

AD-A067 768

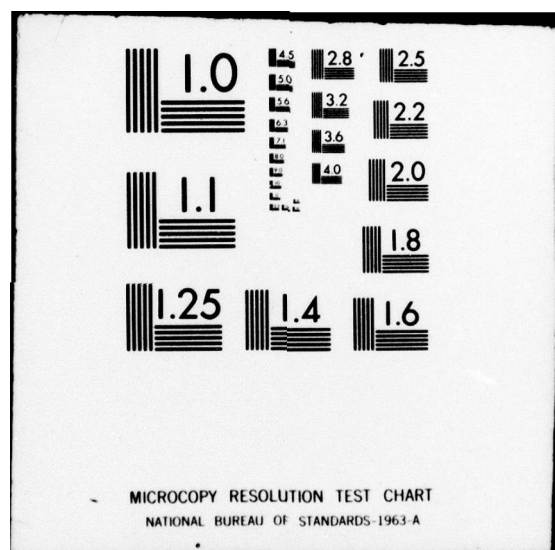
TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/6 12/1
LEAST SQUARES PREDICTION FOR MIXED AUTOREGRESSIVE MOVING AVERAG--ETC(U)
FEB 79 H J NEWTON, M PAGANO N00014-78-C-0599
TR-N-2 NL

UNCLASSIFIED

| OF |
AD
A067 768



END
DATE
FILMED
6 -79
DDC



LEVEL

TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS 77843

INSTITUTE OF STATISTICS
Phone 711-48144

LEAST SQUARES PREDICTION FOR MIXED AUTOREGRESSIVE MOVING AVERAGE TIME SERIES

By M. Joseph Newton and Marcello Pagano
Institute of Statistics, Texas A&M University and
Dept. of Biostatistics, Harvard University, U.S.A.

Technical Report No. M-2
February 1979

Texas A & M Research Foundation
Project No. 3838

"Multiple Time Series Modeling and Time
Series Theoretic Statistical Methods"

Sponsored by the Office of Naval Research

Professor Emanuel Parzen, Principal Investigator

Approved for public release; distribution unlimited.

AD A067268

DDC FILE COPY

79 04 17- 138

347 380 LB

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)		REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER		2. GOVT ACCESSION NO.		3. RECIPIENT'S CATALOG NUMBER	
4. AUTHOR(s)		5. TYPE OF REPORT & PERIOD COVERED		6. PERFORMING ORG. REPORT NUMBER	
7. AUTHORING ORG. NAME AND ADDRESS		8. DISTRIBUTION STATEMENT (of this Report)		9. SECURITY CLASS. (of this Report)	
10. CONTROLLING OFFICE NAME AND ADDRESS		11. DISTRIBUTION STATEMENT (of this Report)		12. SECURITY CLASS. (of this Report)	
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		14. DISTRIBUTION STATEMENT (of this Report)		15. SECURITY CLASS. (of this Report)	
16. DISTRIBUTION STATEMENT (of this Report)		17. DISTRIBUTION STATEMENT (of this Report)		18. SECURITY CLASS. (of this Report)	
19. SUPPLEMENTARY NOTES		20. SUPPLEMENTARY NOTES		21. SUPPLEMENTARY NOTES	
22. AVAILABILITY (Restrictions on access code if necessary and identify by block number)		23. AVAILABILITY (Restrictions on access code if necessary and identify by block number)		24. AVAILABILITY (Restrictions on access code if necessary and identify by block number)	

DD FORM 1473
EDITION OF 1 NOV 68 IS OBSOLETE
1/4 0102-10-914-6401

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LEAST SQUARES PREDICTION FOR MIXED AUTOREGRESSIVE MOVING AVERAGE TIME SERIES

By H. Joseph Newton and Marcello Pagano

*Institute of Statistics, Texas A&M University and
Dept. of Biostatistics, Harvard University, U.S.A.*

Keywords: Mixed Time Series, Toeplitz Matrix, Modified Cholesky Decomposition

ISO Fortran

LANGUAGE

DESCRIPTION AND PURPOSE

Let $y(1), \dots, y(T)$ be a sample realization of a mixed autoregressive moving average process $y(t), t = 0, 1, \dots$ of order (p, q) , i.e., $y(t)$ satisfies

$$\sum_{j=0}^p a(j)y(t-j) = \sum_{k=0}^q b(k)e(t-k), \quad t = 0, 1, \dots$$

for constants $p, q, a(0) = b(0) = 1, a(1), \dots, a(p), b(1), \dots, b(q)$, where $e(\cdot)$ is a white noise time series of zero mean, uncorrelated random variables with variance σ^2 . The zeros of the complex polynomials $g(z) = \sum_{j=0}^p a(j)z^j$ and $h(z) = \sum_{k=0}^q b(k)z^k$ are assumed outside the unit circle.

Subroutine MORD calculates least squares predictors $y(t+v|t)$ of $y(t+v)$ given $y(1), \dots, y(t)$ for

$$v = v_1, \dots, v_L \text{ for } t = t_1, \dots, t_L. \quad (1)$$

NUMERICAL METHOD

The algorithm given by Pagano and Parzen (1973) (based on the work of Whittle (1963)) is used. Let $X(t) = \sum_{j=0}^p a(j)y(t-j)$, $t = p+1, \dots, T$.

Then the $X(\cdot)$ are a sample realization of a pure moving average process of order q and $X^T = (X(p+1), \dots, X(T))$ has the symmetric Toeplitz correlation matrix Γ where

$$\Gamma_{jk} = \begin{cases} 1 & j = k \\ \rho(|j-k|) & |j-k| = 1, \dots, q \\ 0 & |j-k| > q \end{cases}$$

$$\text{and } \rho(v) = \sum_{k=0}^q b(k)b(k+v) / \sum_{k=0}^q b^2(k) = R(|v|)/R(0), \quad |v| \leq q.$$

Let $\Gamma = L\Lambda L^T$ be the modified Cholesky decomposition (Wilkinson (1967)) of Γ , i.e., L is a unit lower triangular $(T-p) \times (T-p)$ matrix (L_{jk}) and D is a $(T-p) \times (T-p)$ diagonal matrix (D_{jj}) . Define $\tilde{\Gamma} = (e(p+1), \dots, e(T))$ by $\tilde{\Gamma} = X(t) - \sum_{k=1}^{T-p} L_{t-p-k} e(t-k)$, $t = p+1, \dots, T$, i.e., $\tilde{\Gamma} = \tilde{X}$.

Then $y(t+v|t) = X(t+v|t) - \sum_{j=1}^p a(j)y(t+j|t)$ where

$$X(t+v|t) = \begin{cases} \sum_{k=0}^q b(k)e(t+v-k), & v = 1, \dots, q \\ 0 & v > q \end{cases}$$

and $y(t|t) = y(t)$ if $t \leq p$.

Thus the algorithm essentially consists of finding successive rows of the matrix L , using as little storage space as possible. This is done by noting:

- 1) the k th row of L has $n_k = \max(k-q-1, 0)$ leading zero elements followed by $n_k = \min(k-1, q)$ elements, followed by a 1, followed by $T-p-n_k-n_k-1$ zeros. Thus only n_k elements need be stored for any row.

- 2) to find the k th row of L , at most the previous q rows are necessary.

- 3) Bauer (1955) has shown that as k gets large, $L_{k,k-j} \rightarrow b(j)$ and $D_{kk} \rightarrow \sigma^2/R(0)$. Thus for a fixed δ , there should be an integer N such that

$$\begin{aligned} |L_{k,k-j} - b(j)| &< \delta \\ |D_{kk} - \sigma^2/R(0)| &< \delta \end{aligned} \quad (2)$$

for $j = 1, \dots, q, k \geq N$. Thus only the first N rows of L need be computed.

MORD calls auxiliary subroutine MORTH to find $e(p+1), \dots, e(T)$ using only a $(q+1) \times (q+1)$ work space for computing L . MORTH is designed so that it can be used also for performing Bauer's algorithm or to find the modified Cholesky decomposition of the symmetric Toeplitz matrix whose first row is $(1, \rho(1), \dots, \rho(q))$.

MORD is designed to perform prediction for pure autoregressive time series if $q = 0$ and for pure moving average time series if $p = 0$.

Subroutine MACV1 (M0, M1A, SIGSQ, R, M0, IFAULT)

Description: MACV1 calculates moving average autocovariances corresponding to moving average parameters.

Formal parameters

M0 Integer input: order of moving average
M1A Real Array (M0) input: moving average coefficients
SIGSQ Real input: variance of $\epsilon(\cdot)$
R Real Array (M0) output: autocovariances for lags 1, ..., M0
M0 Real output: variance of moving average
IFAULT Integer output: faulty indicator, equal to:
0 if proper execution
1 if $M0 < 1$

RESTRICTIONS, TIME, NECESSARY STORAGE

If the zeros of $h(z)$ are not outside the unit circle, the correlation matrix r is not positive definite. This is manifested by a diagonal element of its modified Cholesky decomposition becoming nonpositive. Subroutine MAORTH tests for this by checking for diagonal elements being $< EPS$ (specified in data statement). If one is found, IFAULT is set to 5 and execution stops.

Subroutine MTRD requires $(p+1) + (q+4)(q+1) + 3T + MM$ storage locations. The number of operations is approximately $(t_L - p)(p+q) + M(q+1)/2 + (t_L - t_p + 1)[v_L p + q + (q-1) + \dots + \max(1, q - v_L + 1)]$, where M is the number of rows of L before convergence. M increases as the smallest zero of $h(z)$ approaches the unit circle.

REFERENCES

1. Bauer, F. L. (1955). "Zu den Iterationsverfahren zur Hurwitz-Zerlegung eines Polynoms." Arch. Elek. Übertr., 9, pp 285-290.
2. Pagano, M. and Parzen, E. (1973). "Timesboard: A Time Series Package." Proc. of Comp. Sci. and Stat. 7th Annual Symp. on the Interface, Iowa State Univ., pp 193-197.
3. Whittle, P. (1963). Prediction and Regulation by Linear Least Squares Methods, London: English Universities Press.
4. Wilkinson, J. H. (1967). "The Solution of Ill-Conditioned Linear Equations," In Mathematical Methods for Digital Computers II. (A. Ralston and H. S. Wilf, eds) pp 65-93.

Outline of Algorithm for Subroutine MAORTH

Subroutine MAORTH is used to find a vector $\xi = (\xi(1), \dots, \xi(T))^T$, by $L\xi = Y$, where $Y^T = (Y(1), \dots, Y(T))$ is a sample realization of a moving average process of order q with coefficients $\beta(1), \dots, \beta(q)$ and noise variance σ^2 , and L is the $(T \times T)$ unit lower triangular matrix of the modified Cholesky decomposition $r_{q,T} = LDL^T$ of the $T \times T$ symmetric band Toeplitz correlation matrix $r_{q,T}$ of Y . Thus the (j, k) th element of $r_{q,T}$ is given by

$$(r_{q,T})_{jk} = \begin{cases} 1 & \text{if } j = k \\ \rho(|j - k|) & |j - k| = 1, \dots, q \\ 0 & |j - k| > q \end{cases}$$

where $\rho(v) = R(v)/R(0)$ and

$$R(v) = \sigma^2 \sum_{k=0}^{q-v} \beta(k)\beta(k+v), \quad v = 0, \dots, q$$

(Subroutine MACV1 calculates $R(0), R(1), \dots, R(q)$ given $q, \sigma^2, \beta(1), \dots, \beta(q)$)

Subroutine MAORTH can also be used to calculate $\sigma^2, \beta(1), \dots, \beta(q)$ if $R(0), R(1), \dots, R(q)$ are inputted. Thus if IOPTB = 1, the R 's are input while if IOPTB = 0, the β 's and σ^2 are input.

Further, one need not calculate the c 's if the subroutine is used only for Bauer's algorithm (IOPTB = 1). Thus if IOPTC = 1, the c 's are calculated (and the Y 's are input), while if IOPTC = 0, the c 's are not calculated (and the Y 's need not be inputted). The dimension of E and Y (the mnemonics for ϵ, Y) is IROMS1 which can be 1 if IOPTC = 0.

Finally, subroutine MAORITE can be used to merely find the modified Cholesky decomposition of $\Gamma_{q,T}$ by inputting $R(0)$, $R(1)$, ..., $R(q)$, 0, ..., 0, letting $NQ = T - 1$, $ITER9 = T$, $IOPTS = 1$, and $IOPT8 = 0$.

Let $A = (A_{ij})$ be a real symmetric $(T \times T)$ matrix and $L = (L_{ij})$, $D = \text{diag}(d_1, \dots, d_T)$ be the factors in the Modified Cholesky decomposition of A , i.e., $A = LDL^T$. Then

$$\begin{aligned} L_{11} &= 1, \quad d_1 = A_{11} \\ A_{ki} &= \sum_{j=1}^{i-1} d_j L_{kj} L_{ji} \\ L_{ki} &= \frac{A_{ki} - \sum_{j=1}^{i-1} d_j L_{kj} L_{ji}}{d_i} \quad i < k = 2, \dots, T \\ d_k &= A_{kk} - \sum_{j=1}^{k-1} d_j L_{kj} L_{jk} \end{aligned}$$

For $A = \Gamma_{q,T}$, $L_{ki} = 0$ if $k - i > q$; in fact the nonzero elements of row k of L are

$$L_{k, m_1+1}, \dots, L_{k, m_1+m_2}, \quad L_{kk} = 1$$

where $m_1 = \max(k - q - 1, 0)$, $m_2 = \min(k - 1, q)$.

The equations given above become

$$L_{k, m_1+j} = \frac{\rho(k, m_1+j) - \sum_{i=m_1+1}^{m_1+j-1} d_i L_{ki} L_{ji}}{d_{m_1+j}}, \quad j = 1, \dots, m_2$$

$$d_k = 1 - \sum_{j=1}^{m_2} d_{m_1+j} L_{k, m_1+j}^2$$

From these equations it is possible to note:

$$L_{k, m_1+1} = \frac{\rho(k, m_1+1)}{d_{m_1+1}}$$

Thus in calculating row k of L and d_k , one needs only rows

$m_1 + 2, \dots, m_1 + m_2 = k - 1$ of L and $d_{m_1+1}, \dots, d_{m_1+m_2}$, i.e., at most the previous q rows of L and the previous q elements of D . However, $q + 1$ rows are stored so that the decomposition of $\Gamma_{q, q+1}$ can be obtained.

Let $NQ \equiv q$, $NQP1 \equiv q + 1$. Subroutine MAORITE uses the constant DK and the arrays $D(NQP1)$, $WKL(NQP1)$, $AL(NQP1)$, $NQP1$ to determine the rows of L and the diagonal elements of D :

Step 1: initialize $D(1) = 1$, $AL(1, 1) = 1$, $I = 1$, $NQP1$
Calculate $R(1) = R(1)/M9$, i.e. the autocorrelations

Step K:

Calculate $L_{k, m_1+1}, \dots, L_{k, m_1+m_2}, d_k$ and store in $WKL(1), \dots, WKL(M3)$,
DK

If $K \leq NQP1$: put WKL into K^{th} row of AL , DK into $D(K)$, and go to next row

If $K > NQP1$: shift down by 1 the 2nd through $NQP1^{\text{th}}$ rows of AL and 2nd through $NQP1^{\text{th}}$ elements of D . Put WKL into $NQP1^{\text{th}}$ row of AL and DK into $D(NQP1)$.

Thus the key point of the subroutine is knowing where L_{LJ} and d_L are located in AL and D when calculating the K^{th} step.

If $K \leq NQ + 2$, no shifting of AL and D has been done since shifting is done after calculating WKL , DK and only for $K > NQ + 1$. Thus the I^{th} row of L is in the I^{th} row of AL and d_I is in $D(I)$. If $K = NQ + 2 + M$ the $NQ + 1$ rows of AL are the M^{th} through $(NQ + M)^{\text{th}}$ rows of L . In general then the I^{th} row of L and d_I are in the $(I - NQ)^{\text{th}}$ row of AL and in $D(I - NQ)$, where $NQ = \max(K - NQ - 2, 0)$.

It remains to determine where the j^{th} element of row I of L is located at step K . The elements of the i^{th} row of L that aren't identically zero or one are $L_{i, M_1}, \dots, L_{i, M_2}$ where $M_1 = \max(I - M_0, 1)$. They are stored in the first through $\min(I - 1, M_0)$ elements of row $(I - M_0)$ of AL at step K , i.e., $L_{i, M_1 + M_0 - 1}$ is in $AL(I - M_0, M_1)$ or $L_{i, J}$ is in $AL(I - M_0, J - M_0 + 1)$ at step K .

Thus the equations given above can be written:

For step $K \geq 2$: Let $M_1 = M_1 = \max(K - M_0 - 1, 0)$, $M_2 = M_2 = \min(K - 1, M_0)$,
 $M_3 = \max(K - M_0 - 2, 0)$.

$$WEL(1) = \frac{\rho(K - M_1 - 1)}{d_{M_1 + 1}} = \frac{R(K - M_1 - 1)}{D(M_1 + 1 - M_0)}$$

If $M_3 < 2$, go to d

$$WEL(J) = \frac{\rho(K - M_1 - J) - \sum_{I=1}^{J-1} d_{M_1 + J} L_{I, M_1 + J}^2}{d_{M_1 + J}}$$

$$= \frac{R(K - M_1 - J) - \sum_{I=1}^{J-1} D(M_1 + J - M_0) * WEL(I) * AL(M_1 + J - M_0, M_1 + J - \max(M_1 + J - M_0, 1) + 1)}{D(M_1 + J - M_0)}$$

$J = 2, \dots, M_3$

$$*MK = 1 - \sum_{J=1}^{M_3} d_{M_1 + J} L_{I, M_1 + J}^2$$

$$= 1 - \sum_{J=1}^{M_3} D(M_1 + J - M_0) * WEL(J) * WEL(J)$$

Summary of MAORTH (Let E represent c)

Initialize:
 $EFS = 1.E - 10$
 If $IOPTB = 0$, calculate M_0 , $R(1)$, ..., $R(M_0)$ via MACV1
 Store $\rho(1)$, ..., $\rho(M_0)$ in $R(1)$, ..., $R(M_0)$
 $D(1) = 1$
 $AL(1, 1) = 1$, $I = 1$, $MQP1$
 If $IOPTB = 1$, $E(1) = Y(1)$
 Do 130 $K = 2$, ITERs
 Calculate WEL , DK
 If $IOPTB = 1$, calculate $E(K)$
 If $DK < EFS$, Γ not positive definite. Go to 199
 $K < MQP1$: put WEL , DK into AL , D . Go to 130
 $K > MQP1$: do shifting
 $IOPTB = 1$: check convergence of rows of AL
 Yes: Form $BETA$, $SIGSQ$. Go to 150
 No: Go to 130
 $IOPTB = 0$: check convergence of AL to $BETA$
 Yes: Go to 150
 No: Go to 130
 130 CONTINUE
 If Loop is performed for all K , then the algorithm is done except if
 $IOPTB = 1$ in which case $BETA$, $SIGSQ$ are assigned values from last row
 of AL and D and Go to 199
 150 CONTINUE
 If $IOPTB = 1$, calculate the rest of the E 's using $BETA$
 199 Reform autocovariances
 RETURN

Outline of Algorithm for Subroutine MIXPD

Let $Y(1), \dots, Y(T)$ be a sample realization of length T of a mixed autoregressive moving average time series of order (p, q) with autoregressive parameters $\alpha(1), \dots, \alpha(p)$ and moving average parameters $\beta(1), \dots, \beta(q)$, and σ^2 .

Subroutine MIXPD calculates the least squares predictors

$$\begin{aligned} YPD(1) &= Y(t_f + v_f | t_f), \quad YPD(2) = Y(t_f + v_f + 1 | t_f), \dots, \quad YPD(N1) = Y(t_f + v_f | t_f), \\ YPD(N1 + 1) &= Y(t_f + 1 + v_f | t_f + 1), \quad YPD(N1 + 2) = Y(t_f + 1 + v_f + 1 | t_f + 1), \dots, \\ YPD(2^*N1) &= Y(t_f + 1 + v_f | t_f + 1), \end{aligned}$$

;

$$YPD(N2 - 1)^*N1 + 1 = Y(t_f + v_f | t_f), \dots, \quad YPD(NN) = Y(t_f + v_f | t_f),$$

where $N1 = v_f - v_f + 1$, $N2 = t_f - t_f + 1$, $NN = N1^*N2$.

The predictor $Y(t + v | t)$ is given by

$$Y(t + v | t) = \begin{cases} X(t + v | t) - \sum_{j=1}^p \alpha(j)Y(t + v - j | t) & p \neq 0, q \neq 0 \\ - \sum_{j=1}^p \alpha(j)Y(t + v - j | t) & p \neq 0, q = 0 \\ X(t + v | t) & p = 0, q \neq 0 \\ 0 & p = 0, q = 0 \end{cases}$$

where:

$$(a) \quad Y(t | a) = Y(t)$$

$$\text{if } t \leq a$$

$$(b) \quad X(t + v | t) = \begin{cases} \sum_{k=1}^q \beta(k) \varepsilon(t + v - k) & v = 1, \dots, q \\ 0 & v > q \end{cases}$$

$$(c) \quad \varepsilon = (\varepsilon(p + 1), \dots, \varepsilon(T))^T = L_{q, T-p}^{-1} \bar{X}, \quad \text{where}$$

$L_{q, T-p}$ is the unit lower triangular matrix in the modified Cholesky decomposition of the $(T - p) \times (T - p)$ correlation matrix $\Gamma_{q, T-p} \bar{X}$.

$$(d) \quad \bar{X} = (X(p + 1), \dots, X(T))^T, \quad \text{where}$$

$$X(t) = \sum_{j=0}^p \alpha(j)Y(t - j), \quad t = p + 1, \dots, T.$$

Note that $NPP1 = p + 1$ and $NQP1 = q + 1$ are input rather than p and q so that the dimension of $ALPHA = \alpha$ and $BETA = \beta$ can't be zero without stopping execution.

If $q = 0$, the $X(t + v | t)$ (and thus the ε 's) are not needed.

The predictors $Y(t + v | t)$ satisfy a difference equation for fixed t .

$$\text{Thus } Y(t + 1 | t), \dots, Y(t + v - 1 | t), X(t + 1 | t), \dots, X(t + v - 1 | t),$$

$$Y(t | t) = Y(t), \dots, Y(t - p + 1 | t) = Y(t - p + 1), \text{ and } \varepsilon(t - q + 1),$$

$$\dots, \varepsilon(t) \text{ are required to calculate } Y(t + v_f | t). \quad \text{If } t = q + 1 \leq p + 1$$

or $t - p + 1 < 1$ these values cannot be obtained. This is also true if

$t > T$. Thus if $t < p + q$ or $t < p$ or $t > T$ or $p + q = 0$, the predictors

$Y(t + v_f | t), \dots, Y(t + v_L | t)$ are assigned the value zero.

Summary of MIXPD

If $NQ \neq 0$ calculate $E(NP + 1), \dots, E(NOBS)$ via MAORTH

Do $150 \text{ NT} = \text{NTF}, \text{NTL}$

If $\text{NT} < \text{NP} + \text{NQ}$ or $\text{NT} < \text{NP}$ or $\text{NT} > T$ or $\text{NP} + \text{NQ} = 0$ set predictors equal to zero and go to 150

If $\text{NP} \neq 0$ put $Y(\text{NT} - \text{NP} + 1), \dots, Y(\text{NT})$ into $X(1), \dots, X(\text{NP})$.

Calculate $Y(\text{NT} + 1 | \text{NT}), \dots, Y(\text{NT} + \text{NVL} | \text{NT})$ and store in $X(\text{NP} + 1),$

$\dots, X(\text{NP} + \text{NVL})$.

Put X(NP + NVF), ..., X(NP + NVL) into YPD((NT - NTF)*NI + 1),
 ..., YPD((NT - NTF + 1)*NI), where NI = NVL - NVF + 1.
 150 CONTINUE

```

      SUBROUTINE MXPDI(NPP1,NQP1,ALPHA,BETA,SIGSQ,NOBS,Y,DEL,NTF,
C      INTL,NVF,NVL,NN,IROWS1,IROWS2,AL,D,WKL,R,R0,E,X,YPD,IFAU)
C      THIS SUBROUTINE CALCULATES LEAST SQUARES PREDICTORS FOR A MIXED
C      AUTOREGRESSIVE MOVING AVERAGE PROCESS OF ORDER (NP,NQ)
C
C      DIMENSION ALPHA(NPP1),BETA(NQP1),Y(NOBS),AL(IROWS1,IROWS1),
C      ID(IROWS1),WKL(NQP1),R(NQP1),E(IROWS2),X(NOBS),YPD(NN)
C      DATA ZERO,IOPTR,IOPTE/0.0,0,1/
C
C      TEST FOR INVALID PARAMETERS
C
C      IFAULT=1
C      IF(NPP1.LT.1) GO TO 160
C      IFAULT=2
C      IF(NQP1.LT.1) GO TO 160
C      IFAULT=3
C      IF((INTL-NTF+1)*(NVL-NVF+1).GT.NN) GO TO 160
C      IFAULT=4
C      IF(IROWS1.LT.NQP1) GO TO 160
C
C      FIND E ARRAY
C
C      NQ=NQP1-1
C      NP=NPP1-1
C      NPPNQ=NP+NQ
C      IF(NQ.EQ.0) GO TO 40
C      NX=NOBS-NP
C      DO 20 I=1,NX
C      NPP1=NP+I
C      C=Y(NPP1)
C      DO 10 J=1,NP
C      NPP1M=NPP1-J
C      C=C+ALPHA(J)*Y(NPP1M)
10      CONTINUE
C      X(I)=C
20      CONTINUE

```

```

C      CALL MAORTH(IOPTR,IOPT,E,NQ,DEL,NX,NX,X,BETA,SIGSQ,IROWS1,
C      1R,R0,WKL,D,AL,E,IF1)

```

```

C      IFAULT=5
C      IF(IF1.EQ.5) GO TO 160
C      DO 30 I=1,NX
C      N1=NOBS-I+1
C      N2=N1-NP
C      E(N1)=E(N2)
C 30    CONTINUE

```

```

C      FIND PREDICTORS
C

```

```

C 40    CONTINUE
C      N1=NVL-NVF+1
C      N2=NP+NVF
C      DO 150 NT=NTF,NTL
C      N3=(NT-NTF)*N1
C      NTEMP1=NT-NP-NQ
C      NTEMP=NT-NP
C      IF((NFPNQ.EQ.0).OR.(NT.LT.1).OR.(NT.GT.NOBS)) GO TO 55
C      IF((NP.NE.0).AND.(NT.LT.NPPNQ)) GO TO 55
C      GO TO 60
C 55    CONTINUE
C      DO 50 I=1,N1
C      N3PI=N3+I
C      YPD(N3PI)=ZERO
C 50    CONTINUE
C      GO TO 150
C 60    CONTINUE
C      IF(NP.EQ.0) GO TO 80
C      DO 70 I=1,NP
C      N4=NTEMP+I
C      X(I)=Y(N4)
C 70    CONTINUE

```

```

C 80    CONTINUE
C      DO 130 NV=1,NVL
C      C=ZERO
C      IF((NQ.EQ.0).OR.(NV.GT.NQ)) GO TO 100
C      NTPNV=NT+NV
C      IF((NP.EQ.0).AND.(NTPNV.LE.NQ)) GO TO 100
C      DO 90 K=NV,NQ
C      KK=NTPNV-K
C      C=C+BETA(K)*E(KK)
C 90    CONTINUE
C 100   CONTINUE
C      NPPNV=NP+NV
C      IF(NP.EQ.0) GO TO 120
C      DO 110 I=1,NP
C      NP1=NPPNV-I
C      C=C-ALPHA(I)*X(NP1)
C 110   CONTINUE
C 120   CONTINUE
C      X(NPPNV)=C
C 130   CONTINUE
C      DO 140 NV=1,N1
C      N3PNV=N3+NV
C      N5=N2+NV-1
C      YPD(N3PNV)=X(N5)
C 140   CONTINUE
C 150   CONTINUE

```

```

C      IFAULT=0
C 160   CONTINUE
C      RETURN
C      END
C      SUBROUTINE MAORTH(IOPTR,IOPT,E,NQ,DEL,ITERS,IROWS1,Y,BETA,
C      1SIGSQ,IROWS2,R,R0,WKL,D,AL,E,IFAU)

```

```

C      THIS SUBROUTINE CALCULATES THE MODIFIED CHOLESKY FACTORS OF THE
C      ITERS ORDER CORRELATION MATRIX FOR A MOVING AVERAGE PROCESS OF
C      ORDER NQ. IT IS USED TO CALCULATE THE CORRESPONDING COEFFICIENTS
C      AND/OR AN ORTHOGONAL TRANSFORMATION OF INPUTTED MOVING AVERAGE DATA.
C

```



```

        DIMENSION Y(IROWS1),BETA(NQ),R(NQ),WKL(NQ),D(IROWS2),
        1AL(IROWS2,IROWS2),E(IROWS1)
        DATA ONE/1.0/
        DATA EPS/1.E-10/
C
C     EPS IS CRITERION FOR TESTING FOR ZERO DIAGONAL
C
C     TEST FOR INVALID PARAMETERS
C
        IF(FAULT=1)
        IF(NQ.LT.1) GO TO 299
        IF(FAULT=2)
        IF((IOPTB.NE.0).AND.(IOPTB.NE.1)) GO TO 299
        IF(FAULT=3)
        IF((IOPTB.NE.0).AND.(IOPTB.NE.1)) GO TO 299
        IF(FAULT=4)
        IF(IROWS2.LE.NQ) GO TO 299
C
C     AUTOCORRELATIONS AND INITIALIZATION
C
        IF(IOPTB.EQ.0) CALL MACV1(NQ,BETA,SIGSQ,R,R0,IF1)
        D(1)=ONE
        NQP1=NQ+1
        DO 5 I=1,NQ
        R(I)=R(I)/R0
        5 CONTINUE
        DO 10 I=1,NQP1
        AL(I,I)=ONE
        10 CONTINUE
        IF(IOPTB.EQ.1) E(1)=Y(1)
C
C     ROW K OF FACTORIZATION
C
C     M1 IS NUMBER OF LEADING ZEROS
C
C     M3 IS NUMBER OF ELEMENTS TO CALCULATE
C
C     MR IS SHIFTING FACTOR IN STORAGE OF PREVIOUS ROWS
C
C     INDD+J-1 IS ROW NUMBER OF AL FOR CALCULATING WKL(J)
C

```

```

        DO 130 K=2,ITERS
        M1=MAX0(K-NQ-1,0)
        M3=MIN0(K-1,NQ)
        INDR=K-M1-1
        MR=MAX0(K-NQ-2,0)
        INDD=M1+1-MR
        WKL(1)=R(INDR)/D(INDD)
        IF(M3.LT.2) GO TO 40
        DO 30 J=2,M3
        JM1=J-1
        INDR1=INDR-JM1
        C=R(INDR1)
        INDLR=INDD+JM1
        M4=-MAX0(M1+J-NQ,1)+M1+1
        DO 20 I=1,JM1
        INDD1=INDD+I-1
        INDL1=M4+I
        C=C-D(INDD1)*WKL(I)*AL(INDLR,INDL1)
        20 CONTINUE
        WKL(J)=C/D(INDLR)
        30 CONTINUE
        40 CONTINUE
        DK=ONE
        DO 50 J=1,M3
        INDD=M1+J-MR
        DK=DK-D(INDD)*WKL(J)*WKL(J)
        50 CONTINUE
        IF(IOPTB.EQ.0) GO TO 65
        C=Y(K)
        DO 60 I=1,M3
        INDWKL=M3-I+1
        INDE=K-I
        C=C-WKL(INDWKL)*E(INDE)
        60 CONTINUE
        E(K)=C

```

```

65  CONTINUE
    IFAULT=5
    IF(DK.LT.EPS) GO TO 199
    IF(N.GT.NQP1) GO TO 80
    D(K)=DK
    DO 70 I=1,NQ
        AL(K,I)=WKL(I)
70  CONTINUE
    GO TO 130
80  CONTINUE
    DO 82 I=1,NQ
        IP1=I+1
        D(I)=D(IP1)
        DO 81 J=1,NQ
            AL(I,J)=AL(IP1,J)
81  CONTINUE
82  CONTINUE
    D(NQP1)=DK
    DO 83 I=1,NQ
        AL(NQP1,I)=WKL(I)
83  CONTINUE
C
C  DEL IS CONVERGENCE CRITERION
C
    IF(IOPTR.EQ.0) GO TO 110
    DO 90 I=1,NQ
        IF(ABS(AL(NQP1,I)-AL(NQ,I)).GE.DEL) GO TO 130
90  CONTINUE
    IF(R0*ABS(D(NQP1)-D(NQ)).GE.DEL) GO TO 130
    DO 100 I=1,NQ
        INDWKL=NQ-I+1
        BETA(I)=WKL(INDWKL)
100 CONTINUE
    SIGSQ=R0*D(NQP1)
    IFAULT=0
    GO TO 150

```

```

110 CONTINUE
    DO 120 I=1,NQ
        INDB=NQ-I+1
        IF(ABS(WKL(I)-BETA(INDB)).GE.DEL) GO TO 130
120 CONTINUE
    IF(ABS(D(NQP1)*R0-SIGSQ).LT.DEL) GO TO 150
130 CONTINUE
    IFAULT=6
    IF(IOPTR.EQ.0) GO TO 199
    DO 140 I=1,NQ
        INDWKL=NQ-I+1
        BETA(I)=WKL(INDWKL)
140 CONTINUE
    SIGSQ=R0*DK
    GO TO 199
150 CONTINUE
    IFAULT=0
    IF(IOPTE.EQ.0) GO TO 199
    IF(K.EQ.ITER5) GO TO 199
    KP1=K+1
    DO 170 J=KP1,ITER5
        C=Y(J)
        DO 160 I=1,NQ
            INDE=J-I
            C=C-BETA(I)*E(INDE)
160 CONTINUE
        E(J)=C
170 CONTINUE
    IFAULT=0
199 CONTINUE
    DO 200 I=1,NQ
        R(I)=R(I)*R0
200 CONTINUE
299 RETURN
    END
    SUBROUTINE MACV1(NQ,BETA,SIGSQ,R,R0,IFAU)

```

```

C
C THIS SUBROUTINE CALCULATES AUTOCOVARIANCES OF A MOVING AVERAGE
C PROCESS OF ORDER NQ GIVEN ITS COEFFICIENTS AND RESIDUAL VARIANCE.
C

```

```

      DIMENSION BETA(NQ),R(NQ)
      DATA ONE /1.0/
      IFAULT=1
      IF(NQ.LT.1) GO TO 40
      C=ONE
      DO 10 I=1,NQ
      C=C+BETA(I)*BETA(I)
10    CONTINUE
      R0=SIGSQ*C
      DO 20 IV=1,NQ
      NQMIV=NQ-IV
      C=BETA(IV)
      DO 30 I=1,NQMIV
      IVPI=IV+I
      C=C+BETA(I)*BETA(IVPI)
30    CONTINUE
      R(IV)=C*SIGSQ
20    CONTINUE
C
C
      IFAULT=0
40    RETURN
      END

```