NAG Library Routine Document

F01RJF

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

F01RJF finds the RQ factorization of the complex m by n ($m \leq n$), matrix A, so that A is reduced to upper triangular form by means of unitary transformations from the right.

2 Specification

```
SUBROUTINE F01RJF (M, N, A, LDA, THETA, IFAIL)

INTEGER               M, N, LDA, IFAIL
COMPLEX (KIND=nag_wp) A(LDA,*), THETA(M)
```

3 Description

The m by n matrix A is factorized as

$$A = \begin{pmatrix} R & 0 \end{pmatrix} P^{\mathrm{H}} \quad \text{when } m < n,$$

$$A = RP^{\mathrm{H}} \quad \text{when } m = n,$$

where P is an n by n unitary matrix and R is an m by m upper triangular matrix.

P is given as a sequence of Householder transformation matrices

$$P = P_m \cdots P_2 P_1,$$

the $(m - k + 1)$th transformation matrix, P_k, being used to introduce zeros into the kth row of A. P_k has the form

$$P_k = I - \gamma_k u_k u_k^H,$$

where

$$u_k = \begin{pmatrix} w_k \\ \zeta_k \\ 0 \\ z_k \end{pmatrix}.$$

γ_k is a scalar for which $\mathrm{Re}(\gamma_k) = 1.0$, ζ_k is a real scalar, w_k is a $(k - 1)$ element vector and z_k is an $(n - m)$ element vector. γ_k and u_k are chosen to annihilate the elements in the kth row of A.

The scalar γ_k and the vector u_k are returned in the kth element of THETA and in the kth row of A, such that θ_k, given by

$$\theta_k = (\zeta_k, \mathrm{Im}(\gamma_k)).$$

is in THETA(k), the elements of w_k are in $\mathrm{A}(k,1), \ldots, \mathrm{A}(k,k-1)$ and the elements of z_k are in $\mathrm{A}(k,m+1), \ldots, \mathrm{A}(k,n)$. The elements of R are returned in the upper triangular part of A.

4 References

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

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

5 Parameters

1: M – INTEGER *Input*

On entry: m, the number of rows of the matrix A.

When $\mathrm{M} = 0$ then an immediate return is effected.

Constraint: $\mathrm{M} \geq 0$.

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

Constraint: $\mathrm{N} \geq \mathrm{M}$.

3: A(LDA,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1,\mathrm{N})$.

On entry: the leading m by n part of the array A must contain the matrix to be factorized.

On exit: the m by m upper triangular part of A will contain the upper triangular matrix R, and the m by m strictly lower triangular part of A and the m by $(n-m)$ rectangular part of A to the right of the upper triangular part will contain details of the factorization as described in Section 3.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F01RJF is called.

Constraint: $\mathrm{LDA} \geq \max(1,\mathrm{M})$.

5: THETA(M) – COMPLEX (KIND=nag_wp) array *Output*

On exit: THETA(k) contains the scalar θ_k for the $(m-k+1)$th transformation. If $P_k = I$ then $\mathrm{THETA}(k) = 0.0$; if

$$T_k = \begin{pmatrix} I & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & I \end{pmatrix}, \qquad \mathrm{Re}(\alpha) < 0.0$$

then $\mathrm{THETA}(k) = \alpha$, otherwise THETA($k$) contains θ_k as described in Section 3 and θ_k is always in the range $\left(1.0, \sqrt{2.0}\right)$.

6: 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: $\mathrm{IFAIL} = 0$ 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, M < 0,
or N $<$ M,
or LDA $<$ M.

7 Accuracy

The computed factors R and P satisfy the relation

$$(R0)P^{\mathrm{H}} = A + E,$$

where

$$\|E\| \leq c\epsilon\|A\|,$$

ϵ is the ***machine precision*** (see X02AJF), c is a modest function of m and n, and $\|.\|$ denotes the spectral (two) norm.

8 Further Comments

The approximate number of floating point operations is given by $8m^2(3n - m)/3$.

The first k rows of the unitary matrix P^{H} can be obtained by calling F01RKF, which overwrites the k rows of P^{H} on the first k rows of the array A. P^{H} is obtained by the call:

```
IFAIL = 0
CALL F01RKF('Separate',M,N,K,A,LDA,THETA,WORK,IFAIL)
```

WORK must be a $\max(m-1, k-m, 1)$ element array. If K is larger than M, then A must have been declared to have at least K rows.

Operations involving the matrix R can readily be performed by the Level 2 BLAS routines F06SFF (ZTRMV) and F06SJF (ZTRSV), (see Chapter F06), but note that no test for near singularity of R is incorporated into F06SFF (ZTRMV). If R is singular, or nearly singular then F02XUF can be used to determine the singular value decomposition of R.

9 Example

This example obtains the RQ factorization of the 3 by 5 matrix

$$A = \begin{pmatrix} -0.5i & 0.4 - 0.3i & 0.4 & 0.3 - 0.4i & 0.3i \\ -0.5 - 1.5i & 0.9 - 1.3i & -0.4 - 0.4i & 0.1 - 0.7i & 0.3 - 0.3i \\ -1.0 - 1.0i & 0.2 - 1.4i & 1.8 & 0.0 & -2.4i \end{pmatrix}.$$

9.1 Program Text

```
    Program f01rjfe

!     F01RJF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: f01rjf, nag_wp, x04dbf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
```

```
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lda, m, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), theta(:)
      Character (1)                    :: dummy(1)
!     .. Executable Statements ..
      Write (nout,*) 'F01RJF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      Write (nout,*)
      lda = m
      Allocate (a(lda,n),theta(m))
      Read (nin,*)(a(i,1:n),i=1,m)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
!     Find the RQ factorization of A
      Call f01rjf(m,n,a,lda,theta,ifail)

      Write (nout,*) 'RQ factorization of A'
      Write (nout,*)
      Write (nout,*) 'Vector THETA'
      Write (nout,99999) theta(1:m)
      Write (nout,*)
      Flush (nout)

      Call x04dbf('G',' ',m,n,a,lda,'B','F6.3', &
        'Matrix A after factorization (R is in left-hand upper triangle)','N', &
        dummy,'N',dummy,132,0,ifail)

99999 Format (5(' (',F6.3,',',F6.3,')':))
    End Program f01rjfe
```

9.2 Program Data

```
F01RJF Example Program Data
  3     5                                                              : m, n
( 0.00,-0.50) ( 0.40,-0.30) ( 0.40, 0.00) ( 0.30, 0.40) ( 0.00, 0.30)
(-0.50,-1.50) ( 0.90,-1.30) (-0.40,-0.40) ( 0.10,-0.70) ( 0.30,-0.30)
(-1.00,-1.00) ( 0.20,-1.40) ( 1.80, 0.00) ( 0.00, 0.00) ( 0.00,-2.40)   : a
```

9.3 Program Results

```
 F01RJF Example Program Results

 RQ factorization of A

 Vector THETA
 ( 1.039,-0.101) ( 1.181, 0.381) ( 1.224,-0.000)

 Matrix A after factorization (R is in left-hand upper triangle)
     (  0.788,  0.000) (-0.255,-0.401) (-0.277,-0.277) (-0.285,  0.559) (  0.115,
0.703)
    (  0.040,  0.522) (-2.112,  0.000) (-1.109,-0.555) (  0.128,  0.232) (  0.079,-
0.036)
    (-0.227,  0.227) (  0.045,  0.317) (-3.606,  0.000) (  0.000,-0.000) (  0.000,
0.544)
```