F04EAF – 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

F04EAF calculates the approximate solution of a set of real tridiagonal linear equations with a single right-hand side.

2 Specification

```
SUBROUTINE F04EAF(N, D, DU, DL, B, IFAIL)
INTEGER           N, IFAIL
real              D(N), DU(N), DL(N), B(N)
```

3 Description

F04EAF is based on the Linpack routine SGTSL (see Dongarra *et al.* [1]) and solves the equation

$$Tx = b,$$

where T is a real n by n tridiagonal matrix, by Gaussian elimination with partial pivoting. This routine should only be used if T is known not to be nearly singular since the routine only tests for exactly zero pivots.

4 References

[1] Dongarra J J, Moler C B, Bunch J R and Stewart G W (1979) *LINPACK Users' Guide* SIAM, Philadelphia

[2] Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer-Verlag

5 Parameters

1: N — INTEGER *Input*

On entry: n, the order of the matrix T.

Constraint: $N \geq 1$.

2: D(N) — ***real*** array *Input/Output*

On entry: the diagonal elements of T.

On exit: the diagonal elements of the upper triangular matrix, U, of the factorization of T.

3: DU(N) — ***real*** array *Input/Output*

On entry: the super-diagonal elements of T, stored in DU(2) to DU(n); DU(1) is not used.

On exit: the elements of the first super-diagonal of U, stored in DU(2) to DU(n).

4: DL(N) — ***real*** array *Input/Output*

On entry: the sub-diagonal elements of T, stored in DL(2) to DL(n); DL(1) is not used.

On exit: the elements of the second super-diagonal of U, stored in DL(3) to DL(n).

5: B(N) — ***real*** array *Input/Output*

On entry: the right-hand side vector b.

On exit: B is overwritten by the solution vector x.

6: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

On entry, N < 1.

IFAIL > 1

The (IFAIL−1)th diagonal element of U is zero. Unless IFAIL = N + 1, the factorization of T will not have been completed.

Overflow

If overflow occurs during execution of this routine then either a diagonal element of U, while not being zero, is very small, or an element of b is very large. In the former case T is probably nearly singular and it would be advisable to use F01LEF, while in the latter case the equations should be scaled so that the elements of b are of more moderate size.

Underflow

Any underflows which occur during the execution of this routine are harmless, but users who wish to avoid underflow and whose installation returns the value .TRUE. from X02DAF may prefer to use the combination F01LEF and F04LEF to solve their tridiagonal equations.

7 Accuracy

The computed solution, say $\bar{x}$, of the equations $Tx = b$ will satisfy the equation

$$(T + E)\bar{x} = b,$$

where E can be expected to satisfy a bound of the form

$$\|E\| \leq \alpha\epsilon\|T\|,$$

α being a modest constant and ϵ being the ***machine precision***. The above result implies that the relative error in $\bar{x}$ satisfies the bound

$$\frac{\|\bar{x} - x\|}{\|\bar{x}\|} \leq c(T)\alpha\epsilon,$$

where $c(T)$ is the condition number of T with respect to inversion. Thus if T is nearly singular, x can be expected to have a high relative error.

8 Further Comments

The time taken by the routine is approximately proportional to n.

The tridiagonal equations $T^T x = b$ may be solved by interchanging the parameters DU and DL in the call to this routine.

Users with multiple right-hand sides or users who are uncertain as to whether or not their matrix T is nearly singular, may wish to be aware that the combination F01LEF and F04LEF can also be used to solve tridiagonal equations, but that this combination requires more storage and will usually be slower than F04EAF.

9 Example

To solve the system of tridiagonal equations $Tx = b$, where

$$T = \begin{pmatrix} 3.0 & 2.1 & 0 & 0 & 0 \\ 3.4 & 2.3 & -1.0 & 0 & 0 \\ 0 & 3.6 & -5.0 & 1.9 & 0 \\ 0 & 0 & 7.0 & -0.9 & 8.0 \\ 0 & 0 & 0 & -6.0 & 7.1 \end{pmatrix} \text{ and } b = \begin{pmatrix} 2.7 \\ -0.5 \\ 2.6 \\ 0.6 \\ 2.7 \end{pmatrix}.$$

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.

```
*     F04EAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX
      PARAMETER        (NMAX=100)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, N
*     .. Local Arrays ..
      real             B(NMAX), D(NMAX), DL(NMAX), DU(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F04EAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04EAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      WRITE (NOUT,*)
      IF (N.LT.1 .OR. N.GT.NMAX) THEN
         WRITE (NOUT,99999) 'N is out of range: N = ', N
      ELSE
         READ (NIN,*) (D(I),I=1,N)
         IF (N.GT.1) THEN
            READ (NIN,*) (DU(I),I=2,N)
            READ (NIN,*) (DL(I),I=2,N)
         END IF
         READ (NIN,*) (B(I),I=1,N)
         IFAIL = 1
*
         CALL F04EAF(N,D,DU,DL,B,IFAIL)
*
         IF (IFAIL.NE.0) THEN
            WRITE (NOUT,99999) 'F04EAF fails. IFAIL =', IFAIL
         ELSE
            WRITE (NOUT,*) 'Solution vector'
            WRITE (NOUT,99998) (B(I),I=1,N)
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,5F9.3)
      END
```

9.2 Program Data

```
F04EAF Example Program Data
   5
  3.0   2.3  -5.0  -0.9   7.1
  2.1  -1.0   1.9   8.0
  3.4   3.6   7.0  -6.0
  2.7  -0.5   2.6   0.6   2.7
```

9.3 Program Results

```
F04EAF Example Program Results

Solution vector
   -4.000    7.000    3.000   -4.000   -3.000
```