F 1

Appendix 12

Programs for the Estimation of Autoregressive-
Moving Average Models and data listing.

Five programs for the estimation of autoregressive-moving average models are listed in this volumewith the correspondence to the estimators and chapters of my thesis being determined by the following table.

Program	Estimator	Chapter
1	PH,PH/0	2
2	PH/R	6
3	SYSTEM	5
4	A.F.D.	2
5	A.F.D./S	5

Before reading the descriptions the following points should be noted.

1. As well as comments in this volume there are comment cards contained within the programs themselves.
2. The programs are neither as efficient as they might be nor do they embody good programming practices.
3. Each of the descriptions to follow concentrates upon providing information on the subroutines and data input that must be changed as the model changes. In the program listings a model has been assumed and this is indicated under the EXAMPLE heading.
4. The program length is for the maximum sizes indicated by the cards in the MAIN routine of each program.
5. Where a routine is repeated a description is given only once and backward references are made.

PROGRAM 1

USE: This program estimates a single ARMAX equation by Phillips' estimator and is an application of a numerical derivative Gauss-Newton algorithm to the solution of non-linear equations.

EXAMPLE: The model is $(1-L)(1-L^4)y(t) = (1+\alpha_4 L^4)e(t)$

STRUCTURE: For each type of equation equation estimated 3 subroutines must be modified in the following manner.

DATA 1 - All data must be read into and stored in the program in this subroutine. To do this READ statements (and their associated FORMAT statements) must be supplied. Two arrays - YSTAR AND EXOG - are supplied for this purpose but, if unsuitable, the user must supply DIMENSION statements for any arrays into which he wishes the data to be read. In the sample program 80 observations have been read in under format 8F10.0 and stored in YSTAR. After all data has been read in all variables (including lagged ones) must be stored in the W array: in the example y(t),y(t-1),y(t-4) and y(t-5) are stored in W by cards B15-B18. Note that y(t) must always be set equal to YSTAR.

RESID - Subroutine RESID defines u(t) = B(L)y(t)-C(L)x(t) (for a general ARMAX equation) and then obtains e(t) from A(L)e(t) = u(t). Within the DO loop beginning with card D12 the user must define u(t): in the example first and fourth differencing is required and this is done in D 14.

Array TEMP stores the M.A. parameters so that only the fourth element in TEMP will be non-zero (card D 19). Finally all parameter estimates are stored in BN, and as there is only one parameter (all lagged values of e(t) are ignored i.e. the estimator is PH/0) it must be the M.A. coefficient.

TEST - If it is desired to prevent the roots of $\hat{A}(L) = 0$ from lying within the unit circle subroutine TEST may be invoked. Array COEF is used to construct the polynomial $\hat{A}(L)$. The call to INIT merely initializes COEF (1) at 1.0 and all other elements at zero. As the example contains a fourth order M.A. the parameter (stored in BN(1)) will be the fifth element in COEF. Finally NPP is the order of the polynomial plus one and a call to ROOT must be made. In the example only $\hat{A}(L) = 0$ is tested, but the procedure may be repeated for $\hat{B}(L) = 0$ (although the user will need to duplicate card C 24).

All other subroutines (including the main one) may be left untouched.

CARD INPUT:	Card 1:	cols 1-4	The number of series for analysis with the given model.
	Card 2:	cols 1-4	The number of rows in the data matrix W set up in DATA 1.
		cols 4-8	The number of columns in the data matrix W set up in DATA 1 i.e. the total number of observations to be used.

	cols 8-12	The total number of parameters to be estimated.
	cols 12-16	The order of the moving average.
Card 3	Starting values for the parameters must be read in under FORMAT 8F10.0. The number must be equal to that punched in cols 8-12 of card 2, and no value may be zero.	

All cards following will be data.

SUBROUTINES REQUIRED: EIGEN from I.B.M. Scientific Subroutine Package. If this is not available remove card J 36.

PROGRAM LENGTH: 77K (bytes).

```
C      GENERAL NON-LINEAR LEAST SQUARES FOR SINGLE EQUATIONS
C      USING THE GAUSS NEWTON ALGORITHM
C      ARMAX EQUATION VERSION
C      PHILLIPS / BOX-JENKINS METHOD
C
C
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*4 XX
       DIMENSION BOW(20)
       DIMENSION EXOG(150), YSTAR(150)
       DIMENSION RES(100), REST(150)
       DIMENSION W(1000), D1(100), DD(1500), U(100), UU(3), B(20)
       DIMENSION BB(20), BN(20), U1(100), A(300), D3(20)
       EQUIVALENCE (D1(1),REST(1))
       EQUIVALENCE (U(1),EXOG(1))
       COMMON /PARMS4/ N2,NQ1
       COMMON /SPACE/ XX(300)
C      ------------------------------------------------------------------
C      ------------------------------------------------------------------
C      INPUT PARAMETERS
C
C      M IS THE NUMBER OF ROWS OF THE W ARRAY FORMED IN DATA3
C      N IS THE NUMBER OF COLUMNS OF THE W MATRIX FORMED IN DATA3. IT IS
C       NUMBER OF OBSERVATIONS THAT CAN BE USED IN THE REGRESSION.
C      N1 IS THE NUMBER OF PARAMETERS
C      NCRIT IS THE NUMBER OF SERIES FOR ANALYSIS
C      NQ1 IS THE ORDER OF THE MOVING AVERAGE
C       THE MAXIMUM SIZES OF THE PROGRAM ARE GOVERNED BY THE FOLLOWING
C       CONSTRAINTS
C      N  <  100
C       (M*N) < 1500
C       N1  <  15
C       NQ1 < 30
C      ------------------------------------------------------------------
C      ------------------------------------------------------------------
       READ (1,8) NCRIT
```

```
      DO 7 IJL=1,NCRIT
      READ (1,8) M,N,N1,NQ1
      READ (1,9) (BOW(I),I=1,N1)
      N3=NQ1
      NM=NQ+1
      N2=M
      DO 1 IK=1,N1
      B(IK)=BOW(IK)
1     CONTINUE
      CALL DATA1 (W,YSTAR,EXOG,N,REST,NA)
      WRITE (3,10) (B(I),I=1,N1)
      CALL RESID (W,U,B,N,RES)
      CALL SUMSQ (UU,U,1,N)
      WRITE (3,11) (B(I),I=1,N1)
      WRITE (3,12) UU(1)
C     ITERATION CYCLE BEGINS - FORM DERIVATIVES HERE
      NNN=0
2     CONTINUE
      NNN=NNN+1
      DO 3 I=1,N1
      BN(I)=B(I)
3     BB(I)=B(I)
      CALL DERIVS (DD,D1,U1,N,N1,A,LW,BN,BB,B,W,RES,MW,U,D3)
      CALL SEARCH (BB,B,BN,A,D3,W,U,UU,N,N1,RES,&14)
      WRITE (3,11) (BN(I),I=1,N1)
      CALL RESID (W,U,BN,N,RES)
      CALL SUMSQ (UU,U,1,N)
      WRITE (3,12) UU(1)
      IF (NNN.GT.20) GO TO 6
      DO 4 I=1,N1
      BB(I)=B(I)
4     B(I)=BN(I)
      DO 5 I=1,N1
      CONST=DABS((BN(I)-BB(I))/BB(I))
      IF (CONST.GT..005) GO TO 2
5     CONTINUE
```

```
6     CONTINUE
14    CONTINUE
      WRITE (3,13)
      CALL OUTPUT (BN,N1,U)
      CALL STATS (A,UU,N1,N,BN,BB,B)
      KN=N/3
      CALL HYPOTH (U,U,RES,N,KN,N1)
      CALL FIT (YSTAR,RES,U,REST,N,DD)
7     CONTINUE
      CALL EXIT
C
8     FORMAT (20I4)
9     FORMAT (8F10.0)
10    FORMAT (10F12.4)
11    FORMAT (/' BETA COEFFICIENTS ',5F12.6)
12    FORMAT (/' RESIDUAL SUM OF SQUARES IS',D16.8)
13    FORMAT (/' CONVERGENCE ACHIEVED'/)
      END


      SUBROUTINE DATA1 (W,YSTAR,EXOG,N,REST,NA)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DATA1 SETS UP MATRIX W WITH ALL DATA
C        REQUIRED. THIS MUST INCLUDE ALL LAGGED VALUES
C        AS WELL. THE DEPENDENT VARIABLE IS IN THE FIRST ROW.
C
C
      COMMON /PARMS4/ N2,NQ1
      DIMENSION W(N2,N), YSTAR(1), EXOG(1)
      DIMENSION REST(1)
      READ (1,2) (YSTAR(I),I=1,80)
      DO 1 I=1,N
      W(1,I)=YSTAR(I)
      W(2,I)=YSTAR(I+1)
```

```
      W(3,I)=YSTAR(I+4)                                          B  17
      W(4,I)=YSTAR(I+5)                                          B  18
      YSTAR(I)=W(1,I)                                            B  19
1     CONTINUE                                                   B  20
      RETURN                                                     B  21
C                                                                B  22
2     FORMAT (8F10.0)                                            B  23
      END                                                        B  24-

      SUBROUTINE TESTS (BN,ZETA,KO)                              C   1
      IMPLICIT REAL*8(A-H,O-Z)                                   C   2
C                                                                C   3
C                                                                C   4
C        SUBROUTINE TESTS SETS UP(IN COEF) THE POLYNOMIALS       C   5
C        WHOSE ROOTS ARE TO BE TESTED.                           C   6
C        IF THE POLYNOMIAL TO BE TESTED IS OF ORDER M POL MUST   C   7
C        BE DIMENSIONED AT (M+1)**2. HOWEVER IF THE POLYNOMIAL IS OF C 8
C        HIGH ORDER ESTIMATION SHOULD PROCEED UNDER PH/O AND     C   9
C        THE ROOT PROBLEM SHOULD NOT ARISE.                      C  10
C                                                                C  11
C                                                                C  12
      DIMENSION BN(1)                                            C  13
      DIMENSION ZETA(1)                                          C  14
      DIMENSION COEF(20)                                         C  15
      DIMENSION POL(200)                                         C  16
      DO 1 I=1,20                                                C  17
      COEF(I)=0.0                                                C  18
1     CONTINUE                                                   C  19
      COEF(1)=1.0                                                C  20
      COEF(5)=BN(1)
      NPP=5
      CALL ROOT (NPP,POL,COEF,ZETA,KO)                           C  24
      RETURN                                                     C  25
      END                                                        C  26-
```

```
      SUBROUTINE RESID (W,U1,BN,N,RES)                                  D   1
      IMPLICIT REAL*8(A-H,O-Z)                                          D   2
C                                                                       D   3
C                                                                       D   4
C        SUBROUTINE RESID CALCULATES U(T)=A(L)E(T)(RES) AND SETS        D   5
C        UP M.A. PARAMETERS IN TEMP.                                    D   6
C                                                                       D   7
C                                                                       D   8
      COMMON /PARMS4/ N2,NQ1                                            D   9
      DIMENSION RES(1)                                                  D  10
      DIMENSION W(N2,N), U1(1), BN(1)                                   D  11
      DIMENSION TEMP(30)                                                D  12
      DO 1 I=1,N                                                        D  13
      RES(I)=W(1,I)-W(2,I)-W(3,I)+W(4,I)                                D  14
1     CONTINUE                                                          D  15
      DO 2 IJJ=1,30                                                     D  16
      TEMP(IJJ)=0.0                                                     D  17
2     CONTINUE                                                          D  18
      TEMP(4)=BN(1)                                                     D  19
      NM=NQ1+1                                                          D  21
      CALL EPSVEC (U1,NM,RES,TEMP,N)                                    D  22
      RETURN                                                            D  23
      END                                                               D  24-

      SUBROUTINE OUTPUT (BN,N1,EPSIL)                                   E   1
      IMPLICIT REAL*8(A-H,O-Z)                                          E   2
      DIMENSION EPSIL(1)                                                E   3
      DIMENSION BN(N1)                                                  E   4
C                                                                       E   5
C                                                                       E   6
C        SUBROUTINE OUTPUT ENABLES FINAL ESTIMATES OF THE               E   7
C        PARAMETER VECTOR AND E(T) TO BE OBTAINED                       E   8
C                                                                       E   9
C                                                                       E  10
```

```
      RETURN
      END


      SUBROUTINE SUMSQ (UU,U1,K,N,BN)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BN(1), UU(3), U1(1)
C
C
C        SUBROUTINE SUMSQ COMPUTES THE SUM OF SQUARES(UU)
C
C
      UU(K)=0.0
      DO 1 I=1,N
      UU(K)=UU(K)+U1(I)**2
1     CONTINUE
      RETURN
      END


      SUBROUTINE HYPOTH (X,Y,AUTO,N,K,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE HYPOTH COMPUTES THE CORRELOGRAM OF RESIDUALS
C        THE BOX-PIERCE TEST FOR WHITE NOISE OVER K LAGS
C        OF THE CORRELOGRAM AND THE DURBIN-WATSON STATISTIC.
C
C
      DIMENSION X(1), AUTO(1)
      DIMENSION Y(1)
      DIMENSION STD(100), TTEST(100)
      YMEAN=0.0
      XMEAN=0.0
      DO 1 I=1,N
      YMEAN=YMEAN+Y(I)
```

```
      XMEAN=XMEAN+X(I)
1     CONTINUE
      YMEAN=YMEAN/N
      XMEAN=XMEAN/N
      DO 3 I=1,K
      NK=N-I
      AUTO(I)=0.0
      DO 2 J=1,NK
      AUTO(I)=AUTO(I)+(Y(J)-YMEAN)*(X(I+J)-XMEAN)
2     CONTINUE
3     CONTINUE
      B=0.0
      VAR=0.0
      DO 4 I=1,N
      VAR=VAR+(X(I)-XMEAN)*(Y(I)-YMEAN)
4     CONTINUE
      DO 5 I=1,K
      AUTO(I)=AUTO(I)/VAR
5     CONTINUE
      DO 6 I=1,K
      B=B+AUTO(I)**2
6     CONTINUE
      Q=N*B
      NN=K-N1
      DO 7 J=1,K
      STD(J)=1.0/N
      STD(J)=DSQRT(STD(J))
      TTEST(J)=AUTO(J)/STD(J)
7     CONTINUE
      WRITE (3,10)
      DO 8 I=1,K
      WRITE (3,11) (I,AUTO(I),STD(I),TTEST(I))
8     CONTINUE
      WRITE (3,12) NN,K,Q
      NN=N-1
      DW=0.0
```

```
      DO 9 I=1,NN
      AA=X(I)-X(I+1)
      AA=AA**2
      DW=DW+AA
9     CONTINUE
      DW=DW/VAR
      DW=DW/N
      WRITE (3,13) DW
      RETURN
C
10    FORMAT ('1',5X,'LAG',5X,'AUTOCORRELATION COEFFICIENTS',5X,'STANDAR
     1D ERRORS',5X,'T STATISTICS')
11    FORMAT ('0',5X,I4,13X,F14.6,9X,F14.6,6X,F14.6)
12    FORMAT ('0',10X,'CHI-SQUARE FOR MODEL WITH',I4,'D.F.AND',I4,'  LAG
     1S EQUALS',F10.3)
13    FORMAT (' ',25X,'DURBIN WATSON STATISTIC',F14.6)
      END


      SUBROUTINE FIT (YSTAR,RES,EPSIL,REST,N,DD)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE FIT COMPUTES THE PREDICTED VALUE OF THE
C        DEPENDENT VARIABLE WITHIN THE SAMPLE PERIOD AND
C        ENABLES THESE TO BE PLOTTED FROM DD
C
C
      REAL*4 DD
      DIMENSION YSTAR(1), RES(1), EPSIL(1), REST(1)
      DIMENSION DD(N,3)
      DO 1 I=1,N
      DD(I,1)=I
      DD(I,2)=YSTAR(I)
      DD(I,3)=YSTAR(I)-EPSIL(I)
1     CONTINUE
```

```
      CALL SQUARE (YSTAR,EPSIL,N)
      RETURN
      END


      SUBROUTINE SQUARE (YSTAR,RES,N)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION YSTAR(1), RES(1)
C
C
C        SUBROUTINE SQUARE COMPUTES THE MULTIPLE CORRELATION
C        COEFFICIENT.
C
C
      A=0.0
      B=0.0
      C=0.0
      DO 1 I=1,N
      C=C+YSTAR(I)
      A=A+YSTAR(I)**2
      B=B+RES(I)**2
1     CONTINUE
      C=C**2
      C=C/N
      A=A-C
      R2=B/A
      R2=1-R2
      WRITE (3,2) R2
      RETURN
C
2     FORMAT ('0',30X,'CORRELATION COEFFICIENT',F13.6)
      END


      SUBROUTINE DERIVS (DD,D1,U1,N,N1,A,LW,BN,BB,B,W,RES,MW,U,D3)
      IMPLICIT REAL*8(A-H,O-Z)
```

```
C
C
C        SUBROUTINE DERIVS COMPUTES THE DERIVATIVES FOR THE
C        GAUSS NEWTON ALGORITHM.
C
C
      COMMON /PARMS4/ N2,NQ1
      DIMENSION LW(1), MW(1)
      DIMENSION DD(N1,N), D1(1), U1(1), A(N1,N1), BB(1)
      DIMENSION B(1), W(1), RES(1), BN(1), U(1)
      DIMENSION D3(1)
      DO 2 I=1,N1
      BB(I)=B(I)+.005*B(I)
      CALL RESID (W,D1,BB,N,RES)
      BN(I)=B(I)-.005*B(I)
      CALL RESID (W,U1,BN,N,RES)
      CONST=.01*B(I)
      DO 1 J=1,N
1     DD(I,J)=(D1(J)-U1(J))/CONST
      BB(I)=B(I)
2     BN(I)=B(I)
C     NOW FORM INFORMATION MATRIX
      DO 4 I=1,N1
      DO 4 J=I,N1
      A(I,J)=0.0
      DO 3 K=1,N
3     A(I,J)=A(I,J)+DD(I,K)*DD(J,K)
4     A(J,I)=A(I,J)
      DO 5 I=1,N1
      D3(I)=0.0
      DO 5 J=1,N
5     D3(I)=D3(I)+DD(I,J)*U(J)
      CALL INVERT (A,N1)
      RETURN
C
      END
```

```
      SUBROUTINE SEARCH (BB,B,BN,A,D3,W,U,UU,N,N1,RES,*)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE SEARCH FINDS AN OPTIMAL STEP LENGTH BY QUADRATIC
C        INTERPOLATION.
C
C
      COMMON /PARMS4/ N2,NQ1
      DIMENSION BB(1), B(1), BN(1), A(N1,N1), D3(1), W(1)
      DIMENSION ALX(3), U(1), UU(1)
      DIMENSION RES(1)
      VALUE=UU(1)
      TEST=VALUE*0.001
      DO 1 I=1,N1
      BB(I)=0.0
      DO 1 J=1,N1
1     BB(I)=BB(I)-A(I,J)*D3(J)
C     BB IS THE CHOSEN DIRECTION
      ALX(1)=0.0
      ALX(2)=0.5
      ALX(3)=1.0
      DO 2 I=2,3
      CALL ACCUM (BN,B,BB,ALX(I),N1)
      CALL RESID (W,U,BN,N,RES)
      CALL SUMSQ (UU,U,I,N)
      CALL TESTS (BN,UU,I)
2     CONTINUE
      IF (UU(2).LE.UU(3)) GO TO 3
      ALX(2)=ALX(3)
      UU(2)=UU(3)
3     KIK=0
      IF (UU(2).GT.UU(1)) GO TO 5
4     CONTINUE
```

```
      ALX(3)=2.0*ALX(2)                                K  35
      CALL ACCUM (BN,B,BB,ALX(3),N1)                   K  36
      CALL RESID (W,U,BN,N,RES)                        K  37
      CALL SUMSQ (UU,U,3,N)                            K  38
      CALL TESTS (BN,UU,3)                             K  39
      IF (UU(3).GT.UU(2)) GO TO 7                      K  40
      ALX(2)=ALX(3)                                    K  41
      UU(2)=UU(3)                                      K  42
      KIK=KIK+1                                        K  43
      IF (KIK.LT.10) GO TO 4                           K  44
      RETURN 1
 5    ALX(3)=ALX(2)                                    K  46
      UU(3)=UU(2)                                      K  47
      ALX(2)=ALX(3)*0.5                                K  48
      CALL ACCUM (BN,B,BB,ALX(2),N1)                   K  49
      CALL RESID (W,U,BN,N,RES)                        K  50
      CALL SUMSQ (UU,U,2,N)                            K  51
      CALL TESTS (BN,UU,2)                             K  52
      IF (UU(3).GT.UU(2).AND.UU(2).LT.UU(1)) GO TO 7   K  53
      KIK=KIK+1                                        K  54
      IF (KIK.LT.10) GO TO 5                           K  55
      ALX(2)=-ALX(2)                                   K  56
      CALL ACCUM (BN,B,BB,ALX(2),N1)                   K  57
      CALL RESID (W,U,BN,N,RES)                        K  58
      CALL SUMSQ (UU,U,2,N)                            K  59
      CALL TESTS (BN,UU,2)                             K  60
      IF (UU(2).LT.UU(1)) GO TO 6                      K  61
      RETURN 1
 6    KIK=1                                            K  63
      GO TO 4                                          K  64
 7    JK=0                                             K  65
      ALX(1)=0.0                                       K  66
      JK=JK+1                                          K  67
      CALL QUAD (ALX,UU,ATLEN)                         K  68
      ALX(1)=ATLEN                                     K  69
      CALL ACCUM (BN,B,BB,ALX(1),N1)                   K  70
```

```
      CALL RESID (W,U,BN,N,RES)
      CALL SUMSQ (UU,U,1,N)
      CALL TESTS (BN,UU,1)
      IF (UU(1).LE.UU(2)) GO TO 8
      ALX(1)=ALX(2)
      CALL ACCUM (BN,B,BB,ALX(1),N1)
      CALL RESID (W,U,BN,N,RES)
      CALL SUMSQ (UU,U,1,N)
8     CONTINUE
      TEST2=DABS(UU(1)-VALUE)
      IF (TEST2.LE.TEST) GO TO 9
      RETURN
9     RETURN 1
      END


      SUBROUTINE ACCUM (BN,B,BB,AL,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE ACCUM UPDATES THE PARAMETER ESTIMATES
C
C
      DIMENSION BN(1), B(1), BB(1)
      DO 1 I=1,N1
      BN(I)=B(I)+AL*BB(I)
1     CONTINUE
      RETURN
      END


      SUBROUTINE QUAD (X,U,ATLEN)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE QUAD FITS A QUADRATIC THROUGH 3 POINTS
```

```
C         AND INTERPOLATES THE MINIMUM
C
C
      DIMENSION X(3), U(3)
      IF(U(3).EQ.1.0D35)U(3)=U(1)*1.5
      A=(((U(1)-U(2))*(X(1)-X(3)))-((U(1)-U(3))*(X(1)-X(2))))/((X(1)-X(2
     1))*(X(1)-X(3))*(X(2)-X(3)))
      B=((U(1)-U(2))-(A*(X(1)-X(2))*(X(1)+X(2))))/(X(1)-X(2))
      C=-(A*X(1)*X(1))-(B*X(1))+U(1)
      ATLEN=-B/(A*2)
      ATMIN=-((B*B)/(A*4))+C
      RETURN
C
1     FORMAT (' ',5X,8F10.3)
      END


      SUBROUTINE EPSVEC (EPSIL,M,RES,ROE,T)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C         SUBROUTINE EPSVEC CALCULATES E(T)
C
C
      INTEGER*4 T
      DIMENSION WORK(100)
      DIMENSION EPSIL(1), RES(1), ROE(1)
      N=M-1
      EPSIL(1)=RES(T)
      IF (N.EQ.1) GO TO 3
      DO 2 I=2,N
      KA=I-1
      A=0.0
      DO 1 J=1,KA
      A=A+EPSIL(I-J)*ROE(J)
1     CONTINUE
```

```
      L=T-I+1
      EPSIL(I)=RES(L)-A
2     CONTINUE
C     SECOND STAGE
3     DO 5 I=M,T
      BA=0.0
      L=T-I+1
      DO 4 J=1,N
      BA=BA+ROE(J)*EPSIL(I-J)
4     CONTINUE
      EPSIL(I)=RES(L)-BA
5     CONTINUE
C
      CALL TURN (EPSIL,WORK,T)
      RETURN
      END


      SUBROUTINE STATS (VAR,UU,N1,N,PARM,STD,TTEST)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE STATS COMPUTES THE RESIDUAL VARIANCE,STANDARD
C        DEVIATIONS OF THE PARAMETER ESTIMATES AND T-TESTS
C
C
      DIMENSION VAR(N1,N1), PARM(1), STD(1), TTEST(1)
      DIMENSION UU(1)
      COMMON /PARMS4/ N2,NQ1
      UU(1)=UU(1)/(N-N1)
      WRITE (3,2) UU(1)
      DO 1 I=1,N1
      STD(I)=VAR(I,I)*UU(1)
      STD(I)=DSQRT(STD(I))
      TTEST(I)=DABS(PARM(I)/STD(I))
1     CONTINUE
```

```
      WRITE (3,3)
      WRITE (3,4) (PARM(I),STD(I),TTEST(I),I=1,N1)
      RETURN
C
2     FORMAT (' ',30X,'RESIDUAL VARIANCE IS',F16.7)
3     FORMAT (' ',10X,'PARAMETERS',20X,'STANDARD DEVIATION',25X,'T STATI
     1STIC')
4     FORMAT (' ',5X,F16.7,20X,F16.7,20X,F16.7)
      END


      SUBROUTINE INVERT (A,N)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE INVERT INVERTS A MATRIX A
C
C
      DIMENSION A(N,N), IPIV(20)
      DO 1 I=1,N
1     IPIV(I)=0
      DO 11 I=1,N
      AMAX=0.
      DO 5 J=1,N
      IF (IPIV(J)) 2,2,5
2     IF (DABS(A(J,J)-AMAX)) 4,4,3
3     ICOL=J
      AMAX=DABS(A(J,J))
4     CONTINUE
5     CONTINUE
      IPIV(ICOL)=1
      IF (AMAX-1.0D-50) 6,6,7
6     WRITE (3,12)
      STOP
7     CONTINUE
      AMAX=A(ICOL,ICOL)
```

```
      A(ICOL,ICOL)=1.0
      DO 8 K=1,N
8     A(ICOL,K)=A(ICOL,K)/AMAX
      DO 11 J=1,N
      IF (J-ICOL) 9,11,9
9     AMAX=A(J,ICOL)
      A(J,ICOL)=0.
      DO 10 K=1,N
10    A(J,K)=A(J,K)-A(ICOL,K)*AMAX
11    CONTINUE
      RETURN
C
12    FORMAT (/' SINGULAR MATRIX -- TERMINATE'/)
      END


      SUBROUTINE ROOT (KP,MAT,COEF,ZETA,KO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE ROOT PERFORMS THE ROUTH-HURWICZ TEST
C
C
      REAL*8 MAT
      DIMENSION MAT(KP,KP), COEF(1), ZETA(1)
      DO 1 I=1,KP
      MAT(1,I)=COEF(I)
1     CONTINUE
      DO 3 I=2,KP
      DO 2 J=I,KP
      IJ=J-I
      MAT(I,J)=MAT(I-1,I-1)*MAT(I-1,J-1)-MAT(I-1,KP)*MAT(I-1,KP-IJ)
2     CONTINUE
3     CONTINUE
      DO 4 I=1,KP
      IF(MAT(I,I).LT.0.0D0)ZETA(KO)=1.0D35
```

```
4     CONTINUE
      RETURN
      END


      SUBROUTINE TURN (X,Y,N)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE TURN ENABLES A VECTOR OF OBSERVATIONS TO BE
C        REVERSED.IT IS USED IN EPSVEC.
C
C
      DIMENSION X(1), Y(1)
      DO 1 I=1,N
      Y(I)=X(I)
1     CONTINUE
      DO 2 I=1,N
      X(I)=Y(N-I+1)
2     CONTINUE
      RETURN
      END


      SUBROUTINE WEIGHT (G,N1)
C
C
C        SUBROUTINE WEIGHT DETERMINES IF THE WEIGHTING MATRIX
C        IS POSITIVE DEFINITE,AND,IF SO, WEIGHTS THE
C        DIAGONALS BY THE SHANNO METHOD.
C        SUBROUTINE EIGEN FROM I.B.M.  S.S.P. IS REQUIRED
C
C
      REAL*8 G
      COMMON /SPACE/ B(300)
      DIMENSION G(N1,N1)
```

```
      REAL*4 LAMBDA
      JK=0
      DO 2 I=1,N1
      DO 1 J=1,I
      JK=JK+1
      B(JK)=G(J,I)
    1 CONTINUE
    2 CONTINUE
      CALL EIGEN (B,R,N1,1)
      IK=N1*(N1+1)/2
      EMIN=B(IK)
      EMAX=B(1)
      RATIO=EMAX/EMIN
      WRITE (3,5) EMIN
      WRITE (3,6) RATIO
      LAMBDA=-5*EMIN
      IF (LAMBDA.LE.0.0) GO TO 4
      DO 3 I=1,N1
      G(I,I)=G(I,I)+LAMBDA
    3 CONTINUE
    4 RETURN
C
    5 FORMAT (' ',5X,'MINIMUM EIGENVALUE IS',E13.6)
    6 FORMAT (' ',5X,'RATIO OF MAXIMUM TO MINIMUM EIGENVALUE IS',E13.6)
      END


      SUBROUTINE INIT (COEF)
C
C
C        SUBROUTINE INITIALIZES COEF AT ZERO FOR USE IN TEST
C
C
      DIMENSION COEF(1)
      COEF(1)=1.0
      DO 1 I=1,20
```

```
      COEF(I)=0.0                                        T  10
1     CONTINUE                                           T  11
      RETURN                                             T  12
      END                                                T  13-
```

PROGRAM 2

USE: This program, by building on PROGRAM 1, estimates models in which the disturbance term has a composite form.

EXAMPLE: The model is

$(1-L)y(t) = u(t) = \mu + (1-L)e_1(t) + e_2(t)$.

The parameters are stored in BN in the following order.

BN(1) is the variance ratio $\lambda = \sigma_1^2 / \sigma_2^2$.

BN(2) is the mean μ.

BN(3) is e(-1) .

STRUCTURE: Apart from the changes in PROGRAM 1 the following modifications are required.

RESID - as described for PROGRAM 1, but this special version of RESID must replace that listed earlier. Nevertheless u(t) must still be defined (but not TEMP).

COVAR - This subroutine forms the covariance function $\Gamma_{uu}(Z) = (1-Z)(1-Z^{-1})\sigma_1^2 + \sigma_2^2$. Card B 31 defines (1-Z) and the output (C1) of DERS converts this to $(1-Z)(1-Z^{-1})$. Finally COMB makes all powers of Z positive i.e. $2-Z-Z^{-1}$ becomes $2Z-Z^2-1$ and stores the result in TEMP 1. NPP must be equal to the order of the M.A. for the component under examination. Cards B 36 and B 39 rewrite $\Gamma_{uu}(Z)$ as

$$\Gamma_{uu}(Z) = [(2Z-Z^2-1)\lambda + Z]\sigma_2^2 .$$

CARD INPUT: As for Program 1.

SUBROUTINES REQUIRED: As for Program 1.

PROGRAM LENGTH: 113K (bytes) (Includes PROGRAM 1).

```
      SUBROUTINE RESID (W,U1,BN,N,RES)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE RESID COMPUTES THE RESIDUAL VECTOR AND THE M.A.
C        PARAMETERS FROM FACTORIZATION OF THE COVARIANCE
C        FUNCTION. THE LATTER ARE STORED IN TEMP
C
C
      DIMENSION RES(1)
      COMMON /PARMS4/ N2,NQ1
      DIMENSION W(N2,N), U1(1), BN(1)
      DIMENSION VEC1(60), VEC2(60), HOLD1(30), HOLD2(30)
      DIMENSION WORK1(60), TEMP(30)
      DO 1 I=1,N
      RES(I)=W(1,I)-W(2,I)
      RES(I)=RES(I)-BN(3)
1     CONTINUE
      RES(N)=RES(N)-BN(2)
      DO 2 IJJ=1,20
      TEMP(IJJ)=0.0
2     CONTINUE
      NQ=NQ1
      CALL COVAR (BN,WORK1)
      CALL SORT (VEC1,VEC2,HOLD1,HOLD2,IR,IK,BN,NQ)
      NMM=2*NQ
      IKK=2*IK
      CALL POLY (U1,NMM,VEC1,VEC2,IKK)
      CALL MATCH (WORK1,U1,NMM)
      CALL POLY (TEMP,NQ,HOLD1,HOLD2,IK)
      NM=NQ1+1
      CALL EPSVEC (U1,NM,RES,TEMP,N)
      RETURN
      END
```

```
      SUBROUTINE COVAR (BN,COEF)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE COVAR SETS UP THE COMPOSITE DISTURBANCE
C        TERM. LET THIS HAVE THE FORM U(T)=A(L)E(T)+B(L)V(T)
C        THEN THE PARAMETERS IN A(L) ARE SET UP IN COEF AND BY USE
C        OF DERS AND COMB THE COVARIANCE FUCTION CORRESPONDING TO
C        THIS M.A. COMPONENT IS COMPUTED. THE ORDER OF M.A. OF THE
C        COMPONENT MUST BE SET EQUAL TO NPP.
C        NORMALLY THE COVARIANCES ARE STORED IN TEMP1,TEMP2 IF
C        THERE ARE TWO COMPONENTS WITH TRANSFER FUNCTIONS GIVEN
C        BY A(L) AND B(L). FINALLY ALL OF THE COMPONENTS ARE
C        COMBINED IN COEF BY MULTIPLYING BY THE ELEMENTS IN BN
C        THAT REPRESENT VARIANCE RATIOS
C
C
      DIMENSION BN(1), COEF(1)
      DIMENSION TEMP1(60), TEMP2(60), TEMP3(60)
      DIMENSION C1(30), C2(30), C3(30)
      COMMON /PARMS4/ N2,NQ1
      NQQ=NQ1*2+1
      DO 1 IJ=1,NQQ
      TEMP1(IJ)=0.0
      TEMP2(IJ)=0.0
      TEMP3(IJ)=0.0
    1 CONTINUE
      DO 2 I=1,NQQ
      COEF(I)=0.0
    2 CONTINUE
      COEF(1)=-1.0
      NPP=1
      CALL DERS (COEF,C1,NPP)
      CALL COMB (C1,TEMP1,NPP)
      DO 3 I=1,NQQ
      COEF(I)=TEMP1(I)*BN(1)
```

```
      COEF(2)=COEF(2)+1                                                 B  39
    3 CONTINUE
      AAB=COEF(1)                                                       B  40
      DO 4 IJK=1,NQQ
      COEF(IJK)=COEF(IJK)/AAB                                           B  42
    4 CONTINUE                                                          B  43
      RETURN                                                            B  44
C                                                                       B  45
    5 FORMAT (' ',5X,3E13.6)                                            B  46
      END                                                               B  47-

      SUBROUTINE MATCH (W,U,N),                                         C   1
      IMPLICIT REAL*8(A-H,O-Z)                                          C   2
C                                                                       C   3
C                                                                       C   4
C        SUBROUTINE MATCH COMPARES THE COVARIANCE FUNCTION WHOSE         C   5
C        ROOTS HAVE BEEN DETERMINED WITH THAT COMPUTED FROM THE          C   6
C        CALCULATED ROOTS                                               C   7
C        IF ICOND EQUALS 1 THEY DIFFER IN THE THIRD DECIMAL PLACE       C   8
C                                                                       C   9
C                                                                       C  10
      DIMENSION W(1), U(1)                                              C  11
      DIMENSION X(60)                                                   C  12
      ICOND=0                                                           C  13
      DO 1 I=1,N                                                        C  14
      X(I+1)=U(I)                                                       C  15
    1 CONTINUE                                                          C  16
      NN=N+1                                                            C  17
      DO 2 I=1,NN                                                       C  18
      U(I)=X(I)                                                         C  19
      U(1)=1.0                                                          C  20
      AD=DABS(U(I)-W(I))                                                C  21
      IF (AD.GT.1.0D-3) ICOND=1                                         C  22
    2 CONTINUE                                                          C  23
      WRITE (3,3) ICOND                                                 C  24
```

```
      RETURN
C
3     FORMAT (' ',5X,'POLY MATCH=',I4)
      END


      SUBROUTINE OUTPUT (BN,N1,U)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE OUTPUT ENABLES THE USER TO OBTAIN THE
C        DISTUBANCE VECTOR AND IT ALSO COMPUTES THE M.A. PARAMETERS
C
C
      DIMENSION U(1)
      DIMENSION V1(60), V2(60), H1(30), H2(30), ROE(30)
      DIMENSION BN(N1)
      COMMON /PARMS4/ N2,NQ1
      CALL SORT (V1,V2,H1,H2,IR,IK,BN,NQ1)
      CALL POLY (ROE,NQ1,H1,H2,IK)
      WRITE (3,1) (ROE(I),I=1,NQ1)
      RETURN
C
1     FORMAT (' ',5X,'MOVING AVERAGE PARAMETERS ARE',5F12.6)
      END


      SUBROUTINE POLY (ROE,N,HOLD1,HOLD2,IK)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE POLY COMPUTES THE M. A. PARAMETERS ONCE ALL ROOTS
C        HAVE BEEN DETERMINED
C
C
      DIMENSION POL(100), TEMP(10), APOL(100), ROE(1)
```

```
      DIMENSION HOLD1(1), HOLD2(1)
      POL(1)=1.0
      KQ=IK/2
      KQ=N-KQ
      JK=1
      DO 4 I=1,KQ
      IF (HOLD2(JK).EQ.0.0D0) GO TO 1
      TEMP(1)=1.0
      TEMP(2)=-HOLD1(JK)*2
      TEMP(3)=HOLD1(JK)**2+HOLD2(JK)**2
      CALL PMPY (APOL,IDIMZ,TEMP,3,POL,JK)
      JK=JK+2
      GO TO 2
1     TEMP(1)=1.0
      TEMP(2)=-HOLD1(JK)
      CALL PMPY (APOL,IDIMZ,TEMP,2,POL,JK)
      JK=JK+1
2     CONTINUE
      DO 3 J=1,IDIMZ
      POL(J)=APOL(J)
3     CONTINUE
4     CONTINUE
      DO 5 I=1,N
      ROE(I)=POL(I+1)
5     CONTINUE
      RETURN
      END


      SUBROUTINE SORT (VEC1,VEC2,HOLD1,HOLD2,IR,IK,BN,NQ)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE SORT SEPERATES THE ROOTS INTO A SET INSIDE AND
C        A SET OUTSIDE THE UNIT CIRCLE
C
```

```
C
      DIMENSION BN(1)
      DIMENSION VEC1(1), VEC2(1), HOLD1(1), HOLD2(1)
      DIMENSION COEF(100)
      DIMENSION POL(100)
      DO 1 J=1,100
      COEF(J)=0.0
    1 CONTINUE
      CALL COVAR (BN,COEF)
      M=2*NQ+1
      CALL DPRMM (COEF,VEC1,VEC2,NQ,IR)
      JK=0
      DO 2 I=1,IR
      POL(I)=VEC1(I)**2+VEC2(I)**2
      IF (POL(I).GT.1.0D0) GO TO 2
      JK=JK+1
      HOLD1(JK)=VEC1(I)
      HOLD2(JK)=VEC2(I)
    2 CONTINUE
      IR=JK
      JK=0
      DO 3 I=1,IR
      IF (HOLD2(I).EQ.0.0D0) GO TO 3
      JK=JK+1
    3 CONTINUE
      IK=JK
      RETURN
C
      END


      SUBROUTINE PMPY (Z,IDIMZ,X,IDIMX,Y,IDIMY)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE PMPY MULTIPLYS TWO POLYNOMIALS TOGETHER
```

```
C
C
      DIMENSION Z(1), X(1), Y(1)
      IF (IDIMX*IDIMY) 1,1,2
 1    IDIMZ=0
      GO TO 5
 2    IDIMZ=IDIMX+IDIMY-1
      DO 3 I=1,IDIMZ
 3    Z(I)=0.0
      DO 4 I=1,IDIMX
      DO 4 J=1,IDIMY
      K=I+J-1
 4    Z(K)=X(I)*Y(J)+Z(K)
 5    RETURN
      END


      SUBROUTINE DPRMM (COEF,VEC1,VEC2,NQ,IR)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DPRMM COMPUTES THE REDUCED POLYNOMIAL BY
C        SYNTHETIC DIVISION
C
C
      COMPLEX*16 TEMP
      DIMENSION COEF(1), VEC1(1), VEC2(1), TEMP(100), PARM(10), POL(100)
      JKK=0
      DO 4 I=1,NQ
      NMM=NQ*2+1
      NM=NMM-JKK
      IF (NM.LE.1) GO TO 5
      CALL MULLER (TEMP,COEF,NM,RPART,CPART,ICOND)
      IF (ICOND.EQ.1) GO TO 1
      ALPHA=RPART
      GAMMA=1/RPART
```

```
      VEC1(JKK+1)=ALPHA
      VEC2(JKK+1)=0
      VEC1(JKK+2)=GAMMA
      VEC2(JKK+2)=0
      PARM(1)=ALPHA*GAMMA
      PARM(2)=-(GAMMA+ALPHA)
      PARM(3)=1.0
      CALL PDIV (POL,IP,COEF,NM,PARM,3,1.0D-14,IER)
      JKK=JKK+2
      GO TO 2
1     CONTINUE
      ALPHA=RPART
      BETA=CPART
      H1=ALPHA**2+BETA**2
      GAMMA=ALPHA/H1
      DELTA=-BETA/H1
      VEC1(JKK+1)=ALPHA
      VEC2(JKK+1)=BETA
      VEC1(JKK+2)=ALPHA
      VEC2(JKK+2)=-BETA
      VEC1(JKK+3)=GAMMA
      VEC2(JKK+3)=DELTA
      VEC1(JKK+4)=GAMMA
      VEC2(JKK+4)=-DELTA
      H2=GAMMA**2+DELTA**2
      H3=2*ALPHA
      H4=2*GAMMA
      PARM(1)=H1*H2
      PARM(2)=H2*H3+H1*H4
      PARM(2)=-PARM(2)
      PARM(3)=H1+H2+H3*H4
      PARM(4)=H3+H4
      PARM(4)=-PARM(4)
      PARM(5)=1.0
      CALL PDIV (POL,IP,COEF,NM,PARM,5,1.0D-14,IER)
      JKK=JKK+4
```

```
 2    DO 3 JJ=1,IP
      COEF(JJ)=POL(JJ)
 3    CONTINUE
 4    CONTINUE
 5    CONTINUE
      MM=2*NQ
      IR=MM
      RETURN
C
      END


      SUBROUTINE FUNC (COEF,ROOT,NM,VAL)
      IMPLICIT COMPLEX*16(A-H,O-Z)
C
C        SUBROUTINE FUNC EVALUATES A POLYNOMIAL
C
C
      REAL*8 COEF
      DIMENSION COEF(1)
      VAL=(0.0,0.0)
      DO 1 I=2,NM
      II=I-1
      VAL=VAL+COEF(I)*(ROOT**II)
 1    CONTINUE
      AA=COEF(1)
      VAL=VAL+AA
      RETURN
      END


      SUBROUTINE MULLER (TEMP,COEF,NM,RPART,CPART,ICOND)
      IMPLICIT COMPLEX*16(A-H,O-Z)
C
C
C        SUBROUTINE MULLER FINDS A ROOT OF A POLYNOMIAL BY MULLER'S
```

```
C         METHOD
C
C
      REAL*8 DD,DA,COEF,HH,HA,FJK,RPART,CPART
      DIMENSION TEMP(1), COEF(1), F(100)
      JK=3
      TEMP(1)=(0,0)
      TEMP(2)=(2,0)
      TEMP(3)=(1,0)
1     CALL FUNC (COEF,TEMP(JK-2),NM,F(JK-2))
      CALL FUNC (COEF,TEMP(JK-1),NM,F(JK-1))
      CALL FUNC (COEF,TEMP(JK),NM,F(JK))
      FJK=CDABS(F(JK))
      IF (FJK.LT.1.0D-14) GO TO 4
      DD=CDABS(F(JK-1))
      DD=DABS(DD-FJK)
      IF (DD.LT.1.0D-14) GO TO 4
      H=TEMP(JK)-TEMP(JK-1)
      DD=CDABS(H)
      IF (DD.LT.1.0D-14) GO TO 4
      ALX=H/(TEMP(JK-1)-TEMP(JK-2))
      DELTA=1+ALX
      AA=-2*F(JK)*DELTA
      G=F(JK-2)*ALX**2-F(JK-1)*DELTA**2+F(JK)*(ALX+DELTA)
      GG=G**2-4*F(JK)*DELTA*ALX*(F(JK-2)*ALX-F(JK-1)*DELTA+F(JK))
      GGG=CDSQRT(GG)
      GGA=G+GGG
      DA=CDABS(GGA)
      GGA=G-GGG
      DD=CDABS(GGA)
      IF (DA.GT.DD) GO TO 2
      BB=G-GGG
      GO TO 3
2     BB=G+GGG
3     ALX=AA/BB
      JK=JK+1
```

```
      TEMP(JK)=TEMP(JK-1)+ALX*H
      DA=DREAL(TEMP(JK))
      DD=DIMAG(TEMP(JK))
      DD=DABS(DD)
      IF (DD.LE.0.5D-13) DD=0.0D0
      TEMP(JK)=DCMPLX(DA,DD)
      GO TO 1
    4 CONTINUE
      RPART=DREAL(TEMP(JK))
      CPART=DIMAG(TEMP(JK))
      ICOND=1
      IF (CPART.EQ.0.0D0) ICOND=0
      RETURN
C
      END


      SUBROUTINE PNORM (X,IDIMX,EPS)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE PNORM MORMALIZES THE REMAINDER POLYNOMIAL
C
C
      DIMENSION X(1)
    1 IF (IDIMX) 4,4,2
    2 IF (DABS(X(IDIMX))-EPS) 3,3,4
    3 IDIMX=IDIMX-1
      GO TO 1
    4 RETURN
      END


      SUBROUTINE PDIV (P,IDIMP,X,IDIMX,Y,IDIMY,TOL,IER)
      IMPLICIT REAL*8(A-H,O-Z)
C
```

```
C
C        SUBROUTINE PDIV PERFORMS POLYNOMIAL DIVISION
C
C
      DIMENSION P(1), X(1), Y(1)
C
      CALL PNORM (Y,IDIMY,TOL)
      IF (IDIMY) 5,5,1
1     IDIMP=IDIMX-IDIMY+1
      IF (IDIMP) 2,3,6
C
C     DEGREE OF DIVISOR WAS GREATER THAN DEGREE OF DIVIDEND
C
2     IDIMP=0
3     IER=0
4     RETURN
C
C     Y IS ZERO POLYNOMIAL
C
5     IER=1
      GO TO 4
C
C     START REDUCTION
C
6     IDIMX=IDIMY-1
      I=IDIMP
7     II=I+IDIMX
      P(I)=X(II)/Y(IDIMY)
C
C     SUBTRACT MULTIPLE OF DIVISOR
C
      DO 8 K=1,IDIMX
      J=K-1+I
      X(J)=X(J)-P(I)*Y(K)
8     CONTINUE
      I=I-1
```

```
      IF (I) 9,9,7
C
C     NORMALIZE REMAINDER POLYNOMIAL
C
9     CALL PNORM (X,IDIMX,TOL)
      GO TO 3
      END


      SUBROUTINE COMB (C,TEMP,NQ)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE COMB TRANSFORMS THE ELEMENTS FROM DERS TO A
C        SYMMETRIC COVARIANCE FUNCTION
C
C
      DIMENSION C(1), TEMP(1)
      IQ=NQ-1
      DO 1 I=1,NQ
      II=NQ-I+2
      TEMP(I+IQ)=C(II)
1     CONTINUE
      TEMP(NQ+IQ+1)=C(1)
      DO 2 I=1,NQ
      TEMP(NQ+IQ+I+1)=C(I+1)
2     CONTINUE
      RETURN
      END


      SUBROUTINE DERS (ROE,C,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DERS TAKES A VECTOR OF M.A. PARAMETERS AND
```

```
C          COMPUTES THE ELEMENTS OF THE COVARIANCE FUNCTION
C
C
      REAL*8 MAT
      DIMENSION MAT(1000), ROW(60)
      DIMENSION C(1), ROE(1)
      COMMON /PARMS5/ M
      M=N1+1
      DO 1 I=1,M
      ROW(I)=0.0
1     CONTINUE
      ROW(1)=1.0
      DO 2 I=1,N1
      ROW(I+1)=ROE(I)
2     CONTINUE
      CALL MULT (MAT,ROW,ROW,C)
      RETURN
      END


      SUBROUTINE MULT (MAT,COLJ,COLI,SOL)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C          SUBROUTINE MULT IS USED BY DERS TO CALCULATE COVARIANCES
C
C
      REAL*8 MAT
      COMMON /PARMS5/ M
      DIMENSION MAT(M,M), COLJ(M), COLI(M), SOL(M)
      DO 2 I=1,M
      DO 1 J=1,M
      MAT(I,J)=0.0
1     CONTINUE
2     CONTINUE
      DO 4 I=1,M
```

```
      JK=0
      DO 3 J=1,I
      JK=JK+1
      MAT(J,I)=COLJ(I-JK+1)
3     CONTINUE
4     CONTINUE
C
      DO 6 I=1,M
      SOL(I)=0.0
      DO 5 J=1,M
      SOL(I)=SOL(I)+MAT(I,J)*COLI(J)
5     CONTINUE
6     CONTINUE
      RETURN
      END
```

PROGRAM 3

USE: This program estimates a system of ARMAX equations by Phillips' estimator and is an application of a numerical derivative Gauss-Newton algorithm to the solution of non-linear equations.

EXAMPLE: The model is a 3-equation one.

1. $(1-L)(1-L^4)y_1(t) = (1+\alpha_4 L^4)e_1(t)$

2. $(1-L)(1-L^4)y_2(t) = (1+\alpha_5 L^5+\alpha_6 L^6)e_2(t)$

3. $(1-L)(1-\beta_4 L^4)y_3(t) = (1+\alpha_1 L+\alpha_4 L^4+\alpha_5 L^5+\alpha_6 L^6)e_3(t)$

All pre-period values of $e_1(t)$, $e_2(t)$ and $e_3(t)$ are ignored i.e. effectively set at zero.

STRUCTURE: DATA1 - As with PROGRAM 1 all data is read in and stored in W.

RESID - The structure is similar to PROGRAM 1 except that the residual and M.A. vectors become matrices with each of the columns of REM and TEMPM corresponding to a given equation of the system.

TEST - As for PROGRAM 1.

CARD INPUT Card 1

cols 1-4: Number of columns in the matrix W set up in DATA1.

cols 4-8: Number of rows in the matrix W set up in DATA1.

cols 8-12: Number of parameters to be estimated.

cols 12-16: The total number of observations available. This will differ from the number used owing to lags.

cols 16-20: The number of equations.

cols 20-24: The maximum order of M.A. in the system.

cols 24-28: A variable taking values

=0. The covariance matrix of disturbances will be computed by the program by using the initial parameter estimates.

=1. The covariance matrix must be supplied by the user on the following card.

Card 2 If cols. 24-28 of card 1 have a zero punched in them this card must be deleted. Otherwise the covariance matrix must be punched in under format 8F10.0.

Card 3 The order of the M.A. of each equation must be punched under format 20I4.

Card 4 The initial parameter estimates (format 8F10.0). The number of these must be equal to that punched in cols. 8-12 of card 2 and no values may be zero.

All cards following will be data.

SUBROUTINES REQUIRED: EIGEN from I.B.M. Scientific Subroutine Package. If this is not available omit card O 47.

PROGRAM LENGTH: 104 K (bytes).

```
C          GENERAL NON-LINEAR LEAST SQUARES FOR SYSTEMS OF EQUATIONS
C          USING THE GAUSS-NEWTON ALGORITHM
C          SOLUTION OF A SYSTEM OF ARMAX EQUATIONS BY THE GENERALIZED
C          PHILLIPS METHOD
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RESM(400), TEMPM(100), EPS(400)
      DIMENSION SIGMA(100)
      DIMENSION RES(100), REST(100)
      DIMENSION BOW(20)
      DIMENSION W(2000), D1(400), DD(2000), U(400), UU(3), B(20)
      DIMENSION BB(20), BN(20), U1(100), A(20,20), D3(20)
      EQUIVALENCE (D1(1),REST(1))
      COMMON /PARMS/ NQ(10),NMAX,KKMAX,MG,N2
      READ (1,6) M,N,N1,NA,MG,NMAX,ICOND
      NGG=MG**2
      IF (ICOND.EQ.1) READ (1,7) (SIGMA(I),I=1,NGG)
      READ (1,6) (NQ(I),I=1,MG)
      READ (1,7) (B(I),I=1,N1)
C     ----------------------------------------------------------------
C     ----------------------------------------------------------------
C          INPUT PARAMETERS
C
C          M          TOTAL NUMBER OF VARIABLES AND THE NUMBER OF COLUMNS
C                     IN THE W MATRIX SET UP IN SUBROUTINE DATA1
C          N          THE TOTAL NUMBER OF OBSERVATIONS USED FROM EACH
C                     SERIES
C          N1         THE TOTAL NUMBER OF PARAMETERS
C          NA         THE TOTAL NUMBER OF OBSERVATIONS AVAILABLE. THIS
C                     WILL USUALLY DIFFER FROM NA OWING TO LAGS IN SOME
C                     OF THE VARIABLES
C          MG         THE NUMBER OF EQUATIONS IN THE SYSTEM
C          NMAX       THE MAXIMUM ORDER OF THE M.A. IN THE SYSTEM
C          ICOND      =0   THE COVARIANCE MATRIX OF DISTURBANCES IS
C                          COMPUTED BY THE PROGRAM BY USING THE INITIAL
```

```
C                     PARAMETER ESTIMATES STORED IN B
C                =1   THE COVARIANCE MATRIX IS SUPPLIED BY THE
C                     USER AND IT MUST BE READ IN AS THE THIRD CARD
C                     UNDER FORMAT (8F10.0)
C        NQ      CONTAINS THE ORDER OF EACH M.A. POLYNOMIAL
C        NQ      CONTAINS THE ORDER OF THE M.A. POLYNOMIAL
C                IN EACH EQUATION. IT IS AN (MG*1) VECTOR ANDIS
C                READ IN AS THE SECOND CARD UNDER FORMAT(8F10.0)
C        THE FOLLOWING CONSTAINTS GOVERN THE SIZE OF THE SYSTEM THAT
C        MAY BE ESTIMATED
C        M*N < 2000
C        N < 100
C        N*MG < 400
C        MG < 10
C        N1 < 20
C        MG*NMAX < 100
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
      N2=M
      CALL DATA1 (W,N,NA)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,B,N)
      IF (ICOND.EQ.0) CALL WEIGHT (SIGMA,EPS,N,MG)
      CALL INVERT (SIGMA,MG)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,1,U)
      WRITE (3,8) (B(I),I=1,N1)
      WRITE (3,9) UU(1)
C     ITERATION CYCLE BEGINS - FORM DERIVATIVES HERE
      NNN=0
1     CONTINUE
      NNN=NNN+1
      DO 2 I=1,N1
      BN(I)=B(I)
2     BB(I)=B(I)
      CALL DERIVS (DD,D1,U1,N,N1,A,BN,BB,B,W,RES,U,D3,RESM,TEMPM,SIGMA,E
     1PS)
      CALL SEARCH (BB,B,BN,A,D3,W,U,UU,N,N1,RES,RESM,TEMPM,EPS,DD,SIGMA,
```

```
     1U1,&12)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BN,N)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,1,U)
      WRITE (3,8) (BN(I),I=1,N1)
      WRITE (3,9) UU(1)
      IF (NNN.GT.20) GO TO 5
      DO 3 I=1,N1
      BB(I)=B(I)
3     B(I)=BN(I)
      DO 4 I=1,N1
      CONST=DABS((BN(I)-BB(I))/BB(I))
      IF (CONST.GT..005) GO TO 1
4     CONTINUE
5     CONTINUE
12    CONTINUE
      WRITE (3,10)
      WRITE (3,11)
      WRITE (3,8) (BN(I),I=1,N1)
      WRITE (3,9) UU(1)
      CALL DERIVS (DD,D1,U1,N,N1,A,BN,BB,B,W,RES,U,D3,RESM,TEMPM,SIGMA,E
     1PS)
      CALL COVS (BN,A,N1)
      CALL EXIT
C
6     FORMAT (20I4)
7     FORMAT (8F10.0)
8     FORMAT (/' BETA COEFFICIENTS ',5F12.6)
9     FORMAT (/' RESIDUAL SUM OF SQUARES IS',D16.8)
10    FORMAT (/' CONVERGENCE ACHIEVED'/)
11    FORMAT (/' INFORMATION MATRIX'/)
      END


      SUBROUTINE TEST (BN,ZETA,KO)
      IMPLICIT REAL*8(A-H,O-Z)
C
```

```
C
C        SUBROUTINE TEST CONSTRUCTS THE POLYNOMIALS WHOSE ROOTS ARE
C        TO BE TESTED
C
C
      DIMENSION BN(1), WORK(100), COEF(20), ZETA(1)
      CALL INIT (COEF)
      COEF(1)=1.0
      COEF(5)=BN(1)
      NPP=5
      CALL ROOT (NPP,WORK,COEF,ZETA,KO)
      CALL INIT (COEF)
      COEF(6)=BN(2)
      COEF(7)=BN(3)
      NPP=7
      CALL ROOT (NPP,WORK,COEF,ZETA,KO)
      CALL INIT (COEF)
      COEF(2)=BN(4)
      COEF(5)=BN(5)
      COEF(6)=BN(6)
      COEF(7)=BN(7)
      NPP=7
      CALL ROOT (NPP,WORK,COEF,ZETA,KO)
      RETURN
      END


      SUBROUTINE DECIDE (I,NOEQ)
      NOEQ=1
C
C
C        SUBROUTINE DECIDE DETERMINES THE RELATION BETWEEN THE
C        PARAMETERS AND THE EQUATIONS. PARAMETER I IS PASSED TO IT
C        AND THE EQUATION NUMBER IS RETURNED. THIS IS TO AVOID
C        ZERO MULTIPLICATIONS. IF THERE ARE PARAMETERS COMMON TO A
C        NUMBER OF EQUATIONS MODIFICATION TO SURROUTINE DERIVS WOULD
```

```
C        BE REQUIRED
C
C
      IF (I.GT.1) NOEQ=2
      IF (I.GT.3) NOEQ=3
      RETURN
      END


      SUBROUTINE DATA1 (W,N,NA)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DATA1 READS IN ALL DATA AND FORMS THE
C        MATRIX W CONTAINING ALL VARIABLES IN THE SYSTEM
C
C
      COMMON /PARMS/ NQ(10),NMAX,KKMAX,MG,N2
      DIMENSION W(N,N2)
      DIMENSION HOUSE(100), DUR(100), CLOTH(100), FOOD(100), OTHER(100)
      NRDATA=NA
      READ (1,2) (FOOD(I),I=1,NRDATA)
      READ (1,2) (DUR(I),I=1,NRDATA)
      READ (1,2) (OTHER(I),I=1,NRDATA)
      DO 1 I=1,N
      W(I,1)=FOOD(I)-FOOD(I+1)-FOOD(I+4)+FOOD(I+5)
      W(I,2)=DUR(I)-DUR(I+1)-DUR(I+4)+DUR(I+5)
      W(I,3)=OTHER(I)
      W(I,4)=OTHER(I+1)
      W(I,5)=OTHER(I+4)
      W(I,6)=OTHER(I+5)
1     CONTINUE
      RETURN
C
2     FORMAT (8F10.0)
      END
```

```
      SUBROUTINE RESID (REM,TEMPM,EPS,RES,W,U1,BN,N)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE RESID DEFINES THE RESIDUAL VECTOR AND THE M.A.
C        PARAMETERS. THE LATTER ARE STORED IN TEMP
C
C
      COMMON /PARMS/ NQ(10),NMAX,KKMAX,MG,N2
      DIMENSION REM(N,MG), EPS(N,MG), TEMPM(NMAX,MG)
      DIMENSION W(N,N2)
      DIMENSION BN(1), U1(1), RES(1), TEMP(20)
      DO 1 I=1,N
      REM(I,1)=W(I,1)
      REM(I,2)=W(I,2)
      REM(I,3)=W(I,3)-W(I,4)-BN(8)*W(I,5)+BN(8)*W(I,6)
1     CONTINUE
      DO 3 I=1,MG
      DO 2 J=1,NMAX
      TEMPM(J,I)=0.0
2     CONTINUE
3     CONTINUE
      TEMPM(4,1)=BN(1)
      TEMPM(5,2)=BN(2)
      TEMPM(6,2)=BN(3)
      TEMPM(1,3)=BN(4)
      TEMPM(4,3)=BN(5)
      TEMPM(5,3)=BN(6)
      TEMPM(6,3)=BN(7)
      DO 7 J=1,MG
      DO 4 I=1,N
      RES(I)=REM(I,J)
4     CONTINUE
      NQJ=NQ(J)
```

```
      DO 5 I=1,NQJ
      TEMP(I)=TEMPM(I,J)
    5 CONTINUE
      NM=NQ(J)+1
      CALL EPSVEC (U1,NM,RES,TEMP,N)
      DO 6 I=1,N
      EPS(I,J)=U1(I)
    6 CONTINUE
    7 CONTINUE
      RETURN
      END


      SUBROUTINE WEIGHT (SIGMA,EPS,N,MG)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE WEIGHT CONSTRUCTS AN ESTIMATE OF THE COVARIANCE
C        MATRIX OF THE DISTURBANCES
C
C
      DIMENSION SIGMA(MG,MG), EPS(N,MG)
      DO 3 I=1,MG
      DO 2 J=1,MG
      SIGMA(I,J)=0.0
      DO 1 K=1,N
      SIGMA(I,J)=SIGMA(I,J)+EPS(K,I)*EPS(K,J)
    1 CONTINUE
      SIGMA(I,J)=SIGMA(I,J)/N
    2 CONTINUE
    3 CONTINUE
      RETURN
      END


      SUBROUTINE COVS (PARM,A,N1)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE COVS PRINTS OUT THE FINAL PARAMETER ESTIMATES,
C        THE STANDARD DEVIATIONS AND T STATISTICS
C
C
      DIMENSION PARM(1)
      DIMENSION TTEST(20)
      DIMENSION STD(20)
      DIMENSION A(N1,N1)
      DO 1 I=1,N1
      STD(I)=DSQRT(A(I,I))
      TTEST(I)=PARM(I)/STD(I)
1     CONTINUE
      WRITE (3,2)
      WRITE (3,3) (PARM(I),STD(I),TTEST(I),I=1,N1)
      RETURN
C
2     FORMAT (' ',10X,'PARAMETERS',20X,'STANDARD DEVIATION',25X,'T STATI
     1STIC')
3     FORMAT (' ',5X,F16.7,20X,F16.7,20X,F16.7)
      END


      SUBROUTINE TURN (VEC,REST,N)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE TURN ENABLES A VECTOR TO BE REVERSED
C
C
      DIMENSION VEC(1), REST(1)
      DO 1 I=1,N
      REST(I)=VEC(I)
1     CONTINUE
```

```
      DO 2 I=1,N
      VEC(I)=REST(N-I+1)
    2 CONTINUE
      RETURN
      END

      SUBROUTINE SUMSQ (EPS,SIGMA,UU,N,MG,DD,KO,U)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE COMPUTES THE OBJECTIVE FUNCTIOS
C
C
      DIMENSION EPS(N,MG), SIGMA(MG,MG), DD(MG,MG)
      DIMENSION U(MG,MG)
      DIMENSION UU(1)
      DO 3 I=1,MG
      DO 2 J=1,MG
      DD(I,J)=0.0
      DO 1 K=1,N
      DD(I,J)=DD(I,J)+EPS(K,I)*EPS(K,J)
    1 CONTINUE
    2 CONTINUE
    3 CONTINUE
      DO 6 I=1,MG
      DO 5 J=1,MG
      U(I,J)=0.0
      DO 4 K=1,MG
      U(I,J)=U(I,J)+DD(I,K)*SIGMA(J,K)
    4 CONTINUE
    5 CONTINUE
    6 CONTINUE
      UU(KO)=TRACE(U,MG)
      RETURN
      END
```

```
      REAL FUNCTIONTRACE*8(A,K)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        FUNCTION TRACE COMPUTES THE TRACE OF A MATRIX A
C
C
      DIMENSION A(K,K)
      TRACE=0.0
      DO 1 I=1,K
      TRACE=TRACE+A(I,I)
1     CONTINUE
      RETURN
      END

      SUBROUTINE SEARCH (BB,B,BN,A,D3,W,U,UU,N,N1,RES,RESM,TEMPM,EPS,DD,
     1SIGMA,U1,*)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE SEARCH COMPUTES AN OPTIMAL STEP LENGTH BY
C        QUADRATIC INTERPOLATION
C
C
      COMMON /PARMS/ NQ(10),NMAX,KKMAX,MG,N2
      DIMENSION BB(1), B(1), BN(1), A(N1,N1), D3(1), W(1)
      DIMENSION RESM(N,MG), TEMPM(MG,NMAX), EPS(N,MG), SIGMA(MG,MG)
      DIMENSION DD(1)
      DIMENSION ALX(3), U(N,MG), UU(1)
      DIMENSION U1(1)
      DIMENSION RES(1)
      DO 1 I=1,N1
      BB(I)=0.0
```

```
      DO 1 J=1,N1
1     BB(I)=BB(I)-A(I,J)*D3(J)
C     BB IS THE CHOSEN DIRECTION
      VALUE=UU(1)
      ALX(1)=0.0
      ALX(2)=0.5
      ALX(3)=1.0
      DO 2 I=2,3
      CALL ACCUM (BN,B,BB,ALX(I),N1)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BN,N)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,I,U)
      CALL TEST (BN,UU,I)
2     CONTINUE
      IF (UU(2).LE.UU(3)) GO TO 3
      ALX(2)=ALX(3)
      UU(2)=UU(3)
3     KIK=0
      IF (UU(2).GT.UU(1)) GO TO 5
4     CONTINUE
      ALX(3)=2.0*ALX(2)
      CALL ACCUM (BN,B,BB,ALX(3),N1)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BN,N)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,3,U)
      CALL TEST (BN,UU,3)
      IF (UU(3).GT.UU(2)) GO TO 7
      ALX(2)=ALX(3)
      UU(2)=UU(3)
      KIK=KIK+1
      IF (KIK.LT.10) GO TO 4
      RETURN 1
5     ALX(3)=ALX(2)
      UU(3)=UU(2)
      ALX(2)=ALX(3)*0.5
      CALL ACCUM (BN,B,BB,ALX(2),N1)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BN,N)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,2,U)
```

```
      CALL TEST (BN,UU,2)
      IF (UU(3).GT.UU(2).AND.UU(2).LT.UU(1)) GO TO 7
      KIK=KIK+1
      IF (KIK.LT.10) GO TO 5
      ALX(2)=-ALX(2)
      CALL ACCUM (BN,B,BB,ALX(2),N1)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BN,N)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,2,U)
      CALL TEST (BN,UU,2)
      IF (UU(2).LT.UU(1)) GO TO 6
      RETURN 1
    6 KIK=1
      GO TO 4
    7 ALX(1)=0.0
      CALL QUAD (ALX,UU,ATLEN,UMIN)
      ALX(1)=ATLEN
      CALL ACCUM (BN,B,BB,ALX(1),N1)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BN,N)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,1,U)
      CALL TEST (BN,UU,1)
      IF (UU(1).LE.UU(2)) GO TO 8
      ALX(1)=ALX(2)
      CALL ACCUM (BN,B,BB,ALX(1),N1)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BN,N)
      CALL SUMSQ (EPS,SIGMA,UU,N,MG,DD,1,U)
    8 CONTINUE
      COMT=DABS(VALUE-UU(1))
      VALUE=0.0001*VALUE
      IF (COMT.LE.VALUE) RETURN 1
      RETURN
      END


      SUBROUTINE QUAD (X,U,ATLEN)
      IMPLICIT REAL*8(A-H,O-Z)
C
```

```
C
C         SUBROUTINE QUAD FITS A QUADRATIC THROUGH 3 POINTS AND
C         INTERPOLATES THE MINIMUM
C
C
      DIMENSION X(3), U(3)
      IF(U(3).EQ.1.0D35)U(3)=U(1)*1.5
      A=(((U(1)-U(2))*(X(1)-X(3)))-((U(1)-U(3))*(X(1)-X(2))))/((X(1)-X(2
     1))*(X(1)-X(3))*(X(2)-X(3)))
      B=((U(1)-U(2))-(A*(X(1)-X(2))*(X(1)+X(2))))/(X(1)-X(2))
      C=-(A*X(1)*X(1))-(B*X(1))+U(1)
      ATLEN=-B/(A*2)
      ATMIN=-((B*B)/(A*4))+C
      RETURN
      END

      SUBROUTINE ACCUM (BN,B,BB,AL,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C         SUBROUTINE ACCUM UPDATES THE PARAMETER VECTOR BN
C
C
      DIMENSION BN(1), B(1), BB(1)
      DO 1 I=1,N1
      BN(I)=B(I)+AL*BB(I)
1     CONTINUE
      RETURN
      END

      SUBROUTINE EPSVEC (EPSIL,M,RES,ROE,T)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
```

```
C        SUBROUTINE EPSVEC COMPUTES THE WHITE NOISE DISTURBANCE
C
C
      INTEGER*4 T
      DIMENSION EPSIL(1), RES(1), ROE(1)
      N=M-1
      EPSIL(1)=RES(T)
      IF (N.EQ.1) GO TO 3
      DO 2 I=2,N
      KA=I-1
      A=0.0
      DO 1 J=1,KA
      A=A+EPSIL(I-J)*ROE(J)
1     CONTINUE
      L=T-I+1
      EPSIL(I)=RES(L)-A
2     CONTINUE
C     SECOND STAGE
3     DO 5 I=M,T
      BA=0.0
      L=T-I+1
      DO 4 J=1,N
      BA=BA+ROE(J)*EPSIL(I-J)
4     CONTINUE
      EPSIL(I)=RES(L)-BA
5     CONTINUE
C
      DO 6 I=1,T
      RES(I)=EPSIL(I)
6     CONTINUE
      DO 7 I=1,T
      EPSIL(I)=RES(T-I+1)
7     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DERIVS (DD,D1,U1,N,N1,A,BN,BB,B,W,RES,U,D3,RESM,TEMPM,S
     1IGMA,EPS)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DERIVS COMPUTES THE DERIVATIVES REQUIRED BY
C        THE GAUSS-NEWTON ALGORITHM
C
C
      COMMON /PARMS/ NQ(10),NMAX,KKMAX,MG,N2
      DIMENSION BN(1)
      DIMENSION EPS(N,MG), D1(N,MG), DD(N,N1), A(N1,N1), D3(1), U(N,MG)
      DIMENSION U1(1), RESM(N,MG), BB(1), B(1), W(1)
      DIMENSION RES(1), TEMPM(MG,NMAX)
      DIMENSION SIGMA(MG,MG)
      DIMENSION H(7)
      DO 2 I=1,N1
      CONST1=BN(I)*0.01
      CALL GROUP (BB,BN,N1)
      BB(I)=BB(I)-0.005*BB(I)
      CALL RESID (RESM,TEMPM,D1,RES,W,U1,BB,N)
      CALL SUMSQ (D1,SIGMA,H,N,MG,RESM,1,U)
      CALL GROUP (BB,BN,N1)
      BB(I)=BB(I)+0.005*BB(I)
      CALL RESID (RESM,TEMPM,EPS,RES,W,U1,BB,N)
      CALL SUMSQ (EPS,SIGMA,H,N,MG,RESM,2,U)
      CALL GROUP (BB,BN,N1)
      D3(I)=H(2)-H(1)
      D3(I)=D3(I)/CONST1
      CALL DECIDE (I,NOEQ)
      DO 1 J=1,N
      DD(J,I)=EPS(J,NOEQ)-D1(J,NOEQ)
      DD(J,I)=DD(J,I)/CONST1
1     CONTINUE
2     CONTINUE
```

```
      DO 5 I=1,N1
      DO 4 J=1,I
      CALL DECIDE (I,NE1)
      CALL DECIDE (J,NE2)
      A(I,J)=0.0
      DO 3 K=1,N
      A(I,J)=A(I,J)+DD(K,I)*DD(K,J)*SIGMA(NE1,NE2)
    3 CONTINUE
      A(J,I)=A(I,J)
    4 CONTINUE
    5 CONTINUE
      CALL ADD (A,N1)
      CALL INVERT (A,N1)
      RETURN
      END

      SUBROUTINE ROOT (KP,MAT,COEF,ZETA,KO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE ROOT PERFORMS THE ROUTH-HURWICZ TEST
C
C
      REAL*8 MAT
      DIMENSION MAT(KP,KP), COEF(1), ZETA(1)
      DO 1 I=1,KP
      MAT(1,I)=COEF(I)
    1 CONTINUE
      DO 3 I=2,KP
      DO 2 J=I,KP
      IJ=J-I
      MAT(I,J)=MAT(I-1,I-1)*MAT(I-1,J-1)-MAT(I-1,KP)*MAT(I-1,KP-IJ)
    2 CONTINUE
    3 CONTINUE
      DO 4 I=1,KP
```

```
      IF(MAT(I,I).LT.0.0D0)ZETA(KO)=1.0D35
4     CONTINUE
      RETURN
      END


      SUBROUTINE ADD (G,N1)
C
C
C        SUBROUTINE ADD WEIGHTS THE WEIGHTING MATRIX BY THE SHANNO
C        SCHEME
C
C
      REAL*8 G
      DIMENSION G(N1,N1), B(100)
      REAL*4 LAMBDA
      JK=0
      DO 2 I=1,N1
      DO 1 J=1,I
      JK=JK+1
      B(JK)=G(J,I)
1     CONTINUE
2     CONTINUE
      CALL EIGEN (B,R,N1,1)
      IK=N1*(N1+1)/2
      EMIN=B(IK)
      LAMBDA=-5*EMIN
      EMAX=B(1)
      EE=EMAX/EMIN
      WRITE (3,5) EE
      IF (LAMBDA.LE.0.0) GO TO 4
      DO 3 I=1,N1
      G(I,I)=G(I,I)+LAMBDA
3     CONTINUE
4     RETURN
C
```

```
5     FORMAT (' ',5X,'RATIO OF MAXIMUM TO MINIMUM EIGENVALUE IS',D13.6)
      END

      SUBROUTINE INIT (COEF)
C
C
C        SUBROUTINE INIT INITIALIZES COEF
C
C
      REAL*8 COEF
      DIMENSION COEF(1)
      DO 1 I=1,20
      COEF(I)=0.0
1     CONTINUE
      COEF(1)=1.0
      RETURN
      END

      SUBROUTINE GROUP (BB,BN,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE GROUP INITIALIZES BB
C
C
      DIMENSION BN(1), BB(1)
      DO 1 I=1,N1
      BB(I)=BN(I)
1     CONTINUE
      RETURN
      END

      SUBROUTINE INVERT (A,N)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE INVERT INVERTS A MATRIX A
C
C
      DIMENSION A(N,N), IPIV(20)
      DO 1 I=1,N
1     IPIV(I)=0
      DO 11 I=1,N
      AMAX=0.
      DO 5 J=1,N
      IF (IPIV(J)) 2,2,5
2     IF (DABS(A(J,J)-AMAX)) 4,4,3
3     ICOL=J
      AMAX=DABS(A(J,J))
4     CONTINUE
5     CONTINUE
      IPIV(ICOL)=1
      IF (AMAX-1.0D-50) 6,6,7
6     WRITE (3,12)
      STOP
7     CONTINUE
      AMAX=A(ICOL,ICOL)
      A(ICOL,ICOL)=1.0
      DO 8 K=1,N
8     A(ICOL,K)=A(ICOL,K)/AMAX
      DO 11 J=1,N
      IF (J-ICOL) 9,11,9
9     AMAX=A(J,ICOL)
      A(J,ICOL)=0.
      DO 10 K=1,N
10    A(J,K)=A(J,K)-A(ICOL,K)*AMAX
11    CONTINUE
      RETURN
C
```

```
12    FORMAT (/' SINGULAR MATRIX -- TERMINATE'/)                    T  38
      END                                                           T  39-
```

PROGRAM 4

USE: This program estimates a single ARMA equation in the frequency domain. It is a numerical derivative Newton-Raphson algorithm for the solution of non-linear equations.

EXAMPLE: The model estimated is $(1-L)(1-L^4)y(t) = (1+\alpha_4 L^4)e(t)$.

STRUCTURE: Only two subroutines need changing.

DATA3 - All data is read in and any differencing required is performed. The differenced variable is stored in YSTAR; otherwise the modifications are as for DATA1 in PROGRAM 1.

FORM - The relation between the parameters stored in BN and the A.R. (BETA) and M.A. (ROE) parameters is defined. Card C 17 defines the fourth order M.A. of the example.

CARD INPUT:

Card 1	cols 1-4	Number of series for analysis.
Card 2	cols 1-4	Number of observations available.
	cols 4-8	Number of observations used.
	cols 8-12	Order of A.R. This must be set at 1 if the model has only an M.A. part.
	cols 12-16	Order of M.A. This must be set at 1 if the model has only an A.R. part.
	cols 16-20	Number of parameters to be estimated.
	cols 20-24	Number of frequency bands contained between 0 and π.
Card 3	Initial estimates of the parameters. There must be the same number as in cols. 16-20 of card 2 and no value may be zero.	

All cards following will be data.

SUBROUTINES
REQUIRED: EIGEN from I.B.M. Scientific Subroutine Package.

PROGRAM
LENGTH: 71 K (bytes).

```
C       GENERAL NON-LINEAR LEAST SQUARES FOR SINGLE EQUATIONS
C       USING THE NEWTON-RAPHSON ALGORITHM
C       AITKEN ESTIMATOR IN THE FREQUENCY DOMAIN
C       ARMA EQUATION VERSION
C       THE ARMA MODEL IS B(L)Y(T)=A(L)E(T)
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 P,Q,T,TA
      DIMENSION ARTRAN(150), AMTRAN(150), W(150)
      DIMENSION BN(20), BB(20), B(20), BOW(20)
      DIMENSION COSW(30), SINW(30)
      DIMENSION ROE(60), BETA(60)
      DIMENSION G(100), R(10)
      DIMENSION YSTAR(?00), ZETA(3), REST(300), A(350), STD(10)
      DIMENSION TRANS(150)
      DIMENSION PER(150)
      READ(1,6)NCRIT
1     READ(1,6)IMD,T,P,Q,N1,M
      READ (1,7) (BOW(I),I=1,N1)
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C       INPUT PARAMETERS
C
C
C       IMD        THE TOTAL NUMBER OF OBSERVATIONS IN Y(T)
C       P          THE ORDER OF THE AUTOREGRESSION. THIS MUST ALWAYS
C                  BE SET AT UNITY AT LEAST. IF THE MODEL IS MERELY A
C                  MOVING AVERAGE THEN P=1 IS REQUIRED.
C       T          THE NUMBER OF OBSERVATIONS ON Y(T) USED. THIS
C                  WILL NORMALLY BE IMD-P.
C       Q          ORDER OF THE MOVING AVERAGE. THE SAME RULES APPLY AS
C                  FOR P.
C       NCRIT      THE NUMBER OF SERIES FOR ANALYSIS
C       N1         THE TOTAL NUMBER OF PARAMETERS
C       M          THE NUMBER OF FREQUENCY BANDS
```

```
C          BOW          A VECTOR OF INITIAL ESTIMATES FOR THE PARAMETERS
C                       NO VALUE MAY BE 0.0.
C          THE MAXIMUM SIZE OF THE ARMA MODEL THAT MAY BE ESTIMATED
C          IS GOVERNED BY THE FOLLOWING CONSTRAINTS
C          IMD < 300
C          T < 300
C          P < 30
C          Q < 30
C          N1 < 17
C          M < 150
C     -----------------------------------------------------------------
C     -----------------------------------------------------------------
      NOR=0
      KO=1
      NOR=NOR+1
      DO 2 I=1,N1
      B(I)=BOW(I)
2     CONTINUE
      WRITE (3,9) P,Q
      WRITE (3,8) T
      KO=KO
      NA=IMD
      N=T
      CALL DATA3 (YSTAR,REST,N,NA)
      CALL PERIOD (YSTAR,W,M,N,PER,COSW,SINW)
      DO 3 IJK=1,N1
      BN(IJK)=B(IJK)
      BB(IJK)=B(IJK)
3     CONTINUE
      WRITE (3,11) (BN(I),I=1,N1)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,ZETA,ARTRAN,AMTR
     1AN,PER,1)
      WRITE (3,10) ZETA(1)
      DO 5 IJ=1,20
      CALL DERIVS (BB,BN,R,G,N1,ROE,BETA,P,Q,T,KO,TRANS,W,M,COSW,SINW,ZE
```

```
     1TA,ARTRAN,AMTRAN,PER)
      CALL INVERT (G,N1)
      CALL SEARCH (BB,B,BN,G,R,W,A,ZETA,T,N1,ROE,P,Q,BETA,T,KO,TRANS,M,C
     1OSW,SINW,ARTRAN,AMTRAN,PER,&14)
      WRITE (3,11) (BN(I),I=1,N1)
      WRITE (3,10) ZETA(1)
      DO 4 IJK=1,N1
      B(IJK)=BN(IJK)
      BB(IJK)=BN(IJK)
4     CONTINUE
5     CONTINUE
   14 CONTINUE
      CALL COVAR (G,T,N1,STD,ZETA)
      WRITE (3,11) (BN(I),I=1,N1)
      WRITE (3,12) (STD(I),I=1,N1)
      WRITE (3,13) ZETA(1)
      IF (NOR.LT.NCRIT) GO TO 1
      CALL EXIT
C
6     FORMAT (20I4)
7     FORMAT (8F10.0)
8     FORMAT ('0',45X,'NUMBER OF OBSERVATIONS'/55X,I4)
9     FORMAT ('0',30X,'ORDER OF AUTOREGRESSION',30X,'ORDER OF MOVING
     1 AVERAGE'/40X,I4,50X,I4)
10    FORMAT ('0',5X,'RESIDUAL SUM OF SQUARES IS',D13.6)
11    FORMAT (' ',5X,'PARAMETERS',5F20.6)
12    FORMAT ('0',5X,'STAND DEVS',5F20.6)
13    FORMAT ('0',5X,'STANDARD ERROR OF THE ESTIMATE IS',D13.6)
      END


      SUBROUTINE FORM (BN,ROE,BETA,KKK,N,KO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE FORM DEFINES THE RELATION BETWEEN THE
```

```
C        A.R. PARAMETERS(BETA),THE M.A. PARAMETERS(ALPHA) AND THE
C        PARAMETER VECTOR BN
C
C
      DIMENSION BETA(KKK), ROE(N), BN(1)
      DO 1 I=1,KKK
      BETA(I)=0.0
1     CONTINUE
      DO 2 I=1,N
      ROE(I)=0.0
2     CONTINUE
      ROE(4)=BN(1)
      RETURN
      END


      SUBROUTINE DATA3 (YSTAR,REST,N,NA)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        THE VECTOR Y(T) IS READ IN IN DATA3 AND ANY
C        DIFFERENCING THAT IS REQUIRED IS PERFORMED
C
C
      DIMENSION REST(1), YSTAR(1)
      READ (1,2) (YSTAR(I),I=1,NA)
      DO 1 I=1,NA
      YSTAR(I)=YSTAR(I)-YSTAR(I+1)-YSTAR(I+4)+YSTAR(I+5)
1     CONTINUE
      RETURN
C
2     FORMAT (8F10.0)
      END


      SUBROUTINE TEST (ROE,BETA,P,Q,K,ZETA,KO,BN)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE TEST SETS UP THE A.R. AND M.A. POLYNOMIALS
C        THE ROOTS OF THESE POYNOMIALS ARE TESTED IN ROOT
C        THE MAXIMUM ORDER OF POLYNCMIAL THAT MAY BE TESTED IS
C        DETERMINED BY THE DIMENSION OF POL. IF THE POLYNOMIAL IS OF
C        ORDER M THEN POL MUST BE DIMENSIONED AT (M+1)**2
C        AT A MINIMUM
C
C
      INTEGER*4 P,Q,PP
      DIMENSION BETA(P), ROE(Q), ZETA(1)
      DIMENSION BN(1)
      DIMENSICN COEF(30), POL(900)
      COEF(1)=1.0
      DO 1 I=1,P
      COEF(I+1)=-BETA(I)
1     CONTINUE
      NP=P+1
      CALL ROOT (NP,POL,COEF,ZETA,KO)
      DO 2 I=1,Q
      COEF(I+1)=ROE(I)
2     CONTINUE
      NP=Q+1
      CALL ROOT (NP,PCL,COEF,ZETA,KO)
      RETURN
      END

      SUBROUTINE COVAR (G,T,N1,STD,ZETA)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION G(N1,N1), ZETA(1), STD(1)
C
C
C        SUBROUTINE COVAR COMPUTES THE COVARIANCE MATRIX
```

```
C
C
      INTEGER*4 T
      PI=3.14159265
      ZETA(1)=ZETA(1)/(T-N1)
      ZETA(1)=ZETA(1)*2*PI
      DO 1 I=1,N1
      G(I,I)=G(I,I)*2*ZETA(1)
      G(I,I)=G(I,I)/(2*PI)
      STD(I)=DSQRT(G(I,I))
1     CONTINUE
      RETURN
      END


      SUBROUTINE RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,ZETA,ARTRA
     1N,AMTRAN,PER,EN)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE RESID COMPUTES THE SUM OF SQUARES IN THE FREQUENCY
C        DOMAIN
C
C
      INTEGER*4 T,P,Q,TA
      DIMENSION INDEX(30)
      DIMENSION BETA(P), ROE(Q)
      DIMENSION AMTRAN(M), ARTRAN(M), PER(M)
      DIMENSION TRANS(1), W(1), POLY(30), COSW(1), SINW(1), ZETA(1)
      K9=KO
      DO 1 I=1,P
      POLY(I)=-BETA(I)
1     CONTINUE
      CALL MAKE (POLY,BETA,KO,P,INDEX)
      CALL TRANSF (TRANS,POLY,P,W,M,COSW,SINW,INDEX)
      DO 2 I=1,M
```

```
      ARTRAN(I)=TRANS(I)
2     CONTINUE
      DO 3 I=1,Q
      POLY(I)=ROE(I)
3     CONTINUE
      CALL MAKE (POLY,ROE,KO,Q,INDEX)
      CALL TRANSF (TRANS,POLY,Q,W,M,COSW,SINW,INDEX)
      DO 4 I=1,M
      AMTRAN(I)=TRANS(I)
4     CONTINUE
      ZETA(LN)=0.0
      DO 5 I=2,M
      ZETA(LN)=ZETA(LN)+(PER(I)*ARTRAN(I)/AMTRAN(I))
5     CONTINUE
      BB=PER(1)*ARTRAN(1)/AMTRAN(1)
      BBB=PER(M)*ARTRAN(M)/AMTRAN(M)
      ZETA(LN)=2*ZETA(LN)
      ZETA(LN)=ZETA(LN)+BB
      RETURN
C
      END


      SUBROUTINE DERIVS (BB,BN,D1,D2,N1,ROE,BETA,P,Q,T,KO,TRANS,W,M,COSW
     1,SINW,ZETA,ARTRAN,AMTRAN,PER)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DERIVS COMPUTES THE FIRST AND SECOND DERIVATIVES
C        REQUIRED BY THE NEWTON-RAPHSON ALGORITHM
C
C
      INTEGER*4 T,P,Q
      DIMENSION ZETA(1), ARTRAN(1), AMTRAN(1), PER(1)
      DIMENSION H(5)
      DIMENSION BETA(1), ROE(1), TRANS(1), W(1), COSW(1), SINW(1)
```

```
      DIMENSION BN(1), BB(1), D2(N1,N1), D1(1)
      DO 2 I=1,N1
      DO 1 J=1,I
      CONST1=0.01*BN(I)
      CONST2=BN(J)*0.01
      CALL REFORM (BB,BN,N1)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,H,ARTRAN,AMTRAN,
     1PER,1)
      CALL REFORM (BB,BN,N1)
      BB(I)=BB(I)+0.01*BB(I)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,H,ARTRAN,AMTRAN,
     1PER,2)
      CALL REFORM (BB,BN,N1)
      BB(J)=BB(J)+0.01*BB(J)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,H,ARTRAN,AMTRAN,
     1PER,3)
      CALL REFORM (BB,BN,N1)
      BB(I)=BB(I)-0.01*BB(I)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,H,ARTRAN,AMTRAN,
     1PER,4)
      CALL REFORM (BB,BN,N1)
      BB(J)=BB(J)+0.01*BB(J)
      BB(I)=BB(I)+0.01*BB(I)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,H,ARTRAN,AMTRAN,
     1PER,5)
      D2(I,J)=H(1)-H(3)-H(2)+H(5)
      IF(I.EQ.J)D2(I,I)=H(4)-H(1)+H(2)-H(1)
      D2(I,J)=D2(I,J)/(CONST1*CONST2)
      D1(I)=H(2)-H(1)
      D1(I)=D1(I)/CONST1
      D2(J,I)=D2(I,J)
```

```
1     CONTINUE                                                          G  50
2     CONTINUE                                                          G  51
      CALL WEIGHT (D2,N1)                                               G  54
      RETURN                                                            G  55
C                                                                       G  56
3     FORMAT (' ',5X,8E13.6)                                            G  57
      END                                                               G  58-

      SUBROUTINE WEIGHT (G,N1)                                          H   1
C                                                                       H   2
C                                                                       H   3
C        SUBROUTINE WEIGHT WEIGHTS THE WEIGHTING MATRIX BY  THE SHANNO  H   4
C        SCHEME                                                         H   5
C                                                                       H   6
C                                                                       H   7
      REAL*8 G                                                          H   8
      DIMENSION G(N1,N1), B(50)                                         H   9
      REAL*4 LAMBDA                                                     H  10
      JK=0                                                              H  11
      DO 2 I=1,N1                                                       H  12
      DO 1 J=1,I                                                        H  13
      JK=JK+1                                                           H  14
      B(JK)=G(J,I)                                                      H  15
1     CONTINUE                                                          H  16
2     CONTINUE                                                          H  17
      CALL EIGEN (B,R,N1,1)                                             H  18
      IK=N1*(N1+1)/2                                                    H  19
      EMIN=B(1)
      EMAX=B(IK)
      LAMBDA=-5*EMIN                                                    H  21
      E=EMAX/EMIN
      WRITE(3,5)E                                                       H  22
      IF (LAMBDA.LE.0.0) GO TO 4                                        H  23
      DO 3 I=1,N1                                                       H  24
      G(I,I)=G(I,I)+LAMBDA                                              H  25
```

```
3     CONTINUE
4     RETURN
C
5     FORMAT( ',',5X,'RATIO OF MAXIMUM TO MINIMUM EIGENVALUE IS',D13.6)
      END


      SUBROUTINE REFORM (BB,BN,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE INITIALIZES BB AT BN
C
C
      DIMENSION BB(1), BN(1)
      DO 1 I=1,N1
      BB(I)=BN(I)
1     CONTINUE
      RETURN
      END


      SUBROUTINE SEARCH (BB,B,BN,A,D3,W,U,UU,N,N1,ROE,P,Q,BETA,T,KO,TRAN
     1S,M,COSW,SINW,ARTRAN,AMTRAN,PER,*)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE SEARCH FINDS AN OPTIMAL STEP LENGTH BY QUADRATIC
C        INTERPOLATION
C
C
      INTEGER*4 P,Q,T
      DIMENSION ROE(1), BETA(1), TRANS(1), W(1), COSW(1), SINW(1)
      DIMENSION ARTRAN(1), AMTRAN(1), PER(1)
      DIMENSION BB(1), B(1), BN(1), A(N1,N1), D3(1), ALX(3)
      DIMENSION U(1), UU(3)
```

```
      COM=UU(1)
      DO 1 I=1,N1
      BB(I)=0.0
      DO 1 J=1,N1
1     BB(I)=BB(I)-A(I,J)*D3(J)
C     BB IS THE CHOSEN DIRECTION
      ALX(1)=0.0
      ALX(2)=0.5
      ALX(3)=1.0
      DO 2 I=2,3
      CALL ACCUM (BN,B,BB,ALX(I),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,UU,ARTRAN,AMTRAN
     1,PER,I)
      CALL TEST (ROE,BETA,P,Q,KO,UU,I,BN)
2     CONTINUE
      IF (UU(2).LE.UU(3)) GO TO 3
      ALX(2)=ALX(3)
      UU(2)=UU(3)
3     KIK=0
      IF (UU(2).GT.UU(1)) GO TO 5
4     CONTINUE
      ALX(3)=2.0*ALX(2)
      CALL ACCUM (BN,B,BB,ALX(3),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,UU,ARTRAN,AMTRAN
     1,PER,3)
      CALL TEST (ROE,BETA,P,Q,KO,UU,3,BN)
      IF (UU(3).GT.UU(2)) GO TO 7
      ALX(2)=ALX(3)
      UU(2)=UU(3)
      KIK=KIK+1
      IF (KIK.LT.10) GO TO 4
      RETURN 1
5     ALX(3)=ALX(2)
      UU(3)=UU(2)
```

```
      ALX(2)=ALX(3)*0.5
      CALL ACCUM (BN,B,BB,ALX(2),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,UU,ARTRAN,AMTRAN
     1,PER,2)
      CALL TEST (ROE,BETA,P,Q,KO,UU,2,BN)
      IF (UU(3).GT.UU(2).AND.UU(2).LT.UU(1)) GO TO 7
      KIK=KIK+1
      IF (KIK.LT.10) GO TO 5
      ALX(2)=-ALX(2)
      CALL ACCUM (BN,B,BB,ALX(2),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,UU,ARTRAN,AMTRAN
     1,PER,2)
      CALL TEST (ROE,BETA,P,Q,KO,UU,2,BN)
      IF (UU(2).LT.UU(1)) GO TO 6
      RETURN 1
6     KIK=1
      GO TO 4
7     ALX(1)=0.0
      CALL QUAD (ALX,UU,ATLEN,UMIN)
      ALX(1)=ATLEN
      CALL ACCUM (BN,B,BB,ALX(1),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,UU,ARTRAN,AMTRAN
     1,PER,1)
      CALL TEST (ROE,BETA,P,Q,KO,UU,1,BN)
      IF (UU(1).LE.UU(2)) GO TO 8
      ALX(1)=ALX(2)
      CALL ACCUM (BN,B,BB,ALX(1),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,COSW,SINW,UU,ARTRAN,AMTRAN
     1,PER,1)
8     CONTINUE
      COMM=DABS(COM-UU(1))
      COM=COM*0.0001
```

```
      IF (COMM.LT.COM) RETURN 1                                          J  87
      RETURN                                                             J  88
      END                                                                J  89-


      SUBROUTINE ACCUM (BN,B,BB,AL,N1)                                   K   1
      IMPLICIT REAL*8(A-H,O-Z)                                           K   2
C                                                                        K   3
C                                                                        K   4
C        SUBROUTINE ACCUM UPDATES THE PARAMETER VECTOR                   K   5
C                                                                        K   6
C                                                                        K   7
      DIMENSION BN(1), B(1), BB(1)                                       K   8
      DO 1 I=1,N1                                                        K   9
      BN(I)=B(I)+AL*BB(I)                                                K  10
1     CONTINUE                                                           K  11
      RETURN                                                             K  13
      END                                                                K  16-


      SUBROUTINE QUAD (ALX,UU,AL,UMIN)                                   L   1
      IMPLICIT REAL*8(A-H,O-Z)                                           L   2
C                                                                        L   3
C                                                                        L   4
C        SUBROUTINE QUAD FITS A QUADRATIC THROUGH THREE POINTS           L   5
C        AND INTERPOLATES A MINIMUM                                      L   6
C                                                                        L   7
C                                                                        L   8
      DIMENSION ALX(3), UU(3)                                            L   9
      IF(UU(3).EQ.1.0D35)UU(3)=1.5*UU(1)                                 L  10
      A=(((UU(1)-UU(2))*(ALX(1)-ALX(3)))-((UU(1)-UU(3))*(ALX(1)-ALX(2))) L  11
     1)/((ALX(1)-ALX(2))*(ALX(1)-ALX(3))*(ALX(2)-ALX(3)))                L  12
      B=((UU(1)-UU(2))-(A*(ALX(1)-ALX(2))*(ALX(1)+ALX(2))))/(ALX(1)-ALX( L  13
     12))                                                                L  14
      C=-(A*ALX(1)*ALX(1))-(B*ALX(1))+UU(1)                              L  15
      AL=-B/(A*2)                                                        L  16
```

```
      UMIN=-((B*B)/(A*4))+C
      RETURN
C
1     FORMAT (6D16.8)
      END


      SUBROUTINE ROOT (KP,MAT,COEF,ZETA,KO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE ROOT PERFORMS THE ROUTH-HURWICZ TEST
C
C
      REAL*8 MAT
      DIMENSION MAT(KP,KP), COEF(1), ZETA(1)
      DO 1 I=1,KP
      MAT(1,I)=COEF(I)
1     CONTINUE
      DO 3 I=2,KP
      DO 2 J=I,KP
      IJ=J-I
      MAT(I,J)=MAT(I-1,I-1)*MAT(I-1,J-1)-MAT(I-1,KP)*MAT(I-1,KP-IJ)
2     CONTINUE
3     CONTINUE
      DO 4 I=1,KP
      IF(MAT(I,I).LT.0.0D0)ZETA(KO)=1.0D35
4     CONTINUE
      RETURN
      END


      SUBROUTINE INVERT (A,N)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
```

```
C        SUBROUTINE INVERT INVERTS A MATRIX A
C
C
      DIMENSION A(N,N), IPIV(20)
      DO 1 I=1,N
1     IPIV(I)=0
      DO 11 I=1,N
      AMAX=0.
      DO 5 J=1,N
      IF (IPIV(J)) 2,2,5
2     IF (DABS(A(J,J)-AMAX)) 4,4,3
3     ICOL=J
      AMAX=DABS(A(J,J))
4     CONTINUE
5     CONTINUE
      IPIV(ICOL)=1
      IF (AMAX-1.0D-50) 6,6,7
6     WRITE (3,12)
      STOP
7     CONTINUE
      AMAX=A(ICOL,ICOL)
      A(ICOL,ICOL)=1.0
      DO 8 K=1,N
8     A(ICOL,K)=A(ICOL,K)/AMAX
      DO 11 J=1,N
      IF (J-ICOL) 9,11,9
9     AMAX=A(J,ICOL)
      A(J,ICOL)=0.
      DO 10 K=1,N
10    A(J,K)=A(J,K)-A(ICOL,K)*AMAX
11    CONTINUE
      RETURN
C
12    FORMAT (/' SINGULAR MATRIX -- TERMINATE'/)
      END
```

```
      SUBROUTINE TRANSF (TRANS,POLY,OP,W,M,COSW,SINW,INDEX)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE FORMS THE SQUARED GAIN FUNCTION
C
C
      INTEGER*4 OP
      DIMENSION TRANS(M), POLY(OP), W(M)
      DIMENSION SINW(OP)
      DIMENSION INDEX(1)
      DIMENSION COSW(OP)
      DO 2 I=1,M
      A=0.0
      B=0.0
      DO 1 J=1,OP
      IF (INDEX(J).EQ.0) GO TO 1
      AW=W(I)*J
      COSW(J)=DCOS(AW)
      SINW(J)=DSIN(AW)
      A=A+POLY(J)*COSW(J)
      B=B+POLY(J)*SINW(J)
1     CONTINUE
      A=1.0+A
      A=A**2
      B=B**2
      TRANS(I)=A+B
2     CONTINUE
C
      RETURN
      END

      SUBROUTINE MAKE (POLY,ROE,KO,IQ,INDEX)
C
```

```
C
C         SUBROUTINE MAKE DETERMINES THE NON-ZERO PARAMETERS IN A
C         TRANFER FUNCTION AND STORES THEIR POSITION IN INDEX. THIS
C         AVOIDS MULTIPLICATIONS BY ZEROES
C
C
      REAL*8 POLY,RCE
      DIMENSION POLY(1), ROE(1), INDEX(1)
      DO 1 I=1,IQ
      INDEX(I)=0.0
1     CONTINUE
      DO 3 I=1,IQ
      IF (ROE(I).EQ.0.0) GO TO 2
      INDEX(I)=1
2     CONTINUE
3     CONTINUE
      RETURN
      END

      SUBROUTINE PERIOD (X,W,M,N,PERX,COSW,SINW)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C         SUBROUTINE PERIOD COMPUTES THE PERIODOGRAM FROM
C         AN INPUT VECTOR X
C
C
      DIMENSION X(N), W(M), PERX(M), COSW(1), SINW(1)
      PI=0.314159265358979D1
      DD=2*PI/N
      DO 1 J=1,M
      W(J)=(J-1)*DD
1     CONTINUE
      DO 3 I=1,M
      A=0.0
```

```
      B=0.0
      DO 2 J=1,N
      AW=W(I)*J
      COSWJ=DCOS(AW)
      SINWJ=DSIN(AW)
      A=A+X(J)*COSWJ
      B=B+X(J)*SINWJ
2     CONTINUE
      A=A**2
      B=B**2
      PERX(I)=A+B
      PERX(I)=PERX(I)/(2*PI*N)
3     CONTINUE
      RETURN
      END
```

PROGRAM 5

USE: This program estimates a system of ARMA equations in the frequency domain by the application of a numerical derivative Gauss-Newton algorithm to the solution of non-linear equations.

EXAMPLE: The model estimated is a 3 equation one.

1. $(1-L)(1-L^4)y_1(t) = (1+\alpha_4 L^4)e_1(t)$

2. $(1-L)(1-L^4)y_2(t) = (1+\alpha_5 L^5+\alpha_6 L^6)e_2(t)$

3. $(1-L)(1-\beta_4 L^4)y_3(t) = (1+\alpha_1 L+\alpha_4 L^4+\alpha_5 L^5+\alpha_6 L^6)e_3(t)$

STRUCTURE: Four subroutines must be changed.

DATA3 - As for PROGRAM 4.

FORM - As for PROGRAM 4.

TEST - As for PROGRAM 1.

DECIDE - As for PROGRAM 3.

CARD INPUT: Card 1

Card 1	cols 1-4	Number of observations available.
	cols 4-8	Number of observations used.
	cols 8-12	Maximum order A.R. in the system. If there are only M.A. components this must be 1.
	cols 12-16	Maximum order M.A. in the system. If there are only A.R. components this must be 1.
	cols 16-20	Number of parameters to be estimated.
	cols 20-24	Number of equations.

Card 2 The covariance matrix of the disturbances must be read in under format 8F10.0.

Card 3 The order of the A.R. and M.A. of each equation is read in under format 20I4. This is done equation by equation i.e. the card would be punched

cols 1-4 Order of A.R. of the first equation.

cols 4-8 Order of M.A. of the first equation.

cols 8-12 Order of A.R. of the second equation.

cols 12-16 Order of M.A. of the second equation.

.
.
.
.

As with the maximum orders each value must always be at least unity.

Card 4 Initial estimates of the parameters. There must be the same number as punched in cols 16-20 of card 1 and no value may be zero. The card is read under format 8F10.0.

All cards following will contain data.

SUBROUTINES REQUIRED: None.

PROGRAM LENGTH: 115 K (bytes).

```
C     GAUSS-NEWTON ALGORITHM FOR THE SOLUTION OF SYSTEMS OF EQUATIONS
C     AITKEN ESTIMATOR IN THE FREQUENCY DOMAIN
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX*16 ARTRAN,AMTRAN,TRANS,PER
      COMPLEX*16 WORK1,WORK2
      INTEGER*4 P,Q,T,TA
      DIMENSION ARTRAN(400), AMTRAN(400), W(100)
      DIMENSION WORK1(400), WORK2(1500)
      DIMENSION STD(20), B(20), BN(20), BB(20)
      DIMENSION ROE(60), BETA(60)
      DIMENSION G(400), R(20)
      DIMENSION BOW(20)
      DIMENSION YSTAR(400), ZETA(3), REST(100), A(100)
      DIMENSION PER(400), TRANS(100)
      COMMON /PARMS/ NP(10),NQ(10),MG
      COMMON /PASS/ SIGMA(100)
      READ (1,5) IMD,T,P,Q,N1,MG
      NGG=MG**2
      READ (1,6) (SIGMA(I),I=1,NGG)
      READ (1,5) (NP(I),NQ(I),I=1,MG)
      READ (1,6) (BOW(I),I=1,N1)
C     ----------------------------------------------------------------------
C     ----------------------------------------------------------------------
C     INPUT PARAMETERS
C     IMD        TOTAL NUMBER OF OBSERVATIONS AVAILABLE
C     T          TOTAL NUMBER OF OBSERVATIONS USED AFTER
C                LAGGED VALUES ARE ACCOUNTED FOR
C     P          MAXIMUM ORDER A.R. IN THE SYSTEM
C     Q          MAXIMUM ORDER M.A. IN THE SYSTEM
C     N1         TOTAL NUMBER OF PARAMETERS
C     MG         NUMBER OF EQUATIONS
C     NP         AN MG*1 VECTOR INDICATING THE ORDER OF THE A.R.
C                IN EACH EQUATION. THE ORDER MUST ALWAYS BE SET AT
C                UNITY AS A MINIMUM
```

```
C       NQ          A MG*1 VECTOR INDICATING THE ORDER OF THE M.A.
C                   IN EACH EQUATION. THE ORDER OF EACH EQUATION
C                   MUST BE AT LEAST UNITY
C       SIGMA       THE COVARIANCE MATRIX OF DISTURBANCES
C       BOW         THE INITIAL PARAMETER ESTIMATES
C
C
C       THE SIZE OF SYSTEM THAT MAY BE ESTIMATED IS GOVERNED BY
C       THE FOLLOWING CONSTRAINTS
C       T*MG < 400
C       T*N1 < 1500
C       N1 < 20
C       NQ1 < 30
C       NP1 < 30
C       T < 100
C       MG < 10
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
      CALL INVERT (SIGMA,MG)
      PI=3.14159265
      WRITE (3,8) P,Q
      WRITE (3,7) T
      KO=1
      DO 1 IJ=1,N1
      B(IJ)=BOW(IJ)
1     CONTINUE
      KO=KO
      NA=IMD
      N=T
      M=N
      PI=0.314159265358979D1
      CALL DATA3 (YSTAR,REST,N,NA,MG)
      CALL CON (MG,M,YSTAR,REST,A,W,N,PER)
      MM=Q+1
      DO 2 IJK=1,N1
      BN(IJK)=B(IJK)
```

```
      BB(IJK)=B(IJK)                                                    A  75
2     CONTINUE                                                          A  76
      WRITE (3,10) (BN(I),I=1,N1)
      WRITE (3,9) ZETA(1)
      CALL FORM (BB,ROE,BETA,P,Q,KO)                                    A  77
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,ZETA,ARTRAN,AMTRAN,PER,1) A  78
      DO 4 IJ=1,20                                                      A  79
      CALL DERIVS (BB,BN,R,G,N1,ROE,BETA,P,Q,T,KO,TRANS,W,M,ZETA,ARTRAN, A  80
     1AMTRAN,PER,WORK1,WORK2,SIGMA)                                     A  81
      CALL INVERT (G,N1)                                                A  82
      CALL SEARCH (BB,B,BN,G,R,W,A,ZETA,T,N1,ROE,P,Q,BETA,T,KO,TRANS,M,A A  83
     1RTRAN,AMTRAN,PER,&12)                                             A  84
      WRITE (3,10) (BN(I),I=1,N1)                                       A  85
      WRITE (3,9) ZETA(1)                                               A  86
      DO 3 IJK=1,N1                                                     A  87
      B(IJK)=BN(IJK)                                                    A  88
      BB(IJK)=BN(IJK)                                                   A  89
3     CONTINUE                                                          A  90
4     CONTINUE                                                          A  91
12    CONTINUE
      CALL DERIVS (BB,BN,R,G,N1,ROE,BETA,P,Q,T,KO,TRANS,W,M,ZETA,ARTRAN, A  92
     1AMTRAN,PER,WORK1,WORK2,SIGMA)                                     A  93
      CALL COVAR (G,T,N1,STD,ZETA,BN)                                   A  94
      CALL EXIT                                                         A  97
C                                                                       A  98
5     FORMAT (20I4)                                                     A  99
6     FORMAT (8F10.0)                                                   A 100
7     FORMAT ('0',45X,'NUMBER OF OBSERVATIONS'/55X,I4)                  A 101
8     FORMAT ('0',30X,'ORDER OF AUTOREGRESSION',30X,'ORDER OF MOVING    A 102
     1 AVERAGE'/40X,I4,50X,I4)                                          A 103
9     FORMAT ('0',5X,'RESIDUAL SUM OF SQUARES IS',D13.6)                A 104
10    FORMAT (' ',5X,'PARAMETERS',5F20.6)                               A 105
11    FORMAT ('0',5X,'STANDARD ERROR OF THE ESTIMATE IS',D13.6)         A 106
      END                                                               A 107-
```

```
      SUBROUTINE FORM (BN,ROE,BETA,KKK,N,KO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C        SUBROUTINE FORM DEFINES THE RELATION BETWEEN THE A.R.
C        AND M.A. PARAMETERS AND THE PARAMETER VECTOR BN
C
C
      COMMON /PARMS/ NP(10),NQ(10),MG
      DIMENSION ROE(N,MG), BETA(KKK,MG), BN(1)
      DO 3 J=1,MG
      DO 1 I=1,KKK
      BETA(I,J)=0.0
1     CONTINUE
      DO 2 I=1,N
      ROE(I,J)=0.0
2     CONTINUE
3     CONTINUE
      ROE(4,1)=BN(1)
      BETA(4,3)=BN(8)
      ROE(5,2)=BN(2)
      ROE(6,2)=BN(3)
      ROE(1,3)=BN(4)
      ROE(4,3)=BN(5)
      ROE(5,3)=BN(6)
      ROE(6,3)=BN(7)
      RETURN
      END

      SUBROUTINE DATA3 (YSTAR,REST,[illegible],NA,MG)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DATA3 READS IN ALL DATA AND SETS UP THE MATRIX
C        OF OBSERVATIONS
C
```

```
C
      DIMENSION REST(1), YSTAR(N,MG)
      DIMENSION FOOD(100), DUR(100), OTHER(100)
      READ (1,2) (FOOD(I),I=1,NA)
      READ (1,2) (DUR(I),I=1,NA)
      READ (1,2) (OTHER(I),I=1,NA)
      DO 1 I=1,N
      YSTAR(I,1)=FOOD(I)-FOOD(I+1)-FOOD(I+4)+FOOD(I+5)
      YSTAR(I,2)=DUR(I)-DUR(I+1)-DUR(I+4)+DUR(I+5)
      YSTAR(I,3)=OTHER(I)-OTHER(I+1)
1     CONTINUE
      RETURN
C
2     FORMAT (8F10.0)
      END


      SUBROUTINE TEST (BN,ZETA,K)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE TEST SETS UP THE POLYNOMIALS WHOSE ROOTS ARE
C        TO BE TESTED
C
C
      DIMENSION BN(1), COEF(10)
      DIMENSION POL(100)
      DIMENSION ZETA(1)
      CALL INIT (COEF)
      COEF(5)=BN(1)
      NPP=5
      CALL ROOT (NPP,POL,COEF,ZETA,K)
      CALL INIT (COEF)
      COEF(6)=BN(2)
      COEF(7)=BN(3)
      NPP=7
```

```
      CALL ROOT (NPP,POL,COEF,ZETA,K)
      CALL INIT (COEF)
      COEF(2)=BN(4)
      COEF(5)=BN(5)
      COEF(6)=BN(6)
      COEF(7)=BN(7)
      NPP=7
      CALL ROOT (NPP,POL,COEF,ZETA,K)
      CALL INIT (COEF)
      COEF(2)=BN(8)
      NPP=2
      CALL ROOT (NPP,POL,COEF,ZETA,K)
      RETURN
      END

      SUBROUTINE DECIDF (I,NOEQ)
C
C
C        SUBROUTINE DECIDE INDICATES WHICH EQUATION
C        A GIVEN ELEMENT OF BN IS IN . IT IS USER SUPPLIED.
C
C
      NOEQ=1
      IF (I.GT.1) NOEQ=2
      IF (I.GT.3) NOEQ=3
      RETURN
      END

      SUBROUTINE CON (MG,M,YSTAR,REST,A,W,N,PER)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE EXTRACTS THE OBSERVATIONS FOR EACH EQUATION FROM
C        THE OBSERVATION MATRIX
```

```
C
C
      DIMENSION YSTAR(N,MG), KEST(1), A(1), W(1), PER(N,MG)
      DO 2 I=1,MG
      DO 1 L=1,M
      A(L)=YSTAR(L,I)
1     CONTINUE
      CALL PERIOD (A,W,M,N,PER,I)
2     CONTINUE
      RETURN
      END


      SUBROUTINE INIT (COEF)
C
C
C        SUBROUTINE INIT INITIALIZES COEF
C
C
      REAL*8 COEF
      DIMENSION COEF(1)
      DO 1 I=1,10
      COEF(I)=0.0
1     CONTINUE
      COEF(1)=1.0
      RETURN
      END


      SUBROUTINE COVAR (G,T,N1,STD,ZETA,BN)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE COVAR COMPUTES THE STANDARD DEVIATIONS AND T STATS
C
C
```

```
      DIMENSION G(N1,N1), ZETA(1), STD(1),  BN(1)                    H   8
      INTEGER*4 T                                                     H   9
      PI=3.14159265                                                   H  10
      CALL INVERT (G,N1)                                              H  11
      DO 1 I=1,N1                                                     H  12
      G(I,I)=G(I,I)/(2*PI)                                            H  13
      STD(I)=G(I,I)                                                   H  14
      STD(I)=DSQRT(STD(I))                                            H  15
1     CONTINUE                                                        H  16
      WRITE(3,4)                                                      H  17
      DO 3 I=1,N1                                                     H  18
      WRITE(3,2)BN(I),STD(I)
3     CONTINUE                                                        H  20
      RETURN                                                          H  21
C                                                                     H  22
2     FORMAT(' ',15X,F12.6,15X,F12.6)                                 H  23
4     FORMAT(' ',15X,'PARAMETER',15X,'STANDARD DEVIATION')
      END


      SUBROUTINE RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,ZETA,ARTRAN,AMTRAN,P  I   1
     1ER,LN)                                                          I   2
      IMPLICIT REAL*8(A-H,O-Z)                                        I   3
C                                                                     I   4
C                                                                     I   5
C        SUBROUTINE RESID COMPUTES THE M.A. AND A.R. FREQUENCY        I   6
C        RESPONSE FUNCTIONS                                           I   7
C                                                                     I   8
C                                                                     I   9
      COMPLEX*16 WORK                                                 I  10
      COMPLEX*16 ARTRAN,AMTRAN,TRANS,PER                              I  11
      INTEGER*4 T,P,Q,TA                                              I  12
      DIMENSION INDEX(30)                                             I  13
      DIMENSION WORK(50)                                              I  14
      COMMON /PASS/ SIGMA(100)                                        I  15
      COMMON /PARMS/ NP(10),NQ(10),MG                                 I  16
```

```
      DIMENSION ROE(Q,MG), BETA(P,MG)
      DIMENSION AMTRAN(M,MG), ARTRAN(M,MG), PER(M,MG)
      DIMENSION TRANS(1), W(1), POLY(30), ZETA(1)
      KO=KO
      DO 5 J=1,MG
      NPJ=NP(J)
      DO 1 I=1,NPJ
      POLY(I)=-BETA(I,J)
1     CONTINUE
      CALL MAKE (POLY,BETA,KO,NPJ,INDEX,J,P)
      CALL TRANSF (TRANS,POLY,NPJ,W,M,INDEX)
      DO 2 I=1,M
      ARTRAN(I,J)=TRANS(I)
2     CONTINUE
      NQJ=NQ(J)
      DO 3 I=1,NQJ
      POLY(I)=ROE(I,J)
3     CONTINUE
      CALL MAKE (POLY,ROE,KO,NQJ,INDEX,J,Q)
      CALL TRANSF (TRANS,POLY,NQJ,W,M,INDEX)
      DO 4 I=1,M
      AMTRAN(I,J)=TRANS(I)
4     CONTINUE
5     CONTINUE
      DO 7 I=1,MG
      DO 6 J=1,M
      ARTRAN(J,I)=PER(J,I)*ARTRAN(J,I)/AMTRAN(J,I)
6     CONTINUE
7     CONTINUE
      CALL FUNC (WORK,ARTRAN,MG,M,ZETA,LN,AMTRAN,SIGMA)
      RETURN
C
      END

      SUBROUTINE FUNC (ARTRAN,PER,MG,M,ZETA,LN,AMTRAN,SIGMA)
```

```
      IMPLICIT COMPLEX*16(A-H,O-Z)
C
C
C        SUBROUTINE FUNC COMPUTES TH OBJECTIVE FUNCTION IN THE
C        FREQUENCY DOMAIN
C
C
      REAL*8 SIGMA
      REAL*8 ZETA
      DIMENSION SIGMA(MG,MG), AMTRAN(MG,MG)
      DIMENSION ARTRAN(MG,MG), PER(M,MG), ZETA(1)
      DO 3 I=1,MG
      DO 2 J=1,MG
      ARTRAN(I,J)=0.0
      DO 1 K=1,M
      ARTRAN(I,J)=ARTRAN(I,J)+PER(K,I)*DCONJG(PER(K,J))
1     CONTINUE
2     CONTINUE
3     CONTINUE
      DO 6 I=1,MG
      DO 5 J=1,MG
      AMTRAN(I,J)=0.0
      DO 4 K=1,MG
      AMTRAN(I,J)=AMTRAN(I,J)+ARTRAN(I,K)*SIGMA(K,J)
4     CONTINUE
5     CONTINUE
6     CONTINUE
      AA=DCMPLX(0.0D0,0.0D0)
      DO 7 I=1,MG
      AA=AA+AMTRAN(I,I)
7     CONTINUE
      ZETA(LN)=DREAL(AA)
      RETURN
C
      END
```

```
      SUBROUTINE DERIVS (BB,BN,D1,D2,N1,ROE,BETA,P,Q,T,KO,TRANS,W,M,ZETA
     1,ARTRAN,AMTRAN,PER,WORK1,WORK2,SIGMA)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE DERIVS COMPUTES THE DERIVATIVES REQUIRED BY
C        THE GAUSS-NEWTON ALGORITHM
C
C
      COMPLEX*16 WORK1,WORK2
      COMPLEX*16 AA
      COMPLEX*16 ARTRAN,AMTRAN,TRANS,PER
      INTEGER*4 T,P,Q
      COMMON /PARMS/ NP(10),NQ(10),MG
      DIMENSION SIGMA(MG,MG)
      DIMENSION WORK1(M,MG), WORK2(M,N1)
      DIMENSION ZETA(1), ARTRAN(M,MG), AMTRAN(1), PER(1)
      DIMENSION H(5)
      DIMENSION ROE(1), BETA(1), TRANS(1), W(1)
      DIMENSION BN(1), BB(1), D2(N1,N1), D1(1)
      N=M
      DO 2 I=1,N1
      CONST1=0.01*BN(I)
      CALL REFORM (BB,BN,N1)
      BB(I)=BB(I)-0.005*BB(I)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,H,ARTRAN,AMTRAN,PER,1)
      CALL REFORM (BB,BN,N1)
      BB(I)=BB(I)+0.005*BB(I)
      CALL FORM (BB,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,H,WORK1,AMTRAN,PER,2)
      D1(I)=H(2)-H(1)
      D1(I)=D1(I)/CONST1
      CALL DECIDE (I,NOEQ)
      DO 1 J=1,N
```

```
      WORK2(J,I)=WORK1(J,NDEQ)-ARTRAN(J,NOEQ)
      WORK2(J,I)=WORK2(J,I)/CONST1
1     CONTINUE
2     CONTINUE
      DO 5 I=1,N1
      DO 4 J=1,I
      AA=DCMPLX(0.0D0,0.0D0)
      CALL DECIDE (I,NE1)
      CALL DECIDE (J,NE2)
      DO 3 K=1,N
      AA=AA+WORK2(K,I)*DCONJG(WORK2(K,J))*SIGMA(NE1,NE2)
3     CONTINUE
      D2(I,J)=DREAL(AA)
      D2(J,I)=D2(I,J)
4     CONTINUE
5     CONTINUE
      RETURN
      END


      SUBROUTINE REFORM (BB,BN,N1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BB(1), BN(1)
      DO 1 I=1,N1
      BB(I)=BN(I)
1     CONTINUE
      RETURN
C
      END


      SUBROUTINE SEARCH (BB,B,BN,A,D3,W,U,UU,N,N1,ROE,P,Q,BETA,T,KO,TRAN
     1S,M,ARTRAN,AMTRAN,PER,*)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
```

```
C         SUBROUTINE SEARCH DETERMINES AN OPTIMAL STEP LENGTH BY
C         QUADRATIC INTERPOLATION
C
C
      COMPLEX*16 ARTRAN,AMTRAN,TRANS,PER
      INTEGER*4 P,Q,T
      DIMENSION ROE(1), BETA(1), TRANS(1), W(1)
      DIMENSION ARTRAN(1), AMTRAN(1), PER(1)
      DIMENSION BB(1), B(1), BN(1), A(N1,N1), D3(1), ALX(3)
      DIMENSION U(1), UU(3)
      COM=UU(1)
      DO 1 I=1,N1
      BB(I)=0.0
      DO 1 J=1,N1
1     BB(I)=BB(I)-A(I,')*D3(J)
      ALX(1)=0.0
      ALX(2)=0.5
      ALX(3)=1.0
      DO 2 I=2,3
      CALL ACCUM (BN,B,BB,ALX(I),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,UU,ARTRAN,AMTRAN,PER,I)
      CALL TEST (BN,UU,I)
2     CONTINUE
      IF (UU(2).LE.UU(3)) GO TO 3
      ALX(2)=ALX(3)
      UU(2)=UU(3)
3     KIK=0
      IF (UU(2).GT.UU(1)) GO TO 5
4     CONTINUE
      ALX(3)=2.0*ALX(2)
      CALL ACCUM (BN,B,BB,ALX(3),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,UU,ARTRAN,AMTRAN,PER,3)
      CALL TEST (BN,UU,3)
      IF (UU(3).GT.UU(2)) GO TO 7
```

```
      ALX(2)=ALX(3)
      UU(2)=UU(3)
      KIK=KIK+1
      IF (KIK.LT.10) GO TO 4
      RETURN 1
5     ALX(3)=ALX(2)
      UU(3)=UU(2)
      ALX(2)=ALX(3)*0.5
      CALL ACCUM (BN,B,BB,ALX(2),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,UU,ARTRAN,AMTRAN,PER,2)
      CALL TEST (BN,UU,2)
      IF (UU(3).GT.UU(2).AND.UU(2).LT.UU(1)) GO TO 7
      KIK=KIK+1
      IF (KIK.LT.10) GO TO 5
      ALX(2)=-ALX(2)
      CALL ACCUM (BN,B,BB,ALX(2),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,UU,ARTRAN,AMTRAN,PER,2)
      CALL TEST (BN,UU,2)
      IF (UU(2).LT.UU(1)) GO TO 6
      RETURN 1
6     KIK=1
      GO TO 4
7     ALX(1)=0.0
      CALL QUAD (ALX,UU,ATLEN,UMIN)
      ALX(1)=ATLEN
      CALL ACCUM (BN,B,BB,ALX(1),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,UU,ARTRAN,AMTRAN,PER,1)
      CALL TEST (BN,UU,1)
      IF (UU(1).LE.UU(2)) GO TO 8
      ALX(1)=ALX(2)
      CALL ACCUM (BN,B,BB,ALX(1),N1)
      CALL FORM (BN,ROE,BETA,P,Q,KO)
      CALL RESID (ROE,P,Q,BETA,T,KO,TRANS,W,M,UU,ARTRAN,AMTRAN,PER,1)
```

```
8     CONTINUE
      COMM=DABS(COM-UU(1))
      COM=COM*0.001
      IF (COMM.LT.COM) RETURN 1
      RETURN
C
      END


      SUBROUTINE ACCUM (BN,B,BB,AL,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE ACCUM UPDATES THE PARAMETER VECTOR
C
C
      DIMENSION BN(1), B(1), BB(1)
      DO 1 I=1,N1
      BN(I)=B(I)+AL*BB(I)
1     CONTINUE
      RETURN
      END


      SUBROUTINE QUAD (ALX,UU,AL,UMIN)
C
C
C        SUBROUTINE QUAD FITS A QUADRATIC THROUGH THREE POINTS
C        AND INTERPOLATES A MINIMUM
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALX(3), UU(3)
      IF(UU(3).EQ.1.0D35)UU(3)=1.5*UU(1)
      A=(((UU(1)-UU(2))*(ALX(1)-ALX(3)))-((UU(1)-UU(3))*(ALX(1)-ALX(2)))
     1)/((ALX(1)-ALX(2))*(ALX(1)-ALX(3))*(ALX(2)-ALX(3)))
```

```
      B=((UU(1)-UU(2))-(A*(ALX(1)-ALX(2))*(ALX(1)+ALX(2)))))/(ALX(1)-ALX(
     12))
      C=-(A*ALX(1)*ALX(1))-(B*ALX(1))+UU(1)
      AL=-B/(A*2)
      UMIN=-((B*B)/(A*4))+C
      RETURN
C
1     FORMAT (6D16.8)
      END


      SUBROUTINE ROOT (KP,MAT,COEF,ZETA,KO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE ROOT PERFORMS THE ROUTH-HURWICZ TEST
C
C
      REAL*8 MAT
      DIMENSION MAT(KP,KP), COEF(1), ZETA(1)
      DO 1 I=1,KP
      MAT(1,I)=COEF(I)
1     CONTINUE
      DO 3 I=2,KP
      DO 2 J=I,KP
      IJ=J-I
      MAT(I,J)=MAT(I-1,I-1)*MAT(I-1,J-1)-MAT(I-1,KP)*MAT(I-1,KP-IJ)
2     CONTINUE
3     CONTINUE
      DO 4 I=1,KP
      IF(MAT(I,I).LT.0.0D0)ZETA(KO)=1.0D35
4     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVERT (A,N)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE INVERT INVERTS A MATRIX A
C
C
      DIMENSION A(N,N), IPIV(20)
      DO 1 I=1,N
1     IPIV(I)=0
      DO 11 I=1,N
      AMAX=0.
      DO 5 J=1,N
      IF (IPIV(J)) 2,2,5
2     IF (DABS(A(J,J)-^MAX)) 4,4,3
3     ICOL=J
      AMAX=DABS(A(J,J))
4     CONTINUE
5     CONTINUE
      IPIV(ICOL)=1
      IF (AMAX-1.0D-50) 6,6,7
6     WRITE (3,12)
      STOP
7     CONTINUE
      AMAX=A(ICOL,ICOL)
      A(ICOL,ICOL)=1.0
      DO 8 K=1,N
8     A(ICOL,K)=A(ICOL,K)/AMAX
      DO 11 J=1,N
      IF (J-ICOL) 9,11,9
9     AMAX=A(J,ICOL)
      A(J,ICOL)=0.
      DO 10 K=1,N
10    A(J,K)=A(J,K)-A(ICOL,K)*AMAX
11    CONTINUE
      RETURN
```

```
C
12    FORMAT (/' SINGULAR MATRIX -- TERMINATE'/)
      END

      SUBROUTINE TRANSF (TRANS,POLY,OP,W,M,INDEX)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE TRANSF COMPUTES THE FREQUENCY RESPONSE FUNCTION
C        FOR A TRANSFER FUNCTION
C
C
      COMPLEX*16 TRANS,A
      COMPLEX*16 AD
      INTEGER*4 OP
      DIMENSION TRANS(M), POLY(OP), W(M)
      DIMENSION INDEX(1)
      DO 2 I=1,M
      A=DCMPLX(0.0D0,0.0D0)
      DO 1 J=1,OP
      IF (INDEX(J).EQ.0) GO TO 1
      AW=W(I)*J
      AD=DCMPLX(0.0D0,AW)
      A=A+POLY(J)*CDEXP(AD)
1     CONTINUE
      A=1.0+A
      TRANS(I)=A
2     CONTINUE
C
      RETURN
      END

      SUBROUTINE MAKE (POLY,ROE,KO,IQ,INDEX,J,IP)
      REAL*8 POLY,ROE
```

```
C
C
C        SUBROUTINE MAKE DETERMINES ZERO ELEMENTS IN A TRANSFER
C        FUNCTION AND STORES THEIR POSITIONS IN INDEX
C
C
      COMMON /PARMS/ NP(10),NQ(10),MG
      DIMENSION POLY(1), ROE(IP,MG), INDEX(1)
      DO 1 I=1,IQ
      INDEX(I)=0.0
1     CONTINUE
      DO 3 I=1,IQ
      IF (ROE(I,J).EQ.0.0D0) GO TO 2
      INDEX(I)=1
2     CONTINUE
3     CONTINUE
      RETURN
      END


      SUBROUTINE PERIOD (X,W,M,N,PER,II)
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C        SUBROUTINE PERIOD COMPUTES THE PERIODOGRAM FRO AN INPUT
C        VECTOR X
C
C
      COMPLEX*16 AD,AC,PER
      COMMON /PARMS/ NP(10),NQ(10),MG
      DIMENSION X(1), W(1), PER(M,MG)
      PI=0.314159265358979D1
      DD=2*PI/N
      DO 1 J=1,M
      W(J)=J*DD
1     CONTINUE
```

```
      DO 3 I=1,M
      AD=DCMPLX(0.0D0,0.0D0)
      AC=AD
      DO 2 J=1,N
      AW=W(I)*J
      AD=DCMPLX(0.0D0,AW)
      AC=AC+X(J)*CDEXP(AD)
2     CONTINUE
      PER(I,II)=AC
      AL=DSQRT(2*PI*N)
      PER(I,II)=PER(I,II)/AL
3     CONTINUE
      RETURN
C
      END
```

FOOD	CLOTHING	APPLIANCES	MISC.	NEW MONEY
237.0	146.0	97.0	410.0	26.8
257.0	171.0	111.0	447.0	56.2
263.0	156.0	108.0	443.0	45.2
279.0	186.0	122.0	470.0	46.2
307.0	167.0	131.0	498.0	66.6
342.0	196.0	135.0	557.0	70.2
338.0	156.0	117.0	537.0	42.6
346.0	186.0	117.0	547.0	39.4
346.0	161.0	123.0	524.0	49.2
375.0	202.0	140.0	593.0	63.4
353.0	154.0	124.0	560.0	46.2
366.0	196.0	127.0	580.0	32.0
379.0	177.0	138.0	579.0	58.8
406.0	224.0	163.0	646.0	56.8
380.0	166.0	135.0	610.0	51.4
391.0	212.0	142.0	643.0	61.4
389.0	194.0	156.0	644.0	97.4
424.0	240.0	175.0	680.0	113.2
403.0	180.0	145.0	678.0	74.0
417.0	228.0	153.0	717.0	91.8
423.0	205.0	168.0	702.0	113.2
451.0	250.0	186.0	759.0	146.0
442.0	185.0	154.0	729.0	109.6
443.0	230.0	162.0	756.0	125.8
464.0	206.0	173.0	755.0	169.8
484.0	257.0	199.0	831.0	155.0
458.0	191.0	165.0	788.0	57.4
465.0	231.0	174.0	818.0	55.6
463.0	213.0	190.0	789.0	111.2
490.0	266.0	216.0	883.0	88.2
474.0	191.0	188.0	829.0	59.8
477.0	237.0	191.0	853.0	92.4
482.0	220.0	214.0	840.0	89.0
515.0	270.0	235.0	924.0	117.1
498.0	192.0	192.0	861.0	58.7
497.0	248.0	208.0	899.0	69.9
507.0	235.0	236.0	902.0	99.8
546.0	292.0	265.0	1016.0	109.4
526.0	216.0	230.0	962.0	37.4
532.0	277.0	242.0	1005.0	40.7
551.0	254.0	260.0	1004.0	82.6
587.0	309.0	272.0	1096.0	106.4
556.0	221.0	224.0	992.0	57.3

556.0	278.0	226.0	1009.0	69.1
569.0	251.0	244.0	1001.0	86.0
595.0	302.0	267.0	1099.0	102.8
562.0	225.0	230.0	1053.0	58.9
566.0	286.0	243.0	1101.0	73.8
575.0	258.0	264.0	1106.0	115.6
620.0	318.0	288.0	1214.0	84.0
593.0	229.0	239.0	1141.0	68.0
602.0	299.0	255.0	1173.0	45.3
613.0	278.0	283.0	1207.0	92.5
651.0	343.0	310.0	1317.0	139.8
618.0	254.0	255.0	1217.0	116.0
629.0	323.0	279.0	1288.0	120.5
652.0	297.0	311.0	1299.0	201.3
708.0	369.0	343.0	1423.0	196.8
656.0	264.0	276.0	1344.0	164.1
679.0	341.0	290.0	1393.0	218.0
700.0	312.0	318.0	1318.0	
755.0	379.0	351.0	1503.0	
695.0	275.0	277.0	1417.0	
720.0	350.0	294.0	1477.0	
742.0	321.0	331.0	1485.0	
800.0	401.0	372.0	1625.0	
753.0	292.0	293.0	1539.0	
765.0	375.0	314.0	1606.0	
788.0	352.0	356.0	1619.0	
841.0	429.0	406.0	1781.0	
797.0	306.0	325.0	1709.0	
809.0	389.0	344.0	1757.0	
814.0	357.0	389.0	1738.0	
877.0	445.0	440.0	1923.0	
827.0	325.0	349.0	1851.0	
842.0	414.0	372.0	1923.0	
860.0	380.0	418.0	1934.0	
935.0	472.0	481.0	2133.0	
885.0	345.0	373.0	2032.0	
920.0	449.0	408.0	2108.0	

Data for Equation (7.1a)

C_t	KOD_t	NC_t	P_t	POD_t	YPD
	2824.0		0.975		
151.0	2888.0	169.0	0.978	1.000	2074.0
162.0	2961.0	190.0	0.982	1.000	2093.0
181.0	3050.0	208.0	0.987	1.000	2334.0
218.0	3174.0	256.0	0.992	1.000	2711.0
186.0	3262.0	214.0	1.001	1.000	2240.0
193.0	3354.0	223.0	1.019	1.000	2361.0
199.0	3449.0	236.0	1.030	1.000	2440.0
221.0	3563.0	254.0	1.037	1.005	2937.0
176.0	3630.0	165.0	1.045	1.006	2471.0
172.0	3690.0	164.0	1.051	1.012	2439.0
176.0	3753.0	168.0	1.050	1.023	2549.0
212.0	3849.0	208.0	1.045	1.005	3002.0
179.0	3910.0	181.0	1.044	1.000	2513.0
187.0	3977.0	205.0	1.043	1.000	2602.0
196.0	4051.0	217.0	1.045	0.995	2674.0
234.0	4160.0	243.0	1.046	0.987	3236.0
189.0	4221.0	206.0	1.047	0.989	2683.0
197.0	4288.0	216.0	1.050	0.990	2755.0
217.0	4373.0	244.0	1.052	0.959	2949.0
259.0	4497.0	269.0	1.051	0.961	3565.0
207.0	4566.0	217.0	1.058	0.961	2941.0
224.0	4650.0	238.0	1.068	0.969	3035.0
244.0	4751.0	256.0	1.081	0.955	3174.0
290.0	4894.0	289.0	1.093	0.959	3753.0
228.0	4972.0	232.0	1.101	0.952	3258.0
235.0	5054.0	247.0	1.111	0.953	3277.0
241.0	5140.0	247.0	1.123	0.954	3390.0
298.0	5279.0	261.0	1.137	0.950	3884.0
228.0	5345.0	214.0	1.139	0.947	3344.0
234.0	5415.0	231.0	1.148	0.957	3446.0
247.0	5496.0	235.0	1.153	0.960	3608.0
307.0	5634.0	277.0	1.164	0.964	4267.0
239.0	5700.0	222.0	1.168	0.954	3778.0
250.0	5775.0	254.0	1.183	0.964	3733.0
263.0	5861.0	261.0	1.198	0.966	3786.0
334.0		307.0	1.202	0.967	4477.0

Data for Equations (7.7a) and (7.7b)

L_t	Y_t	P_t	W_t	DE_t
2953.0	3367.0		66.5	0.0
2974.0	3609.0	80.4	70.4	0.0
3001.0	3300.0	78.1	63.1	0.0
3005.0	3554.0	78.5	67.9	0.0
3032.0	3648.0	80.2	70.3	0.0
3061.0	3893.0	81.5	75.6	0.0
3110.0	3635.0	81.0	69.5	0.0
3127.0	3721.0	85.1	75.1	0.0
3158.0	3937.0	83.8	75.0	0.0
3191.0	4075.0	85.7	78.9	0.0
3179.0	3589.0	85.9	72.2	0.0
3133.0	3713.0	84.9	76.5	0.0
3122.0	3747.0	85.7	76.3	0.0
3157.0	4045.0	86.2	80.6	0.0
3204.0	3757.0	86.8	74.6	0.0
3214.0	3913.0	88.6	79.5	0.0
3232.0	4046.0	86.8	78.2	0.0
3258.0	4348.0	86.5	82.6	0.0
3304.0	3964.0	87.5	76.4	0.0
3315.0	4109.0	89.6	81.7	0.0
3342.0	4259.0	89.8	81.2	0.0
3386.0	4677.0	88.9	88.4	0.0
3443.0	4214.0	88.8	79.7	0.0
3461.0	4475.0	91.7	85.8	0.0
3490.0	4627.0	92.4	88.1	0.0
3535.0	5019.0	92.2	93.8	0.0
3593.0	4491.0	93.4	85.9	0.0
3604.0	4777.0	95.6	91.9	0.0
3619.0	4896.0	95.7	93.5	0.0
3653.0	5167.0	95.2	97.1	0.0
3702.0	4677.0	95.2	89.9	0.0
3703.0	4875.0	97.9	95.0	0.0
3706.0	5010.0	98.8	98.8	0.0
3742.0	5341.0	99.1	102.6	0.0
3895.0	4872.0	100.6	96.0	1.0
3902.0	5198.0	101.5	102.7	1.0
3922.0	5338.0	101.9	104.4	1.0
3976.0	5678.0	102.5	109.0	1.0
4015.0	5165.0	104.3	101.9	1.0
4030.0	5477.0	105.7	108.5	1.0
4053.0	5656.0	105.7	109.1	1.0
4115.0	6109.0	106.8	117.4	1.0