F04FFF – NAG Fortran Library Routine Document

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

F04FFF solves the equations $Tx = b$, where T is a real symmetric positive-definite Toeplitz matrix.

2 Specification

```
SUBROUTINE F04FFF(N, T, B, X, WANTP, P, WORK, IFAIL)
INTEGER           N, IFAIL
real              T(0:*), B(*), X(*), P(*), WORK(*)
LOGICAL           WANTP
```

3 Description

This routine solves the equations

$$Tx = b,$$

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 b is an n element vector.

The routine uses the method of Levinson [4], [5]. Optionally, the reflection coefficients for each step may also be returned.

4 References

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

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

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

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

[5] Levinson N (1947) The Weiner RMS error criterion in filter design and prediction *J. Math. Phys.* **25** 261–278

5 Parameters

1: N — INTEGER *Input*

On entry: the order of the Toeplitz matrix T.

Constraint: $\mathrm{N} \geq 0$. When $\mathrm{N} = 0$, then an immediate return is effected.

2: T(0:∗) — ***real*** array *Input*

Note: the dimension of the array T must be at least max(1,N).

On entry: T(i) must contain the value τ_i, $i = 0, 1, \ldots$,N−1.

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

3: B(∗) — ***real*** array *Input*

Note: the dimension of the array B must be at least max(1,N).

On entry: the right-hand side vector b.

4: X(∗) — ***real*** array *Output*

Note: the dimension of the array X must be at least max(1,N).

On exit: the solution vector x.

5: WANTP — LOGICAL *Input*

On entry: WANTP must be set to .TRUE. if the reflection coefficients are required, and must be set .FALSE. otherwise.

6: P(∗) — ***real*** array *Output*

Note. If WANTP = .TRUE., the dimension of the array P must be at least max(1,N−1); otherwise the dimension must be at least 1.

On exit: with WANTP as .TRUE., the ith element of P contains the reflection coefficient, p_i, for the ith step, for $i = 1, 2, \ldots$,N−1. (See Section 8.) If WANTP is .FALSE., then P is not referenced.

7: WORK(∗) — ***real*** array *Workspace*

Note: the dimension of the array WORK must be at least max(1,2∗(N−1)).

8: 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 or gives a warning (see Section 6).

For this routine, because the values of output parameters may be useful even if IFAIL ≠ 0 on exit, users are recommended to set IFAIL to −1 before entry. **It is then 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 specified by the routine:

IFAIL = −1

On entry, N < 0,
or T(0) ≤ 0.0.

IFAIL > 0

The principal minor of order IFAIL of the Toeplitz matrix is not positive-definite to working accuracy. The first (IFAIL−1) elements of X return the solution of the equations

$$T_{\mathrm{IFAIL}-1}x = (b_1, b_2, \ldots, b_{\mathrm{IFAIL}-1})^T,$$

where T_k is the kth principal minor of T.

7 Accuracy

The computed solution of the equations certainly satisfies

$$r = Tx - b,$$

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

$$\|r\| \leq c\epsilon C(T),$$

c being a modest function of n, ϵ being the ***machine precision*** and $C(T)$ is the condition number of T with respect to inversion. This bound is almost certainly pessimistic, but it seems unlikely that the method of Levinson is backward stable, so caution should be exercised when T is ill-conditioned. The following bound on T^{-1} holds,

$$\max\left(\frac{1}{\prod_{i=1}^{n-1}(1-p_i^2)}, \frac{1}{\prod_{i=1}^{n-1}(1-p_i)}\right) \leq \|T^{-1}\|_1 \leq \prod_{i=1}^{n-1}\left(\frac{1+|p_i|}{1-|p_i|}\right).$$

(See Golub and Van Loan [4].) The norm of T^{-1} may also be estimated using routine F04YCF. For further information on stability issues see Bunch [1] and [2], Cybenko [3] and Golub and Van Loan [4].

8 Further Comments

The number of floating-point operations used by this routine is approximately $4n^2$.

If y_i is the solution of the equations

$$T_i y_i = -(\tau_1 \tau_2 \ldots \tau_i)^T,$$

then the partial correlation coefficient p_i is defined as the ith element of y_i.

9 Example

To find the solution of the equations $Tx = b$, where

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

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.

```
*     F04FFF 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 ..
      INTEGER          I, IFAIL, N
      LOGICAL          WANTP
*     .. Local Arrays ..
      real             B(NMAX), P(NMAX-1), T(0:NMAX-1),
     +                 WORK(2*(NMAX-1)), X(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F04FFF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04FFF 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-1)
         READ (NIN,*) (B(I),I=1,N)
         WANTP = .TRUE.
*
         IFAIL = -1
*
         CALL F04FFF(N,T,B,X,WANTP,P,WORK,IFAIL)
*
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Solution vector'
            WRITE (NOUT,99998) (X(I),I=1,N)
            IF (WANTP) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Reflection coefficients'
               WRITE (NOUT,99998) (P(I),I=1,N-1)
            END IF
         ELSE IF (IFAIL.GT.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Solution for system of order', IFAIL - 1
            WRITE (NOUT,99998) (X(I),I=1,IFAIL-1)
            IF (WANTP) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Reflection coefficients'
               WRITE (NOUT,99998) (P(I),I=1,IFAIL-1)
            END IF
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,5F9.4)
      END
```

9.2 Program Data

```
F04FFF Example Program Data

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

9.3 Program Results

```
 F04FFF Example Program Results


 Solution vector
    0.2000   0.0000   0.0000   0.2000
```

```
Reflection coefficients
  -0.7500   0.1429   0.1667
```