NAG Library Routine Document

F01LEF

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

F01LEF computes an LU factorization of a real tridiagonal matrix, using Gaussian elimination with partial pivoting.

2 Specification

```
SUBROUTINE F01LEF (N, A, LAMBDA, B, C, TOL, D, IPIV, IFAIL)
INTEGER             N, IPIV(N), IFAIL
REAL (KIND=nag_wp) A(N), LAMBDA, B(N), C(N), TOL, D(N)
```

3 Description

The matrix $T - \lambda I$, where T is a real n by n tridiagonal matrix, is factorized as

$$T - \lambda I = PLU,$$

where P is a permutation matrix, L is a unit lower triangular matrix with at most one nonzero subdiagonal element per column, and U is an upper triangular matrix with at most two nonzero superdiagonal elements per column.

The factorization is obtained by Gaussian elimination with partial pivoting and implicit row scaling.

An indication of whether or not the matrix $T - \lambda I$ is nearly singular is returned in the nth element of the array IPIV. If it is important that $T - \lambda I$ is nonsingular, as is usually the case when solving a system of tridiagonal equations, then it is strongly recommended that IPIV(n) is inspected on return from F01LEF. (See the parameter IPIV and Section 9 for further details.)

The parameter λ is included in the routine so that F01LEF may be used, in conjunction with F04LEF, to obtain eigenvectors of T by inverse iteration.

4 References

Wilkinson J H (1965) *The Algebraic Eigenvalue Problem* Oxford University Press, Oxford

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: A(N) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the diagonal elements of T.

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

3: LAMBDA – REAL (KIND=nag_wp) *Input*

On entry: the scalar λ. F01LEF factorizes $T - \lambda I$.

4: B(N) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the superdiagonal elements of T, stored in B(2) to B(n); B(1) is not used.

On exit: the elements of the first superdiagonal of U, stored in B(2) to B(n).

5: C(N) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the subdiagonal elements of T, stored in C(2) to C(n); C(1) is not used.

On exit: the subdiagonal elements of L, stored in C(2) to C(n).

6: TOL – REAL (KIND=nag_wp) *Input*

On entry: a relative tolerance used to indicate whether or not the matrix $(T - \lambda I)$ is nearly singular. TOL should normally be chosen as approximately the largest relative error in the elements of T. For example, if the elements of T are correct to about 4 significant figures, then TOL should be set to about 5×10^{-4}. See Section 9 for further details on how TOL is used. If TOL is supplied as less than ϵ, where ϵ is the ***machine precision***, then the value ϵ is used in place of TOL.

7: D(N) – REAL (KIND=nag_wp) array *Output*

On exit: the elements of the second superdiagonal of U, stored in D(3) to D(n); D(1) and D(2) are not used.

8: IPIV(N) – INTEGER array *Output*

On exit: details of the permutation matrix P. If an interchange occurred at the kth step of the elimination, then IPIV$(k) = 1$, otherwise IPIV$(k) = 0$. If a diagonal element of U is small, indicating that $(T - \lambda I)$ is nearly singular, then the element IPIV(n) is returned as positive. Otherwise IPIV(n) is returned as 0. See Section 9 for further details. If the application is such that it is important that $(T - \lambda I)$ is not nearly singular, then it is strongly recommended that IPIV(n) is inspected on return.

9: 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, if you are 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.**

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).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, N < 1.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The computed factorization will satisfy the equation

$$PLU = (T - \lambda I) + E,$$

where

$$\|E\|_1 \leq 9 \times \max_{i \geq j} \left(|l_{ij}|, |l_{ij}|^2 \right) \epsilon \|T - \lambda I\|_1$$

where ϵ is the ***machine precision***.

8 Parallelism and Performance

F01LEF is not threaded by NAG in any implementation.

F01LEF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time taken by F01LEF is approximately proportional to n.

The factorization of a tridiagonal matrix proceeds in $(n - 1)$ steps, each step eliminating one subdiagonal element of the tridiagonal matrix. In order to avoid small pivot elements and to prevent growth in the size of the elements of L, rows k and $(k + 1)$ will, if necessary, be interchanged at the kth step prior to the elimination.

The element IPIV(n) returns the smallest integer, j, for which

$$|u_{jj}| \leq \left\| (T - \lambda I)_j \right\|_1 \times \text{TOL},$$

where $\left\| (T - \lambda I)_j \right\|_1$ denotes the sum of the absolute values of the jth row of the matrix $(T - \lambda I)$. If no such j exists, then IPIV(n) is returned as zero. If such a j exists, then $|u_{jj}|$ is small and hence $(T - \lambda I)$ is singular or nearly singular.

This routine may be followed by F04LEF to solve systems of tridiagonal equations. If you wish to solve single systems of tridiagonal equations you should be aware of F07CAF (DGTSV), which solves tridiagonal systems with a single call. F07CAF (DGTSV) requires less storage and will generally be faster than the combination of F01LEF and F04LEF, but no test for near singularity is included in F07CAF (DGTSV) and so it should only be used when the equations are known to be nonsingular.

10 Example

This example factorizes the tridiagonal matrix T where

$$T = \begin{pmatrix} 3.0 & 2.1 & 0 & 0 & 0 \\ 3.4 & 2.3 & -1.0 & 0 & 0 \\ 0 & 3.6 & -5.0 & 1.9 & 0 \\ 0 & 0 & 7.0 & -0.9 & 8.0 \\ 0 & 0 & 0 & -6.0 & 7.1 \end{pmatrix}$$

and then to solve the equations $Tx = y$, where

$$y = \begin{pmatrix} 2.7 \\ -0.5 \\ 2.6 \\ 0.6 \\ 2.7 \end{pmatrix}$$

by a call to F04LEF. The example program sets $\mathrm{TOL} = 5 \times 10^{-5}$ and, of course, sets $\mathrm{LAMBDA} = 0$.

10.1 Program Text

```
    Program f01lefe

!     F01LEF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f01lef, f04lef, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: lambda, tol
      Integer                          :: ifail, job, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), b(:), c(:), d(:), y(:)
      Integer, Allocatable             :: ipiv(:)
!     .. Executable Statements ..
      Write (nout,*) 'F01LEF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      Allocate (a(n),b(n),c(n),d(n),y(n),ipiv(n))
      Read (nin,*) a(1:n)
      Read (nin,*) b(2:n)
      Read (nin,*) c(2:n)
      tol = 0.00005E0_nag_wp
      lambda = 0.0E0_nag_wp

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f01lef(n,a,lambda,b,c,tol,d,ipiv,ifail)

      If (ipiv(n)/=0) Then
        Write (nout,*) 'Matrix is singular or nearly singular'
        Write (nout,99999) 'Diagonal element', ipiv(n), 'is small'
      Else
        Write (nout,*) 'Details of factorization'
        Write (nout,*)
        Write (nout,*) ' Main diagonal of U'
        Write (nout,99998) a(1:n)
        Write (nout,*)
        Write (nout,*) ' First super-diagonal of U'
        Write (nout,99998) b(2:n)
        Write (nout,*)
        Write (nout,*) ' Second super-diagonal of U'
        Write (nout,99998) d(3:n)
        Write (nout,*)
        Write (nout,*) ' Multipliers'
        Write (nout,99998) c(2:n)
        Write (nout,*)
        Write (nout,*) ' Vector of interchanges'
        Write (nout,99997) ipiv(1:(n-1))

        Read (nin,*) y(1:n)
        job = 1

        ifail = 0
        Call f04lef(job,n,a,b,c,d,ipiv,y,tol,ifail)

        Write (nout,*)
        Write (nout,*) ' Solution vector'
        Write (nout,99998) y(1:n)
      End If

99999 Format (1X,A,I4,A)
99998 Format (1X,8F9.4)
99997 Format (1X,5I9)
    End Program f01lefe
```

10.2 Program Data

```
F01LEF Example Program Data
   5                               : n
  3.0   2.3  -5.0  -0.9   7.1      : a
  2.1  -1.0   1.9   8.0            : b
  3.4   3.6   7.0  -6.0            : c
  2.7  -0.5   2.6   0.6   2.7      : y
```

10.3 Program Results

```
 F01LEF Example Program Results

 Details of factorization

  Main diagonal of U
    3.0000   3.6000   7.0000  -6.0000   1.1508

  First super-diagonal of U
    2.1000  -5.0000  -0.9000   7.1000

  Second super-diagonal of U
    0.0000   1.9000   8.0000

  Multipliers
    1.1333  -0.0222  -0.1587   0.0168

  Vector of interchanges
         0        1        1        1

  Solution vector
   -4.0000   7.0000   3.0000  -4.0000  -3.0000
```