

# NAG Fortran Library Routine Document

## F04MEF

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

### 1 Purpose

F04MEF updates the solution to the Yule–Walker equations for a real symmetric positive-definite Toeplitz system.

### 2 Specification

```
SUBROUTINE F04MEF(N, T, X, V, WORK, IFAIL)
INTEGER          N, IFAIL
real           T(O:N), X(*), V, WORK(*)
```

### 3 Description

This routine solves the equations

$$T_n x_n = -t_n,$$

where  $T_n$  is the  $n$  by  $n$  symmetric positive-definite Toeplitz matrix

$$T_n = \begin{pmatrix} \tau_0 & \tau_1 & \tau_2 & \cdots & \tau_{n-1} \\ \tau_1 & \tau_0 & \tau_1 & \cdots & \tau_{n-2} \\ \tau_2 & \tau_1 & \tau_0 & \cdots & \tau_{n-3} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \tau_{n-1} & \tau_{n-2} & \tau_{n-3} & \cdots & \tau_0 \end{pmatrix}$$

and  $t_n$  is the vector

$$t_n^T = (\tau_1 \tau_2 \cdots \tau_n),$$

given the solution of the equations

$$T_{n-1} x_{n-1} = -t_{n-1}.$$

The routine will normally be used to successively solve the equations

$$T_k x_k = -t_k, \quad k = 1, 2, \dots, n.$$

If it is desired to solve the equations for a single value of  $n$ , then routine F04FEF may be called. This routine uses the method of Durbin (see Durbin (1960) and Golub and van Loan (1996)).

### 4 References

Bunch J R (1985) Stability of methods for solving Toeplitz systems of equations *SIAM J. Sci. Statist. Comput.* **6** 349–364

Bunch J R (1987) The weak and strong stability of algorithms in numerical linear algebra *Linear Algebra Appl.* **88/89** 49–66

Cybenko G (1980) The numerical stability of the Levinson–Durbin algorithm for Toeplitz systems of equations *SIAM J. Sci. Statist. Comput.* **1** 303–319

Durbin J (1960) The fitting of time series models *Rev. Inst. Internat. Stat.* **28** 233

Golub G H and van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

## 5 Parameters

- 1: N – INTEGER *Input*  
*On entry:* the order of the Toeplitz matrix  $T$ .  
*Constraint:*  $N \geq 0$ . When  $N=0$ , then an immediate return is effected.
- 2: T(0:N) – *real* array *Input*  
*On entry:* T(0) must contain the value  $\tau_0$  of the diagonal elements of  $T$ , and the remaining N elements of T must contain the elements of the vector  $t_n$ .  
*Constraint:* T(0) > 0.0. Note that if this is not true, then the Toeplitz matrix cannot be positive-definite.
- 3: X(\*) – *real* array *Input/Output*  
**Note:** the dimension of the array X must be at least  $\max(1, N)$ .  
*On entry:* with  $N > 1$  the  $(n - 1)$  elements of the solution vector  $x_{n-1}$  as returned by a previous call to this routine. The element X(N) need not be specified.  
*Constraint:*  $|X(N - 1)| < 1.0$ . Note that this is the partial (auto)correlation coefficient, or reflection coefficient, for the  $(n - 1)$ th step. If the constraint does not hold, then  $T_n$  cannot be positive-definite.  
*On exit:* the solution vector  $x_n$ . The element X(N) returns the partial (auto)correlation coefficient, or reflection coefficient, for the  $n$ th step. If  $|X(N)| \geq 1.0$ , then the matrix  $T_{n+1}$  will not be positive-definite to working accuracy.
- 4: V – *real* *Input/Output*  
*On entry:* with  $N > 1$  the mean square prediction error for the  $(n - 1)$ th step, as returned by a previous call to this routine.  
*On exit:* the mean square prediction error, or predictor error variance ratio,  $\nu_n$ , for the  $n$ th step. (See Section 8 and the Introduction to Chapter G13.)
- 5: WORK(\*) – *real* array *Workspace*  
**Note:** the dimension of the array WORK must be at least  $\max(1, N - 1)$ .
- 6: IFAIL – INTEGER *Input/Output*  
*On entry:* IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.  
*On exit:* IFAIL = 0 unless the routine detects an error (see Section 6).  
 For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

## 6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = -1

- On entry,  $N < 0$ ,
- or  $T(0) \leq 0.0$ ,
- or  $N > 1$  and  $|X(N - 1)| \geq 1.0$ .

IFAIL = 1

The Toeplitz matrix  $T_{n+1}$  is not positive-definite to working accuracy. If, on exit, X(N) is close to unity, then the principal minor was probably close to being singular, and the sequence  $\tau_0, \tau_1, \dots, \tau_N$  may be a valid sequence nevertheless. X returns the solution of the equations

$$T_n x_n = -t_n,$$

and V returns  $v_n$ , but it may not be positive.

## 7 Accuracy

The computed solution of the equations certainly satisfies

$$r = T_n x_n + t_n,$$

where  $\|r\|_1$  is approximately bounded by

$$\|r\|_1 \leq c\epsilon \left( \prod_{i=1}^n (1 + |p_i|) - 1 \right),$$

$c$  being a modest function of  $n$ ,  $\epsilon$  being the *machine precision* and  $p_k$  being the  $k$ th element of  $x_k$ . This bound is almost certainly pessimistic, but it has not yet been established whether or not the method of Durbin is backward stable. For further information on stability issues see Bunch (1985), Bunch (1987), Cybenko (1980) and Golub and van Loan (1996). The following bounds on  $\|T_n^{-1}\|_1$  hold:

$$\max \left( \frac{1}{v_{n-1}}, \frac{1}{\prod_{i=1}^{n-1} (1 - p_i)} \right) \leq \|T_n^{-1}\|_1 \leq \prod_{i=1}^{n-1} \left( \frac{1 + |p_i|}{1 - |p_i|} \right),$$

where  $v_n$  is the mean square prediction error for the  $n$ th step. (See Cybenko (1980).) Note that  $v_n < v_{n-1}$ . The norm of  $T_n^{-1}$  may also be estimated using routine F04YCF.

## 8 Further Comments

The number of floating-point operations used by this routine is approximately  $4n$ .

The mean square prediction errors,  $v_i$ , is defined as

$$v_i = (\tau_0 + t_{i-1}^T x_{i-1}) / \tau_0.$$

Note that  $v_i = (1 - p_i^2)v_{i-1}$ .

## 9 Example

To find the solution of the Yule–Walker equations  $T_k x_k = -t_k$ ,  $k = 1, 2, 3, 4$  where

$$T_4 = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 4 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix} \quad \text{and} \quad t_4 = \begin{pmatrix} 3 \\ 2 \\ 1 \\ 0 \end{pmatrix}.$$

### 9.1 Program Text

**Note:** the listing of the example program presented below uses *bold italicised* terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      F04MEF Example Program Text
*      Mark 15 Release. NAG Copyright 1991.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER       (NIN=5, NOUT=6)
      INTEGER          NMAX
      PARAMETER       (NMAX=100)
```

```

*   .. Local Scalars ..
   real          V
   INTEGER       I, IFAIL, K, N
*   .. Local Arrays ..
   real          T(0:NMAX), WORK(NMAX-1), X(NMAX)
*   .. External Subroutines ..
   EXTERNAL      F04MEF
*   .. Executable Statements ..
   WRITE (NOUT,*) 'F04MEF Example Program Results'
*   Skip heading in data file
   READ (NIN,*)
   READ (NIN,*) N
   WRITE (NOUT,*)
   IF ((N.LT.0) .OR. (N.GT.NMAX)) THEN
     WRITE (NOUT,99999) 'N is out of range. N = ', N
   ELSE
     READ (NIN,*) (T(I),I=0,N)
*
     DO 20 K = 1, N
*
       IFAIL = 0
*
       CALL F04MEF(K,T,X,V,WORK,IFAIL)
*
       WRITE (NOUT,*)
       WRITE (NOUT,99999) 'Solution for system of order', K
       WRITE (NOUT,99998) (X(I),I=1,K)
       WRITE (NOUT,*) 'Mean square prediction error'
       WRITE (NOUT,99998) V
   20   CONTINUE
     END IF
     STOP
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,5F9.4)
   END

```

## 9.2 Program Data

F04MEF Example Program Data

```

4           :Value of N
4.0  3.0  2.0  1.0  0.0  :End of vector T

```

## 9.3 Program Results

F04MEF Example Program Results

```

Solution for system of order  1
-0.7500
Mean square prediction error
 0.4375

Solution for system of order  2
-0.8571  0.1429
Mean square prediction error
 0.4286

Solution for system of order  3
-0.8333  0.0000  0.1667
Mean square prediction error
 0.4167

Solution for system of order  4
-0.8000  0.0000 -0.0000  0.2000
Mean square prediction error
 0.4000

```