Chapter 8

Basic Algorithms and Program Listings

The computer listings of the basic inductive network structures for multilayer, combinatorial and harmonical techniques, and their computational aspects are given here. Multilayer algorithm uses a multilayered network structure with linearized input arguments and generates simple partial functionals. Combinatorial algorithm uses a single-layered structure with all combinations of input arguments including the full description. Harmonical algorithm follows the multilayered structure in obtaining the optimal harmonic trend with nonmultiple frequencies for oscillatory processes. One can modify these source listings as per his/her needs. These programs run on microcomputers and SPARC stations of SUN microsystems. To some extent they were also previously given for NORD-100/500 systems [88].

1 COMPUTATIONAL ASPECTS OF MULTILAYERED ALGORITHM

The basic schematic functional flow of the multilayered inductive learning algorithm is given in Chapters 2 and 7.

As the multilayer network procedure is more repetitive in nature, it is important to consider the algorithm in modules and facilitate repetitive characteristics. The most economical way of constructing the algorithm is to provide three main modules: (i) the first module is for computations of common terms in the conditional symmetric matrix of the normal equations for all input variables. This is done at the beginning of each layer with all fresh input variables entering into the layer using the training set, (ii) the second module is for generating the partial functions by forming the symmetric matrices of the normal equations for all pairs of input variables, for estimating their coefficients, for computing the values of the threshold objective functions on the testing set, and for memorizing the information of coefficients and input variables of the best functions (this is done for each layer), and (iii) the third module is for computing the coefficients of the optimal model by recollecting the information from the associated units.

To initiate the program one has to specify the control parameters:

Ml	—	no. of input variables
N	—	total no. of data points
PE	—	percentage of points on training and testing sets; $50 < PE < 100$; if $PE = 80$, then $A = 80\%$, $B = 80\%$, and $C = 20\%$

PM	—	no. of layers
ALPHA	—	weightage used in the combined criterion as C = ALPHA*C1 + (1-ALPHA)*C2, where C indicates the combined criterion (c2), Cl indicates the minimum-bias criterion, C2 indicates the regularity criterion, and $0 \leq$ ALPHA ≤ 1
CHO(I), I - 1,PM	—	freedom-of-choice at each layer of *PM* layers
FF	—	choice of optimal models at the end (FF > 1)

The values of these parameters are supplied through the file "param.dat." The file "input.dat" supplies the output and input data measurements.

The "input.dat" file is to be supplied according to the specified reference function. If the reference function is a linear function (for example, (Ml = 6)), then

$$y_1 = a_0 + a_1x_1 + a_2x_2 + \cdots + a_6x_6, \tag{8.1}$$

where a are the coefficients; $x_1, \cdots, x_6$ are the inputs to the network; and y_1 is the desired output variable. One has to supply the data file with *N* rows of points as

y_1	x_1	x_2	x_3	x_4	x_5	x_6

If the reference function is a nonlinear function (for example, ($M1$ = 5)), then

$$y_1 = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 + a_5x_1x_2, \tag{8.2}$$

where a are the coefficients; $x_1, x_2, x_1^2, x_2^2, x_1x_2$ are the inputs to the network, and y_1 is the desired output variable. One has to supply the data file with *N* rows of points as

y_1	x_1	x_2	x_1^2	x_2^2	x_1x_2

The higher-ordered terms are to be calculated and supplied in the file. Data sets A and B are separated according to the dispersion analysis.

In the first module, common terms in the conditional matrix XH is computed using the P2 input variables and the output variable Y. P1 and PU indicate the number of functions to be selected at the first layer and number of the layer, correspondingly.

In the second module, it forms the matrices (HM1, HM2, HM3) of normal equations for each pair of input variables J and I, and estimates the weights or coefficients (KO1, KO2, KO3) using the data sets A, B, and W (=A∪B), correspondingly. All partial functions are evaluated by the combined criterion. It stores the information on coefficients (KOE) and input variables (NK) of the best Pl nodes. Subroutine RANG is used to arrange all values in ascending order. Standard subroutine GAUSS is used to estimate the coefficients of each partial function.

Futhermore, the estimated outputs (YY) of Pl functions are calculated to send it to the next layer. To repeat the above two modules, we have to convert the outputs (YY) as inputs (XX) and initialize with fresh control parameters of the layer—the number of the layer PU is updated as PU+1, the number of input arguments P2 is equated to Pl, and the number of functions to be selected (freedom-of-choice) is taken from CHO(PU) as specified at the beginning. This procedure is repeated until PU becomes the number of specified layers (PM).

Modules 1 and 2 with the subroutine NM, help in forming normal equations for each pair in a more economical of utilizing computer time.

In the third module, it recollects the information for the function that has achieved global minimum or FF functions. The parameter PDM is calculated in advance as an indicator of the number of original input arguments *u* activating in the function at a particular layer—in the first layer PDM = 2 and in consecutive layers PDM = PDM*2. The coefficients and number of input arguments of the optimal function are computed using the stored information from KOE and NK.

The program listing and the sample output for a chosen example are given below.

1.1 Program listing

```
C
C*******************************************
C THIS FORTRAN VERSION IS DEVELOPED BY H. MADALA
C*******************************************
C  MULTILAYER INDUCTIVE LEARNING ALGORITHM
C
C    MAIN PROGRAM
C
      INTEGER N,M,M1,PE,PM,N1,I,J,K,S,P,R,T,GG,PN,
     1       FF,SH,PU,YP,Pl,BM,P2,NI,PDM,
     2       PL,NL,EG,SS,MH,MH1,MH2,IFAIL
      REAL XS,XM,OSH,TL,TX,YB,C,C1,C2,YM,AL,OL,H21,H22,Y3,Y11,
     1       Y22,CTROO
      REAL CML  (30,10),X(15,200),Y(l,200),KX(15),AX(200),
     1     XX(15,200),KO1(15),KO2(15),KO3(15),KO4(15),CM(30),
     2     HM1(15,16),HM2(15,16),HM3(15,16),CMM(30,10),
     3     KOE(30,10,20),CT(15),CTRO(15),D2(15),AY(200),
     4     XH(15,10,10),YY(20,200),SK(20),A(256),AD  (256),
     5      D22(200)
      INTEGER NPP  (200),NPl(200),NP2(200),NOl(200),NO2(200),
     1       CHO(10),NK(30,10,20),NC(30),ND(15),ST(20,5),
     2      NDD(200),AN  (256),AND (256),OB(200,5)
C
       OPEN(1,FILE='param.dat')
       OPEN(8,FILE='input.dat')
       OPEN(3,FILE='output.dat')
C***************
C INITIALIZATION
C***************
       READ(1,*)M1,N,PE,PM,ALPHA
       READ(l,*)(CHO(I),I=l,PM), FF
       XS =PE*N
       PE =INT(XS/100.)
C*********************************************************
C  M1 - NO. OF INPUT VARIABLES
C  N  - NO. OF DATA OBSERVATIONS
C  PE - PERCENTAGE OF TOTAL PTS. ON TRAIN AND TESTING SETS
C  PM - NO. OF LAYERS
C  (CHO(I), I =l,PM) - CHOICE OF MODELS AT EACH LAYER
C  FF - CHOICE OF OPTIMAL MODELS AT THE END
C*********************************************************
       M=1
       DO 91 I=l,N
       READ(8,*)Y(1,I),(X(J,I),J=1,M1)
   91 CONTINUE
C
   92  FORMAT (2X,'CONTROL PARAMS:'/2X,'______________'/)
   95  FORMAT (3x,'NO.OF INPUT VARIABLES (Ml)  ',I2)
   97  FORMAT (3x,'NO.OF DATA POINTS (N)  ',I3)
   99  FORMAT (3X,'PERCENTAGE OF TRAIN AND TEST POINTS (PE) ',I2)
```

```
 100   FORMAT (3X,'NO.OF LAYERS (PM) ',I2)
 102   FORMAT (3X,'WEIGHTAGE VALUE IN COMBINED CRIT (ALPHA) ',F3.1)
 104   FORMAT (3X,'FREEDOM-OF-CHOICE AT EACH LAYER(CHO) ',10I3)
 106   FORMAT (3X,'NO.OF OPTIMAL MODELS (FF) ',I2)
 108   FORMAT (3X,'NO.OF OUTPUT VARIABLES (M) ',I2)
 110   FORMAT (//)
 120   FORMAT (2X,10E10.3)
 125   FORMAT (1X,'PERFORMANCE OF THE NET:'/1X,'------------------'/)
 130   FORMAT (2X,'EQUATION NUMBER= ',I2/)
 140   FORMAT (3X,'LAYER=',I4,2X,'SELECTED DESCRIPTION=',I5)
 150   FORMAT (5X,'ERROR GAUSS='I4)
 160   FORMAT (5X,'COMBINED ERROR BEST= ',E10.3,4X,'WORST= ',E10.3)
 165   FORMAT (5x,'RESIDUAL MSE= ',E10.3,'AT THE BEST COMBINED NODE')
 170   FORMAT (5X,'RESIDUAL MSE   BEST= ',E10.3,4X,'WORST= ',E10.3)
 175   FORMAT (1X,'OPTIMAL MODELS:'/1X,'--------------'/)
 180   FORMAT (2X,'MODEL',I3,1X,'(LAYER ',I2,3X,'COMBINED=',E10.3,1X,
     1                    'MIN BIAS=',E10.3,1X,'MSE=',E10.3,1X,')')
 190   FORMAT (2X,'COEFFICIENTS=',/2X,E12.3)
 200   FORMAT (/(2X,10I10))
 210   FORMAT (2X,10E10.3)
 220   FORMAT (7X,' ---------------------------------------')
 230   FORMAT (10X,'---------------------')
 240   FORMAT(2X,'Y=')
 250   FORMAT(2X,'X=')
 260   FORMAT(/13X,'MULTI   L A Y E R E D   ALGORITHM'//)
C
       WRITE(3,260)
       WRITE (3,92)
       WRITE(3,95)M1
       WRITE(3,97)N
       WRITE(3,99)PE
       WRITE(3,100)PM
       WRITE(3,102)ALPHA
       WRITE(3,104) (CHO(I),I=1,PM)
       WRITE(3,106)FF
       WRITE(3,108)M
C
         PN=0
         S=M1+PN
                P=S
         N1=N
C
                P=M1
                S=M1
                CHO(0)=M1
                 WRITE(3,240)
                DO 71 J=1,M
                WRITE (3,120) (Y(J,I),I=1,N1)
  71            CONTINUE
               WRITE (3,250)
               WRITE (3,120) ((X(I,J), I=1,M1),J=1,N1)
C***********************************************************
C NORMALIZATION AND RANGE OF DATA AS PER DISPERSION ANALYSIS
C***********************************************************
         DO 5 J=1,S
         DO 3 I=1,N1
  3      AX(I)=ABS(X(J,I))
         CALL NORM(AX,N1,XS)
         KX(J)=XS
         DO 4 I=1,N1
```

```
  4      X(J,I)=X(J,I)/XS
  5      CONTINUE
         DO 6 I=1,N1
  6      Y(1,I)=Y(1,I)/KX(1)
             NI=0
             BM=CHO(0)
                DO 7 I=1,PM
                IF (BM.LT.CHO(I))BM=CHO(I)
  7      CONTINUE
                YP=1
  8      P2=CHO(0)
              P1=CHO(1)
             DO 9 I=1,BM
              DO 9 J=1,PM
  9      NK (I,1,J)=0
                WRITE (3,110)
        WRITE(3,125)
                WRITE (3,130)YP
         IF(P2.EQ.P)THEN
         DO 10 I=1,P
 10      ND(I)=P-I+1
         GOTO 13
         ENDIF
                DO 12 J=1,P
                D2 (J)=0.0
                DO 11 I=1,N1
 11       D2 (J)=D2(J)+X(J,I)*Y(YP,I)
 12       D2(J)=ABS (D2(J))
                CALL RANG (D2,ND,P)
 13       CONTINUE
                DO 14 J=1,P2
                DO 14 I=1,N1
             I1=P-J+1
                MH1=ND(I1)
 14       XX(J,I)=X(MH1,I)
                PU=1
                PDM=2
C***************************************************************
C FIRST MODULE TO CALCULATE COMMON TERMS IN CONDITIONAL MATRICES
C***************************************************************
 15             DO 16 I=1,N1
                D22(I)=0.0
                DO 16 J=1,P2
 16       D22(I)=D22(I)+XX(J,I)**2
                CALL RANG(D22,NDD,N1)
                DO 17 I=1,PE
                NP1 (I)=NDD(I)
            I1=N1-I+1
 17       NP2 (I)=NDD(I1)
                CALL OPE (NP1,NO1,PE,N1)
                CALL OPE (NP2,NO2,PE,N1)
            EG=0
                K=0
                DO 18 I=1,PE
                SH=NP1 (I)
                DO 18 J=1,PE
         IF(SH.EQ.NP2(J))THEN
         K=K+1
         NPP(K)=SH
         GOTO 18
```

```
       ENDIF
 18     CONTINUE
       IF(PU.EQ.1)THEN
       DO 19 I=1,N1
 19    AY(I)=ABS(Y(YP,I))
       CALL FMAX(AY,N1,YM,I)
       ENDIF
              R=N1-PE
              Y3=0.0
              Y22=0.0
              DO 74 K=1,N1
              Y3=Y3+Y(YP,K)
              Y22=Y22+Y(YP,K)**2
 74       CONTINUE
        Y22 =SQRT(Y22)
              DO 20 J=1,R
              OB(J,1)=NO1(J)
 20     OB(J,2)=NO2(J)
              DO 21 J=1,P2
              XH (J,1,3)=0.0
              XH (J,2,3)=0.0
              XH (J,3,3)=0.0
              DO 75 K=1,N1
              XH (J,1,3)=XH(J,1,3)+XX(J,K)
              XH (J,2,3)=XH(J,2,3)+XX(J,K)**2
              XH (J,3,3)=XH(J,3,3)+XX(J,K)*Y(YP,K)
 75     CONTINUE
              DO 21 T=1,2
              XH(J,1,T)=0.0
              XH(J,2,T)=0.0
              XH(J,3,T)=0.0
              DO 76 K=1,R
              MH=OB(K,T)
              XH(J,1,T)=XH(J,1,T)+XX(J,MH)
              XH(J,2,T)=XH(J,2,T)+XX(J,MH)**2
              XH(J,3,T)=XH(J,3,T)+XX(J,MH)*Y(YP,MH)
 76    CONTINUE
 21     CONTINUE
              XS=0.0
              XM=0.0
              DO 22 I=1,R
              MH1=NO1(I)
              MH2=NO2(I)
              XS=XS+Y(YP,MH1)
 22     XM=XM+Y(YP,MH2)
C***************************************************************
C SECOND MODULE FOR FORMING THE CONDITIONAL MATRICES FOR EACH
C   PARTIAL FUNCTION
C***************************************************************
              SH=1
              J=0
 23     J=J+1
              I=J+1
 24     HM1(1,1)=R
              HM2(1,1)=R
              HM1(1,4)=XS
              HM2(1,4)=XM
              H21=0.0
              H22=0.0
              DO 77 K=1,R
```

```
               MH1=NO1(K)
               MH2=NO2(K)
               H21=H21+XX(J,MH1)*XX(I,MH1)
               H22=H22+XX(J,MH2)*XX(I,MH2)
  77       CONTINUE
               HM1(2,3)=H21
               HM1(3,2)=H21
               HM2(2,3)=H22
               HM2(3,2)=H22
               HM2(1,4)=XM
               CALL NM (HM1,XH,1,J,I)
               CALL NM (HM2,XH,2,J,I)
               DO 25 K=1,3
               DO 25 S=1,4
 25       HM3(K,S)=HM1(K,S)+HM2(K,S)
C******************************
C     ESTIMATING COEFFICIENTS
C******************************
               CALL GAUSS(HM1,3,4,KO1,IFAIL)
               IF (IFAIL.EQ.0)GO TO 29
               CALL GAUSS(HM2,3,4,KO2,IFAIL)
               IF(IFAIL.EQ.0)GO TO 29
               CALL GAUSS (HM3,3,4,KO3,IFAIL)
               IF (IFAIL.EQ.0)GO TO 29
C*********************************************
C     COMPUTING THE VALUES OF EXTERNAL CRITERIA
C*********************************************
               C1=0.0
               C2=0.0
C*****************************************************
C  C1 - MEAN SQUARED MINIMUM BIAS ERROR ON TOTAL POINTS
C  C2 - MEAN SQUARED RESIDUAL ERROR ON EXAMIN SET
C  C  - ROOT MEAN COMBINED ERROR  OF (C1 + C2)
C*****************************************************
               DO 78 S=1,N1
          C1=C1+(KO1(1)-KO2(1)+(KO1(2)-KO2(2))*XX(J,S)+(KO1(3)-
     1     KO2(3))*XX(I,S))**2
 78     CONTINUE
               C1=C1/(Y22**2)
        Y11 =0.0
               MH1=2*PE-N1
               DO 79 S=1,MH1
               MH=NPP(S)
        C2=C2+(Y(YP,MH)-KO3(1)-KO3(2)*XX(J,MH)-KO3(3)*XX(I,MH))**2
        Y11 =Y11+Y(YP,MH)**2
 79    CONTINUE
               C2=C2/Y11
        C = SQRT( ALPHA*C1 + (1-ALPHA)*C2)
C
        CALL NM(HM3,XH,3,J,I)
               HM3(1,1)=N1
             HM3(1,4)=Y3
               HM3(2,3)=0.0
               DO 80 K=1,N1
               HM3(2,3)=HM3(2,3)+XX(J,K)*XX(I,K)
 80     CONTINUE
               HM3(3,2)=HM3(2,3)
               CALL GAUSS(HM3,3,4,KO4,IFAIL)
               IF(IFAIL.EQ.0)GO TO 29
               IF(SH.GT.P1)GO TO 27
```

```
                  CM(SH)=C
                  DO 26 K=1,3
 26         KOE(SH,K,PU)=KO4(K)
                  CMM(SH,1)=C1
                  CMM(SH,2)=C2
                  NK(SH,2,PU)=J
                  NK(SH,3,PU)=I
                  IF(SH.EQ.P1)CALL RANG(CM,NC,P1)
                  SH=SH+1
                  GO TO 30
 27        MH1=NC(P1)
                  IF(C.GT.CM(MH1))GO TO 30
                  GG=NC(P1)
                  CMM(GG,1)=C1
                  CMM(GG,2)=C2
                  CM(MH1)=C
                  DO 28 K=1,3
 28         KOE(MH1,K,PU)=KO4(K)
                  NK(MH1,2,PU)=J
                  NK(MH1,3,PU)=I
                  CALL RANG(CM,NC,P1)
                  GO TO 30
 29         EG=EG+1
 30         I=I+1
                  IF(I.LE.P2)GO TO 24
                  IF(J.LT.P2-1)GO TO 23
                  DO 33 S=1,P1
                  OSH=0.0
                  DO 32 J=1,N1
                  YB=KOE(S,1,PU)
                  DO 31 I=2,3
                  MH1=NK(S,I,PU)
 31         YB=YB+KOE(S,I,PU)*XX(MH1,J)
 32         OSH=OSH+(Y(YP,J)-YB)**2
C                 OSH=SQRT(OSH/N1)/Y22
           OSH =SQRT(OSH)/Y22
           IF(S.EQ.1)THEN
           TX=OSH
           TL=OSH
           ENDIF
           IF(NC(1).EQ.S)THEN
           AL=OSH
           IF(PU.EQ.1)THEN
           OL=OSH
           PL=1
           NL=S
           ENDIF
           IF(OL.GE.OSH)THEN
           OL=OSH
           PL=PU
           NL=S
           ENDIF
           ENDIF
           IF(OSH.LT.TL)TL=OSH
           IF(OSH.GT.TX)TX=OSH
 33                  CONTINUE
C*******************************************************
C PRINTING THE PERFORMANCE OF THE NETWORK AT EACH LAYER
C*******************************************************
                  MH1=NC(1)
```

```
                MH2=NC(P1)
                WRITE(3,140)PU,P1
                WRITE(3,150)EG
                WRITE(3,160)CM(MH1),CM(MH2)
                WRITE(3,170)TL,TX
                WRITE(3,165)AL
                WRITE(3,230)
 34          PDM=2*PDM
            DO 35 J=1,P1
            DO 35 I=1,N1
            YY(J,I)=KOE(J,1,PU)
            DO 35 S=2,3
 35         YY(J,I)=YY(J,I)+XX(NK(J,S,PU),I)*KOE(J,S,PU)
            DO 36 J=1,P1
            DO 36 I=1,N1
 36         XX(J,I)=YY(J,I)
            IF(PU.EQ.1)THEN
            DO 39 I=1,FF
            IF(I.LE.P1)THEN
            CML(I,1)=CM(NC(I))
            CML(I,2)=PU
            CML(I,3)=NC(I)
            DO 38 J=1,2
 38         CML(I,J+3)=CMM(NC(I),J)
            ELSE
            CML(I,1)=10000.
            ENDIF
 39         CONTINUE
            ELSE
            K=1
 40         I=1
            C=CML(1,1)
            DO 41 J=2,FF
            IF(CML(J,1).GT.C)THEN
            C=CML(J,1)
            I=J
            ENDIF
 41         CONTINUE
            IF(C.LE.CM(NC(K)))GOTO 43
            CML(I,2)=PU
            CML(I,1)=CM(NC(K))
            DO 42 J=1,2
 42         CML(I,J+3)=CMM(NC(K),J)
            CML(I,3)=NC(K)
            K=K+1
            IF(K.LE.P1)GOTO 40
 43         ENDIF
                IF(PU.EQ.PM)GO TO 44
                PU=PU+1
                P2=P1
                P1=CHO(PU)
                GO TO 15
C***********************************************
C THIRD MODULE TO RECOLLECT THE OPTIMAL MODELS
C***********************************************
 44         WRITE(3,110)
            WRITE(3,175)
            SS=0
            PDM=PDM-1
 45         SS=SS+1
```

```
 46      PU=CML(SS,2)
         NCDGE=CML(SS,3)
         K=0
         DO 47 I=1,10
         ST(I,1)=0
         ST(I,2)=0
         SK(I)=0
 47      CONTINUE
       WRITE(3,180)SS,INT(CML(SS,2)),CML(SS,1),
      1SQRT(CML(SS,4)),SQRT(CML(SS,5))
         K=0
         DO  48 I=0,P
         CTRO(I)=0.0
 48      CT(I)=0.0
                DO 49 I=1,PDM
                A(I)=0.0
                AD(I)=0.0
                AN(I)=0
 49      AND(I)=0
                DO 50 I=1,3
                A(I)=KOE(NCDGE,I,PU)
 50      AN(I)=NK(NCDGE,I,PU)
                IF(PU.EQ.1)GO TO 55
                AD(1)=A(1)
 88      SH=1
         DO 53 I=2,PDM
         IF(A(I).NE.0)THEN
         IF(AN(I).EQ.0)THEN
         SH=SH+1
         AD(SH)=A(I)
         AND(SH)=0
         ELSE
         DO 86 S=1,3
         AD(SH+S)=A(I)*KOE(AN(I),S,PU-1)
         AND(SH+S)=NK(AN(I),S,PU-1)
 86      CONTINUE
         SH=SH+3
         ENDIF
         ENDIF
 53      CONTINUE
                DO 54 I=2,PDM
                A(I)=AD(I)
                AN(I)=AND(I)
 54      CONTINUE
         PU=PU-1
         IF(PU.GT.1)GOTO 88
 55      CONTINUE
                DO 56 I=1,PDM
                S=AN(I)
                CT(S)=CT(S)+A(I)
 56      CONTINUE
                DO 57 I=1,P
               IP1=P-I+1
                MH=ND(IP1)
                CTRO(MH)=CT(I)
 57      CONTINUE
                CTRO(0)  =CT(0)*KX(YP)
                CTROO  =CTRO(0)
                WRITE (3,190)CTROO
              MPN=M1+PN
```

```
                DO 60 J=1,MPN
                IF(CTRO(J).NE.0.0)THEN
                CTRO(J)=CTRO(J)*KX(YP)/KX(J)
                K=K+1
                ST(K,1)=J
                SK(K)=CTRO(J)
                   IF(K.EQ.10)THEN
                 WRITE(3,200)(ST(K,1),K=1,10)
                 WRITE(3,210)(SK(K),K=1,10)
                 DO 61 K=1,10
                ST(K,1)=0
                 SK(K)=0
 61           CONTINUE
                 K=0
                 ENDIF
                ENDIF
 60            CONTINUE
                IF(K.NE.0)THEN
                WRITE(3,200)(ST(I,1),I=1,K)
                WRITE(3,210)(SK(I),I=1,K)
               ENDIF
             WRITE(3,220)
                IF(SS.LT.FF)GO TO 45
                YP=YP+1
                IF(YP.LE.M)GO TO 8
                close(3)
                close(8)
                close(1)
                STOP
                END
```

Subroutines used

```
C
                SUBROUTINE FMAX(X,N,XM,K)
                DIMENSION X(200)
                REAL XM
                INTEGER N,K,I
                XM=X(1)
                K=1
                DO 1 I=2,N
                IF (XM.GE.X(I))GOTO 1
                XM=X(I)
                K=I
  1       CONTINUE
                RETURN
                END
C
C
                SUBROUTINE NORM(XN,N,P)
                DIMENSION XN(200)
                INTEGER N,K
                REAL P,XM
                CALL FMAX (XN,N,XM,K)
                P=1.0
  1       P=P*10
                IF(P.GT.XM)GO TO 2
                GO TO 1
  2       P=P/10
```

```
      IF(P.LT.XM)GO TO 3
      GO TO 2
 3    P=P*10
      RETURN
      END
C
C
      SUBROUTINE RANG(X,NP,N)
      DIMENSION X(200),XD(200)
      INTEGER NP(200),ND(200)
      INTEGER N,K,I,N1
      REAL XM
      DO 1 I=1,N
      XD(I)=X(I)
 1    ND(I)=I
      N1=N
 2    CALL FMAX(XD,N1,XM,K)
      NP(N1)=ND(K)
      K1=K+1
      DO 3 I=K1,N1
      XD(I-1)=XD(I)
 3    ND(I-1)=ND(I)
      N1=N1-1
      IF(N1.GE.2)GO TO 2
      NP(1)=ND(1)
      RETURN
      END
C
C
      SUBROUTINE NM(HM,XH,T,J,I)
      INTEGER T,S,R
      DIMENSION XH(15,10,10),HM(15,16)
      S=2
      R=J
 1    HM(1,S)=XH(R,1,T)
      HM(S,1)=HM(1,S)
      HM(S,S)=XH(R,2,T)
      HM(S,4)=XH(R,3,T)
      S=S+1
      R=I
      IF(S.EQ.3)GO TO 1
      RETURN
      END
C
C
      SUBROUTINE OPE(NP,NO,PE,N1)
      INTEGER I,J,Z,PE
      INTEGER NP(200),NO(200)
      Z=0
      I=1
 1    DO 2 J=1,PE
      IF(I.EQ.NP(J))GO TO 3
 2    CONTINUE
      Z=Z+1
      NO(Z)=I
 3    I=I+1
      IF (I.LE.N1) GO TO 1
      RETURN
      END
C
```

```
C
      FUNCTION RND(S2)
        R1=(S2+3.14159)*5.04
      Rl=Rl-INT(Rl)
      S2=R1
        RND=R1
      RETURN
      END
C
C
      SUBROUTINE  GAUSS(A,N,L,X,IF)
      DIMENSION A(15,16),X(15)
      IF=1
      NN=N-1
      DO 99 K=1,NN
      J=K
      KK=K+1
      DO 100  I=KK,N
      IF(ABS(A(J,K)).LT.ABS(A(I,K)))J=I
 100  CONTINUE
      IF(J.EQ.K)GOTO  11
      DO 300 I=l,L
      T=A(K,I)
      A(K,I)=A(J,I)
      A(J,I)=T
 300  CONTINUE
  11  DO 88 J=KK,N
      IF(A(K,K).EQ.0.)GOTO 13
      D=-A(J,K)/A(K,K)
      DO 400 I=l,L
      A(J,I)=A(J,I)+D*A(K,I)
 400  CONTINUE
  88  CONTINUE
  99  CONTINUE
      IF(A(N,N).EQ.O.)GOTO  13
      X(N)=A(N,L)/A(N,N)
      NN=N-1
      DO 500 J=l,NN
      K=N-J
      SUM=0.0
      NNN=N-K
      DO 200 JJ=1,NNN
      M=K+JJ
      SUM=SUM+A(K,M)*X(M)
 200  CONTINUE
      IF(A(K,K).EQ.O.)GOTO  13
      X(K)=(A(K,L)-SUM)/A(K,K)
 500  CONTINUE
      GOTO 14
  13  IF=0
  14  RETURN
      END
C
```

1.2 Sample output

Example. The output data is generated from the equation:

$$y = 0.433 - 0.095x_1 + 0.243x_2 + 0.35x_1^2 - 0.18x_1x_2 + \epsilon,$$

where x_1, $X2$ are randomly generated input variables, y is the output variable computed from the above equation, and ϵ is the noise added to the data. The data file "input.dat" is prepared correspondingly.

The control parameters are supplied in the file "param.dat"

```
5  100  75  7  0.5
10  10  10  10  10  10  10  8
```

The parameters take the values as Ml =5, N =100, PE =75, PM =7, ALPHA =0.5, CHO(l) =10, CHO(2) =10, ..., CHO(7) =10, and FF =8.

The program creates the output file "output.dat" with the results.

The results are given first with the control parameters, then the performance of the network at each layer that include the values of the combined criterion for the best and the worst models, the values of the residual mean-square error (MSE) for the best and the worst models, and the residual MSE value for the best model according to the combined criterion. The value of ERROR GAUSS indicates the number of singular nodes, if any in the layer, and the SELECTED DESCRIPTION is the freedom-of-choice at each layer. The EQUATION NUMBER indicates the number of the output variable. It is fixed as one (M = 1) because it is dealt with as a single output equation. This can be changed to a number of output equations and the program is modified accordingly.

The coefficient values of optimal models as a number specified for FF are displayed with the constant term and the numbers of input variables with the layer number and the values of the criteria. The second model in the list, obtained at the seventh layer, is the best among all according to the combined criterion; this is read as

$$y = 0.433 - 0.0948x_1 + 0.248x_2 + 0.340x_1^2 - 0.00593x_2^2 - 0.167x_1x_2. \qquad (8.3)$$

The output is written in the file "output.dat" as below:

```
          M U L T I     L   A   Y   E   R      ALGORITHM


 CONTROL PARAMS:
 --------------

  NO.OF INPUT VARIABLES (M1)  5
  NO.OF DATA POINTS (N) 100
  PERCENTAGE OF TRAIN AND TEST POINTS (PE) 75
  NO.OF LAYERS (PM)  7
  WEIGHTAGE VALUE IN COMBINED CRIT (ALPHA) 0.5
  FREEDOM-OF-CHOICE AT EACH LAYER(CHO)   10 10 10 10 10 10 10
  NO.OF OPTIMAL MODELS (FF)  8
  NO.OF OUTPUT VARIABLES (M)  1



PERFORMANCE OF THE NET:
--------------------

 EQUATION NUMBER=  1

  LAYER=   1  SELECTED DESCRIPTION=   10
    ERROR GAUSS=   0
    COMBINED ERROR BEST=  0.644E-01    WORST=  0.275E+00
    RESIDUAL MSE   BEST=  0.304E-01    WORST=  0.961E-01
    RESIDUAL MSE=  0.304E-01 AT THE BEST COMBINED NODE
```

```
          ---------------------
 LAYER=   2  SELECTED DESCRIPTION=   10
   ERROR GAUSS=   0
   COMBINED ERROR BEST=  0.202E-01    WORST=  0.538E-01
   RESIDUAL MSE   BEST=  0.160E-01    WORST=  0.334E-01
   RESIDUAL MSE=  0.177E-01 AT THE BEST COMBINED NODE
          ---------------------
 LAYER=   3  SELECTED DESCRIPTION=   10
   ERROR GAUSS=   0
   COMBINED ERROR BEST=  0.196E-01    WORST=  0.223E-01
   RESIDUAL MSE   BEST=  0.113E-01    WORST=  0.194E-01
   RESIDUAL MSE=  0.173E-01 AT THE BEST COMBINED NODE
          ---------------------
 LAYER=   4  SELECTED DESCRIPTION=   10
   ERROR GAUSS=   0
   COMBINED ERROR BEST=  0.108E-01    WORST=  0.162E-01
   RESIDUAL MSE   BEST=  0.592E-02    WORST=  0.117E-01
   RESIDUAL MSE=  0.608E-02 AT THE BEST COMBINED NODE
          ---------------------
 LAYER=   5  SELECTED DESCRIPTION=   10
   ERROR GAUSS=   0
   COMBINED ERROR BEST=  0.614E-02    WORST=  0.127E-01
   RESIDUAL MSE   BEST=  0.470E-02    WORST=  0.878E-02
   RESIDUAL MSE=  0.509E-02 AT THE BEST COMBINED NODE
          ---------------------
 LAYER=   6  SELECTED DESCRIPTION=   10
   ERROR GAUSS=   0
   COMBINED ERROR BEST=  0.593E-02    WORST=  0.861E-02
   RESIDUAL MSE   BEST=  0.392E-02    WORST=  0.509E-02
   RESIDUAL MSE=  0.418E-02 AT THE BEST COMBINED NODE
          ---------------------
 LAYER=   7  SELECTED DESCRIPTION=   10
   ERROR GAUSS=   0
   COMBINED ERROR BEST=  0.496E-02    WORST=  0.664E-02
   RESIDUAL MSE   BEST=  0.349E-02    WORST=  0.418E-02
   RESIDUAL MSE=  0.362E-02 AT THE BEST COMBINED NODE
          ---------------------


OPTIMAL MODELS:
--------------

 MODEL  1 ( LAYER  7   COMBINED= 0.599E-02 MIN BIAS= 0.713E-02
                            MSE= 0.457E-02 )
 COEFFICIENTS=
    0.431E+00

       1          2          3          4          5
 -0.813E-01 0.245E+00 0.326E+00-0.614E-02-0.161E+00
         -------------------------------------
 MODEL  2 ( LAYER  7   COMBINED= 0.496E-02 MIN BIAS= 0.566E-02
                            MSE= 0.415E-02 )
 COEFFICIENTS=
    0.433E+00

       1          2          3          4          5
 -0.948E-01 0.248E+00 0.340E+00-0.593E-02-0.167E+00
         -------------------------------------
 MODEL  3 ( LAYER  7   COMBINED= 0.550E-02 MIN BIAS= 0.654E-02
```

```
                               MSE= 0.420E-02 )
COEFFICIENTS=
   0.433E+00

          1          2          3          4          5
-0.941E-01 0.250E+00 0.339E+00-0.749E-02-0.168E+00
      --------------------------------------
MODEL  4 ( LAYER  7   COMBINED= 0.570E-02 MIN BIAS= 0.685E-02
                               MSE= 0.423E-02 )
COEFFICIENTS=
   0.432E+00

          1          2          3          4          5
-0.937E-01 0.250E+00 0.339E+00-0.795E-02-0.168E+00
      --------------------------------------
MODEL  5 ( LAYER  7   COMBINED= 0.580E-02 MIN BIAS= 0.663E-02
                               MSE= 0.483E-02 )
COEFFICIENTS=
   0.431E+00

          1          2          3          4          5
-0.813E-01 0.245E+00 0.326E+00-0.619E-02-0.161E+00
      --------------------------------------
MODEL  6 ( LAYER  6   COMBINED= 0.593E-02 MIN BIAS= 0.702E-02
                               MSE= 0.458E-02 )
COEFFICIENTS=
   0.431E+00

          1          2          3          4          5
-0.812E-01 0.245E+00 0.326E+00-0.617E-02-0.161E+00
      --------------------------------------
MODEL  7 ( LAYER  7   COMBINED= 0.576E-02 MIN BIAS= 0.696E-02
                               MSE= 0.421E-02 )
COEFFICIENTS=
   0.432E+00

          1          2          3          4          5
-0.923E-01 0.251E+00 0.338E+00-0.828E-02-0.169E+00
      --------------------------------------
MODEL  8 ( LAYER  7   COMBINED= 0.578E-02 MIN BIAS= 0.700E-02
                               MSE= 0.423E-02 )
COEFFICIENTS=
   0.432E+00

          1          2          3          4          5
-0.915E-01 0.251E+00 0.338E+00-0.863E-02-0.169E+00
      --------------------------------------
```

2 COMPUTATIONAL ASPECTS OF COMBINATORIAL ALGORITHM

The algorithm given is for a single-layered structure. The mathematical description of a system is represented as a reference function in the form of discrete Volterra series in multivariate data and finite-difference equations in time series data.

$$y = a_0 + \sum_{i=1}^{l} a_i x_i + \sum_{i=1}^{l} \sum_{j=1}^{l} a_{ij} x_i x_j + \sum_{i=1}^{l} \sum_{j=1}^{l} \sum_{k=1}^{l} a_{ijk} x_i x_j x_k + \cdots$$

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots, \tag{8.4}$$

where y and x_i are the desired and input variables in the first polynomial; / is the number of input variables; y_t is the desired output at the time t; $y_{t-1}, y_{t-2}, \cdots$ are the delayed arguments of the output as inputs in the finite-difference scheme.

The combinatorial algorithm frames all combinations of partial functions from the given reference function. If the reference function is a linear function; for example,

$$y = f(x_1, x_2) = a_0 + a_1x_1 + a_2x_2, \tag{8.5}$$

then it generates

$$y = a_0, y = a_1x_1, y = a_2x_2, y = a_0 + a_1x_1,$$

$$y = a_0 + a_2x_2, y = a\backslash x\backslash + a_2x_2, \text{ and } y = a_0 + a\backslash x\backslash + a_2x_2. \tag{8.6}$$

Suppose there are $m(= 3)$ parameters in the reference function, then the total combinations are $2^m - 1 (= 7)$. The "structure of functions" is used to generate these partial models.

a_2	a_1	a_0
0	0	1
0	1	0
1	0	0
0	1	1
1	0	1
1	1	0
1	1	1

where each row indicates a partial function with its parameters represented by "1," the number of rows indicates the total number of units, and the number of columns indicates total number of parameters in the full description. This matrix is referred further in forming the normal equations.

The weights are estimated for each partial equation by using the least squares technique with a training data set at each unit and computed at its threshold measure according to the external criterion using the test set. Then the unit errors are compared with each other and the better functions are selected for their output responses and evaluated further.

For simplicity, the external criteria used in this algorithm are the minimum-bias, regularity, and combined criterion of minimum-bias and regularity.

Three ways of splitting data are used here: sequential, alternative, and dispersion analysis. The user can choose one of them or experiment with them for different types of splittings.

The program works for time series data as well as multivariate data. If it is time series data, the user has to specify the number of autoregressive terms in the finite-difference function and supply the "input.dat" file with the time series data. If it is multivariate data, one has to specify the number of input variables and supply the "input.dat" file with the rows of the data points for output and input variables.

The program listing and an example with the sample output are given below.

2.1 Program listing

```
C
C***********************************************************************
C THIS PROGRAM IS THE RESULT OF EFFORTS FROM VARIOUS GRADUATE STUDENTS
C  AND RESEARCH PROFESSIONALS AT THE COMBINED CONTROL SYSTEMS GROUP OF
C  INSTITUTE OF CYBERNETICS, KIEV (UKRAINE)
```

```
C*************************************************************************
C    SINGLE LAYER COMBINATORIAL INDUCTIVE ALGORITHM
C
C      M   - TOTAL NO.OF DATA POINTS
C      MP  - NO.OF POINTS IN TEST SET
C      MA1 - NO.OF POINTS IN EXAMIN SET
C      IT - ORDER OF THE MODEL
C      L   - NO.OF INPUT VARIABLES
C      NB - FREEDOM OF CHOICE (NO.OF BEST MODELS AT THE OUTPUT)
C      IH - NO.OF DISCRETE POINTS IN SIGNAL DATA
C      G(IH)- DISCRETE SIGNAL DATA
C      LM - SELECTION CRITERION NO.
C      IS = -1 - DATA IS SPLITTED ON THE BASIS OF STD.DEVIATIONS
C         =  0 - DATA IS SPLITTED ALTERNATELY
C         =  1 - DATA IS SPLITTED SEQUENTIALLY
C      Y1(M) - DEPENDENT VARIABLE (OUTPUT VECTOR)
C      X1(M,L) - INDEPENDENT VARIABLES (INPUT MATRIX)
C      Y(M) - OUTPUT VECTOR AFTER SEPARATION OF DATA
C      X(M,L) - INPUT MATRIX AFTER SEPARATION OF DATA
C
C SUBROUTINE DATA - WHICH SUPPLIES THE DISCRETE SIGNAL
C                   DATA G(IH)
C SUBROUTINE FORM - WHICH FORMS THE OUTPUT VECTOR Y1(M)AND
C                   THE INPUT MATRIX  X1(M,L)FROM THE
C                         DISCRETE SIGNAL G(IH). THIS IS MAINLY
C                         FOR FORMING FINITE-DIFFERENCE
C                         EQUATIONS
C
C***********************************************************************
C
C   MAIN PROGRAM
C
           DIMENSION D(100)
           INTEGER NP(9)
           COMMON /GAMA/G(100)
           COMMON /X1Y1/X1(100,15),Y1(100)
           COMMON /XYUD/X(100,15),Y(100),UD(100,15)
           COMMON /PS/NB,N,PS(15,16)
           COMMON /INIT/M,MP,MA1,L
C
             OPEN(3,FILE='results.dat')
             OPEN(8,FILE='innl.dat')
C
        WRITE(3,12)
  12    FORMAT(8X,'SINGLE  L A Y E R E D  COMBINATORIAL ALGORITHM'///)
             WRITE (*,230)
 230         FORMAT(2X,'GIVE TOTAL DISCRETE POINTS')
        READ(*,*)IH
C
         WRITE(*,235)
 235     FORMAT(2X, 'TIME SERIES (1)/ MULTIVARIATE DATA (2)??')
         READ(*,301)IIH
         IF (IIH.EQ.2) GOTO 350
C
        CALL DATA(IH)
        WRITE(3,240)
 240    FORMAT(2X,'DATA:')
             WRITE (3,100) (G(I),I=1,IH)
 100         FORMAT (3X,5F12.2)
        WRITE(3,303)
```

```
 303     FORMAT(//)
         WRITE(*,300)
 300     FORMAT(//3X,'Give No of AR terms in model= ')
         READ(*,301)L
 301     FORMAT(I2)
             CALL FORM(IH,M,L)
             IF (IIH.EQ.1) GOTO 355
C
 350       M =IH
           WRITE(*,245)
 245       FORMAT(2X, 'GIVE NO.OF INPUT VARIABLES??')
           READ(*,301)L
           DO 91 I =1,M
           READ(8,*) Y1(I), (X1(I,J), J=1,L)
  91       CONTINUE
C
 355     WRITE(3,250)M
         WRITE(*,250)M
 250     FORMAT(//2X,'TOTAL NO.OF DATA PTS. =',I3//)
         WRITE(*,280)
 280     FORMAT(2X,'GIVE NO.OF TRAINING PTS??')
         READ(*,290)ME
 290     FORMAT(I2)
         WRITE(*,260)
 260     FORMAT(2X,'GIVE NO.OF TESTING PTS??')
         READ(*,270)MP
 270     FORMAT(I2)
         MA1=M-(MP+ME)
         IF(MA1.LE.0)MA1=0
         WRITE(*,999)
 999     FORMAT(1H$,'DATA SETS BY (-1 DISP, 0 ALTER, 1 SEQUEN)?')
         READ(*,220)IS
 220     FORMAT(I2)
C
              YM=0.0
              DO 5 I=1,M
              YM=YM+Y1(I)
   5          CONTINUE
              YM=YM/M
              IF(IS)15,16,17
  15          DO 7 I=1,M
   7          Y1(I)=(Y1(I)-YM)/YM
              DO 8 I=1,L
              XM=0.0
              DO 9 J=1,M
   9          XM=XM+X1(J,I)
              XM=XM/M
              DO 10 J=1,M
  10          X1(J,I)=(X1(J,I)-XM)/XM
   8          CONTINUE
              DO 11 I=1,M
              D(I)=Y1(I)**2
              DO 13 J=1,L
  13          D(I)=D(I)+X1(I,J)**2
              D(I)=D(I)/(L+1)
  11          CONTINUE
              CALL RANG (D,NP,M)
              DO 14 I=1,M
              I2=M-I+1
              I1=NP(I2)
```

```
            Y(I)=Y1(I1)
            DO 14 J=1,L
            X(I,J)=X1(I1,J)
  14        CONTINUE
            GO TO 3
  16        I1=0
            DO 18 L1=1,2
            DO 18 I=L1,M,2
            I1=I1+1
            Y(I1)=Y1(I)
            DO 18 J=1,L
            X(I1,J)=X1(I,J)
  18        CONTINUE
            GO TO 3
  17        DO 19 I=1,M
            Y(I)=Y1(I)
            DO 19 J=1,L
            X(I,J)=X1(I,J)
  19        CONTINUE
   3        CONTINUE
            CALL COMBI
        NOB=NB
            STOP
            END
C
```

Subroutines used

```
        SUBROUTINE DATA(IH)
        COMMON /GAMA/G(100)
        DO 300 I=1,IH
        READ(8,100)G(I)
 100     format(f12.6)
 300    CONTINUE
        RETURN
        END
C
            SUBROUTINE FORM(IH,M,L)
            COMMON /GAMA/G(100)
            COMMON /X1Y1/X(100,15),Y(100)
            M1=0
            L1=L+1
            DO 2 I=L1,IH
            M1=M1+1
            Y(M1)=G(I)
            DO 1 J=1,L
            IJ=I-J
   1        X(M1,J)=G(IJ)
   2        CONTINUE
            M=M1
            RETURN
            END
C
            SUBROUTINE COMBI
            REAL KCH,IQ
        DIMENSION OS(16),OA(16),FS(15,16),FS1(15,16),
     1      ID(15),P(15),P1(15),IA(15),IP(15)
        COMMON /XYUD/X(100,15),Y(100),UD(100,15)
        COMMON /PS/NB,N,PS(15,16)
```

```
      COMMON /INIT/M,MP,MA1,L
65     FORMAT(/2X,'MODEL ORDER (IT)=',I3/2X,'NO INPUT VAR.(L)=',
    1 I3/2X,'TOTAL NO.PTS.(M)=',I3/2X,'N0.PTS.TESTSET(MP)=',I3/2X,
    2 'NO.PTS.EXAM.SET (MA1)=',I3/)
           WRITE(*,64)
64      FORMAT(2X,'GIVE ORDER OF THE MODEL??')
           READ(*,*)IT
           WRITE (3,65)IT,L,M,MP,MA1
           N=1
           DO 38 J1=1,L
38         N=N*(IT+J1)/J1
           KCH=2.**N-1
           WRITE(3,50)N,KCH
           WRITE(*,50)N,KCH
50     FORMAT (/4X, 'NO.TERMS IN FULL MODEL=',I3/4X,
    1   'NO.PARTIAL MODELS=',F12.0/)
      WRITE(*,320)
320   FORMAT(///2X,'NO OF OPTIMAL MODELS (NB)??')
      READ(*,330)NB
330   FORMAT(I2)
      WRITE(3,321)NB
321   FORMAT(//2X,'NO OF OPTIMAL MODELS = ',I2)
C**********************************
C - FORMING CONDITIONAL EQUATIONS
C**********************************
           N1=N+1
           MA=M-MA1
           MO=MA-MP
           MPR=MO+1
C**********************************
C   STRUCTURE OF FULL POLYNOMIAL
C**********************************
           CALL FORD(IT,L,M,N,IP)
            WRITE(*,100)
100   FORMAT(1H$,'GIVE SELECT CRIT(1-REGUL,2-MINBIAS,3-COMBINED)?')
      READ(*,101)LM
101   FORMAT(I2)
C**********************************
C    FORMING NORMAL EQUATIONS
C**********************************
           CALL NOS(N,N1,M,1,MO,FS)
           CALL NOS(N,N1,M,MPR,MA,FS1)
C************************************
C    SORTING OF PARTIAL DESCRIPTIONS
C************************************
           IQ=0.0
C*********************************************
C    CALCULATION OF COEFFICIENTS OF THE MODELS
C*********************************************
41         IQ=IQ+1
           CALL DICH(IQ,ID,N,2)
           KB=0
           DO 60 I4=1,N
60         KB=KB+ID(I4)
           KB1=KB+1
           CALL PAP(ID,N,N1,KB,KB1,FS,P,IA)
           CALL PAP(ID,N,N1,KB,KB1,FS1,P1,IA)
C***********************************************
C    VALIDATION OF MODELS BY SELECTING CRITERION
C***********************************************
```

```
          IF(LM-2)92,93,92
92        OSH=0.0
          DO 54 J=MPR,MA
          Z=0.0
          DO 55 I=1,N
55        Z=Z+P(I)*UD(J,I)
54        OSH=OSH+(Z-Y(J))**2
          OSH1=SQRT(OSH)/MP
          IF (LM-2)51,93,93
93        OSH=0.0
          DO 56 J3=1,MA
          Z=0.0
          AF=0.0
          DO 57 I3=1,N
          Z=Z+P(I3)*UD(J3,I3)
57        AF=AF+P1(I3)*UD(J3,I3)
56        OSH=OSH+(Z-AF)**2
          OSH2=SQRT(OSH)/MA
          IF(LM-2)51,52,53
51        OSH=OSH1
          GO TO 59
52        OSH=OSH2
          GO TO 59
53        OSH=OSH1+OSH2
C*********************************
C     SELECTION OF THE NB BEST MODELS
C*********************************
59        IF (IQ-NB)42,42,43
42        JF=IQ
          GO TO 47
43        IF(NB-1)45,44,45
44        R5=OS(1)
          GO TO 49
45        CALL FMAX(OS,NB,R5,JF)
49        IF(OSH-R5)47,41,41
47        OS (JF)=OSH
          DO 48 I5=1,N
48        PS (I5,JF)=P(I5)
          IF (IQ.LT.KCH)GO TO 41
C*****************************************************
C     SELECTION CRITERION FOR SORTING OUT THE BEST MODELS
C*****************************************************
          IF(LM-2)88,89,90
88        WRITE(3,85)
85        FORMAT(/4X,'SORTING OUT BY REGULARITY CRITERION')
          GOTO 91
89        WRITE(3,84)
84        FORMAT (/4X,'SORTING OUT BY MINIMUM-BIAS CRITERION')
          GOTO 91
90        WRITE(3,80)
80        FORMAT(/4X,'SORTING OUT BY COMBINED CRITERION')
91        CONTINUE
          WRITE(3,75)
75        FORMAT(4X,'DEPTH OF THE MINIMUM')
          WRITE(3,68)(OS(K),K=1,NB)
C**********************************
C     ADAPTATION OF THE COEFFICIENTS
C**********************************
          DO 76 K=1,NB
          DO 71 I6=1,N
```

```
          IF (PS(I6,K))72,73,72
73        ID(I6)=0
          GO TO 71
72        ID(I6)=1
71        CONTINUE
          CALL NOS(N,N1,M,1,MA,FS)
          KB=0
          DO 70 I7=1,N
70        KB=KB+ID(I7)
          KB1=KB+1
          CALL PAP(ID,N,N1,KB,KB1,FS,P,IA)
          DO 58 I8=1,N
58        PS(I8,K)=P(I8)
          OSH=0.0
          AF=0.0
          DO 77 J=1,M
          Z=0.0
          DO 78 I=1,N
78        Z=Z+PS(I,K)*UD(J,I)
          OSH=OSH+(Z-Y(J))**2
          AF=AF+Y(J)**2
          IF(J-MA)77,79,77
79        R5=OSH
          R1=AF
77        CONTINUE
          R7=R5/R1
          OS(K)=SQRT(R7)
          IF (MA1)83,83,86
83        OA(K)=0.0
          GO TO 76
86        OA(K)=SQRT((OSH-R5)/(AF-R1))
76        CONTINUE
C**************************************************
C    PRINTING OUT THE PARAMETERS OF BEST MODELS
C**************************************************
          WRITE (3,67)
67        FORMAT(4X,'COEFFICIENTS:')
          DO 94 J=1,NB
94        WRITE(3,69) (PS(I,J),I=1,N)
69        FORMAT (8F10.3)
          WRITE (3,95)
95        FORMAT (4X,'MSE AFTER ADAPTATION')
          WRITE (3,68) (OS(K),K=1,NB)
68        FORMAT (2X,5E12.3)
          WRITE (3,87)
87        FORMAT (4X,'ERROR ON THE EXAMIN SET')
          WRITE (3,68) (OA (K),K=1,NB)
          RETURN
          END
C
C
          SUBROUTINE FMAX(G,JE,C,M)
          DIMENSION G(16)
          C=G(1)
          M=1
          I=2
20        IF (C-G(I))21,22,22
21        C=G(I)
          M=I
22        I=I+1
```

```
      IF(I-JE)20,20,23
23    RETURN
      END
C
      SUBROUTINE PAP(ID,N,N1,IS,IS1,FS,P,IA)
      DIMENSION ID(15),FS(15,16),P(15),IA(15)
      DIMENSION QN(15,16),R(15)
      K=0
      DO 34 I=1,N
      P(I)=0.0
      IF (ID(I)) 35,34,35
35    K=K+1
      IA(K)=I
      QN(K,IS1)=FS(I,N1)
34    CONTINUE
      DO 36 I=1,IS
      DO 36 J=1,IS
      L1=IA(I)
      L2=IA(J)
36    QN(I,J)=FS(L1,L2)
      CALL GAUSS(QN,IS,IS1,R)
      DO 37 K=1,IS
      L3=IA(K)
37    P(L3)=R(K)
      RETURN
      END
C
      SUBROUTINE DICH(JQ,ID,JN,JS)
      DIMENSION ID(15)
      REAL JQ,JL
      JL=JQ
      DO 11 I=1,JN
11    ID(I)=0
      IF(JS-1)15,19,15
15    I=0
      JN1=JN+1
16    I=I+1
      IF(JS-JL)17,17,18
17    JC=JL/JS
      L1=JN1-I
      ID(L1)=JL-JC*JS
      JL=JC
      GO TO 16
18    L2=JN1-I
      ID(L2)=JL
19    RETURN
      END
C
      SUBROUTINE FORD(ICT,L,M,N,IP)
      REAL IC
      DIMENSION IP(15)
      COMMON /XYUD/X(100,15),Y(100),UD(100,15)
      WRITE (3,24)
24    FORMAT(4X,'STRUCTURE OF THE FULL POLYNOMIAL')
      IC=0.0
      JF=0
      ICT1=ICT+1
25    CALL DICH(IC,IP,L,ICT1)
      IC=IC+1
      IS=0
```

```
      DO 26 J1=1,L
26    IS=IS+IP(J1)
      IF(IS-ICT)27,27,25
27    JF=JF+1
28    FORMAT(5X,17I3)
      WRITE(3,28)(IP(J),J=1,L)
      DO 32 I=1,M
      UD(I,JF)=1.0
      IF(JF-1)32,32,81
81    DO 31 J=1,L
      IF(IP(J))31,31,82
82    UD(I,JF)=UD(I,JF)*X(I,J)**IP(J)
31    CONTINUE
32    CONTINUE
      IF(IP(1)-ICT)25,30,30
30    RETURN
      END
C
      SUBROUTINE NOS(N,N1,ML,MB,M1,FS)
      DIMENSION FS(15,16)
      COMMON /XYUD/X(100,15),Y(100),UD(100,15)
      DO 31 I=1,N
      FS(I,N1)=0.0
      DO 31 J=MB,M1
31    FS(I,N1)=FS(I,N1)+UD(J,I)*Y(J)
      DO 32 I1=1,N
      DO 32 J1=1,N
      FS(I1,J1)=0.0
      DO 32 K=MB,M1
32    FS(I1,J1)=FS(I1,J1)+UD(K,I1)*UD(K,J1)
      RETURN
      END
C
      SUBROUTINE RANG(X,NP,N)
      DIMENSION X(100),XD(100)
      INTEGER NP(100),ND(100)
      DO 1 I=1,N
      XD(I)=X(I)
1     ND(I)=I
      N1=N
2     CALL FMAX(XD,N1,XM,K)
      NP(N1)=ND(K)
      K1=K+1
      DO 3 I=K1,N1
      XD(I-1)=XD(I)
3     ND(I-1)=ND(I)
      N1=N1-1
      IF (N1.GE.2)GO TO 2
      NP(1)=ND(1)
      RETURN
      END
C
   SUBROUTINE GAUSS(A,N,L,X)
   DIMENSION A(15,16),X(15)
   L=N+1
   NN=N-1
   DO 88 K=1,NN
   J=K
   KK=K+1
   DO 100 I=KK,N
```

```
          IF(ABS(A(J,K)).LT.ABS(A(I,K)))J=I
  100     CONTINUE
          IF(J.EQ.K)GOTO 11
          DO 300 I=1,L
          T=A(K,I)
          A(K,I)=A(J,I)
          A(J,I)=T
  300     CONTINUE
   11     DO 88 J=KK,N
          IF(A(K,K).EQ.0.)GOTO 600
          D=-A(J,K)/A(K,K)
          DO 88 I=1,L
          A(J,I)=A(J,I)+D*A(K,I)
   88     CONTINUE
          IF(A(N,N).EQ.0.)GOTO 600
          X(N)=A(N,L)/A(N,N)
          NN=N-1
          DO 500 J=1,NN
          K=N-J
          SUM=0.0
          NNN=N-K
          DO 200 JJ=1,NNN
          M=K+JJ
          SUM=SUM+A(K,M)*X(M)
  200     CONTINUE
          IF(A(K,K).EQ.0.)GOTO 600
          X(K)=(A(K,L)-SUM)/A(K,K)
  500     CONTINUE
  600     RETURN
          END
```

2.2 Sample outputs

Example.

I. Here the case of multivariate data is considered. The output data is generated from the equation:

$$y = 0.433 - 0.095\,x_1 + 0.243\,x_2 + 0.35\,x_1^2 - 0.18\,x_1\,x_2 + \epsilon,$$

where x_1, x_2 are randomly generated input variables y is the output variable, and ϵ is the noise added to the data. The "input.dat" file is arranged for 100 measured points with the values of $y, x_1, x_2, x_1^2, x_2^2, x_1x_2$.

y	x_1	x_2	x_1^2	x_2^2	x_1x_2

The initial control parameters of the program are fed through the terminal as it asks inputting the values, starting with

```
  GIVE TOTAL DISCRETE POINTS
100

  TIME SERIES (1)/MULTIVARIATE DATA (2)??
2

  GIVE NO.OF INPUT VARIABLES??
5
```

```
 GIVE NO.OF TRAINING PTS??
30

 GIVE NO.OF TESTING PTS??
15

 DATA SPLITTING BY (-1 DISP, 0 ALTER, 1 SEQUEN)??
1

 GIVE ORDER OF THE MODEL??
1
```

Then it on the screen displays information to the user on how to feed further information:

```
    NO.OF TERMS IN FULL MODEL = 6
    NO.OF PARTIAL MODELS = 63
```

The user has to feed further data such as the number of optimal models to be selected and the selection criterion to be used.

```
 NO.OF OPTIMAL MODELS (NB)??
8

 GIVE SELECT CRIT (1-REGUL, 2-MINBIAS, 3-COMBINED)?
1
```

The output is written in a file "results.dat" given here:

```
        SINGLE  L A Y E R E D  COMBINATORIAL ALGORITHM

 TOTAL NO.OF DATA PTS. =100

 MODEL ORDER (IT)=  1
 NO INPUT VAR.(L)=  5
 TOTAL NO.PTS.(M)=100
 N0.PTS.TESTSET(MP)= 15
 NO.PTS.EXAM.SET (MA1)=  5

   NO.TERMS IN FULL MODEL=  6
   NO.PARTIAL MODELS=          63.

 NO OF SELECT MODELS =  8
   STRUCTURE OF THE FULL POLYNOMIAL
      0   0   0   0   0
      0   0   0   0   1
      0   0   0   1   0
      0   0   1   0   0
      0   1   0   0   0
      1   0   0   0   0

   SORTING OUT BY REGULARITY CRITERION
   DEPTH OF THE MINIMUM
    0.647E-04   0.652E-04   0.219E-02   0.364E-02   0.352E-02
    0.219E-02   0.394E-02   0.409E-02
```

```
COEFFICIENTS:
 0.434     -0.180      0.000      0.350      0.243     -0.095
 0.434     -0.180      0.000      0.350      0.243     -0.095
 0.417     -0.192      0.005      0.266      0.242      0.000
 0.442      0.000      0.000      0.174      0.161      0.000
 0.437      0.000     -0.030      0.173      0.190      0.000
 0.416     -0.191      0.000      0.265      0.247      0.000
 0.458      0.000     -0.033      0.293      0.196     -0.127
 0.463      0.000      0.000      0.292      0.163     -0.126
MSE AFTER ADAPTATION
 0.469E-03   0.470E-03   0.116E-01   0.306E-01   0.303E-01
 0.116E-01   0.260E-01   0.264E-01
ERROR ON THE EXAMIN SET
 0.516E-03   0.527E-03   0.901E-02   0.268E-01   0.266E-01
 0.900E-02   0.182E-01   0.182E-01
```

The STRUCTURE OF THE FULL POLYNOMIAL helps to read the coefficients in order. For example, the first row indicates the constant term; the second row which contains 1 at the fifth column indicates that the second coefficient corresponds to the fifth variable; similarly, the third row for the fourth variable, and so on until the last row indicates the coefficient of first variable.

The COEFFICIENTS are given for eight optimal models; they are given according to the order of STRUCTURE OF THE FULL POLYNOMIAL as $a_0, a_5, a_4, a_3, a_2,$ and a_1. The DEPTH OF THE MINIMUM for regularity criterion, MSE AFTER ADAPTATION, and ERROR ON THE EXAMIN SET are given for each model in the order. The first model is the best one among all; this is read as

$$y = 0.434 - 0.180x_1x_2 + 0.0x_2^2 + 0.350x_1^2 + 0.243x_2 - 0.095x_1 \tag{8.7}$$

II. The above example can also be solved alternatively by forming the "input.dat" with the variables $y, x_1,$ and x_2 as

y	x_1	x_2

.

The control parameter values are the same as above, except the number of variables and the value of the order of the model which must be fed as

```
  GIVE NO.OF INPUT VARIABLES??
2

  GIVE ORDER OF THE MODEL??
2
```

Then the output in "results.dat" is shown below:

```
        SINGLE  L A Y E R E D  COMBINATORIAL ALGORITHM

  TOTAL NO.OF DATA PTS.  =100

  MODEL ORDER  (IT)=  2
  NO INPUT VAR.(L)=  2
  TOTAL NO.PTS.(M)=100
  NO.PTS.TESTSET(MP)= 15
  NO.PTS.EXAM.SET  (MA1)=  5

    NO.TERMS IN FULL MODEL=  6
```

```
NO.PARTIAL MODELS=        63.

NO OF SELECT MODELS =  8
  STRUCTURE OF THE FULL POLYNOMIAL
     0  0
     0  1
     0  2
     1  0
     1  1
     2  0

  SORTING OUT BY REGULARITY CRITERION
  DEPTH OF THE MINIMUM
   0.364E-02   0.646E-04   0.219E-02   0.651E-04   0.394E-02
   0.352E-02   0.409E-02   0.219E-02
  COEFFICIENTS:
   0.442      0.161      0.000      0.000      0.000      0.174
   0.434      0.243      0.000     -0.095     -0.180      0.350
   0.417      0.242      0.005      0.000     -0.192      0.266
   0.434      0.243      0.000     -0.095     -0.180      0.350
   0.458      0.196     -0.033     -0.127      0.000      0.293
   0.437      0.190     -0.030      0.000      0.000      0.173
   0.463      0.163      0.000     -0.126      0.000      0.292
   0.416      0.247      0.000      0.000     -0.191      0.265
  MSE AFTER ADAPTATION
   0.306E-01   0.469E-03   0.116E-01   0.470E-03   0.260E-01
   0.303E-01   0.264E-01   0.116E-01
  ERROR ON THE EXAMIN SET
   0.268E-01   0.516E-03   0.901E-02   0.527E-03   0.182E-01
   0.266E-01   0.182E-01   0.900E-02
```

Notice the change in the order of the coefficients. The first row of the STRUCTURE OF THE POLYNOMIAL indicates that the first coefficient term is the constant term; the second row indicates that the second coefficient term corresponds to the variable x_2; the third row indicates that the third coefficient term corresponds to the variable x_2^2; the fourth row indicates that the fourth coefficient term corresponds to the variable x_1; the fifth row corresponds to the variable x_1x_2; and the sixth row indicates the variable x_1^2. The second model is the best optimal model among the eight models; this is read as

$$y = 0.434 + 0.243x_2 + 0.0x_2^2 - 0.095x_1 - 0.180x_1x_2 + 0.350x_1^2. \tag{8.8}$$

3 COMPUTATIONAL ASPECTS OF HARMONICAL ALGORITHM

This is used mainly to identify the harmonical trend of oscillatory processes [127]. It is assumed that the effective reference functions of such processes are in the form of a sum of harmonics with nonmultiple frequencies. This means that the harmonical function is formed by several sinusoids with arbitrary frequencies which are not necessarily related.

Let us suppose that function $f(t)$ is the process having a sum of m harmonic components with distinct frequencies $w_1, w_2, \cdots, w_m$.

$$f(t) = \mathcal{A}_0 + \sum_{k=1}^{m}[\mathcal{A}_k \sin(w_kt) + \mathcal{B}_k \cos(w_kt)], \tag{8.9}$$

where $\mathcal{A}_0$ is the constant term; $\mathcal{A}_k$ and $\mathcal{B}_k$ are the coefficients; and $w_i \neq w_j,\ i \neq j,\ 0 < w_i < \pi,\ i = 1, 2, \cdots, m$. The process has discrete data points of interval length of N $(1 < t < N)$.

A balance relation is derived using the trigonometric properties for a fixed point i and any p;

$$\sum_{p=0}^{m-1} \mu_p \left[f(i+p) + f(i-p)\right] = f(i+m) + f(i-m), \tag{8.10}$$

where $\mu_0, \mu_1, \cdots, \mu_{m-1}$ are the weighing coefficients. This is considered a balance relation of the process and is used as an objective function

$$b_i = \left[f(i+m) + f(i-m)\right] - \sum_{p=0}^{m-1} \mu_p \left[f(i+p) + f(i-p)\right]. \tag{8.11}$$

If the process is expressed exactly in terms of a given sum of harmonic components, then $b_i = 0$; i.e., the discrete values of $f(t)$ which are symmetric with respect to a point $i\,(m+1 \leq i \leq N-m)$ satisfy the balance relation. The coefficients μ_p are independent of i. It is possible to determine uniquely the coefficients $\mu_p,\ p = 0, 1, \cdots, m-1$ from the balance relation for $i = m+1, \cdots, N-m$. $(N-m) - (m+1) \geq m-1$; i.e., $N \geq 3m$.

The standard trigonometric relation which is used in deriving the balance relation,

$$\mu_0 + \sum_{k=1}^{m-1} \mu_p \cos(pw_k) = \cos(mw_k) \tag{8.12}$$

helps in obtaining the frequencies w_k. This could be formed as mth degree algebraic equation in $\cos w$:

$$\mathcal{D}_m(\cos w)^m + \mathcal{D}_{m-1}(\cos w)^{m-1} + \cdots + \mathcal{D}_1(\cos w) + \mathcal{D}_0 = 0, \tag{8.13}$$

where $\mathcal{D}_i,\ i = 0, 1, \cdots, m$ are the functions of μ_p.

Substituting the values of μ_p, the above equation can be solved for m frequencies w_k of harmonics by using the standard numerical techniques. Various combinations of the harmonic components are formed with the frequencies w_k. The coefficients $\mathcal{A}_0, \mathcal{A}_k$, and $\mathcal{B}_k$ are estimated for each combination by using the least-squares technique. The best combination as an optimal trend is selected according to the value of the balance criterion.

The algorithm functions as below:

The discrete data is to be supplied as training set A and testing set B; one can allot a separate checking set C for examining the final optimal trend; i.e., $N = N_A + N_B + N_C$. The maximum number of harmonics is chosen as M_{max} ($< N/3$). The coefficients μ_p are estimated by using the least squares technique by forming the balance equations with the training set. The system of equations has the form:

$$\sum_{p=0}^{m-1} \mu_p \left[y(i+p) + y(i-p)\right] = y(i+m) + y(i-m);$$

$$i = m+1, \cdots, N_A - m. \tag{8.14}$$

By substituting the values of μ_p in the above mth order polynomial in $\cos w$, the frequencies are estimated; the m roots of the polynomial uniquely determine the m frequencies w_k. These frequencies are fed through the input layer of multilayer structure where the complete sifting of harmonic trends would take place according to the inductive principle of self-organization. This is done by a successive increase in the number of terms of the harmonic components $m = 1, m = 2, m = 3, \cdots$ until $m = M_{max}$. The linear normal equations are constructed in the first layer for any $1 < m < M_{max}$ number of harmonics. The coefficients $\mathcal{A}_0, \mathcal{A}_k$, and $\mathcal{B}_k$ are estimated for all the combinations based on the training set using the least squares technique; the balance functions are then evaluated. The best trends are selected. The output error residuals of the best trends are fed forward as inputs to the second layer. This procedure is repeated in all subsequent layers. The complexity of the model increases layer by layer as long as the value of the "imbalance" decreases. The optimal trend is the total combination of the harmonical components obtained from the layers. The performance of the optimal trend is tested on the checking set **C**.

The program listing and sample outputs for an example are given below.

3.1 Program listing

```
C
C***********************************************************************
C THIS PROGRAM IS THE RESULT OF EFFORTS FROM VARIOUS GRADUATE STUDENTS
C  AND RESEARCH PROFESSIONALS AT THE COMBINED CONTROL SYSTEMS GROUP OF
C  INSTITUTE OF CYBERNETICS, KIEV (UKRAINE)
C***********************************************************************
C
C     HARMONICAL INDUCTIVE LEARNING ALGORITHM
C
C
C       N - NO.OF TRAINING SET POINTS
C       NP - NO.OF TEST SET POINTS
C       NE - NO.OF EXAMIN SET POINTS
C       PT - NO.OF PREDICTION POINTS
C       JFM - MAX NO.OF FREQUENCIES
C       JF - FREEDOM OF CHOICE
C       NRM - NO.OF SERIES IN HARMONICAL TREND
C       NN = N+NP+NE
C       NPT = NN+PT
C       G(NN) - DISCRETE SIGNAL DATA
C       APR(NPT) - HARMONICAL MODEL VALUES
C       MA - NO.OF LAG POINTS FOR SMOOTHING PROCEDURE(MOVING AVERAGE
C             VALUE). IF IT IS ONE, DATA REMAINS SAME
C
C***********************************************************************
C   MAIN PROGRAM
C
        INTEGER PT
        DIMENSION GY(120)
        COMMON /AB/G(120)
C
          OPEN(3,FILE='output.dat')
          OPEN(8,FILE='ts.dat')
C
        WRITE(3,4)
4       FORMAT(5X,'       L A Y E R E D  HARMONICAL ALGORITHM'/)
C
        WRITE(*,110)
110      FORMAT(3X,'GIVE NO.OF TRAIN, TEST & EXAM PTS?')
        READ(*,*)N,NP,NE
        NN=N+NP+NE
        WRITE(*,112)
```

```
112      FORMAT(3X,'GIVE NO.OF PRED PTS??')
        READ(*,*)PT
        NPT=NN+PT
        READ(8,*)(G(I),I=1,NN)
        FAX=G(1)
        DO 5 I=2,NN
        IF(G(I).GT.FAX)FAX=G(I)
  5     CONTINUE
         DO 6 I=1,NN
         G(I)=G(I)/FAX
  6      CONTINUE
        WRITE(*,222)
222     FORMAT(3X,'GIVE MOVING AVERAGE VALUE (=1 or >1)?')
        READ(*,111)MA
111     FORMAT(I2)
        WRITE(*,333)
333     FORMAT(3X,'HOW MANY SERIES?')
        READ(*,111)NRM
        WRITE(*,114)
114     FORMAT(3X,'GIVE MAX NO.OF FREQS(<=15)??')
        READ(*,*)JFM
        JF2=2*JFM+2
        WRITE(*,115)
115     FORMAT(3X,'GIVE FREEDOM OF CHOICE(< MAX FREQS)??')
        READ(*,*)JF
        SMA=0.0
        DO 7 I=1,MA
  7     SMA=SMA+G(I)
        SMA=SMA/MA
        GY(1)=SMA
        IX=1
        MHR=MA+1
        DO 8 I=MHR,NN
        IX=IX+1
        IX1=IX-1
        IMA=I-MA
        GY(IX)=GY(IX1)+(G(I)-G(IMA))/MA
  8     CONTINUE
        DO 9 I=1,IX
        G(I)=GY(I)
  9     CONTINUE
        CALL HARMAN(N,NP,NE,NN,PT,JF,JFM,NRM,0,1,JF2,NPT)
        STOP
        END
C
```

Subroutines used

```
        SUBROUTINE WB(N1,M,M1,IER,KA)
        COMMON /BC/X(160),Y(160),Y1(31),Y2(31),A(31),C(31,32),W(15)
        N=N1-2
         M1=M+1
        NM=N-M
        DO 1 I=1,N
 1      Y(I)=X(I+2)-X(I)
        DO 2 J=1,M1
        Y1(J)=0.0
        DO 2 I=1,M
 2      C(I,J)=0.0
```

```
      DO 3 I=M1,NM
      K=I-M
      R=K
      E=1.0/R
      DO 4 J=1,M1
      I1=I+J-1
      I2=I-J+1
      Y2(J)=Y1(J)+Y(I1)+Y(I2)
      IF(KA-0)4,10,4
10    Y2(J)=Y2(J)-Y1(J)
 4    Y1(J)=Y2(J)
 8    DO 5 K1=1,M
      DO 5 J=K1,M1
      E1=Y2(K1)*Y2(J)
      IF(KA-2)5,11,5
11    E1=E1*E
 5    C(K1,J)=C(K1,J)+E1
      IF(KA-2)3,12,12
12    K=K-1
      IF(K-0)13,3,13
13     DO 7 J=1,M1
      I1=I+J-1-K
      I2=I-J+1-K
 7     Y2(J)=Y2(J)-Y(I1)-Y(I2)
      GOTO 8
 3     CONTINUE
      IF(M-1)14,77,14
14    DO 6 I=2,M
      I1=I-1
      DO 6 J=1,I1
 6    C(I,J)=C(J,I)
77    CALL GAUSS(C,M,M1,A,IER)
      RETURN
      END
C
      SUBROUTINE COEF(M,N,IER)
      COMMON /BC/Y(160),Y1(160),WK(31),B(31),A(31),HM(31,32),W(15)
      K=2*M
      K1=K+1
      DO 1 I=1,K1
      HM1=0.0
      IF(I-K)2,2,3
 2    AI=I
      BI=(AI+1.25)/2.
      II=INT(BI)
      BI=(AI+0.1)/2.
      AI=INT(BI)
      TI=BI-AI
      DO 4 J=I,K
      AJ=J
      BJ=(AJ+1.25)/2.
      JJ=INT(BJ)
      BJ=(AJ+0.1)/2.
      AJ=INT(BJ)
      TJ=BJ-AJ
      W1=W(II)-W(JJ)
      W2=W(II)+W(JJ)
      IF(II-JJ)6,5,6
 5    IF(ABS(TI-TJ)-0.01)8,30,30
30    S1=0.0
```

```
      GOTO 9
 8    S1=N
      GOTO 9
 6    AN=N
      CN=AN*W1/2.
      BN=W1/2.
      S1=SIN(CN)/SIN(BN)
 9    AN=N
      CN=AN*W2/2.
      BN=W2/2.
      S2=SIN(CN)/SIN(BN)
      AN=N+1
      BN=AN*W1/2.
      CN=AN*W2/2.
      CN1=COS(BN)
      CN2=COS(CN)
      SN1=SIN(BN)
      SN2=SIN(CN)
      IF(TI-0.25)11,10,10
10    IF(TJ-0.25)13,12,12
12    HM(I,J)=S1*CN1-S2*CN2
      GOTO 40
13    HM(I,J)=S2*SN2+S1*SN1
      GOTO 40
11    IF(TJ-0.25)15,14,14
14    HM(I,J)=S2*SN2-S1*SN1
      GOTO 40
15    HM(I,J)=S1*CN1+S2*CN2
40    HM(I,J)=0.5*HM(I,J)
 4   CONTINUE
      IF(TI-0.25)17,16,16
16    Y1(1)=SIN(W(II))
      Y1(2)=SIN(2*W(II))
      GOTO 18
17    Y1(1)=COS(W(II))
      Y1(2)=COS(2*W(II))
18    WK1=COS(W(II))
      DO 19 J=3,N
      Y1(J)=2.*WK1*Y1(J-1)-Y1(J-2)
19    HM1=HM1+Y1(J)*Y(J)
      HM(I,K+2)=HM1+Y1(1)*Y(1)+Y1(2)*Y(2)
      IF(TI-0.25)21,20,20
20    AN=N+1
      AN=AN*W(II)/2.
      H1=SIN(AN)
      GOTO 22
21    AN=N+1
      AN=AN*W(II)/2.
      H1=COS(AN)
22    AN=N
      BN=W(II)/2.
      CN=AN*BN
      HM(I,K1)=H1*SIN(CN)/SIN(BN)
      GOTO 24
 3   HM(I,K1)=N
      H1=0.
      DO 23 J=1,N
23    H1=H1+Y(J)
      HM(I,K+2)=H1
24    IF(I-2)1,25,25
```

```
25      I1=I-1
        DO 26 J=1,I1
26      HM(I,J)=HM(J,I)
1      CONTINUE
        K11 = K1+1
        CALL GAUSS(HM,K1,K11,B,IER)
        RETURN
        END
C
        SUBROUTINE WB1(M,M1,IER)
        COMMON /BC/YB(160),AP(160),WK(31),B(31),A(31),C(31,32),W(15)
        M1 = M+1
        DO 1 I=1,M
        DO 1 J=1,M1
        AJ=J-1
        AJ=AJ*W(I)
1       C(I,J)=COS(AJ)
        CALL GAUSS(C,M,M1,A,IER)
        RETURN
        END
C
        SUBROUTINE RANG(N,B)
        DIMENSION B(15)
        DO 1 I=1,N
        I1=I+1
        IF(I1-N)7,7,3
7       DO 1 J=I1,N
        IF(B(I)-B(J))1,1,2
2       R=B(I)
        B(I)=B(J)
        B(J)=R
1       CONTINUE
3       RETURN
        END
C
C
        SUBROUTINE HARMAN(N,NP,NE,NN,PT,F,FM,NRM,KA,IP,F2,NPT)
        INTEGER F,FM,PT,F2
        REAL IB(6)
        DIMENSION IST(6),PA(15,120),PA1(15,120),APR(160)
        COMMON /AB/G(120)
        COMMON /TIN/TIN(15,48)
        COMMON /BC/YB(160),AP(160),WK(31),B(31),A(31),C(31,32),W(15)
C
100     FORMAT(//)
101     FORMAT(5X,'FREEDOM OF CHOICE',I3/)
102     FORMAT(5X,'MAX NO.OF FREQUENCIES',I3/)
103     FORMAT(5X,'MAX.NO.OF SERIES',I3/)
104     FORMAT(5X,'LENGTH OF EXAMINING SET (C)',I4/)
105     FORMAT(5X,'LENGTH OF TESTING SET (B)',I4/)
106     FORMAT(5X,'LENGTH OF TRAINING SET (A)',I4/)
107     FORMAT(5X,'NO.OF PREDICTION POINTS',I4/)
109     FORMAT(/)
110     FORMAT(2X,7F11.3)
111     FORMAT(2X,'TIME SERIES')
112     FORMAT(10X,'OPTIMAL TREND',/,10X,'------- -----')
113     FORMAT(3X,'SERIES',I3)
114     FORMAT(3X,'NO.OF FREQUENCIES',I3)
115     FORMAT(3X,'FREE TERM',F13.5)
116     FORMAT(3X,'FREQ',12X,'COEFFS A',9X,'COEFFS B',8X,'AMPLITUDE')
```

```
117     FORMAT(F10.7,3F17.6)
118     FORMAT(2X,'ACTUAL VALUES:')
119     FORMAT(5F16.6)
120     FORMAT(2X,'ESTIMATED VALUES:')
121     FORMAT(5X,'PREDICTED VALUES:')
122     FORMAT(I8,2F28.5)
123     FORMAT(I8,F53.5)
124     FORMAT(/)
127     FORMAT(11X,'NO CORRECT DECISION')
C
        NK=N+NP+NE
        N1=N+NP
         NKT=NK+PT
        PI=3.1415926535/2.
        WRITE(3,100)
        WRITE(3,106)N
        WRITE(3,105)NP
        WRITE(3,104)NE
        WRITE(3,102)FM
        WRITE(3,101)F
        WRITE(3,107)PT
        WRITE(3,103)NRM
        WRITE(*,109)
        WRITE(*,111)
        WRITE(*,110)(G(I),I=1,NN)
        NR=1
1       IT=0
2       IT=IT+1
        M=0
3       M=M+1
        MP=2*M
        DO 4 I=1,NK
        IF(NR-1)6,6,5
6       YB(I)=G(I)
        GOTO 4
5       YB(I)=PA(IT,I)
4       CONTINUE
        CALL WB(N,M,M1,IER,KA)
        IF(IER)77,998,77
77      CALL FRIQ(M,M1)
        DO 7 J=1,M
        AN=1.-WK(J)**2
        BN=WK(J)/SQRT(AN)
7       W(J)=PI-ATAN(BN)
        CALL WB1(M,M1,IER)
        IF(IER)78,999,78
78      CALL COEF(M,N,IER)
        IF(IER)79,997,79
79      B1=0.
        B2=0.
        B3=0.
        D1=0.
         D2=0.
         D3=0.
        M1=M+1
        NKM=NK-M
        DO 11 I=M1,NKM
        R=0.
        DO 12 J=1,M
        I1=I+J-1
```

```
      I2=I-J+1
12    R=R+A(J)*(YB(I1)+YB(I2)-2*B(MP+1))
      IM=I+M
       MI=I-M
      R=(YB(IM)+YB(MI)-R-2*B(MP+1))**2
      IF(I-(N-M))80,80,13
80    B1=B1+R
      GOTO 11
13    IF(I-(N1-M))81,81,14
81    B2=B2+R
      GOTO 11
14    B3=B3+R
11    CONTINUE
      AN=N-MP
      BN=B1/AN
      IB(1)=SQRT(BN)
      AN=NP
      BN=B2/AN
      IB(2)=SQRT(BN)
      DO 15 I=1,MP
      R=0.0
      DO 16 J=1,M
      AI=I
      D=W(J)*AI
      J2 = 2*J
      J21 = J2-1
16    R=R+B(J21)*SIN(D)+B(J2)*COS(D)
      D1=D1+(YB(I)-B(MP+1)-R)**2
15   AP(I)=R
      DO 17 I=M1,NKM
      I1=I-M
      R=-AP(I1)
      DO 18 J=1,M
      I1=I+J-1
      I2=I-J+1
18    R=R+A(J)*(AP(I1)+AP(I2))
      I2=I+M
      AP(I2)=R
      D=(YB(I2)-R-B(MP+1))**2
      IF(I2-N)82,82,19
82    D1=D1+D
      GOTO 17
19    IF(I2-N1)83,83,20
83    D2=D2+D
      GOTO 17
20    D3=D3+D
17    CONTINUE
      AN=N
      BN=D1/AN
      IB(4)=SQRT(BN)
      AN=NP
      BN=D2/AN
      IB(5)=SQRT(BN)
      IF(NE)21,21,22
21    IB(3)=0.
      IB(6)=0.
      GOTO 23
22    IB(3)=SQRT(B3/NE)
      IB(6)=SQRT(D3/NE)
23    IF(IT-1)25,84,25
```

```
84      IF(M-F)24,24,25
24      KP=(NR-1)*8+1
        IF(NR-1)26,26,27
26      TIN(M,KP)=0.
        GOTO 28
27      TIN(M,KP)=IT
28      TIN(M,KP+1)=M
        DO 29 I=1,6
        KS=KP+1+I
29      TIN(M,KS)=IB(I)
        DO 30 I=1,NK
30      PA1(M,I)=YB(I)-AP(I)-B(MP+1)
        GOTO 34
25      R=0.
        IZ=0
        DO 31 I=1,F
        KP=(NR-1)*8+IP+2
        D=TIN(I,KP)
        IF(R-D)85,85,31
85      R=D
        IZ=I
31      CONTINUE
55      IF(R-IB(IP))34,34,86
86      DO 32 I=1,NK
32      PA1(IZ,I)=YB(I)-AP(I)-B(MP+1)
        KP=(NR-1)*8+1
        DO 33 I=1,6
        KS=KP+1+I
33      TIN(IZ,KS)=IB(I)
        TIN(IZ,KP)=IT
        TIN(IZ,KP+1)=M
        IF(NR-1)34,87,34
87      TIN(IZ,KP)=0.0
34      IF(M-FM)3,88,88
88      IF(NR-1)89,35,89
89      IF(IT-F)2,35,35
35      CALL PRI(NR,IP,F)
        NR=NR+1
        DO 136 J=1,F
        DO 136 I=1,NK
136     PA(J,I)=PA1(J,I)
        IF(NR-NRM)1,1,90
90      WRITE (3,100)
        WRITE (3,112)
        IZ=1
        NR=1
        P1=TIN(1,IP+2)
        DO 36 I=1,NRM
        KS=(I-1)*8+IP+2
        DO 36 J=1,F
        D=TIN(J,KS)
        IF(D-P1)91,36,36
91      NR=I
        P1=D
        IZ=J
36      CONTINUE
        KP=(NR-1)*8+2
        IST(NR)=TIN(IZ,KP)
        I1=NR-1
        IF(I1)92,382,92
```

```
92      CONTINUE
        DO 37 I=1,I1
        I2=NR-I
        KS=I2*8+1
        IZ=TIN(IZ,KS)
        KS=(I2-1)*8+2
37      IST(I2)=TIN(IZ,KS)
382     DO 38 I=1,NKT
        APR(I)=0.0
        IF(I-NK)39,39,40
39      YB(I)=G(I)
        GO TO 38
40      YB(I)=0.0
38      CONTINUE
381     IZ=1
41      M=IST(IZ)
        MP=2*M
        CALL WB(N,M,M+1,IER,KA)
        IF(IER)999,999,42
42      CALL FRIQ(M,M+1)
        DO 43 J=1,M
        AN=1.0-WK(J)**2
        BN=WK(J)/SQRT(AN)
43      W(J)=PI-ATAN(BN)
        CALL RANG(M,W)
        CALL WB1(M,M+1,IER)
        IF(IER)93,998,93
93      CONTINUE
        CALL COEF(M,N,IER)
        IF(IER)94,997,94
94      CONTINUE
        WRITE(3,113)IZ
        WRITE(3,115)B(MP+1)
        WRITE(3,114)M
        WRITE(3,116)
        DO 46 I=1,M
        I2 = I*2
        I21 = I2-1
        BN=B(I21)**2+B(I2)**2
        P1=SQRT(BN)
46      WRITE(3,117)W(I),B(I21),B(I2),P1
        DO 47 I=1,MP
        R=0.0
        DO 48 J=1,M
        AI=I
        D=W(J)*AI
        J2 = J*2
        J21 = J2-1
48      R=R+B(J21)*SIN(D)+B(J2)*COS(D)
        APR(I)=APR(I)+R+B(MP+1)
47      AP(I)=R
        M1=M+1
        NKM=NK+PT-M
        DO 53 I=M1,NKM
        I1=I-M
        R=-AP(I1)
        DO 49 J=1,M
        IJ1=I+J-1
        IJ2=I-J+1
49      R=R+A(J)*(AP(IJ1)+AP(IJ2))
```

```
      I2=I+M
      AP(I2)=R
53    APR(I2)=APR(I2)+AP(I2)+B(MP+1)
      DO 50 I=1,NK
50    YB(I)=YB(I)-AP(I)-B(MP+1)
      IZ=IZ+1
      IF(IZ-NR)41,41,95
95    CONTINUE
      WRITE(3,100)
      WRITE(3,118)
      WRITE(3,110)(G(I),I=1,NN)
      WRITE(3,109)
      WRITE(3,120)
      WRITE(3,110)(APR(I),I=1,NN)
      GM=0.0
      DO 54 IH=1,NN
      GM=GM+G(IH)
54    CONTINUE
      GM=GM/NN
      CN=0.0
      CD=0.0
      DO 10 IH=1,NN
      CK=G(IH)-APR(IH)
      CN=CN+CK**2
10    CD=CD+(G(IH)-GM)**2
      CK=SQRT(CN/CD)
      WRITE(3,133)CK
133   FORMAT(/5X,'RESIDUAL SUM OF SQUARES =',5X,E18.7/)
      WRITE(3,100)
      WRITE(3,121)
      I1=N1+1
      I2=NK+PT
      DO 51 I=I1,I2
      IF(I-NK)96,96,52
96    CONTINUE
      WRITE(3,122)I,G(I),APR(I)
      GO TO 51
52    WRITE(3,123)I,APR(I)
51    CONTINUE
      GO TO 1001
999   WRITE(*,124)
      GO TO 1000
998   WRITE(*,124)
      GO TO 1000
997   WRITE(*,124)
1000  WRITE(*,127)
1001  RETURN
      END
C
C
      SUBROUTINE PRI(NR,IP,F)
      INTEGER F
      DIMENSION SERV(6)
      COMMON /TIN/TIN(15,48)
10    FORMAT(//,1X,'SERIES',I2)
11    FORMAT(2X,'TRNO',2X,'FRNO',4X,'BAL A',
     1 6X,'BAL B',6X,'BAL C',6X,'ERR A',6X,'ERR B',6X,
     2 'ERR C',/)
12    FORMAT(3X,I3,2X,I4,6E11.3)
13    FORMAT(3X,'--------------------------------------------')
```

```
      K=1
      KP=(NR-1)*8+IP+2
      P=TIN(1,KP)
      IF(F-1)7,4,7
7     DO 1 I=2,F
      IF(TIN(I,KP)-P)2,1,1
2     P=TIN(I,KP)
      K=I
1     CONTINUE
4     WRITE(3,10)NR
      WRITE(3,11)
      KP=(NR-1)*8+1
      DO 3 I=1,F
      DO 5 J=1,6
      KS=KP+1+J
5     SERV(J)=TIN(I,KS)
      MT=TIN(I,KP)
      MF=TIN(I,KP+1)
      WRITE(3,12)MT,MF,(SERV(J),J=1,6)
      IF(I-K)3,6,3
6     WRITE(3,13)
3     CONTINUE
      RETURN
      END
C
C
      SUBROUTINE NEW(N1,N,MAX,EPS,EPS1)
      DIMENSION DB(31)
      COMMON /BC/YB(160),AP(160),C(31),AD(31),A(31),FI(31,32),W(15)
      DO 11 I=1,N1
11    DB(I)=AD(I)
      DO 12 I=1,N1
      I1=N1+1-I
12    AD(I1)=DB(I)
      I=N
      J=1
      N2=N1
1     IF(I-1)20,20,2
2     R=1.0
      M=0
      DO 3 I1=1,I
3     DB(I1)=(N2-I1)*AD(I1)
      F2=1.0
4     CALL FUNC(AD,N2,R,F)
      CALL FUNC(DB,I,R,F1)
      IM=M+1
      IF(ABS(F1)-EPS1)7,7,8
8     F2=F1
7     R=R-F/F2
      M=M+1
      IF(M-MAX)10,5,5
10    IF(ABS(F)-EPS)5,5,4
5     C(J)=R
      J=J+1
      DO 6 I1=1,I
6     AD(I1)=AD(I1)+AD(I1-1)*R
      I=I-1
      N2=N2-1
      GO TO 1
20    C(J)=-AD(2)/AD(1)
```

```
      RR=-AD(2)/AD(1)
      RETURN
      END
C
C
      SUBROUTINE FUNC(A,N1,R,F)
      DIMENSION A(31)
      N=N1-1
      F=A(1)
      DO 1 I=1,N
1     F=F*R+A(I+1)
      RETURN
      END
C
      SUBROUTINE FRIQ(M,M1)
      COMMON /BC/YB(160),AP(160),WK(31),CO(31),A(31),FI(31,32),W(15)
      M1 = M+1
      DO 1 I=1,M
      DO 1 J=1,M1
1     FI(I,J)=0.0
      FI(1,2)=1.0
      IF(M-1)11,27,11
11    FI(2,1)=-1.0
      FI(2,3)=2.0
      IF(M-2)2,2,12
12    DO 3 I=3,M
      I1=I+1
      DO 4 J=2,I1
4     FI(I,J)=2*FI(I-1,J-1)-FI(I-2,J)
3     FI(I,1)=-FI(I-2,1)
2     M2=M-1
      DO 5 I=1,M2
      DO 5 J=1,M1
5     FI(M,J)=FI(M,J)-FI(I,J)*A(I+1)
27    FI(M,1)=FI(M,1)-A(1)
      DO 6 I=1,M1
6     CO(I)=FI(M,I)
      EP=0.000001
      EPS2=0.000001
      EPS3=0.0001
      MAX=25
      EPS=0.000001
      EPS1=0.001
      ETA=0.00001
      DO 66 I=1,M
66    WK(I)=0
      CALL NEW(M1,M,MAX,EPS,EPS1)
      RETURN
      END
C
      SUBROUTINE GAUSS(A,N,L,X,KGA)
      DIMENSION A(31,32),X(31)
      KGA = 1
      L = N+1
      NN=N-1
      DO 99 K=1,NN
      J=K
      KK=K+1
      DO 100 I=KK,N
      IF(ABS(A(J,K)).LT.ABS(A(I,K)))J=I
```

```
 100     CONTINUE
         IF(J.EQ.K)GOTO 11
         DO 300 I=1,L
         T=A(K,I)
         A(K,I)=A(J,I)
         A(J,I)=T
 300    CONTINUE
  11     DO 88 J=KK,N
         IF(A(K,K).EQ.0.)GOTO 600
         D=-A(J,K)/A(K,K)
         DO 400 I=1,L
         A(J,I)=A(J,I)+D*A(K,I)
 400     CONTINUE
  88     CONTINUE
  99     CONTINUE
         IF(A(N,N).EQ.0.)GOTO 600
         X(N)=A(N,L)/A(N,N)
         NN=N-1
         DO 500 J=1,NN
         K=N-J
         SUM=0.0
         NNN=N-K
         DO 200 JJ=1,NNN
         M=K+JJ
         SUM=SUM+A(K,M)*X(M)
 200     CONTINUE
         IF(A(K,K).EQ.0.)GOTO 600
         X(K)=(A(K,L)-SUM)/A(K,K)
 500     CONTINUE
         GOTO 800
600      KGA = 2
         WRITE(*,700)
700      FORMAT(5X,' SINGULAR')
800      RETURN
         END
```

3.2 Sample output

Example. The time series data sample is supplied with a file "ts.dat." The data corresponds to the air-temperature data that is collected at an interval of one day. The control parameters are fed as input:

```
  GIVE NO.OF TRAIN, TEST & EXAM PTS?
45 1 1
  GIVE NO.OF PRED PTS??
5
  GIVE MOVING AVERAGE VALUE (=1 or >1)?
1
  HOW MANY SERIES?
3
  GIVE MAX NO.OF FREQS(<=15)??
8
  GIVE FREEDOM OF CHOICE(< MAX FREQS)??
7
```

One can choose the MOVING AVERAGE VALUE to smooth out the noises in the data; if it is 1, then it takes the data as it is. SERIES indicates the number of layers in the algorithm. Usually, one or two layers are sufficient to obtain the optimal trend. Even if the user chooses more number of layers, it selects the optimal trend from the layer where it achieves the global minimum of the balance relation. MAX NO.OF FREQS which has the limit of less than or equal to 15 indicates the maximum number of distinct frequencies M_{max} to be determined. FREEDOM OF CHOICE denotes the number of optimal trends to be selected at each layer.

The performance of the algorithm is given for each layer. The values of the balance function for training, testing, and examining sets (BAL A, BAL B, BAL C) and their error values (ERR A, ERR B, ERR C) are given correspondingly for each selected trend. The best trends or combinations of the freedom-of-choice are shown. The best one among them according to the balance relation on training set (BAL A) is underlined. TRNO indicates the trend number or combination number from the previous layer and FRNO indicates the number of harmonical components in the current trend. For example, the optimum trend underlined for SERIES 1 has seven frequencies (see output below). The best trend underlined for SERIES 2 has also seven (FRNO =7) harmonical components. This is based on the seventh trend or combination (TRNO =7) of the SERIES 1. Similarly, the best trend in SERIES 3 has one frequency (FRNO =1) and is based on the second trend or combination (TRNO = 2) of the SERIES 2.

The OPTIMAL TREND is collected starting from the SERIES, where the global minimum on the balance relation (BAL A) is achieved, to the first layer. For the output given below, the global minimum is achieved at the SERIES 3 with the value of BAL A equal to 0.101E+01; it has one harmonical component. This is the follow up of the second combination (TRNO = 2) of the SERIES 2. The second combination of the SERIES 2 has eight harmonical components and is the follow up of the sixth trend (TRNO = 6) of the SERIES 1. The sixth one in the SERIES 1 has six harmonic components. This means that the recollected information of the optimal trend includes six harmonical components from the SERIES 1, eight from the SERIES 2, and one from the SERIES 3 along with a FREE TERM from each SERIES; the OPTIMAL TREND is printed giving the values of the FREE TERMs, the frequencies (FREQ), and the coefficients (COEFFS A and B) at each layer along with the AMPLITUDE values. This is represented as

$$\hat{y}_t = \sum_{j=1}^{s}[A_{0j} + \sum_{k=1}^{m_j}(A_{jk}\sin(w_{jk}t) + B_{jk}\cos(w_{jk}t))], \tag{8.15}$$

where $\hat{y}_t$ is the estimated output value; s denotes the number of series in the optimal trend; $m_j, j = 1, 2, \cdots, s$ denote the number of harmonic components at each series; A_{0j} is the free term at jth SERIES; A_{jk} and B_{jk} are the estimated coefficients of the kth component of the jth SERIES; and w_{jk} are the corresponding frequency components.

ACTUAL and ESTIMATED VALUES are given for comparison and the RESIDUAL SUM OF SQUARES (RSS) is computed as

$$\text{RSS} = \sum_{i=1}^{N}\frac{(y_i - \hat{y}_i)^2}{(y_i - \bar{y})^2} \leq 1, \tag{8.16}$$

where y and $\hat{y}$ are the actual and estimated values and $\bar{y}$ is the average value of the time series.

The PREDICTED VALUES are given as specified using the optimal trend; this includes the predictions for the points N_C.

The output is written in the file "output.dat" below.

```
                    L A Y E R E D  HARMONICAL ALGORITHM
```

LENGTH OF TRAINING SET (A) 45

LENGTH OF TESTING SET (B) 1

LENGTH OF EXAMINING SET (C) 1

MAX NO. OF FREQUENCIES 8

FREEDOM OF CHOICE 7

NO. OF PREDICTION POINTS 5

MAX. NO. OF SERIES 3

SERIES 1

TRNO	FRNO	BAL A	BAL B	BAL C	ERR A	ERR B	ERR C
0	1	0.464E+01	0.620E+00	0.709E+01	0.455E+01	0.131E+01	0.365E+01
0	2	0.651E+01	0.381E+01	0.654E+01	0.427E+01	0.304E+01	0.687E+01
0	8	0.408E+01	0.149E+02	0.358E+01	0.271E+01	0.628E+01	0.950E+01
0	4	0.607E+01	0.650E+01	0.555E+01	0.419E+01	0.300E+00	0.462E+01
0	5	0.486E+01	0.994E+01	0.512E+00	0.442E+01	0.278E+01	0.548E+01
0	6	0.373E+01	0.883E+01	0.320E+01	0.354E+01	0.133E+01	0.111E+01
0	7	0.356E+01	0.121E+02	0.463E+01	0.296E+01	0.522E+01	0.588E+01

SERIES 2

TRNO	FRNO	BAL A	BAL B	BAL C	ERR A	ERR B	ERR C
7	7	0.215E+01	0.606E+01	0.360E+01	0.158E+01	0.401E+01	0.454E+01
6	8	0.236E+01	0.575E+01	0.385E+01	0.919E+00	0.443E+00	0.207E+00
7	8	0.254E+01	0.829E+01	0.338E+01	0.101E+01	0.447E+01	0.407E+01
6	7	0.252E+01	0.673E+01	0.588E+01	0.157E+01	0.275E+01	0.152E+01
3	7	0.235E+01	0.885E+01	0.203E+01	0.183E+01	0.681E+01	0.902E+01
3	5	0.261E+01	0.981E+01	0.151E+01	0.190E+01	0.842E+01	0.972E+01
7	6	0.255E+01	0.809E+01	0.313E+01	0.258E+01	0.671E+01	0.732E+01

SERIES 3

TRNO	FRNO	BAL A	BAL B	BAL C	ERR A	ERR B	ERR C
3	3	0.120E+01	0.457E+01	0.164E+01	0.909E+00	0.435E+01	0.443E+01
2	4	0.133E+01	0.236E+01	0.490E+00	0.784E+00	0.170E+00	0.929E+00
3	2	0.133E+01	0.563E+01	0.359E+01	0.971E+00	0.467E+01	0.428E+01
2	3	0.123E+01	0.171E+01	0.226E-01	0.838E+00	0.386E+00	0.150E-01
2	2	0.116E+01	0.159E+01	0.115E+01	0.874E+00	0.101E+01	0.200E+00
3	8	0.116E+01	0.456E+01	0.503E+00	0.537E+00	0.361E+01	0.256E+01
2	1	0.101E+01	0.596E-01	0.132E+01	0.902E+00	0.323E+00	0.389E+00

OPTIMAL TREND

SERIES 1
FREE TERM -0.56199
NO. OF FREQUENCIES 6

FREQ	COEFFS A	COEFFS B	AMPLITUDE
0.2369936	-1.056414	1.915627	2.187610

```
0.7902706          -2.265249          -1.351049          2.637553
1.0355266          -0.320283           1.655817          1.686509
1.8367290          -0.274392          -0.120682          0.299759
2.1455603           1.113026           0.479222          1.211809
2.5376661           0.573313          -0.212797          0.611531
  SERIES  2
  FREE TERM       -0.09219
  NO.OF FREQUENCIES   8
  FREQ              COEFFS A           COEFFS B          AMPLITUDE
0.1195246          -3.281033          -2.040643          3.863858
0.6629882           1.209835          -0.435315          1.285768
0.9145533          -1.877773          -0.696096          2.002644
1.3779728          -0.100550          -0.039555          0.108051
1.8496013          -0.052124          -0.297579          0.302110
2.0773623           0.101575           0.242814          0.263203
2.3273549           0.492773           0.068364          0.497493
2.7066665           0.342581          -0.085725          0.353144
  SERIES  3
  FREE TERM       -0.00055
  NO.OF FREQUENCIES   1
  FREQ              COEFFS A           COEFFS B          AMPLITUDE
1.8217989           0.012065           0.247733          0.248027


ACTUAL VALUES:
    -5.000    -10.000     -1.000     -1.500     -1.000      2.000     -8.500
   -12.500    -10.000     -9.000     -4.000      0.000     -0.250     -5.000
    -7.500     -8.000     -7.000     -2.000      2.000      1.000      2.000
     2.000      2.500      3.000      1.750      1.000      0.000      1.000
     4.000      8.000      6.000      2.500      1.500     -2.500     -0.250
     3.000      0.000      3.500      3.000     -0.250     -2.000      1.750
    -0.250      1.000      4.000      1.000      3.000

ESTIMATED VALUES:
    -3.638     -8.640     -1.738     -0.811     -0.339      1.102     -7.365
   -12.481    -11.739     -9.539     -4.904      1.292     -0.541     -6.119
    -7.006     -8.557     -6.686     -0.882      0.505      0.457      2.355
     1.528      2.405      4.178      2.263      0.802      1.061      1.741
     3.362      8.525      5.694      1.881      2.656     -3.908     -0.646
     3.155      0.691      3.938      1.978     -0.462     -0.409     -0.207
    -0.136      1.278      2.909      1.323      2.611

  RESIDUAL SUM OF SQUARES =          0.1963205E+00


 PREDICTED VALUES:
     47                     3.00000                  2.61100
     48                                              0.34542
     49                                             -2.28130
     50                                             -0.90668
     51                                             -1.17158
     52                                              1.71091
```
