NAG Library Routine Document

F04FEF

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

F04FEF solves the Yule–Walker equations for a real symmetric positive definite Toeplitz system.

2 Specification

```
SUBROUTINE F04FEF (N, T, X, WANTP, P, WANTV, V, VLAST, WORK, IFAIL)

INTEGER             N, IFAIL
REAL (KIND=nag_wp) T(0:N), X(N), P(*), V(*), VLAST, WORK(N-1)
LOGICAL             WANTP, WANTV
```

3 Description

F04FEF solves the equations

$$Tx = -t,$$

where T is the n by n symmetric positive definite Toeplitz matrix

$$T = \begin{pmatrix} \tau_0 & \tau_1 & \tau_2 & \dots & \tau_{n-1} \\ \tau_1 & \tau_0 & \tau_1 & \dots & \tau_{n-2} \\ \tau_2 & \tau_1 & \tau_0 & \dots & \tau_{n-3} \\ . & . & . & & . \\ \tau_{n-1} & \tau_{n-2} & \tau_{n-3} & \dots & \tau_0 \end{pmatrix}$$

and t is the vector

$$t^{\mathrm{T}} = (\tau_1, \tau_2 \dots \tau_n).$$

The routine uses the method of Durbin (see Durbin (1960) and Golub and Van Loan (1996)). Optionally the mean square prediction errors and/or the partial correlation coefficients for each step can be returned.

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 (KIND=nag_wp) array *Input*

On entry: T(0) must contain the value τ_0 of the diagonal elements of T, and the remaining N elements of T must contain the elements of the vector t.

Constraint: $T(0) > 0.0$. Note that if this is not true, then the Toeplitz matrix cannot be positive definite.

3: X(N) – REAL (KIND=nag_wp) array *Output*

On exit: the solution vector x.

4: WANTP – LOGICAL *Input*

On entry: must be set to .TRUE. if the partial (auto)correlation coefficients are required, and must be set to .FALSE. otherwise.

5: P($*$) – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array P must be at least $\max(1, N)$ if WANTP = .TRUE., and at least 1 otherwise.

On exit: with WANTP as .TRUE., the ith element of P contains the partial (auto)correlation coefficient, or reflection coefficient, p_i for the ith step. (See Section 8 and Chapter G13.) If WANTP is .FALSE., then P is not referenced. Note that in any case, $x_n = p_n$.

6: WANTV – LOGICAL *Input*

On entry: must be set to .TRUE. if the mean square prediction errors are required, and must be set to .FALSE. otherwise.

7: V($*$) – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array V must be at least $\max(1, N)$ if WANTV = .TRUE., and at least 1 otherwise.

On exit: with WANTV as .TRUE., the ith element of V contains the mean square prediction error, or predictor error variance ratio, v_i, for the ith step. (See Section 8 and Chapter G13.) If WANTV is .FALSE., then V is not referenced.

8: VLAST – REAL (KIND=nag_wp) *Output*

On exit: the value of v_n, the mean square prediction error for the final step.

9: WORK(N – 1) – REAL (KIND=nag_wp) array *Workspace*

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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, because for this routine the values of the output parameters may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

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).

Note: F04FEF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= -1$

On entry, N < 0,
or T(0) ≤ 0.0.

IFAIL > 0

The principal minor of order (IFAIL $+ 1$) of the Toeplitz matrix is not positive definite to working accuracy. If, on exit, x_{IFAIL} is close to unity, then the principal minor was close to being singular, and the sequence $\tau_0, \tau_1, \ldots, \tau_{\mathrm{IFAIL}}$ may be a valid sequence nevertheless. The first IFAIL elements of X return the solution of the equations

$$T_{\mathrm{IFAIL}} x = -(\tau_1, \tau_2, \ldots, \tau_{\mathrm{IFAIL}})^{\mathrm{T}},$$

where T_{IFAIL} is the IFAILth principal minor of T. Similarly, if WANTP and/or WANTV are true, then P and/or V return the first IFAIL elements of P and V respectively and VLAST returns v_{IFAIL}. In particular if IFAIL $=$ N, then the solution of the equations $Tx = -t$ is returned in X, but τ_{N} is such that $T_{\mathrm{N+1}}$ would not be positive definite to working accuracy.

7 Accuracy

The computed solution of the equations certainly satisfies

$$r = Tx + t,$$

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

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

c being a modest function of n and ϵ being the ***machine precision***. This bound is almost certainly pessimistic, but it has not yet been established whether or not the method of Durbin is backward stable. If $|p_n|$ is close to one, then the Toeplitz matrix is probably ill-conditioned and hence only just positive definite. For further information on stability issues see Bunch (1985), Bunch (1987), Cybenko (1980) and Golub and Van Loan (1996). The following bounds on $\|\mathrm{T}^{-1}\|_1$ hold:

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

Note: $v_n < v_{n-1}$. The norm of T^{-1} may also be estimated using routine F04YDF.

8 Further Comments

The number of floating point operations used by F04FEF is approximately $2n^2$, independent of the values of WANTP and WANTV.

The mean square prediction error, v_i, is defined as

$$v_i = (\tau_0 + (\tau_1 \tau_2 \ldots \tau_{i-1}) y_{i-1}) / \tau_0,$$

where y_i is the solution of the equations

$$T_i y_i = -(\tau_1 \tau_2 \ldots \tau_i)^{\mathrm{T}}$$

and the partial correlation coefficient, p_i, is defined as the ith element of y_i. Note that $v_i = \left(1 - p_i^2\right) v_{i-1}$.

9 Example

This example finds the solution of the Yule–Walker equations $Tx = -t$, where

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

9.1 Program Text

```
    Program f04fefe

!     F04FEF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: f04fef, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)                :: vlast
      Integer                           :: ifail, n
      Logical                           :: wantp, wantv
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable   :: p(:), t(:), v(:), work(:), x(:)
!     .. Executable Statements ..
      Write (nout,*) 'F04FEF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      Write (nout,*)

      Allocate (p(n),t(0:n),v(n),work(n-1),x(n))
      Read (nin,*) t(0:n)
      wantp = .True.
      wantv = .True.

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 1
      Call f04fef(n,t,x,wantp,p,wantv,v,vlast,work,ifail)

      If (ifail==0) Then
        Write (nout,*)
        Write (nout,*) 'Solution vector'
        Write (nout,99998) x(1:n)
        If (wantp) Then
          Write (nout,*)
          Write (nout,*) 'Reflection coefficients'
          Write (nout,99998) p(1:n)
        End If
        If (wantv) Then
          Write (nout,*)
          Write (nout,*) 'Mean square prediction errors'
```

```
          Write (nout,99998) v(1:n)
        End If
      Else If (ifail>0) Then
        Write (nout,*)
        Write (nout,99999) 'Solution for system of order', ifail
        Write (nout,99998) x(1:ifail)
        If (wantp) Then
          Write (nout,*)
          Write (nout,*) 'Reflection coefficients'
          Write (nout,99998) p(1:ifail)
        End If
        If (wantv) Then
          Write (nout,*)
          Write (nout,*) 'Mean square prediction errors'
          Write (nout,99998) v(1:ifail)
        End If
      Else
        Write (nout,99997) ifail
      End If

99999 Format (1X,A,I5)
99998 Format (1X,5F9.4)
99997 Format (1X,' ** F04FEF returned with IFAIL = ',I5)
    End Program f04fefe
```

9.2 Program Data

```
F04FEF Example Program Data

   4                            : n
   4.0  3.0  2.0  1.0  0.0      : vector T
```

9.3 Program Results

```
 F04FEF Example Program Results


 Solution vector
   -0.8000   0.0000  -0.0000   0.2000

 Reflection coefficients
   -0.7500   0.1429   0.1667   0.2000

 Mean square prediction errors
    0.4375   0.4286   0.4167   0.4000
```