C IM(1:13/2) WHERE I IS LESS THAN OR EQUAL TO J.

C

*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

C

IMPLICIT REAL*8 (A-H,O-Z)
C DUMMY FORCES PROPER ALIGNMENT IN COMMON AREA
COMMON /BLEN, /DUMM, IJK1(1023), RL1(1035), RLZ1(1331)
COMMON /IMY, IX
REAL*4 (L,7), LABEL(9)
INTEGER IXNUMB(1100)
LOGICAL CORREL, NNUMB
DIMENSION X(I(1),JTX(100),I(1000)),XTX(100)
COMMON /SCRLNK/, DUMM, NP, &TX, ITM, NUMBR, RSQ1, RSO2
DIMENSION XPAR(10), ILON(10), IPL(10)
SORTD(1)=SORTD(4)

C

60I FORMAT(111)
12345 CONTINUE
READ(5,499,END=54321) LABEL
499 FORMAT(999)
READ(5,500) NNP, ITM, M, IDPT, IX, ICOR, IXNUMB
500 FORMAT(15,19,1X,11,1X,11)
CORREL=True,
IF I(8,NE,0) CORREL=TRUE,
READ(5,501) NLIST(1), ILON(I1), IPL(I1), I=1,NLIST)
501 FORMAT(115)
XNUMB=True,
IF I(8,NE,0) XNUMB=TRUE,
IF (ITM,NE,1) GO TO 5005
READ(5,503) XMNUMB, IXNUMB(1), I=I,MXNUMB)
503 FORMAT(15,5X,9G0(1028)
5005 CONTINUE
READ(5,502) FNT
502 FORMAT(108)
WRITE(6,50305) LABEL, NNP, ITM, M
50305 FORMAT(1) DATA DESCRIPTION : "NAB"=-1.6X,
1 = NO. OF OBSERVATIONS = 15/01, 5X;
2 = NO. OF INDEPENDENT VARIABLES = 10X (EXCLUDES INTERCEPT) = 3X,15",0",2X="SUBSET SIZE = 7","1","15/01,5X; 
4 = NO. OF "A" VARIABLES DESIRED = 6X;
IF (IN,NE,8) IN=S
IF (NP,GT,100) STOP

C

*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

C

INITIALIZE VALUES

C

*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*
C
IP=NP+1
NBP=NP+2
NBA=MM*(NB+1)/2
NBB=NB*(NB+1)/2
ANIP=FLOAT(N-1P)
AA=DOTP(1Z,..NZ,ANIP)
AR=2./NZ./ANIP+1
XTX(1)=N
DO 1 I=2,78
1 XTX(I)=0.D
C
C*********************************************************************************************
C READ DATA AND FORM XTX
C*********************************************************************************************
C IS INPUT CORRELATION MATRIX ?
IPI (=NOT.CORREL ) GO TO 99099
WRITE(6,99001)
99001 FORMAT('1INPUT CORRELATION MATRIX*/')
L=2
DO 99000 I=1,IP
L=L+1+I
L=L+I-1
READ(I6,FMT) (XTX(J,J)=L,L+2)
WRITE(6,99002) (XTX(J,J)=L,L+2)
99002 FORMAT('9E18.9')
99000 CONTINUE
GO TO 99999
99999 WRITE(6,99099)
99199 FORMAT('1INPUT DATA MATRIX*/')
DO 10 J=1,N
READ(I6,FMT) (X(J,J)=2,IP)
DO 5 J=J(J+1)/2
XTX(JSUB+J)=XTX(JSUB+J)+X(J)
DO 5 J=J(J+1)/2
XTX(JSUB+J)=XTX(JSUB+J)+X(J)
IF(I(J,J) .GE. 1) WRITE(6,600) (X(J,J)=2,IP)
10 CONTINUE
600 FORMAT('9E18.9')
DO 105 J=J(J+1)/2
JSUB=J(J+1)/2
DO 105 J=J(J+1)/2
JSUB=J(J+1)/2
JJSUB=JSUB+J
JJSUB=JJSUB+J
105 CONTINUE
JSUB1=JSUB+1
JSUBJ=JSUB+JJ
XXM=N
XNUM=XTX(JJSUBJ)-XTX(JSUB1)+XTX(JSUB1)/XXN
DENOM=(XTX(JJSUBJ)-XTX(JSUB1))*2/XXM*(XTX(JJSUBJ)-XTX(JSUB1))
1JSUB1=2/XXM
105 XTX(JJSUB1)=XNUM/(DOSRT(DENOM))
   DO 205 J=2,IP
      JSUB=J*J-1/J/2
      205 CONTINUE
   DO 251 J=2,IP
      XTX(I*J+I)=0.
      DO 251 K=1,J-1
         111 XTX(I*J+I)=XTX(I*J+I)
      251 CONTINUE

C**************************************************************************************
C**************************************************************************************
C**************************************************************************************
C C                99990 CONTINUE
C C                IF(IOPT. NE.2) GO TO 50
C C                IF(CORRELATE) GO TO 50
C C    99888 FORMAT(*,CORRELATION MATRIX OF STANDARDIZED DATA/*)
C C    DO 60 I=1,IP
C C    WRITE(6,99888)
C C    60 WRITE(6,640) (XTX(K+J,J*1),K=1,IP)
C C    640 FORMAT(*,7E10.9)
C C**************************************************************************************
C**************************************************************************************
C**************************************************************************************
C C                50 CONTINUE
C C                IF(XNUMB ) GO TO 60000
C C                NUMBA=NUMBA*XTX(NP+1,IT,N)
C C                L1=1
C C                L2=1
C C                XNUMB(I)=NUMBA
C C                GO TO 66666
C C    60000 WRITE(*,60000)
C C    66666 WRITE(*,60000)
C C**************************************************************************************
C**************************************************************************************
C**************************************************************************************
C FOR A PROCEDURE DESCRIPTION SEE
C COMPUTATIONAL PROCEDURES
C (B) ESTIMATION OF C INVERSE
FACTOR XTX INTO LTL SUCH THAT L IS LOWER TRIANGULAR.

60000 CONTINUE
WRITE(*,60100) NXXNUMB
61116 FORMAT(/' WILL PROCESS',IS,' SPECIFIED REGRESSION SUBSET(S)',/)
L=1
L2=NXXNUMB
C SAVE XTX FOR PROCESSING EXTRA NUMBERS
NXXTX=IP*(IP+1)/2
ON 16661 I=1,NXXTX
16661 XTX(I)=XTX(I)
60660 DO 77777 KA=L,L2
NUMBR=NUMBS(KK)
IF(I.NE.NUMBR)CALL SUBSET(NUMBR,HP,IT)
C IF(KK.GT.1)GO TO 60666
CALL LUDECP(XTX,L,IP,DET1,DET2,ERROR)
IF(ERROR.NE.129)GO TO 20
WRITE(*,610)
610 FORMAT(' XTX IS SINGULAR')
STOP
20 CONTINUE
DO 49 I=1,(P)
K=IP(I+1)/2
49 L(K)=L(K)*L(I)
C
C**************************************************************************************************
C
C CALCULATE THE 'A' VARIABLES.
C
C GENERATE THE INDEPENDENT VARIABLE COLUMNS FROM THE NORMAL
C RANDOM NUMBER GENERATOR.
C
C**************************************************************************************************
C
60600 CONTINUE
DO 100 LL=1,N
AP=GAUSSI(0.,0.,0.)
XTX(NBB)=ANP*(AP**AA+AB)**3
DO 10 I=1,IP
XI=GAUSSI(0.,0.,0.)
XTX(NBA)=XTX(NBA)*XI**2
XTX(NBA+I)=0.0
K=I*(I-1)/2
10 X(NBA+I)=XTX(NBA+I)+L(K+J)*XI
CALL SCRIA
IF(NUMBR.EQ.845014520 )GO TO 77777

CALL VSORTA(A,M)
WRITE(6,2553)
2020 FORMAT(E15.0)
DO 200 1=1,NLIST
10 I=I+1
200 WRITE(6,2554) (A(I),I=1,NLIST)
2553 RESTORE KTX
7777 IF (.NOT.XNUMB) GO TO 7777
DO 7777 I=1,NKTX
7777 CONTINUE
7777 CONTINUE
GO TO 12365
210 WRITE(6,2550) 1A,IB
630 FORMAT(/1X,'INVALID RANGE FOR OUTPUT OF A VARIABLES')
GO TO 12365
54321 CONTINUE
WRITE(6,55544)
55544 FORMAT('NORMAL TERMINATION: NO MORE DATA SETS')
STOP
END
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(I),POW(11)
DATA POW1, 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(6,11) P,(SET(I),I=1,T)
   1 FORMATTED SPECIFIED SUBSET FROM THE 'P' INDEPENDENT VARIABLES
     /*6X,*(INTERCEPT ALREADY INCLUDED) ******9,11161
C
RETURN
END
FUNCTION NUMBEX, P, T, N)

FUNCTION NUMB FINDS THE INDEPENDENT VARIABLES OF THE
GIVEN SUBSET SIZE T TO BE USED IN FINDING
A = IL.RSU(W)/I1-PSUDEL1) IN THE MAINLINE
PROGRAM.

SEE COMPUTATIONAL PROCEDURES
(A) DETERMINATION OF K (PAGE 6-7)

XTX IS X TRANSPOSE X WHERE X IS THE INPUT DATA.
XTX IS IN SYMMETRIC STORAGE MODE.

P IS THE NUMBER OF INDEPENDENT VARIABLES.
T IS THE GIVEN SUBSET SIZE.
N IS THE NUMBER OF OBSERVATIONS.

****NOTE**** KM(K+1)/2 MUST NOT EXCEED THE DIMENSION
OF GAMMA AND GINV.  K IS THE NUMBER OF PERMUTATIONS.
P THINGS TAKEN T AT A TIME.

C IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XTX(N), X(I,II), Y(I,II), Z(I,II)
DIMENSION ZONE(I,II), ZONE(I,II)
DIMENSION C(I,II), R(I,II), E(I,II)
INTEGER PERM(I,II), PA, PA, TA, SELECT(I,II)

C I/DUMMY PERMICES ALIGNMENT
COMMON /BLOCK/ I/DUMMY, PERM, GAMMA(I=ID), GINV(IID)
EQUIVALENCE (Y(I,II),GINV(I,II),X(I,II),GINV(I,II))
EQUIVALENCE (ZONE(I,II),GINV(I,II),X(I,II),GINV(I,II))
EQUIVALENCE (XTX(I,II),GINV(I,II),X(I,II),GINV(I,II))
EQUIVALENCE (C(I,II),GINV(I,II))
PA, PA,
TA=T+1
AA=N-2*T-2
AD=N-1
CALL PRMTA(PA,TA,K)
GAMMA(I)=1.0

C FOR EACH I=1,....K AND J=1,....K THE PROGRAM
C DETERMINS X,Y AND XY WHERE XY REFERS TO THOSE ELEMENTS
C OF XTX WHICH HAVE THE ROW COORDINATES PERM(I,1),....
C PERM(I,TA) AND THE COLUMN COORDINATES PERM(J,1),....
C PERM(J,TA).  Y IS FORMED THE SAME WAY EXCEPT THAT I IS
C REPLACED BY J.

C NUMBEX001
C NUMBEX002
C NUMBEX003
C NUMBEX004
C NUMBEX005
C NUMBEX006
C NUMBEX007
C NUMBEX008
C NUMBEX009
C NUMBEX010
C NUMBEX011
C NUMBEX012
C NUMBEX013
C NUMBEX014
C NUMBEX015
C NUMBEX016
C NUMBEX017
C NUMBEX018
C NUMBEX019
C NUMBEX020
C NUMBEX021
C NUMBEX022
C NUMBEX023
C NUMBEX024
C NUMBEX025
C NUMBEX026
C NUMBEX027
C NUMBEX028
C NUMBEX029
C NUMBEX030
C NUMBEX031
C NUMBEX032
C NUMBEX033
C NUMBEX034
C NUMBEX035
C NUMBEX036
C NUMBEX037
C NUMBEX038
C NUMBEX039
C NUMBEX040
C NUMBEX041
C NUMBEX042
C NUMBEX043
C NUMBEX044
C NUMBEX045
C NUMBEX046
C NUMBEX047
C NUMBEX048
C NUMBEX049
C NUMBEX050
XY is formed by the (TA by TA) submatrix of XTX with the row coordinates PERM(1), ..., PERM(j, TA) and the column coordinates PERM(j, 1), ..., PERM(j, TA).

DO 200 I=1, K
   IM1=I-1
   INDEX=I*(I+1)/2
   GAMMA(INDEX)=L.

DETERMINE X

DO 50 LOOP=1, TA
   INDEX=PERM1, LOOP)
   DO 50 LOOP=LOOP, TA
   INDEX=PERM1, LOOP)
   INDEX=INDEX(INDEX, INDEX)
   X(LOOP, LOOP)=XTX(ISUB)
   X(LOOP, LOOP)=XTX(ISUB)
50 CONTINUE

DO 150 J=1, IM

DETERMINE Y AND XY

DO 55 LOOP=1, TA
   INDEXA=PERM(j, LOOP)
   DO 55 LOOP=LOOP, TA
   INDEXA=PERM(j, LOOP)
   ISUB=INDEX(INDEXA, INDEX)
   Y(LOOP, LOOP)=XTX(ISUB)
   Y(LOOP, LOOP)=XTX(ISUB)
55 CONTINUE

FIND ZONE SUCH THAT X*ZONE=XY
IF(SING.EQ.0) GO TO 60
WRITE(*,610) T,J
610 FORMAT(I,'MATRX X IS SINGULAR',5X,2I5)
STOP

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

C TAKE TRANSPOSE OF XY MATRIX.
********************************************************************

60 DO 65 LOOP=1,T
   LOOPB=LOOP+1
   DO 65 LOOPA=LOOPB,T
       SAVE=XY(LOOP,LOOPA)
       XY(LOOP,LOOPA)=XY(LOOPA,LOOP)
   65
   XY(LOOPA,LOOP)=SAVE

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

C FIND XTWO SUCH THAT Y*XTWO=XY.
********************************************************************

CALL LINEQ(I,Y,X,ZTWO,11,T,A,SING,C,R,E)
IF(SING.EQ.0) GO TO 70
WRITE(*,-620) I,J
70 CONTINUE
STOP

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

C FIND TRACE OF ZONE * ZTWO.

PERFORM OPERATION ON IT AND STORE IN GAMMA.

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

C SUM=0.D0
DO 75 LOOP=1,T
   DO 75 LOOPB=1,T
      SUM=SUM+ZONE(I,LOOP)*ZTWO(LOOPB,LOOP)
   75 INDEX(I)=SUM/AB
   GAMMA(I)=INDEX(I)*(TA+SUM)/AB

CONTINUE
200 CONTINUE
C
C
********************************************************************
C TAKE INVERSE OF GAMMA
C*******************************************************************************
C CALL LINV1PGAMMA,X,GINV,7,X,IER1
 IF (IER1 .NE. 129) GO TO 210
 WRITE(6,630) 
 630 FORMAT(12G15.7,1X,GAMMA DID NOT HAVE AN INVERSE')
STOP
C*******************************************************************************
C FIND THE SMALLEST ELEMENT ON THE DIAGONAL OF GAMMA
C INVERSE.
C*******************************************************************************
C 210 INDEX=1
 VALUE=GINV(1)
 DD=220 LOOP=2,K
 LOOPA=LOOP*(LOOP-1)/2+LOOP
 IF (GINV(L,JPAJ+J-VALUE) .GE. VALUE) GO TO 220
 VALUE=GINV(LOOPA)
 INDEX=LOOP
 LOOP=LOOP-1
 CONTINUE
C*******************************************************************************
C THE ELEMENTS (MINUS 1) OF THE (INDEX)TH PERMUTATION IN
C PERM (2,...,T,A) 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.)
C*******************************************************************************
C NUMB=0
 DO 230 I=2,T,A
 J=PERM(INDEX,I)-2
 SELECT(I)=J+1
 NUMB=NUMB+2**J
230 WRITE(6,90000) P(SELECT(I),J+1)
90000 FORMAT(12I5) ' INDEPENDENT VARIABLES**6X,1 (INTERCEPT ALREADY INCLUDED)' 
RETURN
END
FUNCTION INDX(I,J)

C***************************************************************************
C
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***************************************************************************

C
INDX=I+J+IABS(I-J)/2
INDX=INDX*(INDX-1)/2
INDX=INDX*(I+J-IABS(I-J))/2
RETURN
END

INDEX0001
INDEX0002
INDEX0003
INDEX0004
INDEX0005
INDEX0006
INDEX0007
INDEX0008
INDEX0009
INDEX0010
INDEX0011
INDEX0012
INDEX0013
INDEX0014
INDEX0015
INDEX0016
SUBROUTINE SCNA
IMPLICIT REAL*4 (A-H,O-Z)

SUBROUTINE SCNA IS A REGRESSION ANALYSIS PROGRAM WRITTEN
BY GEORGE M. PUNNIVAL 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 ATA MATRIX IS NOW IN SYMMETRIC STORAGE MODE.
THIRD INSTEAD OF OUTPUTING THE R-SQUARES THE PROGRAM
STORES THEM ALONG WITH THE BIT PATTERN CORRESPONDING TO THE INDEPENDENT VARIABLES USED.

1111 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 ONES 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 THEREFORE THE BIT
PATTERN OF JNUM GIVES THE INDEPENDENT VARIABLES IN
THE REGRESSION.

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

REAL*8 JSAVE(1035), KSAVE

DIMENSION MN110, MM110, JSAVE(11231)
DIMENSION N111, N66(78), A(11,11,11)
COMMON /SCRLK/ JNUMMm,N66, IT, NUM1, RSQ1, RSQ2
COMMON /BLOCK/ NUMMM, JSAVE, JSAVE, A

DATA MN110, MM110, JSAVE, A
1 IF ( N.GT.10 ) STOP
100 DD 100 I=1,10
100 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
NAN+1
NBX+2
NBA=NBR(NB-1)/2
SS+10/DSQRT(45-1)-SS(NBA+1)-SS(NBA+1)/SS(11)
DO 1 I=1,NA
ISUB=I-1(I-1)/2
NTI=I-1-0
B=SS(ISUB+1)/SS(1)
A(INA,1,1)=SS(B)-SS(NBA+1)-SS(NBA+1)
DO 1 J=1,NA
JSUM=J(J-J-1)/2
A(INA,1,J)=B*SS(JSUB+1)-SS(JSUB+1)
1 CONTINUE
A(INA,1,1)=0.
NTINAI-1
JNUM=0
11 DO 20 I=1,N
IFI NTAI=EQ.0 ) GO TO 21
JNUM=JNUM-1
NTAI=0
20 CONTINUE
10 LIMAX=MIN(N)
LIMAX=MIN(N)
ASQ=JSAVE(LIMAX)
DO 60 LOOP=LIMAX,LIMB
IFI JNUM.EQ.0 )SAVE=LOOP) ) GO TO 61
60 CONTINUE
WRITE(6,600) NUMB
600 FORMAT(1X,10) THE CORRECT BIT PATTERN COULD NOT BE FOUND : *283
NUMB = NA5014520
RETURN
61 ASQ=JSAVE(LOOP)
RETURN
21 NTI=1
JNUM=JNUM+1
JNUM=JNUM+1
K=1
DO 22 J=K,NA
IFI NTI(J) EQ.1 ) GO TO 30
22 CONTINUE
30 DO 31 L=1,1
DO 31 L=1,1
DO 31 H=1,1
31 CONTINUE
KSAV=AA1,1,11,01
MMJNUM=MM(JNUM)+1
LIMAX=MM(JNUM)
LIMAX=MM(JNUM)
IFI LIMAX.EQ.1 )SAVE=1 ) GO TO 50
LIMAX=LIMAX+1
RETURN
DO 45 LOOP=LIM1=LIM2
LOOP=LIM2=LIM1=LOOP
1 IF SAVE(Loop)=SAVE(Loop-1) I GO TO 55

45 SAVE(Loop)=SAVE(Loop-1)
50 LOOP=MIN1=NUM
55 SAVE(Loop)=SAVE(Loop)=NUM
GO TO 11
END
SUBROUTINE PRMUT(P,T,K)

PRMUT IS A SUBROUTINE WHICH DETERMINES THE PERMUTATIONS ON A SEQUENCE OF NUMBERS. THE PERMUTATION IS TAKEN ON IP THINGS TAKEN T AT A TIME.

BECAUSE THE REGRESSION ANALYSIS PROGRAM SCRA 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 IT-1 AT A TIME DONE ON THE INTEGERS 2,3,...,P.

THE USAGE OF THE SUBROUTINE IS CALL PRMUT(P,T,K,PERM)

P IS THE NUMBER OF INDEPENDENT VARIABLES PLUS 1.
T IS THE SUBSET SIZE BEING CONSIDERED PLUS 1.
K IS THE RESULTING NUMBER OF PERMUTATIONS.
PERM IS THE ARRAY WHICH STORES THE DETERMINED PERMUTATIONS.
ALL ARGUMENTS ARE INTEGERS.

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

COMMON /BLOCK/ IDUMMY, PERN
INTEGER P,T,PERMN93,111,PM,TM
PM=P-1

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

CALCULATE K

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

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

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

FILL OUT THE ARRAY PERM.

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

I=1
DO 15 LOOP=1,T
15 PERM(I,LOOP)=LOOP
C 20 I=1+1
IF(I.EQ.K+1) GO TO 100
GO TO 50 LOOP=1,T
50 PERM(I,LOOP)=PERM(I-1,LOOP)
IF(PERM(I,T).EQ.P) GO TO 55
PERM(I,T)=PERM(I,T)+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
L=L+1
DO 75 LOOP=LA,P
75 PERM(I,LOOP)=PERM(I,LOOP)+1
GO TO 20
C 100 CONTINUE
RETURN
END
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: NEWHAN AND GOOD: THE GENERATION OF RANDOM VARIATES.
C
REAL*8 GAUSS,SD,AM
LOGICAL FIRST
COMMON /IHX/ IX
DATA FIRST/.FALSE./
C IF (.NOT.FIRST) GO TO 100
C CALL RANDU(IHX,IX,Y)
IX=IX+1
CALL RANNI(IHX,IX,Y)
IX=IX+1
C S=SIGN(-2.*ALOG(1.))
P=0.28332867
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
SUBROUTINE LINEQ1(A,B,X,N,D,NR,SR,S,G,E)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(N,D),B(N,NR),X(N,NR),C(N,NR),R(N),E(N)
INTEGER S
ABS10=DABS(D)
AMAX1(F,G)=AMAX1(F,G)

SUBROUTINE LINEQ1

DECKNAME - LINEQ1

PURPOSE
SOLVES THE REAL MATRIX EQUATION AX=B WITH NR RIGHT-HAND SIDES.

USAGE
CALL LINEQ1(A,B,X,N,D,NR,S)

DESCRIPTION OF PARAMETERS
A - (N X N) REAL COEFFICIENT MATRIX.
B - (N X NR) REAL RIGHT-HAND SIDE ARRAY.
X - (N X NR) REAL ARRAY FOR RETURN OF SOLUTION VECTORS.
ND - THE NUMBER OF ROWS FOR THE ARRAYS A, B, AND X.
N - THE NUMBER OF EQUATIONS TO BE SOLVED.
NR - THE NUMBER OF RIGHT-HAND SIDES TO BE SOLVED.
S - INTEGER VARIABLE RETURNED NON-ZERO ONLY IF MATRIX A IS SINGULAR TO MACHINE ACCURACY.

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 DYNAPIR.

METHOD
THE MATRIX A IS FACTORED INTO LOWER AND UPPER TRAPEZIAL MATRICES L AND U AND THEN THE EQUATIONS LI=B AND UXZ 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.
FORM EQUILIBRATION FACTORS IN VECTOR E.

DO 12 I=1,N
   P=0.0
   DO 10 J=1,N
      C(I,J)=A(I,J)
      10 P=P+MAX1(ABS(C(I,J)),P)
   IF(P.EQ.0.0) GO TO 73
   12 E(I,J)=1.0/P

FACTOR COEFFICIENT MATRIX WITH PARTIAL PIVOTING.

M=1
14 MM=M-1
   P=0.0
   DO 22 J=K,N
      C(I,J)=ARITH(C(I,J),MM,C(I,J),N,C(I,J),1)
      Q=E(J)*ABS(C(I,J))
      IF(P.GE.Q) GO TO 22
      P=Q
      K=I
   22 CONTINUE
   IF(M.EQ.N) GO TO 73
   IF(M.EQ.K) GO TO 30
   DO 28 J=1,N
      P=C(J,J)
      C(J,J)=C(I,J)
52 C(I,J)=P
   28 E(I,J)=C(I,J)
   30 E(M,H)=C(M,H)
   IF(M.EQ.N) GO TO 37
   MM=M+1
   DO 34 J=MP,N
      C(J,J)=ARITH(C(J,J),MM,C(J,J),N,C(J,J),1)
      GO TO 14
   34 MM=MP
   GO TO 37

BACK SUBSTITUTE RIGHT-HAND SIDES WITH ITERATIVE REFINEMENT.

37 IF(M.EQ.N) GO TO 71
   DO 43 M=1,N
      P=0.0
      Q=0.0
      DO 43 I=1,N
R(I,J)=R(I,M)

43 X(I,M)=0.0

44 DO 48 I=1,N

K=I(I)

T=R(K)

R(K)=R(I)

48 R(I)=ARITH(R(I-1),C(I,J),N,R(I),1)/C(1,1)

IF(N .GT. 0) GO TO 52

51 R(I)=ARITH(R(I),N-1,C(I,J),T(I),N,R(I),1)

52 T(I)=1

IF(I .GT. 0) GO TO 51

END

59 X(I,N)=X(I,M)+R(I)

C TEST FOR CONVERGENCE OF ITERATIVE REFINEMENT.

C IF(P .EQ. 0.0) GO TO 68

C IF(Q .EQ. 0.0) Q=P

C IF ( I.SCAL .LT. 0.0 .AND. P.GT.T(I) ) GO TO 68

C DO 65 I=1,N

65 R(I)=ARITH(R(I),N,K(I),1,0.0,N,R(I),1)

GO TO 44

67 IF ( I.SCAL .LT. 0.0 .AND. I .LT. 1.0 ) GO TO 73

68 CONTINUE

C SET NON-SINGULAR/SINGULAR FLAG AND RETURN.

C S=0

RETURN

C 71 S=1

WRITE(*,700) Q,P,N

C FORMAT(*-MATRIX SINGULAR IN LINEQ I Q, P, N = *,5X)

ON(50,6,19)

RETURN

C END
FUNCTION ARITH(C,N,A,KA,B,KB)
C
C CALLED FROM LINEQ1
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(KA,N),B(KB,N)
T=C
IF(N.EQ.0) GO TO 5
DD 4 I=1,N
4 T=T-A(I+1,1)*B(I+1,1)
5 ARITH=T
RETURN
END
SUBROUTINE VSORTN (A,LA)
C
FUNCTION VSORTN - SORT ARRAYS BY ABSOLUTE VALUE
C
USAGE
- CALL VSORTN (A,LA)
- CALL VSORTA (A,LA)
C
PARAMETERS
A - ON INPUT, CONTAINS THE ARRAY TO BE SORTED
- ON OUTPUT, A CONTAINS THE SORTED ARRAY
LA - INPUT VARIABLE CONTAINING THE NUMBER OF
- ELEMENTS IN THE ARRAY TO BE SORTED
C
PRECISION - SINGLE
C
AUTHOR/IMPLEMENTER - N.E. ROSTEN
C
LANGUAGE - FORTRAN
C
LATEST REVISION - DECEMBER 7, 1970
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(I1),I(U1),I(U2)
C
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
LA=LA
10 IF (I .EQ. J) GO TO 55
15 IF (R .GT. 0.890437) GO TO 20
R=R+3.90625E-2
GO TO 25
20 R=R-2.1075
25 K=I
C
I=I+1,J=J+1,R
T=A(I)
C
IF (A(I) .LE. T) GO TO 30
A(I)=A(I)
A(I)=T
T=A(I)
C
30 L=J
C
SELECT A CENTRAL ELEMENT OF THE
ARRAY AND SAVE IT IN LOCATION T
C
IF FIRST ELEMENT OF ARRAY IS GREATER
THAN T, INTERCHANGE WITH T
C
IF LAST ELEMENT OF ARRAY IS LESS THAN VSORT010
IF A(I) .GE. T) GO TO 40
A(I)=A(J)
A(J)=T
T=A(I)
GO TO 40
35 IF (A(I) .LE. T) GO TO 40
A(I)=A(J)
A(J)=T
GO TO 40
T, INTERCHANGE WITH T
VSORT051
VSORT052
VSORT053
VSORT054
VSORT055
IF FIRST ELEMENT OF ARRAY IS GREATER THAN T, INTERCHANGE WITH T
VSORT056
VSORT057
VSORT058
VSORT059
VSORT060
VSORT061
VSORT062
VSORT063
VSORT064
VSORT065
FIND AN ELEMENT IN THE SECOND HALF OF THE ARRAY WHICH IS SMALLER THAN T
VSORT066
VSORT067
VSORT068
FIND AN ELEMENT IN THE FIRST HALF OF THE ARRAY WHICH IS GREATER THAN T
VSORT070
VSORT071
VSORT072
VSORT073
VSORT074
VSORT075
INTERCHANGE THESE ELEMENTS
SAVE UPPER AND LOWER SUBSCRIPTS OF THE ARRAY YET TO BE SORTED
VSORT077
VSORT078
VSORT079
VSORT080
VSORT081
VSORT082
VSORT083
VSORT084
VSORT085
VSORT086
BEGIN AGAIN ON ANOTHER PORTION OF THE UNSORTED ARRAY
VSORT087
VSORT088
50 L=L-1
IF (A(L) .GT. T) GO TO 40
40 K=L+1
IF (A(K) .LT. T) GO TO 45
C IF (K .LE. L) GO TO 35
C IF (L-1) .LE. J-K) GO TO 50
10 U=M+L
J=L
M=M+1
GO TO 60
55 K=M+1
IF (M .EQ. 0) RETURN
I=I+1
J=I+1
60 IF (I .LT. L) GO TO 25
IF (I .EQ. I) GO TO 10
70 A(K)=A(I)
K=K+1
IF (I .LT. J) GO TO 70
A(K+1)=T
GO TO 65
END
SUBROUTINE LUDECP (A,UL,N,D1,D2,IER)

FUNCTION
- CHOLESKY DECOMPOSITION OF A MATRIX
- SYMMETRIC STORAGE MODE

USAGE
- CALL LUDECP (A,UL,N,D1,D2,IER)

PARAMETERS A - A IS N X N POSITIVE DEFINITE MATRIX AND IS
- IN SYMMETRIC STORAGE MODE.
UL - THE RESULT L OF THIS ROUTINE IS STORED IN THE LUDECP9
- MATRIX UL. UL MAY OCCUPY THE SAME STORAGE AS A.
- THE MAIN DIAGONAL VALUES OF UL ARE STORRED INRECI.
- PROCAL FORM.
N - ORDER OF A
D1 - A NUMEBR COMPUTED IN THE SUBROUTINE SUCH THAT
- THE DETERMINANT OF A = D1*2.**D2
D2 - SEE D1
IER - ERROR PARAMETER
- TERMINAL ERROR = 128 * 11

N = 1 INDICATES THAT THE MATRIX IS
- SINGULAR

PRECISION
- SINGLE

REDD. IMSL ROUTINES - UPTST

AUTHOR/IMPLEMENTER - U.G. JOHNSON/FW. CHOU

LANGUAGE - FORTRAN

LATEST REVISION - MAY 10, 1972

IMPLICIT REAL*(A-H,O-Z)
DIMENSION A(1:1),UL(1)
DATA ZERO,ONE,FOUR,SIXTH,SIXTH/0.00001.0,4.,6.,16./LUDECP10
I*I+I
ABS(D)=DABS(D)
D1=ONE
D2=ZERO
IP=0
IER=0
DO 40 I=1,N
40 IQ=IP+1
IR=0
DO 35 J=1,N
35 K=K+1
IF (IP.LT.IQ) GO TO 10
30 40 9  CONTINUE
10 IR=IR+1
IP=IP+1
IF (I.NE.J) GO TO 30
D1 = D1*X
IF (X.NE.0.000) GO TO 15
D2 = ZERO
GO TO 45
15 IF (ABS(D1).LT.ONE) GO TO 20
D1 = D1 + SIXTH
D2 = D2 + FOUR
GO TO 15
20 IF (ABS(D1).GE.SIXTH) GO TO 25
D1 = D1 + SIXTH
D2 = D2 + FOUR
GO TO 20
25 IF (X.LT.0.000) GO TO 45
UL(IIP) = 1.00/DSQRT(X)
GO TO 35
30 UL(IIP) = X * UL(IIP)
35 CONTINUE
40 CONTINUE
GO TO 9005
45 IER = 129
9000 CONTINUE
CALL UERFST (IER,6,MLUDECP)
9005 RETURN
END
SUBROUTINE UERTST(IER,NAME)  
C      FUNCTION   - ERROR MESSAGE GENERATION  
C      USAGE     - CALL UERTST(IER,NAME)  
C      PARAMETERS IER   - ERROR PARAMETER, IER = N WHERE N =  
C             - 12A: IMPLIES TERMINAL ERROR  
C             - 44: IMPLIES WARNING WITH FIX  
C             - 32: IMPLIES WARNING  
C             - N = ERROR CODE RELEVANT TO CALLING ROUTINE  
C             - THE LITERAL STRING IDENTIFYING THE NAME OF  
C             - THE CALLING ROUTINE FROM A TO 6 ALPHABETIC  
C             - CHARACTERS  
C      NAME      - PETER SWEDEN  
C      MODULE    - FORTTRAN  
C      LATEST REVISION - JANUARY 19, 1971  
REAL I,TYP,NAME  
DIMENSION ITPY(2,1,12),EQUIV  
INTEGER WTH,WARF,TER,PRINTR  
EQUIVALENCE (EQUIV,IWARF,EWARF),(EIT(10),TERM)  
DATA ITPY /BHUMAN,TERM,ERR,TWTH,RIFIX/  
*  
*     IMPLIES WARNING  
*     IMPLIES WARN  
*     IMPLIES DEFINED  
*     IER / 32.12.44.44  
*     DATA PRINTR / 8/  
*     IF (IER .GE. WARN) GO TO 5  
C     NON-DEFINED  
C     GO TO 02  
C     TERMINAL  
C     GO TO 20  
C     WARNING(WITH FIX)  
C     WARNING  
C     EXTRACT NAME  
C     PRINT ERROR MESSAGE  
C     NAME = NAME  
C     FORMAT *** I N IER(IER) *** ,AM,43,AM,44,44,12  
RETURN  
END
SUBROUTINE LEEQTP (A,M,N,IB,B,LDGT,WKAREA,IER)
IMPLICIT REAL*8 (A-H,O-Z)

FUNCTION
mode = SPACE ECONOMIZER SOLUTION

USAGE
- CALL LEETIP (A,M,N,IB,B,LDGT,WKAREA,IER)

PARAMETERS
A - THE COEFFICIENT MATRIX OF THE EQUATION
B - AX = B, WHERE A IS ASSUMED TO BE POSITIVE
DEFINITE IF SIZE N X N AND IS IN SYMMETRIC STORAGE MODE, A IS REPLACED BY THE N X N

LATEST REVISION
- MAY 19, 1972

1010 CONTINUE
N - NUMBER OF COLUMNS IN MATRIX B
N = ORDER OF A AND NUMBER OF ROWS IN B

B - MATRIX OF THE RIGHT HAND SIDE OF THE EQUATION
AX = B

C - WORK AREA OF DIMENSION GREATER THAN OR
C = SINGLE
EQUAL TO N
C = ERROR PARAMETER
C = TERMINAL ERROR = 128 + N
C = INDICATES THAT LEEQTP FAILED TO
C = FIND A SOLUTION, A IS SINGULAR

DIMENSION
A(1),B(1,1),WKAREA(1)

INITIALIZE IER

IER = 0
CALL LUEECP (A,M,N,DL,DZ,IER)

IF (IER .NE. 0) GO TO 15

PERFORM ELIMINATION
DO 10 I=1,M
CALL LUEMP (A,M,N,DL,DZ,IER)
10 CONTINUE
GO TO 9005

IER = 129
9005 CONTINUE

CALL UERTTP (IER,6,LEETIP)
RETURN
END
SUBROUTINE LINVIP (A,N,AINV,IDGT,WKAREA,IER)

C
C
FUNCTION
C
C
USAGES
C
C
PARAMETERS
A
C
N
C
AINV
C
IDGT
C
WKAREA
C
IER
C
C
DIMENSION A(1:N),AINV(1:N),WKAREA(1:N)
C
DATA IER=0
C
C
K=N
C
DO 25 I=1,N
C
L=L+1
C
WKAREA(L)=ZERO
C
IF (J.EQ.I) WKAREA(I)=ONE
C
5 CONTINUE
C
IF (IER.NE.1) GO TO 10
C
DECOMPOSE A
C
CALL LEQIP (A,1:N,AINV(N),K,WKAREA,IER)
C
IF (IER.GT.0) GO TO 15
C
GO TO 30
C
C
COMPUTE THE INVERSE AND MOVE INTO
C
ARRAY AINV
C
CALL LUELMP (A,N,WKAREA(K+1),N,WKAREA(K+1),IER)
C
K1 = K1+1/2 +1
C
DO 20 J=1,N
C
20 CONTINUE
C
GO TO 9005
C
9000 CONTINUE
C
CALL UERST (IER,6,LINVIP)
C
9005 RETURN
C
END
SUBROUTINE LUELHP (A,B,N,X)
C
FUNCTION
  - ELIMINATION PART OF SOLUTION OF A*X = B -
  SYMMETRIC STORAGE MODE
C
USAGE
  - CALL LUELHP (A,B,N,X)
C
PARAMETER
  A - THE RESULT L, COMPUTED IN THE ROUTINE
  *LUDECP*, WHERE L IS LOWER TRIANGULAR,
  THE MAIN DIAGONAL ELEMENTS OF L ARE STORED
  IN RECIPROCAL FORM
  B - VECTOR OF LENGTH N ON THE RIGHT HAND SIDE
  OF THE EQUATION A*X = B
  N - ORDER OF A AND THE LENGTH OF B AND X
  X - THE RESULTANT SOLUTION, X
C
PRECISION
  - SINGLE
C
AUTHOR/IMPLEMENTER - Q.G. JOHNSON/E.W. CHOU
C
LANGUAGE
  - FORTRAN
C

C LATEST REVISION - MAY 10, 1972

C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(I),B(I),X(I)
C
IP=1
DO 15 I=1,N
  IO=I-1
  T=X(I)
  IF (IJ,G.E.1) GO TO 10
  DO 5 K=1,IO
    T = T - A(I,K)*X(K)
    IP=IP+1
  5 CONTINUE
  S = T/X(I)
  IP=IP+1
  X(I)=S
  IP=IP+1
15 CONTINUE
C
SOLUTION OF UX = Y
C
DO 30 J=1,N
  I=J-1
  IP=IP+1
  IQ=I+1
  T=X(I)
  IF (J,J.GT.1) GO TO 25
  JJ=0
  DO 20 K=1,Q
    JJ=JJ+1
    T = T - A(I,K)*X(JJ)
    I=I+1
  20 CONTINUE
  30 CONTINUE
RETURN
END

LUELMO01
LUELMO02
LUELMO03
LUELMO04
LUELMO05
LUELMO06
LUELMO07
LUELMO08
LUELMO09
LUELMO10
LUELMO11
LUELMO12
LUELMO13
LUELMO14
LUELMO15
LUELMO16
LUELMO17
LUELMO18
LUELMO19
LUELMO20
LUELMO21
LUELMO22
LUELMO23
LUELMO24
LUELMO25
LUELMO26
LUELMO27
LUELMO28
LUELMO29
LUELMO30
LUELMO31
LUELMO32
LUELMO33
LUELMO34
LUELMO35
LUELMO36
LUELMO37
LUELMO38
LUELMO39
LUELMO40
LUELMO41
LUELMO42
LUELMO43
LUELMO44
LUELMO45
LUELMO46
LUELMO47
LUELMO48
LUELMO49
LUELMO50
LUELMO51
LUELMO52
LUELMO53
LUELMO54
LUELMO55