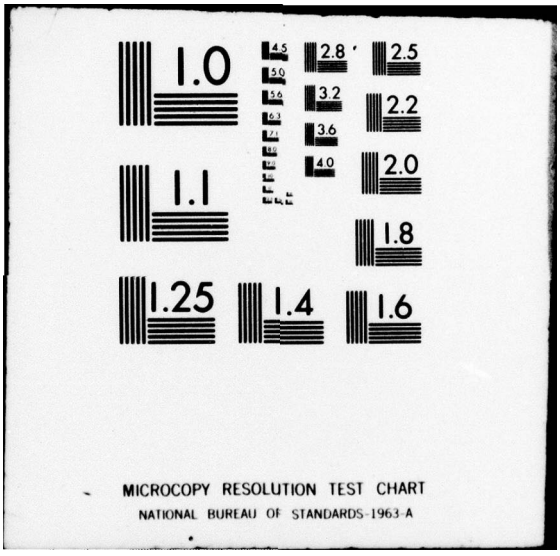
AD-A067 768 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 12/1
LEAST SQUARES PREDICTION FOR MIXED AUTOREGRESSIVE MOVING AVERAG--ETC(U)
FEB 79 H J NEWTON, M PAGANO N00014-78-C-0599
UNCLASSIFIED TR-N-2 NL

END
DATE FILMED
6-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS 77843

Institute of Statistics
Phone 713-845-3141

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.

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 A067768

DDC FILE COPY

DDC RECEIVED APR 20 1979 C

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: Technical Report, No. M-2 (2)	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Least Squares Prediction for Mixed Autoregressive Moving average Time Series, (6)	5. TYPE OF REPORT & PERIOD COVERED: Technical
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): H. Joseph Newton Marcello Pagano (10)	8. CONTRACT OR GRANT NUMBER(s): N00014-78-C-0599 (15)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Texas A&M University, Institute of Statistics, College Station, TX 77843	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Code 436, Arlington, VA 22217 (11)	12. REPORT DATE: February 1979 / 13. NUMBER OF PAGES: 21
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 12p.	15. SECURITY CLASS (of this report): Unclassified / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited. (14) TR-N-2	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report): NA	
18. SUPPLEMENTARY NOTES: NA	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Mixed time series, Toeplitz matrix, Modified Cholesky decomposition	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number): A computer subroutine is given for computing predictors of any specified horizon's ahead.	

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-LF-014-6601

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

LANGUAGE

ISO Fortran

DESCRIPTION AND PURPOSE

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

$$\sum_{j=0}^{p} \alpha(j)y(t-j) = \sum_{k=0}^{q} \beta(k)\varepsilon(t-k) , \qquad t = 0, \pm 1, \ldots$$

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

Subroutine MXPD calculates least squares predictors y(t + v|t) of y(t + v) given y(1), ..., y(t) for

$$v = v_F, \ldots, v_L \quad \text{for} \quad t = t_F, \ldots, t_L . \tag{1}$$

NUMERICAL METHOD

The algorithm given by Pagano and Parzen (1973) (based on the work of Whittle (1963)) is used. Let $X(t) = \sum_{j=0}^{p} \alpha(j)y(t-j)$, $t = p+1, \ldots, T$. Then the X(·) are a sample realization of a pure moving average process of order q and $\underline{X}^T = (X(p+1), \ldots, X(T))$ has the symmetric Toeplitz correlation matrix Γ where

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

and $\rho(v) = \sum_{k=0}^{q} \beta(k)\beta(k+|v|) / \sum_{k=0}^{q} \beta^2(k) \equiv R(|v|)/R(0)$, $|v| \le q$.

Let $\Gamma = LDL^T$ be the modified Cholesky decomposition (Wilkinson (1967)) of Γ, i.e., L is a unit lower triangular (T - p) × (T - p) matrix (L_{jk}) and D is a (T - p) × (T - p) diagonal matrix (D_{ij}). Define $\underline{e}^T = (e(p+1), \ldots, e(T))$ by $e(t) = X(t) - \sum_{k=1}^{\min(t-p-1,q)} L_{t-p,t-p-k}\, e(t-k)$, $t = p+1, \ldots, T$, i.e., $L\underline{e} = \underline{X}$.

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

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

and y(r|s) = y(r) if $r \le s$.

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:

i) the k^{th} row of L has $m_1 = \max(k-q-1, 0)$ leading zero elements followed by $m_2 = \min(k-1, q)$ elements, followed by a 1, followed by T - p - m_1 - m_2 - 1 zeros. Thus only m_2 elements need be stored for any row.

ii) to find the k^{th} row of L, at most the previous q rows are necessary.

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

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

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

MXPD calls auxilliary subroutine MAORTH to find e(p + 1), ..., e(T) using only a (q + 1) × (q + 1) work space for computing L. MAORTH 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, ρ(1), ..., ρ(q)).

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

STRUCTURE

SUBROUTINE MXPD (NPP1, NQP1, ALPHA, BETA, SIGSQ, NOBS, Y, DEL, NTF, NTL, NVF, NVL, NN, IROWS1, IROWS2, AL, D, WKL, R, RØ, E, X, YPD, IFAULT)

Formal parameters

NPP1	Integer	input:	p + 1
NQP1	Integer	input:	q + 1
ALPHA	Real Array (NP)	input:	coefficients of autoregressive part
BETA	Real Array (NQ)	input:	coefficients of moving average part
SIGSQ	Real	input:	variance of $\varepsilon(\cdot)$
NOBS	Integer	input:	number of observations
Y	Real Array (NOBS)	input:	array of time series observations
DEL	Real	input:	convergence criterion: see (2)
NTF	Integer	input:	t_P: see (1)
NTL	Integer	input:	t_L: see (1)
NVF	Integer	input:	v_P: see (1)
NVL	Integer	input:	v_L: see (1)
NN	Integer	input:	(NTL - NTF + 1)(NVL - NVF + 1)
IROWS1	Integer (> q)	input:	row dimension of AL and dimension of D, WKL, R in DIMENSION statement of calling program
IROWS2	Integer	input:	dimension of E in DIMENSION statement of calling program, given by: ≥ 1 if NQP1 = 1 ≥ NOBS if NQP1 > 1
AL	Real Array (NQP1, NQP1)	workspace:	
D	Real Array (NQP1)	workspace:	
WKL	Real Array (NQP1)	workspace:	
R	Real Array (NQP1)	output:	moving average autocovariances of lag 1, ..., q corresponding to BETA, SIGSQ
RØ	Real	output:	moving average variance corresponding to BETA, SIGSQ
E	Real Array (IROWS2)	workspace:	
X	Real Array (NOBS)	workspace:	
YPD	Real Array (NN)	output:	predictors stored as $y(t_P + v_P \mid t_P)$, ..., $y(t_P + v_L \mid t_P)$, ..., $y(t_L + v_P \mid t_L)$, ..., $y(t_L + v_L \mid t_L)$
IFAULT	Integer	output:	fault indicator, equal to: 0 if proper execution 1 if NPP1 < 1 2 if NQP1 < 1 3 if NN < (NTL - NTF + 1)(NVL - NVF + 1) 4 if IROWS1 < NQP1 5 if correlation matrix singular

Auxiliary algorithms

SUBROUTINE MAORTH (IOPTB, IOPTE, NQ, DEL, ITERS, IROWS1, Y, BETA, SIGSQ, IROWS2, R, RØ, WKL, D, AL, E, IFAULT)

Formal parameters

IOPTB	Integer	input:	option switch equal to: 1 if coefficients are to be calculate 0 if coefficients are inputted
IOPTE	Integer	input:	option switch equal to: 1 if e's are to be calculated 0 if e's not to be calculated
NQ	Integer	input:	order of moving average
DEL	Real	input:	convergence criterion: see (2)
ITERS	Integer	input:	order of correlation matrix
IROWS1	Integer	input:	dimension of Y and E arrays, give by: ≥ 1 if IOPTE = 0 ≥ ITERS if IOPTE = 1
Y	Real Array (IROWS1)	input	(If IOPTE = 1) or workspace (If IOPTE = 0): data array
BEAT	Real Array (NQ)	input	(If IOPTB = 0) or output (If IOPTB = 1): moving average coefficients
SIGSQ	Real	input	(If IOPTB = 0) or output (If IOPTB = 1): variance of $\varepsilon(\cdot)$
IROWS2	Integer (> NQ)	input:	row dimension of AL and dimension of D in DIMENSION statement of calling program
R	Real Array (NQ)	input	(If IOPTB = 1) or output (If IOPTB = 0): moving average autocovariances for lags 1, ..., NQ corresponding to BETA, SIGSQ
RØ	Real	input	(If IOPTB = 1) or output (If IOPTB = 0): moving average variance corresponding to BETA, SIGSQ
WKL	Real Array (NQ)	workspace:	
D	Real Array (NQ + 1)	workspace:	
AL	Real Array (NQ + 1, NQ + 1)	workspace:	
E	Real Array (IROWS1)	output	(If IOPTE = 1) or workspace (If IOPTE = 0)
IFAULT	Integer	output:	faculty indicator, equal to: 0 if proper execution 1 if NQ < 1 2 if IOPTB ≠ 0 or 1 3 if IOPTE ≠ 0 or 1 4 if IROWS2 < NQ 5 if matrix not positive definite 6 if convergence not reached

SUBROUTINE MACV1 (MQ, BETA, SIGSQ, R, RØ, IFAULT)

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

Formal parameters

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

RESTRICTIONS, TIME, NECESSARY STORAGE

If the zeros of h(z) are not outside the unit circle, the correlation matrix Γ 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 executive stops.

Subroutine MXPD requires $(p + 1) + (q + 4)(q + 1) + 3T + MN$ storage locations. The number of operations is approximately $(t_L - p)(p + q) + Mq(q + 1)/2 + (t_L - t_F + 1)[v_L p + q + (q - 1) + \ldots + \max(1, q - v_L + 1)]$, 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

Bauer, F. L. (1955). "E in Direktes Iterationsverfahren zur Hurwitz-Zerlegung eines Polynoms.," Arch. Elek. Ubertr., 9, pp 285-290.

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.

Whittle, P. (1963). Prediction and Regulation by Linear Least Squares Methods, London: English Universities Press.

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 $\underset{\sim}{\varepsilon} = (\varepsilon(1), \ldots, \varepsilon(T))^T$, by $L\underset{\sim}{\varepsilon} = \underset{\sim}{Y}$, where $\underset{\sim}{Y}^T = (Y(1), \ldots, Y(T))$ is a sample realization of a moving average process of order q with coefficients $\beta(1), \ldots, \beta(q)$ and noise variance σ^2, and L is the $(T \times T)$ unit lower triangular matrix of the modified Cholesky decomposition $\Gamma_{q,T} = LDL^T$ of the $T \times T$ symmetric band Toeplitz correlation matrix $\Gamma_{q,T}$ of $\underset{\sim}{Y}$. Thus the (j, k)th element of $\Gamma_{q,T}$ is given by

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

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

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

(Subroutine MACV1 calculates R(0), R(1), ..., R(q) given q, σ^2, $\beta(1), \ldots, \beta(q)$)

Subroutine MAORTH can also be used to calculate σ^2, $\beta(1), \ldots, \beta(q)$ if R(0), R(1), ..., 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 ε's if the subroutine is used only for Bauer's algorithm (IOPTB = 1). Thus if IOPTE = 1, the ε's are calculated (and the Y's are input), while if IOPTE = 0, the ε's are not calculated (and the Y's need not be inputted). The dimension of E and Y (the mnemonics for ε, Y) is IROWS1 which can be 1 if IOPTE = 0.

Finally, subroutine MAORTH 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, ITERS= T, IOPTB = 1, and IOPTE = 0.

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

$$L_{11} = 1 \; ; \; d_1 = A_{11}$$

$$L_{ki} = \frac{A_{ki} - \sum_{j=1}^{i-1} d_j L_{kj} L_{ij}}{d_i} \qquad i < k = 2, \ldots, T$$

$$d_k = A_{kk} - \sum_{j=1}^{k-1} d_j L_{kj} L_{kj}$$

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

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

where $m_1 = \max(k - q - 1, 0)$, $m_3 = \min(k - 1, q)$.

The equations given above become

$$L_{k,m_1+j} = \frac{\rho(k-m_1-j) - \sum_{i=m_1+1}^{m_1+j-1} d_i L_{k,i} L_{m_1+j,i}}{d_{m_1+j}} \qquad j = 1, \ldots, m_3$$

$$d_k = 1 - \sum_{j=1}^{m_3} d_{m_1+j} L^2_{k,m_1+j}$$

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 $m_1 + 2, \ldots, m_1 + m_3 = k - 1$ of L and $d_{m_1+1}, \ldots, d_{m_1+m_3}$, 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 ≡ q, NQP1 ≡ q + 1. Subroutine MAORTH 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(I, I) = 1, I = 1, NQP1
Calculate R(I) = R(I)/R0, i.e. the autocorrelations

Step K :

Calculate $L_{K,m_1+1}, \ldots, L_{K,m_1+m_3}$, d_k and store in WKL(1), ..., WKL(M3), DK

If K ≤ 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 2^{nd} through $NQP1^{th}$ rows of AL and 2^{nd} through $NQP1^{th}$ elements of D. Put WKL into $NQP1^{th}$ row of AL and DK into D(NQP1).

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

If K ≤ 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 $(NQ + M)^{th}$ rows of L. In general then the I^{th} row of L and d_I are in the $(I - MR)^{th}$ row of AL and in D(I - MR), where MR = 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 $L_{I,MM}, \ldots, L_{I,I-1}$, where $MM = \max(I - MQ, 1)$. They are stored in the first through $\min(I - 1, MQ)^{th}$ elements of row $(I - MR)$ of AL at step K, i.e., $L_{I,MM+M-1}$ is in $AL(I - MR, M)$ or L_{IJ} is in $AL(I - MR, J - MM + 1)$ at step K.

Thus the equations given above can be written:

For step $K \geq 2$: Let $M1 \equiv m_1 = \max(K - MQ - 1, 0)$, $M3 \equiv m_3 = \min(K - 1, MQ)$,

$MR = \max(K - MQ - 2, 0)$.

$$WKL(1) \equiv \frac{\rho(K-m_1-1)}{d_{m_1+1}} = \frac{R(K-M1-1)}{D(M1+1-MR)}$$

If $M3 < 2$, go to *

$$WKL(J) \equiv \frac{\rho(K-m_1-J) - \sum_{I=1}^{J-1} d_{m_1+I} L_{K,m_1+I} L_{m_1+J,I}}{d_{m_1+J}}$$

$$= \frac{R(K-M1-J) - \sum_{I=1}^{J-1} D(M1+I-MR)*WKL(I)*AL(M1+J-MR,M1+I-\max(M1+J-MQ,1)+1)}{D(M1+J-MR)},$$

$J = 2, \ldots, M3$

$$*DK \equiv 1 - \sum_{J=1}^{m_3} d_{m_1+J} L^2_{K,m_1+J}$$

$$= 1 - \sum_{J=1}^{M3} D(M1 + J - MR)*WKL(J)*WKL(J)$$

Summary of MAORTH (Let E represent ε)

Initialize:

```
EPS = 1.E - 10
If IOPTB = 0, calculate RØ, R(1), ..., R(MQ) via MACV1
Store ρ(1), ..., ρ(MQ) in R(1), ..., R(MQ)
D(1) = 1
AL(I, I) = 1, I = 1, MQP1
If IOPTE = 1, E(1) = Y(1)
```

Do 130 K = 2, ITERS

```
Calculate WKL, DK
If IOPTE = 1, calculate E(K)
If DK < EPS, Γ_{q,T} not positive definite.  Go to 199
      K ≤ MQP1 :  put WKL, 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 IOPTE = 1, calculate the rest of the E's using BETA

199 Reform autocovariances
    RETURN
```

Outline of Algorithm for Subroutine MXPD

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

Subroutine MXPD calculates the least squares predictors

$YPD(1) = Y(t_F + v_F|t_F)$, $YPD(2) = Y(t_F + v_F + 1|t_F)$, ..., $YPD(M1) = Y(t_F + v_L|t_F)$,

$YPD(M1 + 1) = Y(t_F + 1 + v_F|t_F + 1)$, $YPD(M1 + 2) = Y(t_F + 1 + v_F + 1|t_F + 1)$, ...,

$YPD(2*M1) = Y(t_F + 1 + v_L|t_F + 1)$,

$\vdots$

$YPD((M2 - 1)*M1 + 1) = Y(t_L + v_F|t_L)$, ..., $YPD(MM) = Y(t_L + v_L|t_L)$,

where $M1 = v_L - v_F + 1$, $M2 = t_L - t_F + 1$, $MM = M1*M2$.

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) $Y(r|s) = Y(r)$ if $r \leq s$

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

(c) $\underline{\varepsilon} = (\varepsilon(p + 1), \ldots, \varepsilon(T))^T = L_{q,T-p}^{-1}\, \underline{X}$, 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 $\Gamma_{q,T-p}$ $\underline{X}$.

(d) $\underline{X} = (X(p + 1), \ldots, X(T))^T$, where

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

Note that NPP1 = p + 1 and NQP1 = q + 1 are input rather than p and q so that the dimension of ALPHA = α and BETA = β 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. Thus $Y(t + 1|t)$, ..., $Y(t + v - 1|t)$, $X(t + 1|t)$, ..., $X(t + v - 1|t)$, $Y(t|t) = Y(t)$, ..., $Y(t - p + 1|t) = Y(t - p + 1)$, and $\varepsilon(t - q + 1)$, ..., $\varepsilon(t)$ are required to calculate $Y(t + v_F|t)$. 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 $Y(t + v_F|t)$, ..., $Y(t + v_L|t)$ are assigned the value zero.

Summary of MXPD

If NQ ≠ 0 calculate E(NP + 1), ..., E(NOBS) via MAORTH

Do 150 NT = NTF, NTL
  If NT < NP + NQ or NT < NP or NT > T or NP + NQ = 0 set predictors equal to zero and go to 150

  If NP ≠ 0 put Y(NT - NP + 1), ..., Y(NT) into X(1), ..., X(NP).

  Calculate Y(NT + 1|NT), ..., Y(NT + NVL|NT) and store in X(NP + 1), ..., X(NP + NVL).

```
      Put X(NP + NVF), ...., X(NP + NVL) into YPD((NT - NTF)*N1 + 1),
     ...., YPD((NT - NTF + 1)*N1), where N1 = NVL - NVF + 1.
  150 CONTINUE
```

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

```
C
      CALL MAORTH(IOPTR,IOPTE,NQ,DEL,NX,NX,X,BETA,SIGSQ,IROWS1,
     1R,RO,WKL,D,AL,E,IF1)
C
      IFAULT=5
      IF(IF1.EQ.5) GO TO 160
      DO 30 I=1,NX
      N1=NOBS-I+1
      N2=N1-NP
      E(N1)=E(N2)
   30 CONTINUE
C
C    FIND PREDICTORS
C
   40 CONTINUE
      N1=NVL-NVF+1
      N2=NP+NVF
      DO 150 NT=NTF,NTL
      N3=(NT-NTF)*N1
      NTEMP1=NT-NP-NQ
      NTEMP=NT-NP
      IF((NPPNQ.EQ.0).OR.(NT.LT.1).OR.(NT.GT.NOBS)) GO TO 55
      IF((NP.NE.0).AND.(NT.LT.NPPNQ)) GO TO 55
      GO TO 60
   55 CONTINUE
         DO 50 I=1,N1
         N3PI=N3+I
         YPD(N3PI)=ZERO
   50    CONTINUE
      GO TO 150
   60 CONTINUE
      IF(NP.EQ.0) GO TO 80
         DO 70 I=1,NP
         N4=NTEMP+I
         X(I)=Y(N4)
   70    CONTINUE

   80 CONTINUE
         DO 130 NV=1,NVL
         C=ZERO
         IF((NQ.EQ.0).OR.(NV.GT.NQ)) GO TO 100
         NTPNV=NT+NV
      IF((NP.EQ.0).AND.(NTPNV.LE.NQ)) GO TO 100
            DO 90 K=NV,NQ
            KK=NTPNV-K
            C=C+BETA(K)*E(KK)
   90       CONTINUE
  100    CONTINUE
         NPPNV=NP+NV
         IF(NP.EQ.0) GO TO 120
            DO 110 I=1,NP
            NP1=NPPNV-I
            C=C-ALPHA(I)*X(NP1)
  110       CONTINUE
  120    CONTINUE
         X(NPPNV)=C
  130    CONTINUE
         DO 140 NV=1,N1
         N3PNV=N3+NV
         N5=N2+NV-1
         YPD(N3PNV)=X(N5)
  140    CONTINUE
  150 CONTINUE
C
      IFAULT=0
  160 CONTINUE
      RETURN
      END
      SUBROUTINE MAORTH(IOPTR,IOPTE,NQ,DEL,ITERS,IROWS1,Y,BETA,
     1SIGSQ,IROWS2,R,RO,WKL,D,AL,E,IFAULT)
C
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
      IFAULT=1
      IF(NQ.LT.1) GO TO 299
      IFAULT=2
      IF((IOPTB.NE.0).AND.(IOPTB.NE.1)) GO TO 299
      IFAULT=3
      IF((IOPTE.NE.0).AND.(IOPTE.NE.1)) GO TO 299
      IFAULT=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(IOPTE.EQ.1) E(1)=Y(1)
C
C   ROW K OF FACTORIZATION
C   M1 IS NUMBER OF LEADING ZEROS
C   M3 IS NUMBER OF ELEMENTS TO CALCULATE
C   MR IS SHIFTING FACTOR IN STORAGE OF PREVIOUS ROWS
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
            INDLC=M4+I
            C=C-D(INDD1)*WKL(I)*AL(INDLR,INDLC)
   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(IOPTE.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(K.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(IOPTB.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(IOPTB.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.ITERS) GO TO 199
      KP1=K+1
      DO 170 J=KP1,ITERS
      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,IFAULT)
```

```
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
```