match no. 12 pp. 135-167 1981

MOLECULAR INTERACTIONS IN BIOLOGICAL SYSTEMS. III[4]. THE SIBIS AND HIBIS PROGRAMS. APPLICATIONS.

O.Dragomir-Filimonescu[a)], I.Moţoc[b)] and I.Muscutariu[c)]

a) Computer Research Center
str. Gh. Lazar no. 9, R-1900 Timişoara,
ROMANIA

b) Chemistry Research Center (*)
Bv. M. Viteazu no. 24, R-1900 Timişoara,
ROMANIA

c) University of Timişoara, Department of Physics,
R-1900 Timişoara, ROMANIA

(Received: February 1981)

Summary. The SIBIS and HIBIS programs implement our algorithms developed to quantify the steric (SIBIS) and hydrophobic (HIBIS) interactions in biological systems. The paper presents the flowcharts, the complete listing in FORTRAN of the programs, and details the structure of the input/output data. Illustratively, one performs SIBIS and HIBIS calculations for a serie of twenty inhibitors (triazines) of L.casei dehydropholate reductase, good results being reported. Finally, the algorithm CCNEX, which teaches the computer to recognize connected graphs, is given.

(*) all correspondence could be sent at this address

1. The SIBIS and HIBIS Programs

The two algorithms (SIBIS - Steric Interactions in BIological Systems and HIBIS - Hydrophobic Interactions in BIological Systems) were discussed and applied to drug - receptor interactions in previous papers (i.e., refs. 1-3, and 4, respectively).

The present paper contains the flowchart, the complete listing of the programs and the structure of the input/output data. An application of these algorithms to a serie of twenty inhibitors of L.casei dihydropholate reductase is also given.

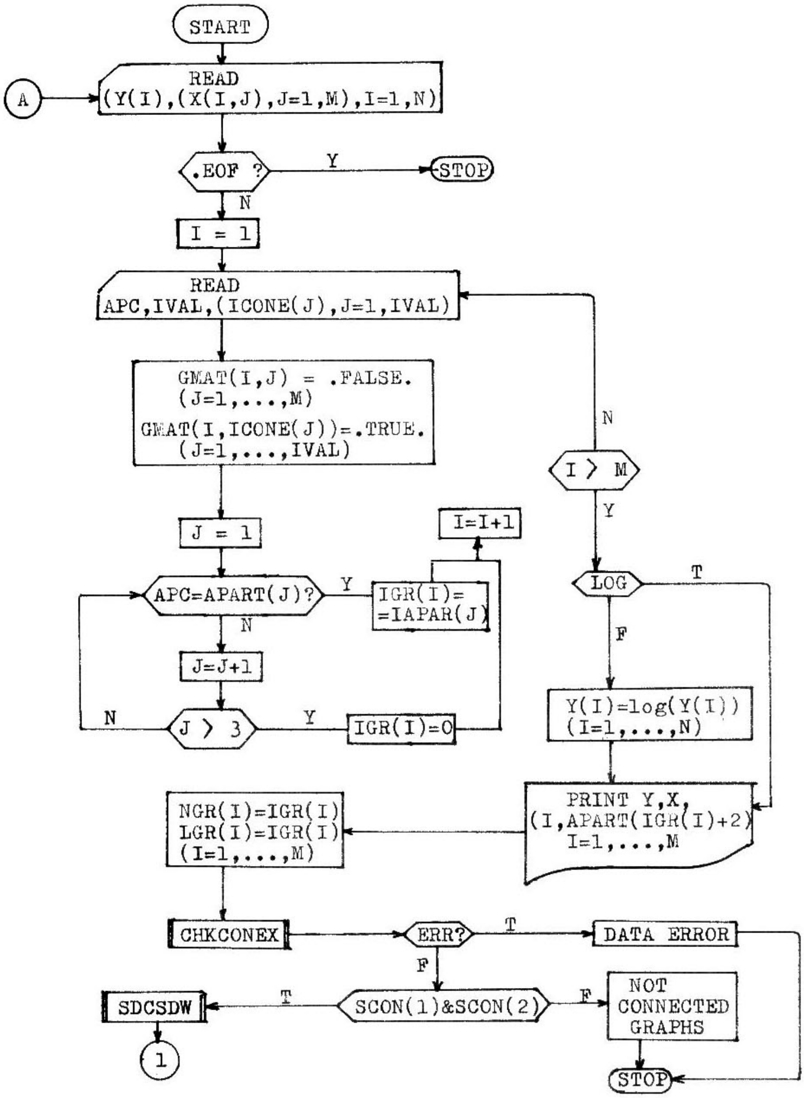
The main difference between SIBIS and HIBIS consists in the objective functions to be optimized, namely:

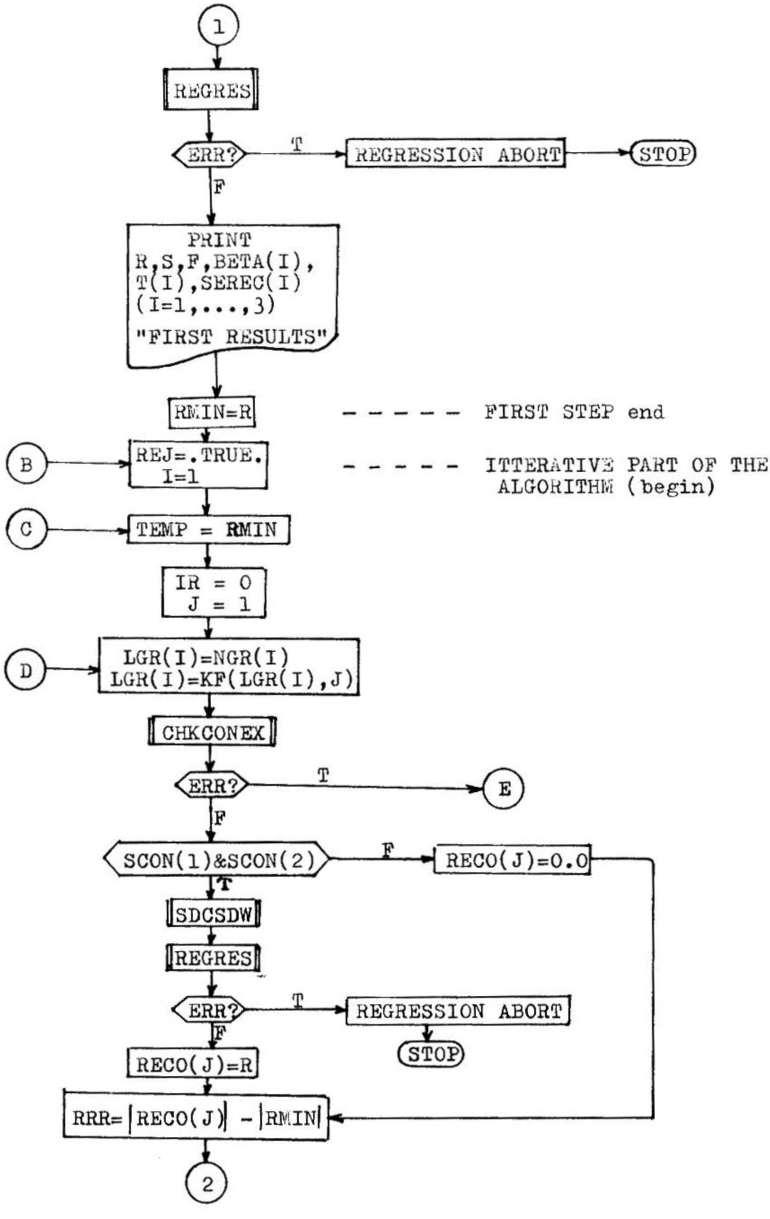
$$BR = a + b\, SD^{*}_{c} + c\, SD^{*}_{w} \qquad \text{(SIBIS)}$$

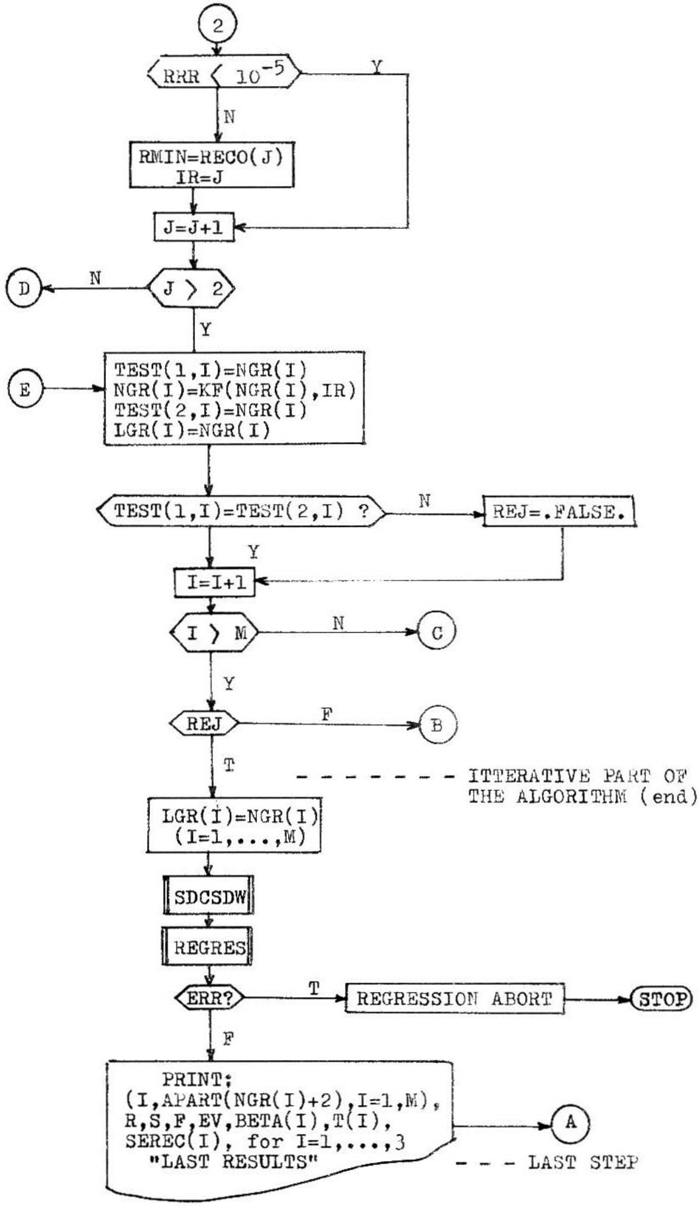
$$BR = m + n\, f + p\, f^{2} \qquad \text{(HIBIS)}$$

(BR stands for the Biological Response). The mathematical tool is the same (i.e., the least squares technique with the subsidiary condition to preserve connected some subgraphs of the ⟨IRS⟩ (Investigated Receptor Space) graph). Accordingly, the SIBIS and HIBIS programs have the same flowchart given below. The complete listing of the programs is given in Appendix 1. Appendix 2 contains the algorithm CONEX, which teaches the computer to recognize connected graphs.

SIBIS and HIBIS - the flowchart




1
REGRES
ERR?
T
REGRESSION ABORT
STOP
F
PRINT
R,S,F,BETA(I),
T(I),SEREC(I)
(I=1,...,3)
"FIRST RESULTS"
RMIN=R
- - - - - FIRST STEP end
B
REJ=.TRUE.
I=1
- - - - - ITTERATIVE PART OF THE ALGORITHM (begin)
C
TEMP = RMIN
IR = 0
J = 1
D
LGR(I)=NGR(I)
LGR(I)=KF(LGR(I),J)
CHKCONEX
ERR?
T
E
F
SCON(1)&SCON(2)
F
RECO(J)=0.0
T
SDCSDW
REGRES
ERR?
T
REGRESSION ABORT
STOP
F
RECO(J)=R
RRR= |RECO(J)| - |RMIN|
2



2
RRR < 10^-5
Y
N
RMIN=RECO(J)
IR=J
J=J+1
D
N
J > 2
Y
E
TEST(1,I)=NGR(I)
NGR(I)=KF(NGR(I),IR)
TEST(2,I)=NGR(I)
LGR(I)=NGR(I)
TEST(1,I)=TEST(2,I) ?
N
REJ=.FALSE.
Y
I=I+1
I > M
N
C
Y
REJ
F
B
T
- - - - - - - - ITTERATIVE PART OF THE ALGORITHM (end)
LGR(I)=NGR(I)
(I=1,...,M)
SDCSDW
REGRES
ERR?
T
REGRESSION ABORT
STOP
F
PRINT;
(I,APART(NGR(I)+2),I=1,M),
R,S,F,EV,BETA(I),T(I),
SEREC(I), for I=1,...,3
"LAST RESULTS"
A
- - - LAST STEP

The structure of the input data is detailed below.

The SIBIS and HIBIS programs together with the POLYREG program (which performs multivariate regressional analysis) are parts of a unique program called "ABSIN" (Analysis of Biological Structures INteractions).

Input data for this program are composed from an unlimited number of specific input data for each of its parts. One can select the subprogram by a Keyword, namely: SIBIS-LV (SIBIS - Linear Version), SIBIS-NL (SIBIS - Non Linear version), HIBIS, and POLYREG.

A commented example of input data is given at the end of Appendix I .

2. Applications

The I_{50} values for the inhibition of L.casei dihydropholate reductase by triazines of type I are taken from ref. 5 (see Table 1).

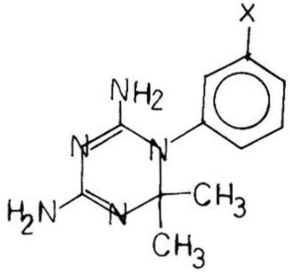


Fig.1.

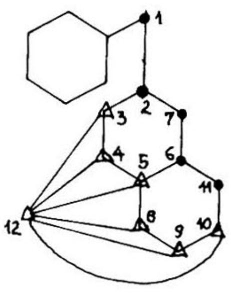


Fig.2.

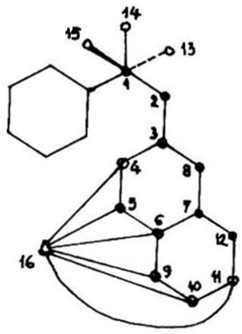


Fig.3.

The starting ⟨IRS⟩ for SIBIS calculations is

$\langle IRS \rangle_S^{init} = [c(1-3,7,8,11,12);w(4-6,9,10,13-15);i(16)]$

(see Figure 1; c-type subgraph corresponds to the structure of compound 20 from Table 1). - Explanations for the significance of c, w, r and i are given in refs. 1 and 4 -.

The starting $\langle IRS \rangle$ for HIBIS calculations is

$\langle IRS \rangle_H^{init} = [r(1,2,6,7,11);i(3-5,8,9)]$

(see Figure 2). The two optimization algorithms offer the following results:

$\langle IRS \rangle_S^{opt} = [c(1-5,7,8,11,12);w(6,9,10,13-15);i(16)]$

$$\log 1/I_{50} = \underset{(t=30.383)}{2.964(\pm 0.098)} + \underset{(t=8.787)}{0.261(\pm 0.030)}SD_c^{*} - \underset{(t=4.674)}{0.382(\pm 0.082)}SD_w^{*} \quad (1)$$

($r = 0.789$, $s = 0.413$, $F = 8.801$), and, respectively,

$\langle IRS \rangle_H^{opt} = [r(1-7,10,11);i(8,9,12)]$

$$\log 1/I_{50} = \underset{(t=34.227)}{3.017(\pm 0.088)} + \underset{(t=6.137)}{0.368(\pm 0.060)}f + \underset{(t=1.340)}{0.031(\pm 0.023)}f^2 \quad (2)$$

($r = 0.832$, $s = 0.373$, $F = 11.987$)

One may combine the equations (1) and (2) into one equation, resulting the equation

$$\log 1/I_{50} = \underset{(t=37.393)}{2.963(\pm 0.079)} + \underset{(t=5.926)}{0.143(\pm 0.024)}SD_c^{*} - \underset{(t=-3.445)}{0.228(\pm 0.066)}SD_w^{*} + \underset{(t=4.883)}{0.276(\pm 0.056)}f \quad (3)$$

($r = 0.875$, $s = 0.325$, $F = 12.266$)

The f^2 term has been rejected because it is not statistically significant (i.e. $t_{19,0.05} = 2.093 > 1.340$).

Table 1. Inhibition constants fot the inhibition of L.casei DHFR by Triazines of type (I)

No.	X	log $1/I_{50}$	SD_c^*	SD_w^*	$\sum f_i$
1	SO_2NH_2	1.82	2.2	2.0	-1.480
2	$CONH_2$	2.47	2.0	1.0	-1.109
3	H	2.64	0.0	0.0	0.167
4	$COCH_3$	2.87	2.0	1.0	-0.115
5	CH_3	3.07	1.0	0.0	0.691
6	OCH_3	3.12	2.0	0.0	0.262
7	OH	3.19	1.0	0.0	-0.309
8	$C(CH_3)_3$	3.20	2.0	2.0	2.150
9	$COOC_2H_5$	3.21	4.0	1.0	0.975
10	SO_2F	3.21	2.0	2.0	-1.488
11	F	3.29	0.8	0.0	0.382
12	CF_3	3.29	1.8	1.6	1.323
13	Cl	3.45	1.2	0.0	0.925
14	NO_2	3.56	2.0	1.0	-0.066
15	Br	3.69	1.3	0.0	1.121
16	I	3.73	1.7	0.0	1.430
17	$O(CH_2)_2OC_6H_5$	3.74	7.0	3.0	1.574
18	$OCH_2C_6H_5$	4.20	7.0	1.0	2.163
19	$O(CH_2)_3CH_3$	4.20	5.0	0.0	1.846
20	$(CH_2)_5CH_3$	4.96	6.0	0.0	3.295

References

1. I.Motoc and O.Dragomir-Filimonescu, Eur.J.Med.Chem., in press
2. I.Motoc and O.Dragomir-Filimonescu, Math.Chem., This issue
3. I.Motoc, Can.J.Pharm.Sci., in press
4. I.Motoc and O.Dragomir-Filimonescu, Math.Chem., This issue
5. R.J.A.Walsh, K.R.H.Wooldridgem, D.Jackson and J.Gilmour, Eur.J.Med.Chem., 12, 495 (1977).

APPENDIX 1

```
C
C
C     ************************************************************
C     *                                                          *
C     *              SSSS   II  BBBBB   II   SSSS                *
C     *             SS   S  II  BB   B  II  SS   S               *
C     *             SS      II  BB   B  II  SS                   *
C     *              SS     II  BBBBB   II   SS                  *
C     *                SS   II  BB   B  II     SS                *
C     *             S   SS  II  BB   B  II  S   SS               *
C     *              SSSS   II  BBBBB   II   SSSS                *
C     *                                                          *
C     ************************************************************
C
C
C
C                            * AUTHORS *
C                            ***********
C  * OCTAVIAN  DRAGOMIR-FILIMONESCU *          * IOAN MOTOC *
C  **********************************          **************
C
C    COMPUTER RESEARCH CENTER                 CHEMISTRY RESEARCH CENTER
C    STR. GH. LAZAR NO. 9,                    BV. M. VITEAZU NO. 24,
C    R - 1900  TIMISOARA,                     R - 1900  TIMISOARA,
C
C    R O M A N I A                            R O M A N I A
C
C
C
* SEGMENT GRAMAT,VALORI,REGPAR
      COMMON /GRAMAT/ WMAT(50,50),GMAT(50,50)
      COMMON /VALURI/ X(100,50),Y(100),IGR(50),NGR(50),LGR(50)
     1,SCUN(3),SD(100,3),TEST(2,50),N,M
      COMMON /REGPAR/ BETA(21),T(21),SEREC(21),R,S,F,EXPVAR
     1,SERY,SERES
     2,VEPS
C
      DIMENSION  ICONE(50)
      DIMENSION     RECO(2)
      DIMENSION TITL(20),FRD(20),FWR(20)
      DIMENSION MARK(50)
      DIMENSION APART(3)
      DIMENSION IAPAR(3)
      DIMENSION EG(3)
      INTEGER  CHOICE(2),  STDCH(2,4)
      DOUBLE PRECISION BETA,VEPS
      LOGICAL*1   AVY, AVX, AVIGR
      LOGICAL*1 MARK,MRK
      LOGICAL*1 GMAT,WMAT,REJ,SCON,NPRINT,LOG
      LOGICAL*1 ERR
      LOGICAL*1  EG
      INTEGER  TEST,DOI
      INTEGER CR,LP
      DATA CR /105/
      DATA LP /108/
      DATA NMAX, MMAX, NPRINT, DOI / 100, 50, .TRUE., 2 /
      DATA APART, IAPAR / 'C','I','W',-1,0,1 /
      DATA  REC, CAV /'R', 'C' /
      DATA STDCH/'HIBI','S   ','SIBI','S-LV','SIBI','S-NL','POLY',
     1'REG '/
C
      KF(IND1,IND2) = MOD ((IND1 + 1 + IND2), 3) -1
C
      VEPS = 0.1D-71
```

```
      LMX = 60
      IPAG = 1
100   CONTINUE
      READ(CR,200,END=990) CHOICE, AVY, AVX, AVIGR
      DO 110 I = 1, 4
         DO 109 J = 1, 2
            IF(CHOICE(J) .NE. STDCH(J,I)) GOTO 110
109      CONTINUE
         INDRUN = I
         GOTO 111
110   CONTINUE
      WRITE(LP,113)CHOICE
      STOP ' WRONG CHOICE PARAMETER'
111   CONTINUE
112   FORMAT('1',10X,12('=')/11X,'I ',2A4,' I'/11X,12('-'),10X,
     1'PAG.',I5//11X,10A4/11X,10A4//)
113   FORMAT(//11X,2A4/11X,7('-')/)
      IF (INDRUN .NE. 4)  GOTO  114
      CALL  POLYREG
      GOTO  100
114   CONTINUE
      IF (INDRUN .GT. 2)  GOTO  115
      IRUN = 2
      IRN = 3
      GOTO  116
115   IRUN = 3
      IRN = 4
116   CONTINUE
      IF (INDRUN .EQ. 1)  GOTO  117
      APART(1) = CAV
      GOTO  118
117   APART(1) = REC
118   CONTINUE
      ITER = 0
      IF(.NOT.(AVY .OR. AVX .OR. AVIGR)) GOTO 610
      READ(CR,2,END=1000) TITL
      DO 17 I = 1, 3
         MARK(I) = .FALSE.
17    CONTINUE
      MRK = .TRUE.
      READ(CR,1,END=1000) N,M,LOG
1     FORMAT (I3,I2,L1)
      IF(N.GT.NMAX)GOTO 2000
      IF(M.GT.MMAX)GOTO 2000
      WRITE(LP,112) CHOICE, IPAG, TITL
      IPAG = IPAG + 1
      READ(CR,2)       FWR
      PRINT            20
2     FORMAT(20A4)
200   FORMAT(2A4,3L1)
20    FORMAT(11X,'** VECTOR <Y> AND MATRIX <X>'//)
21    FORMAT (I2,A1,(38I2))
25    FORMAT(11X,
     1'VERTEX',I3,' <',A1,'> CONNECTION:',(T38,10(I3,';')))
      IF (.NOT. AVY) GOTO 253
      READ(CR,2)       FRD
      READ (CR,FRD) (Y(I),I=1,N)
C
      IF ( LOG )  GOTO 61
      DO 6 I = 1, N
6     Y(I) = ALOG10(Y(I))
61    CONTINUE
253   IF (.NOT. AVX) GOTO 305
      READ(CR,2) FRD
```

```
3       READ (CR,FRD,END=305) I, (X(I,J), J = 1, M)
        IF (I .GT. N)  GOTO 2012
        GOTO  3
305     CONTINUE
        DO 30 I = 1, N
           WRITE (LP,FWR) Y(I),(X(I,J),J=1,M)
30      CONTINUE
        IF (.NOT. AVIGR) GOTO 610
        WRITE(LP,112) CHOICE, IPAG, TITL
        IPAG = IPAG + 1
        PRINT 31
31      FORMAT(///11X,' ** CONNECTIVITY MATRIX **'//)
32      READ(CR,21,END=5) I, APC, IVAL, (ICONE(J), J = 1, IVAL)
        PRINT 25,I,APC,(ICONE(J),J=1,IVAL)
        IF ( I .GT. M )  GOTO  2016
        DO 4 J = 1, M
4       GMAT(I,J) = .FALSE.
        DO 41 J = 1, IVAL
        GMAT(I, ICONE(J)) = .TRUE.
41      CONTINUE
        DO 42 J = 1, 3
           IF(APC .NE. APART(J)) GOTO 42
           IGR(I) = IAPAR(J)
           MARK(I) = .TRUE.
        GOTO  32
42      CONTINUE
        IGR(I) = 0
        GOTO  32
5       CONTINUE
        DO 51 I = 1, M
           IF ( MARK(I) ) GOTO 51
           MRK = .FALSE.
51      CONTINUE
        IF ( MRK ) GOTO 52
        WRITE(LP,510)
510     FORMAT(//11X,'WRONGLY PARAMETRIZED IGR')
        GOTO 100
52      CONTINUE
C
610     CONTINUE
C
        WRITE(LP,112) CHOICE, IPAG, TITL
        IPAG = IPAG + 1
        PRINT 77,(I,APART(IGR(I)+2),I=1,M)
C
        DO 7 I = 1, M
        NGR(I) = IGR(I)
7       LGR(I) = IGR(I)
71      FORMAT(/11X,'RESULTS OF THE FIRST REGRESSION:'/)
72      FORMAT(/11X,'R =',F7.5,';  S =',F7.5,
       1';  F =',F9.3,';  EV =',F7.4/11X,51('-')//
       212X,'I    BETA(I)         T(I)          ERR.(I)'/
       312X,'-    -------         ----          -------'/)
720     FORMAT(11X,I2,3F11.3)
73      FORMAT(/11X,'RESULTS OF THE LAST  REGRESSION:'/)
75      FORMAT(//11X,'RESULTS FOR ITTERATION NO.',I5//)
C
        CALL CHKCONEX (ERR, EG)
C
        IF(ERR.AND.(INDRUN.NE.1)) GOTO 76
        IF((INDRUN.EQ.1).AND..NOT.EG(3)) GOTO  2100
        IF((INDRUN.EQ.1).AND.(EG(1).OR.EG(2))) GOTO 76
C
        IF ( .NOT. (SCON(1) .AND. SCON(2)) ) GOTO 2010
```

```
C
      CALL SOCSOW(INDRUN)
C
      CALL REGRES(Y, SD, N,  IRUN,  NPRINT, ERR)
      IF ( ERR ) GOTO    1050
C
      PRINT 71
      PRINT 72,R,S,F,EXPVAR
      DO 750 I = 1, IRN
         IZ = I - 1
         PRINT 720, IZ, BETA(I), T(I), SEREC(I)
750   CONTINUE
C
      RMIN = R
C
      GOTO 8
76    WRITE(LP, 760)
760   FORMAT(/11X,'*DATA ERROR* - CHECK THE DATA SET')
      GOTO 100
77    FORMAT(//11X,'INITIAL PARTITION OF THE GRAPH IS:'//
     C(11X,5(I6,' => ',A1,2X)))
78    FORMAT(//11X,'FINAL PARTITION OF THE GRAPH IS:'//
     C(11X,5(I6,' => ',A1,2X)))
C
8     REJ = .TRUE.
      ITER = ITER+1
C
      DO 15 I = 1, M
C
      TEMP = RMIN
      IR = 0
C
      DO  13  J = 1, 2
C
      LGR(I) = NGR(I)
      LGR(I) = KF( LGR(I), J )
C
      IF(INDRUN .GT. 1) GOTO 800
      IF(LGR(I) .EQ. 1) GOTO 13
800   CONTINUE
C
      CALL CHKCONEX (ERR, EG)
C
      IF(ERR.AND.(INDRUN .NE.1)) GOTO 131
      IF((INDRUN.EQ.1).AND..NOT.EG(3)) GOTO 2100
      IF((INDRUN.EQ.1).AND.(EG(1).OR.EG(2))) GOTO 11
      IF ( .NOT. (SCON(1) .AND. SCON(2)) ) GOTO 11
      CALL SOCSOW(INDRUN)
      CALL REGRES(Y, SD, N,  IRUN,  NPRINT, ERR)
      IF ( ERR ) GOTO    1050
      RECO(J) = R
      GOTO 12
11    RECO(J) = 0.0
12    RRR = ABS(RECO(J)) - ABS(RMIN)
      IF ( RRR .LT. 0.00001 )  GOTO 13
      RMIN = RECO(J)
      IR = J
C
13    CONTINUE
C
131   CONTINUE
C
      TEST(1,I) = NGR(I)
      NGR(I) = KF( NGR(I), IR )
```

```
      TEST(2,I) = NGR(I)
      LGR(I) = NGR(I)
C
15    CONTINUE
C
      DO 16 I = 1, M
      IF (TEST(1,I) .NE. TEST(2,I) ) REJ = .FALSE.
16    CONTINUE
C
      IF ( .NOT. REJ ) GOTO 8
C
      DO 9 J = 1, M
9     LGR(J) = NGR(J)
C
      CALL SDCSDW(INDRUN)
      CALL REGRES(Y, SD, N,  IRUN,  NPRINT, ERR)
      IF ( ERR ) GOTO    1050
C
      PRINT 79,ITER
79    FORMAT(//,11X,I4,' ITTERATIONS PERFORMED'//)
C
      PRINT 78,(I,APART(NGR(I)+2),I=1,M)
      PRINT  73
      PRINT 72,R,S,F,EXPVAR
      DO 980 I = 1, IRN
         IZ = I - 1
         PRINT 720, IZ, BETA(I), T(I), SEREC(I)
980   CONTINUE
C
      GOTO 100
990   CONTINUE
1000  STOP '  OUTPUT OK'
1050  WRITE(LP,     1051)
      GOTO 100
1051  FORMAT(/11X,    'ABANDONED RUN'//)
2000  WRITE(LP,     2001)
      GOTO 100
2001  FORMAT(/11X,    'WRONG <N> OR <M>'//)
2010  WRITE(LP,     2011)
      GOTO 100
2011  FORMAT(/11X,    'NOT CONNECTED GRAPHS'//)
2016  WRITE(LP,2018)
2013  READ(CR,2,END = 100) TITL
      GOTO  2013
2014  FORMAT(/11X,'WRONG INDEX,IN --- <X>'/)
2012  WRITE(LP,2014)
2017  READ(CR,2,END = 100) TITL
      GOTO  2017
2018  FORMAT(/11X,'WRONG INDEX,IN --- <IRS>'/)
C
2100  WRITE(LP, 2101)
2101  FORMAT(/11X,'WRONG <IRS>')
      GOTO 100
      END
* SEGMENT REGPAR
C
C     ******************************************************************
C     *                                                                *
C     *     PPPPP    OOOO    LL     YY    Y   RRRRR   EEEEEE   GGGG    *
C     *     PP   P  OO   O   LL      YY  Y    RR   R  EE      GG   G   *
C     *     PP   P  OO   O   LL       YYY     RR   R  EE      GG       *
C     *     PPPPP   OO   O   LL        YY     RRRRR   EEEE    GG GGG   *
C     *     PP      OO   O   LL        YY     RR R    EE      GG   G   *
C     *     PP       OOOO    LLLLLL    YY     RR  RR  EEEEEE   GGGG    *
```

```
C      *                                                                  *
C      *******************************************************************
C
       SUBROUTINE POLYREG
       COMMON /REGPAR/ BETA(21),T(21),SEREC(21)
      1,R,S,F,EXPVAR,SERY,SERES
      2,VEPS
       INTEGER IY,IX(20),KY,KX(20),CR
       INTEGER*4 NAMX(2,21),NYNA(2)
       REAL X(100,21),YMAN(100),XMAN(100,20),FORM(20),TITLE(40)
       DOUBLE PRECISION BETA,VEPS
       LOGICAL*1 NPR,ERR,PLSZ
       LOGICAL*1  CHKN
       DATA NPR / .FALSE. /
       DATA NYNA /4HDEP.,4HVAR./
       DATA CR,LP / 105, 108 /
       DATA LINMAX / 55 /
       IPAG = 1
1      READ(CR,2,END=100)TITLE
2      FORMAT(20A4)
3      FORMAT(/21X,'INPUT DATA'/21X,10('*')/)
4      FORMAT('1'///21X,11('=')/21X,'I POLYREG I'
      1/21X,11('='),5X,'PAG.',I5//(15X,20A4))
5      READ(CR,6,END=1)N,K,PLSZ,VEPS,VDLZ
C/     PRINT 300,N,K,PLSZ,VEPS,VDLZ
C/300  FORMAT(//10X,I3,I2,L1,2G12.6//)
6      FORMAT(I3,I2,L1,2G12.6)
       IF(N .GT. 100)GOTO 200
       IF(K .GT. 20) GOTO 200
       KP1 = K + 1
       READ(CR,7,END=205)((NAMX(I,J),I=1,2),J=1,KP1)
       READ(CR,7)FORM
7      FORMAT(20A4)
       DO 9 I = 1, N
          READ(CR,FORM,END=205)(X(I,J),J=1,KP1)
          IF( .NOT. PLSZ ) GOTO 9
          DO 8 J = 1, KP1
             IF(X(I,J) .NE. 0.0) GOTO 8
             X(I,J) = VDLZ
8         CONTINUE
9      CONTINUE
       NLIN = 9
       WRITE(LP,4)IPAG,TITLE
       IPAG = IPAG + 1
       N1 = N - 1
       DO 11 I = 1, N1
          I1 = I + 1
          DO 101 J = I1, N
             DO 10 M = 1, KP1
                IF(X(I,M) .NE. X(J,M)) GOTO 101
10           CONTINUE
             WRITE(LP,12)
             NLIN = NLIN + 4
             GOTO 13
101       CONTINUE
11     CONTINUE
12     FORMAT(' '/11X,'WARNING ] --> IDENTICAL POINTS'//)
13     WRITE(LP,3)
       NLIN = NLIN + 4
14     FORMAT(L1,I2,21(I2,I1))
       LMIN = 1
       LMAX = 5
15     IF(LMAX .GT. N) LMAX = N
       IF(LMIN .GT. N) GOTO  22
```

```
      IF ( NLIN .LE. LINMAX - KP1 - 2 ) GOTO 152
      NLIN = 9
      WRITE(LP,4)IPAG,TITLE
      IPAG = IPAG + 1
152   CONTINUE
      WRITE(LP,18)(I,I = LMIN, LMAX)
18    FORMAT(/9X,'NAME\POINT',5(I3,8X))
      WRITE(LP,91)
91    FORMAT(' ')
19    FORMAT(' ',T10,2A4,2X,5G11.4)
      NLIN = NLIN + 3
      DO 21 J = 1, KP1
         WRITE(LP,19)(NAMX(I,J),I=1,2)
     1,                (X(I,J),I=LMIN,LMAX)
         NLIN = NLIN + 1
21    CONTINUE
      LMIN = LMIN + 5
      LMAX = LMAX + 5
      GOTO 15
22    WRITE(LP,23)
23    FORMAT(' '/11X,'END OF INPUT DATA'/11X,17('-'))
      NRUN = 1
24    READ(CR,14,END=5) CHKN,LIM,KY,IY,(KX(I),IX(I),I=1,LIM)
      IF(KY .GT. KP1)GOTO 202
      IF(LIM .EQ. 0) GOTO 202
      IF(KY .EQ. 0)  GOTO 202
      DO 27 I = 1, LIM
         IF(KX(I) .GT. KP1) GOTO 202
         IF(KX(I) .EQ. 0)   GOTO 202
27    CONTINUE
      WRITE(LP,4)IPAG,TITLE
      IPAG = IPAG + 1
      WRITE(LP,26)NRUN
26    FORMAT(' '/11X,'RUN NO.',I4/11X,11('-')//)
      NRE = N
      IN = IY + 1
      WRITE(LP,40)
      WRITE(LP,25)(NAMX(I,KY),I=1,2),IY
25    FORMAT(11X,' Y      - ',    2A4,' - CONTROL INDEX ->',I3)
40    FORMAT(/11X,'VARIABLES USED IN THIS RUN:'/)
41    FORMAT(11X,'X(',I2,') - ',2A4,' - CONTROL INDEX ->',I3)
      J1 = KY
      GOTO(111,113,115,117,119,121,123,125,127,129), IN
111   DO 112 I = 1, N
      IF ( .NOT. CHKN ) GOTO 1110
      IF(X(I,J1).GT.0.0)GOTO 1110
      IF(NRE .GE. I)  NRE = I - 1
      GOTO        39
1110  CONTINUE
         YMAN(I)   =   ALOG10( X(I,J1) )
112   CONTINUE
      GOTO 39
113   DO 114 I = 1, N
         YMAN(I)   =   X(I,J1)
114   CONTINUE
      GOTO 39
115   DO 116 I = 1, N
         YMAN(I)   =   X(I,J1) ** 2
116   CONTINUE
      GOTO 39
117   DO 118 I = 1, N
         YMAN(I)   =   X(I,J1) ** 3
118   CONTINUE
      GOTO 39
```

```
119     DO 120 I = 1, N
           YMAN(I)  =  X(I,J1) ** 4
120     CONTINUE
        GOTO 39
121     DO 122 I = 1, N
        IF ( .NOT. CHKN ) GOTO 1210
        IF(X(I,J1).NE.0.0)GOTO 1210
        IF(NRE .GE. I)  NRE = I - 1
        GOTO          39
1210    CONTINUE
           YMAN(I)  =  1.0 / X(I,J1)
122     CONTINUE
        GOTO 39
123     DO 124 I = 1, N
        IF ( .NOT. CHKN ) GOTO 1230
        IF(X(I,J1).NE.0.0)GOTO 1230
        IF(NRE .GE. I)  NRE = I - 1
        GOTO          39
1230    CONTINUE
           YMAN(I)  =  1.0 / X(I,J1) ** 2
124     CONTINUE
        GOTO 39
125     DO 126 I = 1, N
        IF ( .NOT. CHKN ) GOTO 1250
        IF(X(I,J1).GT.0.0)GOTO 1250
        IF(NRE .GE. I)  NRE = I - 1
        GOTO          39
1250    CONTINUE
           YMAN(I)  =  - ALOG10( X(I,J1) )
126     CONTINUE
        GOTO 39
127     DO 128 I = 1, N
           YMAN(I)  =  EXP( X(I,J1) )
128     CONTINUE
        GOTO 39
129     DO 130 I = 1, N
           YMAN(I)  =  EXP( -X(I,J1) )
130     CONTINUE
39      CONTINUE
        DO 55 J = 1, LIM
           J1 = KX(J)
           IN = IX(J) + 1
           GOTO(131,133,135,137,139,141,143,145,147,149), IN
131        DO              132 I = 1, N
        IF ( .NOT. CHKN ) GOTO 1310
        IF(X(I,J1).GT.0.0)GOTO 1310
        IF(NRE .GE. I)  NRE = I - 1
        GOTO          151
1310    CONTINUE
              XMAN(I,J)=ALOG10(X(I,J1))
132        CONTINUE
151        WRITE(LP,41)J,(NAMX(I,J1),I=1,2),IX(J)
           GOTO 55
133        DO              134 I = 1, N
              XMAN(I,J)=X(I,J1)
134        CONTINUE
           GOTO 151
135        DO              136 I = 1, N
              XMAN(I,J)=X(I,J1)**2
136        CONTINUE
           GOTO 151
137        DO              138 I = 1, N
              XMAN(I,J)=X(I,J1)**3
138        CONTINUE
```

```
          GOTO 151
139       DO              140 I = 1, N
             XMAN(I,J)=X(I,J1)**4
140       CONTINUE
          GOTO 151
141       DO              142 I = 1, N
      IF ( .NOT. CHKN ) GOTO 1410
      IF(X(I,J1).NE.0.0)GOTO 1410
      IF(NRE .GE. I)  NRE = I - 1
      GOTO          151
1410  CONTINUE
             XMAN(I,J)=1.0/X(I,J1)
142       CONTINUE
          GOTO 151
143       DO              144 I = 1, N
      IF ( .NOT. CHKN ) GOTO 1430
      IF(X(I,J1).NE.0.0)GOTO 1430
      IF(NRE .GE. I)  NRE = I - 1
      GOTO          151
1430  CONTINUE
             XMAN(I,J)=1.0/X(I,J1)**2
144       CONTINUE
          GOTO 151
145       DO              146 I = 1, N
      IF ( .NOT. CHKN ) GOTO 1450
      IF(X(I,J1).GT.0.0)GOTO 1450
      IF(NRE .GE. I)  NRE = I - 1
      GOTO          151
1450  CONTINUE
             XMAN(I,J)=-ALOG10(X(I,J1))
146       CONTINUE
          GOTO 151
147       DO              148 I = 1, N
             XMAN(I,J)=EXP(X(I,J1))
148       CONTINUE
          GOTO 151
149       DO              150 I = 1, N
             XMAN(I,J)=EXP(-X(I,J1))
150       CONTINUE
          GOTO 151
55    CONTINUE
      IF (NRE .LE. 0) GOTO 207
      WRITE(LP,160) NRE
160   FORMAT(//11X,'THIS RUN WILL CONSIDER',I4,'  VALID POINTS')
      CALL REGRES(YMAN,XMAN,NRE,LIM,NPR,ERR)
      IF( ERR ) GOTO 98
      WRITE(LP,87)NRUN
87    FORMAT(//11X,'END OF RUN NO.',I5/11X,19('='))
88    NRUN = NRUN + 1
      GOTO 24
98    WRITE(LP,90)NRUN
90    FORMAT(/11X,'*** REGRESSION CANNOT BE PERFORMED'
     6/15X,'WITH THESE DATA !)'//11X'*** RUN NO.',I4,
     2' IS SKIPED')
      GOTO 88
100   RETURN
200   WRITE(LP,203)
201   READ(CR,7,END=5)FORM
      GOTO 201
207   WRITE(LP,208)
202   WRITE(LP,204)LIM,KY,IY,(KX(I),IX(I),I=1,LIM)
      GOTO 88
203   FORMAT(//'    *** WRONG <N> OR <K>')
204   FORMAT(//'    *** WRONG RUN PARAM.'//4X,I3,(T7,7(I5,I2)))
```

```
205     WRITE(LP,206)
206     FORMAT(//'    *** WRONG DATA'//)
        GOTO 201
208     FORMAT(//'    *** WRONG ASSOCIATION OF RUN PARAMETERS')
        END
* SEGMENT GRAMAT,VALORI
        SUBROUTINE CHKCONEX (ERR, GV)
C
        COMMON /GRAMAT/ WMAT(50,50),GMAT(50,50)
        COMMON /VALORI/ X(100,50),Y(100),IGR(50),NGR(50),LGR(50)
       1,SCON(3),SD(100,3),TEST(2,50),N,M
        DIMENSION GV(3)
        LOGICAL*1 WMAT,GMAT,REJ,SCON,CONEX
        LOGICAL*1 ER,ERR
        LOGICAL*1  GV
C
        ERR = .FALSE.
        DO 100 I = 1, 3
C
        GV(I) = .FALSE.
        IG = I - 2
        NF = 0
        NN = 0
C
           DO 90 J = 1, M
C
           IF(LGR(J) .NE. IG) GOTO 70
C
              DO 60 K = 1, M
C
              IF(LGR(K) .NE. IG) GOTO 50
              WMAT(J,K) = GMAT(J,K)
              GOTO 60
50            WMAT(J,K) = .FALSE.
C
60            CONTINUE
C
           NN = NN + 1
           IF(NF .NE. 0) GOTO 90
           NF = J
           GOTO 90
C
70            DO 80 K = 1, M
C
80            WMAT(J,K) = .FALSE.
C
90         CONTINUE
C
        SCON(I) = CONEX ( NF, NN, ER )
C
C
        IF ( .NOT. ER )  GOTO  100
C
        ERR = .TRUE.
        GV(I) = .TRUE.
C
C
100     CONTINUE
C
        RETURN
        END
* SEGMENT GRAMAT
        FUNCTION  CONEX ( NF, NN, ERR )
        COMMON /GRAMAT/ CONNECT(50,50),GMAT(50,50)
```

```
      DATA NNM /50/
      LOGICAL*1 CONEX, CONNECT, MARK(50)
      LOGICAL*1 STAR(50), ARROW(50), KMNS, GMAT
      LOGICAL*1 ERR
C
      ERR = .FALSE.
      IF(NN .GT. 1) GOTO 10
      IF ( NN .EQ. 0 ) GOTO 9
      CONEX = .TRUE.
      RETURN
9     CONEX = .TRUE.
      ERR = .TRUE.
      RETURN
10    CONTINUE
C
      DO 1 I = 1, NNM
      MARK(I) = .FALSE.
      STAR(I) = .FALSE.
1     ARROW(I) = .FALSE.
      MARK(NF) = .TRUE.
      STAR(NF) = .TRUE.
      ARROW(NF) = .TRUE.
      DO 2 I = 1, NNM
      IF(.NOT.CONNECT(I, NF)) GOTO 2
      MARK(I) = .TRUE.
      STAR(I) = .TRUE.
2     CONTINUE
      CONEX = .FALSE.
3     CONTINUE
      DO 5 I = 1, NNM
      IF(.NOT.(MARK(I).AND.STAR(I).AND..NOT.ARROW(I))) GOTO 5
      ARROW(I) = .TRUE.
      DO 4 J = 1, NNM
      IF(.NOT.CONNECT(J, I)) GOTO 4
      MARK(J) = .TRUE.
4     CONTINUE
5     CONTINUE
      KMNS = .FALSE.
      DO 6 I = 1, NNM
      IF(.NOT.(MARK(I).AND..NOT.STAR(I))) GOTO 6
      KMNS = .TRUE.
      STAR(I) = .TRUE.
6     CONTINUE
      IF(KMNS) GOTO 3
      NCARD = 0
      DO 8 I = 1, NNM
      IF(MARK(I).AND.STAR(I).AND.ARROW(I)) GOTO 7
      IF(MARK(I).OR.STAR(I).OR.ARROW(I))STOP 'TEST IN ERROR'
      GOTO 8
7     NCARD = NCARD + 1
8     CONTINUE
      IF ( NCARD .EQ. NN ) CONEX = .TRUE.
      RETURN
      END
* SEGMENT VALORI
      SUBROUTINE SDCSDW(IR)
C
      COMMON /VALORI/ X(100,50),Y(100),IGR(50),NGR(50),LGR(50)
     1,SCUN(3),SD(100,3),TEST(2,50),N,M
      DIMENSION SDC(100),SDW(100)
      DIMENSION  SDWSQ(100)
      EQUIVALENCE (SD(1,1),SDC(1))
      EQUIVALENCE (SD(1,2),SDW(1))
      EQUIVALENCE (SD(1,3), SDWSQ(1))
```

```
C
      DO 20 I = 1, N
C
      SDC(I) = 0.0
      SDW(I) = 0.0
      SDWSW(I) = 0.0
C
         DO 10 J = 1, M
C
         IF(LGR(J) .GT. 0) SDW(I) = SDW(I) + X(I,J) * LGR(J)
         IF(LGR(J) .LT. 0) SDC(I) = SDC(I) - X(I,J) * LGR(J)
C
10       CONTINUE
C
20    CONTINUE
C
      DO 30 I = 1, N
         SDWSQ(I) = SDW(I) ** 2
         IF(IR.EQ.1)  SDW(I) = SDC(I) ** 2
30    CONTINUE
      RETURN
      END
* SEGMENT REGPAR
      SUBROUTINE REGRES(Y,X,N,L,NPR,VER)
      COMMON /REGPAR/ BETA(21),T(21),SEREC(21),R,S,F
     1,EXPVAR,SERY,SERES
     2,VEPS
      DIMENSION Y(100),X(100,L),Q(100),YR(100),XS(20)
     1,XSSW(20),XYS(20),XM(20),A(21,22)
      DIMENSION AA(21,22)
      DOUBLE PRECISION AA,BETA,VEPS
      LOGICAL*1 ER, NPR, VER
C
C
C     REGRESSION
C     **********
C
C
      VER = .FALSE.
      YS = XSIGMA(Y,N)
      YSSW = XYSIGMA(Y,Y,N)
      DO 20 J = 1, L
         DO 10 I = 1, N
            Q(I) = X(I,J)
10       CONTINUE
         XS(J)   =   XSIGMA(Q,N)
         XSSQ(J)= XYSIGMA(Q,Q,N)
         XYS(J) = XYSIGMA(Q,Y,N)
20    CONTINUE
      FLN = FLOAT(N)
      FLK = FLOAT(L)
      ND = L + 1
      NDPL = ND + 1
      DO 30 I = 1, ND
      DO 30 J = 1, ND
         A(I,J) = 0.0
30    CONTINUE
      A(1,NDPL) = YS
      A(1,1) = FLN
      DO 40 I = 2, ND
         J = I - 1
         A(I, NDPL) = XYS(J)
         A(1,I) = XS(J)
         A(I,1) = XS(J)
```

```
40      CONTINUE
        DO 52 I = 2, ND
           I1 = I - 1
           DO 51 J = 2, ND
              J1 = J - 1
              DO 50 IP = 1, N
                 A(I,J) = A(I,J) + X(IP,I1) * X(IP,J1)
50            CONTINUE
51         CONTINUE
52      CONTINUE
C
C
C       THE INTERCORRELATION MATRIX
C       ****************************
C
C
C
        IF ( NPR ) GOTO 200
        PRINT 60
60      FORMAT(' '/T10,'REGRESSION RESULTS:'//
       1/T10,'INTERCORRELATION MATRIX: => INDEX 1 IS (Y)'/)
        IF ( ND .LE. 10) GOTO 130
        PRINT 70,(I, I= 1, ND)
70      FORMAT(' '/9X,20I6/)
        DO 110 I = 1, ND
           IM = I - 1
           IF (I .NE. 1 ) GOTO 90
           Q(1) = 1.0
           DO 80 J = 2, ND
              JM = J - 1
              Q(J) = (FLN * XYS(JM) - YS * XS(JM)) /
       1                SQRT((FLN * YSSQ - YS * YS) *
       2                     (FLN * XSSQ(JM) - XS(JM) ** 2))
80         CONTINUE
        GOTO 101
90         Q(1) = (FLN * XYS(IM) - YS * XS(IM)) /
       1             SQRT((FLN * YSSQ - YS * YS ) *
       2                  (FLN * XSSQ(IM) - XS(IM) ** 2))
           DO 100 J = 2, ND
              JM = J - 1
              Q(J) = (FLN * A(I,J) - XS(IM) * XS(JM)) /
       1                SQRT((FLN * A(I,I) - XS(IM) ** 2) *
       2                     (FLN * A(J,J) - XS(JM) ** 2))
100        CONTINUE
101     CONTINUE
           PRINT 120, I,(Q(J), J = 1, ND)
110     CONTINUE
120     FORMAT(' '/4X,I2,5X,20F6.2)
        GO TO 200
130     PRINT 140,(I, I = 1, ND)
140     FORMAT(' '/5X,10I12/)
        DO 180 I = 1, ND
           IM = I - 1
           IF(I .NE. 1) GOTO 160
           Q(1) = 1.0
           DO 150 J = 2, ND
              JM = J - 1
              Q(J) = (FLN * XYS(JM) - YS * XS(JM)) /
       1                SQRT((FLN * YSSQ - YS * YS) *
       2             (FLN * XSSQ(JM) - XS(JM) ** 2))
150        CONTINUE
        GOTO 171
160        Q(1) = (FLN * XYS(IM) - YS * XS(IM)) /
       1             SQRT((FLN * YSSQ - YS * YS) *
       2                  (FLN * XSSQ(IM) - XS(IM) ** 2))
```

```
      DO 170 J = 2, ND
         JM = J - 1
         Q(J) = (FLN * A(I,J) - XS(IM) * XS(JM))/
     1               SQRT((FLN * A(I,I) - XS(IM) ** 2) *
     2               (FLN * A(J,J) - XS(JM) ** 2))
170   CONTINUE
171   CONTINUE
      PRINT 190, I, (Q(J), J = 1, ND)




180   CONTINUE
190   FORMAT('   '/4X,I2,6X,10(F8.4,4X))
C
C
C     REGRESSION COEFICIENTS
C
C     *********************
C
200   CONTINUE
      DO 202 I = 1, ND
         DO 201 J = 1, NDPL
            AA(I,J) = A(I,J)
201      CONTINUE
202   CONTINUE
      CALL CROUT (AA,BETA,ND,ER,VEPS)
      IF ( ER ) GOTO 320
      DO 210 I = 1, N
         YR(I) = BETA(1)
210   CONTINUE
      DO 230 J =1, L
         DO 220 I = 1, N
            YR(I) = YR(I) + BETA(J+1) * X(I,J)
220      CONTINUE
230   CONTINUE
      DO 240 I = 1, N
         Q(I) = Y(I) - YR(I)
240   CONTINUE
      YM = YS / FLN
      YMR = XSIGMA(YR,N) / FLN
      DO 250 I = 1, L
         XM(I) = XS(I) / FLN
250   CONTINUE
      SFN = XYSIGMA(Q,Q,N)
      R1 = 0.0
      R2 = 0.0
      R3 = 0.0
      DO 260 I = 1, N
         R1 = R1 + (Y(I) - YM) * (YR(I) - YMR)
         R2 = R2 + (Y(I) - YM) ** 2
         R3 = R3 + (YR(I) - YMR) ** 2
260   CONTINUE
C
C
C     GLOBAL PARAMETERS
C     *****************
C
C     OF THE REGRESSION
C     *****************
C
C
C
      R = R1 / SQRT(R2 * R3)
      S = SFN / (FLN - 1.0)
      S = SQRT(S)
      F = (FLN - FLK - 2.0) * R * R /
     1    (FLK + 1.0) / (1.0 - R * R)
      EXPVAR = 1.0 - (SFN / (FLN - FLK - 1.0))
```

```
     1            / (R2 / (FLN - 1.0))
      SERY = R2 / (FLN - 1.0)
      SERES = SFN / (FLN - FLK - 1.0)
      NDEGFR = N - ND
C
C
C     STATISTICS FOR THE REGRESSIONAL EQUATION
C     ****************************************
C
C
      SEREC(1) = SQRT(SERES / FLN)
      T(1) = BETA(1) / SEREC(1)
      DO 270 I = 2, ND
         IM = I -1
         SEREC(I) = SQRT(SERES / XSSQ(IM))
         T(I) = BETA(I) / SEREC(I)
270   CONTINUE
      SERY = SQRT(SERY)
      SERES = SQRT(SERES)
      IF (¬NPR ) RETURN
      PRINT 280, R,S,F,EXPVAR,SERY,SERES,L,NDEGFR
280   FORMAT(' '//T10,'REGRESSION GLOBAL PARAMETERS:'//
     1T10,'         R =',F15.5/
     2T10,'         S =',F15.5/
     3T10,'         F =',F15.5/
     4T10,'        EV =',F15.5/
     5T10,'      SERY =',F15.5/
     6T10,'     SERES =',F15.5/
     7T10,'DEG. FR. =',I4,'   AND   ',I4//)
      PRINT 290
290   FORMAT(T10,'REGRESSION COEFICIENTS:'//
     1T25,'      VALUE        STANDARD ERROR   STUDENT TEST'/)
      NRZ = 0
      PRINT 300, NRZ,BETA(1),SEREC(1),T(1)
300   FORMAT(T10,'BETA(',I2,')',7X,3(F11.4,4X))
      DO 310 I = 1, L
         IM = I + 1
         PRINT 300, I,BETA(IM),SEREC(IM),T(IM)
310   CONTINUE
      RETURN
320   PRINT 330
330   FORMAT(T10,'** WARNING ] -> DATA EROR'//
     1T10,'THE REGRESSION CANNOT BE PERFORMED **')
      VER = .TRUE.
      RETURN
      END
      SUBROUTINE CROUT (A,BETA,N,ER,V)
      DIMENSION A(21,22),BETA(21)
      DOUBLE PRECISION A,BETA,V
      DOUBLE PRECISION PHANTOM,COM,S
      LOGICAL*1 ER
C/    PRINT 100
C/100 FORMAT(//'        ***** CROUT ****' )
      ER = .FALSE.
      DO 1 I = 1, N
         IF ( A(I,I) ** 2 .LE. V ) GOTO 12
1     CONTINUE
      NMIN = N - 1
      NPL  = N + 1
C/    DO 101 I = 1, N
C/       PRINT 102, (A(I,J),J=1,NPL)
C/101 CONTINUE
C/102 FORMAT(10G12.5)
C/    PRINT 103,N,V
```

```
C/103 FORMAT('      N=',I4,'   V=',G10.4)
      IF(N .GT. 1) GOTO 3
      IF ( N .LT. 1 ) GOTO 12
      BETA(1) = A(1,2) / A(1,1)
      RETURN
3     DO 9 K = 1, NMIN
         KPL = K + 1
         L = K
         DO 4 I = KPL, N
            IF(A(I,K)**2 .GT. A(L,K) ** 2) L = I
4        CONTINUE
         IF(L .EQ. K) GOTO 6
         DO 5 J = K, NPL
            PHANTOM = A(K,J)
            A(K,J) = A(L,J)
            A(L,J) = PHANTOM
5        CONTINUE
6        DO 8 I = KPL, N
            IF ( A(K,K) ** 2 .LE. V ) GOTO 13
            COM = A(I,K) / A(K,K)
            DO 7 J = KPL, NPL
               A(I,J) = A(I,J) - COM * A(K,J)
7           CONTINUE
8        CONTINUE
9     CONTINUE
      IF ( A(N,N) ** 2 .LE. V ) GOTO  14
      BETA(N) = A(N,NPL) / A(N,N)
      I = NMIN
10    IPL = I + 1
      S = 0.0
      DO 11 J = IPL, N
         S = S + A(I,J) * BETA(J)
11    CONTINUE
      IF ( A(I,I) ** 2 .LE. V ) GOTO  15
      BETA(I) = (A(I,NPL) - S) / A(I,I)
      I = I - 1
      IF (I .GE. 1) GOTO 10
      RETURN
12    CONTINUE
13    CONTINUE
14    CONTINUE
15    CONTINUE
      ER = .TRUE.
      RETURN
      END
      FUNCTION XSIGMA(X,N)
      DIMENSION X(N)
      XSIGMA = 0.0
      DO 1 I = 1, N
         XSIGMA = XSIGMA + X(I)
1     CONTINUE
      RETURN
      END
      FUNCTION XYSIGMA(X,Y,N)
      DIMENSION X(N),Y(N)
      XYSIGMA = 0.0
      DO 1 I = 1, N
         XYSIGMA = XYSIGMA + X(I) * Y(I)
1     CONTINUE
      RETURN
      END
```

INPUT DATA FOR SIBIS

```
 N
       → SIBIS-Linear Version   AVY   = .TRUE. =  vector Y will be read
SIBIS-LVTTI                     AVX   = .TRUE. =   -//-  X // -
- 34 - SERIE → TITLE            AVIGR = .TRUE. =    IRS will be read
3424T  K  ← Y's are already logarithmed
(11X,F7.2,(T20,24F4.1)) ← output FORMAT for (Y(I),(X(I,J),J=1,K))
(16F4.2) ← input FORMAT for Y
 522 600 610 700 800 492 555 559 674 700 770 485 505 549 470 510
 560 600 650 730 748 749 755 768 352 540 566 585 589 600 650 660
 740 755                              "Y"
(I2,24F2.1) ← input Format for (I,(X(I,J),J=1,K))
 11 1 1 1 1 1 1                         1
 21 1 1 1 1 1 1 1 1 1 1     1           1
 31 1 1 1 1 1 1       1 1 1 1           1
 41 1 1 1 1 1 1 1 1 1 1                 1
 51 1 1 1 1 1 1 1 1 1 1                 1
 61 1       1 1 1 1 1 1       1 1 1 1 121
 71 1       1 1 1 1 1 1       1 1 1 1   1 1 1 1
 81 1       1 1 1 1 1 1       1 1 1 1   1
 91 12      1 1 1 1 1 1       1 1 1 1   1
101 1       1 1 1 1 1 1       1 1 1 1   1 1
111 1       1 1 1 1 1 1       1 1 1 1   1
121 1 1 1 1 1 1 1 1 1 1                 1            "X"
131 1 1 1 1 1 1 1 1 1 1                 1 1
141 1 1 1 1 1 1 1 1 1 1                 1
151 1     121 1 1 1 1 1       12    12121
161 1     121 1 1 1 1 1       12      121
171 1     121 1 1 1 1 1               121
181 1     171 1 1 1 1 1                 1
191 1     121 1 1 1 1 1 12              1
201 1     121 1 1 1 1 1             12  1
211 1     121 1 1 1 1 1       12        1
```

```
221 1     121 1 1 1 1 1       12    12  1
231 1     121 1 1 1 1 1             12  1 1
241 1     1 1 1 1 1 1 1       12        1
251 1       1 1 1 1 1 1                 1 1 1 1
261 1       1 1 1 1 1 1 12          12  1
271 1       1 1 1 1 1 1                 1
281 1       1 1 1 1 1 1                 1 1
291 1       1 1 1 1 1 1                 1
301 1       1 1 1 1 1 1        8        1
311 1       1 1 1 1 1 1       12        1
321 1       1 1 1 1 1 1       13        1
331 1       1 1 1 1 1 1 17              1
341 1       1 1 1 1 1 1 12    12        1
```

"X" (continued)

```
.EOF   <-- end of data (for X)
 1C 2 220   <-- vertex 1, type "c" connected with 2 vertices,
                                      namely 2 and 20
 2C 3 1 3 1
 3C 2 2 4
 4C 3 3 5 6
 5W 3 4 614
 6C 4 4 5 711   <-- vertex 6, type "c" connected with 4 vertices,
                                      namely 4,5,7 and 11
 7C 3 2 6 8
 8C 3 7 919
 9C 3 81018
10C 4 9111516
11C 3 61012
12W 3111314
13W 21214
14W 3 51215
15W 21016
16W 410151724
17W 3161824
18W 3 91724
```

⟨IRS⟩ - input FORMAT - (I2,A1,(38I2))

```
19W 3 1 824
20W 2 121
21W 22022
22W 22123
23W 22224
24I 516171O1923
.EOF
SIBIS-NL
```

⟨IRS⟩ - (continued)

.EOF ← end of data (for SIBIS-LV) (it was a complete set of input data)

SIBIS-NL: SIBIS - Non Linear version (AVY = AVX = AVIGR = .FALSE. The run will be performed with the same input data).

- For HIBIS, input data have the same structure.

INPUT DATA FOR POLYREG

POLYFUNCTIONAL REGRESSION ANALYSIS

```
POLYREG
       HI-AIB  }
 N     ******  }  <--- TITLE of the regression
10 5<-M
HI      L   .   B1     B2     B3     B4 }<-- variables
(6F5.3) <--- input FORMAT for (X(I,J),J=1,M+1)
  447  3    152  19   19   204      ^
 1023 411   152  19   19   297      |
 1523 505   152  19   19   349      |
 1154 411   204  276  316  316      |
 2023 617   152  19   19   442      |
 1951 505   152  19   316  421    "X" - each card is a row
 1879 505   19   276  316  349      |
 1724 411   259  286  286  297      |
 2253 711   152  19   19   494      |
 2379 617   152  19   316  442      v
T 1 11 21 <----------- This RUN parameters card means:
                              - DATA will be verified;
T 1 11 31    RUN PARAMETERS   - monoparametric regression;
T 1 11 41                     - "Y" is "HI"
                              - "X" is "B1"
T 1 11 51
T 1 11 61
.EOF <---- end of run parameters - new data are required under
                                          the same TITLE
.EOF <---- end of data under this TITLE
.EOF <---- end of data for POLYREG
```

after this ".EOF" card, one can ask a new computation specifying a keyword and a new data set, or finish the execution of the program with a new ".EOF" card!

APPENDIX 2

Teach the Computer to Recognize Connected Graphs.

The algorithm CONEX.

The algorithm is based on the idea that starting with a given vertex of a connected graph, one may reach any vertex of the graph.

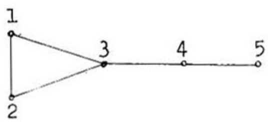
For brevity, the algorithm is described in ALGOL.

```
function  CONEX (G,K,N)
begin
   for  i = 1 step 1 until N
      do
         begin
            V[m][i] := false;
            V[*][i] := false;
            V[†][i] := false;
         end
      end
   V[m][K] := true; V[*][K] := true; V[†][K] := true;
   for  i = 1 step 1 until N
      do
         begin
            if G[i,K]
               then
                  V[m][i] := true;
                  V[*][i] := true;
            fi
         end
      end
1: for  i = 1 step 1 until N
      do
         begin
            if  V[m][i] &  V[*][i] & ¬V[†][i]
               then
                  V[†][i] := true;
```

```
                for  j = 1 step 1 until N
                   do
                      begin
                         if  G [i,j]
                            then
                               V[m][j] := true;
                         fi
                      end
                   end
         fi
      end
   end
KNMS := false;
for  i = 1 step 1 until N
   do
      begin
         if  V[m][i] & ¬ V[*][i]
            then
               KNMS := true;
               V[m][i]:= true;
         fi
      end
   end
if  KNMS
   then
      go to 1;
   else
      CARD := 0;
      for  i = 1 step 1 until N
         do
            begin
               if  V[m][i] & V[*][i] & V[↑][i]
                  then
                     CARD := CARD + 1;
               fi
            end
         end
```

```
            if  CARD ≠ N
              then
                 return  false;
              else
                 return  true;
            fi
        fi
    end  CONEX;
```

Illustratively, let us consider two applications of the algorithm:

A) 

$G = (X,\Gamma)$; Card $X = 5$

$$\begin{vmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix} = \begin{vmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{vmatrix}$$

The $V^{[m]}$, $V^{[*]}$ and $V^{[\uparrow]}$ vectors, for each step, reads:

$V^{[\uparrow]}$	
$V^{[*]}$	
$V^{[m]}$	0 0 0 0 0

(start with the vertex 1)

⇓

$V^{[\uparrow]}$	↑ ↑
$V^{[*]}$	* *
$V^{[m]}$	0 1 1 0 0

==>

$V^{[\uparrow]}$	↑ ↑ ↑
$V^{[*]}$	* * *
$V^{[m]}$	1 1 1 0 0

⇓

$V^{[\uparrow]}$	↑ ↑ ↑
$V^{[*]}$	* * * *
$V^{[m]}$	1 1 1 1 0

⇐=

$V^{[\uparrow]}$	↑ ↑ ↑ ↑
$V^{[*]}$	* * * *
$V^{[m]}$	1 1 1 1 0

⇓

⇓

$V^{[\uparrow]}$	↑ ↑ ↑ ↑ ↑
$V^{[*]}$	* * * * *
$V^{[m]}$	1 1 1 1 1

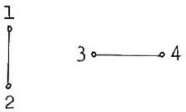
==>

$V^{[\uparrow]}$	↑ ↑ ↑ ↑ ↑
$V^{[*]}$	* * * * *
$V^{[m]}$	1 1 1 1 1

There is no entry of $V^{[m]}$ equal to 1 with corresponding $V^{[*]}$ and $V^{[\uparrow]}$ entries different from * and ↑ , respectively. The computing procedure is stopped, and one totalizes the 1's of the $V^{[m]}$:

$\sum V^{[m]} = 5 = \text{Card } X$ ==> the graph G is connected.

B)



$G = (X,\Gamma)$; Card X = 4.

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

$V^{[\uparrow]}$	
$V^{[*]}$	
$V^{[m]}$	0 0 0 0

(start with the vertex 2)

⇓

$V^{[\uparrow]}$	↑
$V^{[*]}$	*
$V^{[m]}$	1 0 0 0

==>

$V^{[\uparrow]}$	↑ ↑
$V^{[*]}$	* *
$V^{[m]}$	1 1 0 0

$V^{[\uparrow]}$	↑ ↑
$V^{[*]}$	* *
$V^{[m]}$	1 1 0 0

$\sum V^{[m]} = 2 \neq \text{Card } X = 4$;
so:
the graph G is not connected.