NAG Library Routine Document

F01RGF

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

F01RGF reduces the complex m by n ($m \leq n$) upper trapezoidal matrix A to upper triangular form by means of unitary transformations.

2 Specification

```
SUBROUTINE F01RGF (M, N, A, LDA, THETA, IFAIL)
INTEGER                M, N, LDA, IFAIL
COMPLEX (KIND=nag_wp) A(LDA,*), THETA(M)
```

3 Description

The m by $n(m \leq n)$ upper trapezoidal matrix A given by

$$A = \begin{pmatrix} U & X \end{pmatrix},$$

where U is an m by m upper triangular matrix, is factorized as

$$A = \begin{pmatrix} R & 0 \end{pmatrix} P^{\mathrm{H}},$$

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

$$P_k = \begin{pmatrix} I & 0 \\ 0 & T_k \end{pmatrix},$$

where

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

$$u_k = \begin{pmatrix} \zeta_k \\ 0 \\ z_k \\ cr \end{pmatrix},$$

γ_k is a scalar for which $\mathrm{Re}(\gamma_k) = 1.0$, ζ_k is a real scalar and z_k is an $(n - m)$ element vector. γ_k, ζ_k and z_k are chosen to annihilate the elements of the kth row of X and to make the diagonal elements of R real.

The scalar γ_k and the vector u_k are returned in the kth element of the array 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) and the elements of z_k are in A$(k, m+1), \ldots,$ A(k, n). The elements of R are returned in the upper triangular part of A.

For further information on this factorization and its use see Section 6.5 of Golub and Van Loan (1996).

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 Arguments

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: $\mathrm{A}(\mathrm{LDA}, *)$ – 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 upper trapezoidal 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 $(n - m)$ upper trapezoidal part of A 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 F01RGF is called.

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

5: $\mathrm{THETA}(\mathrm{M})$ – COMPLEX (KIND=nag_wp) array *Output*

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

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

then $\mathrm{THETA}(k) = \alpha$, otherwise $\mathrm{THETA}(k)$ contains θ_k as described in Section 3 and $\mathrm{Re}(\theta_k)$ 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 argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The computed factors R and P satisfy the relation

$$\begin{pmatrix} R & 0 \end{pmatrix} P^{\mathrm{H}} = A + E,$$

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 Parallelism and Performance

F01RGF 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 approximate number of floating-point operations is given by $8 \times m^2(n - m)$.

10 Example

This example reduces the 3 by 4 matrix

$$\begin{pmatrix} 2.4 & 0.8+0.8i & -1.4+0.6i & 3.0-1.0i \\ 0 & 1.6 & 0.8+0.3i & 0.4+0.5i \\ 0 & 0 & 1.0 & 2.0-1.0i \end{pmatrix}$$

to upper triangular form.

10.1 Program Text

```
    Program f01rgfe

!     F01RGF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f01rgf, 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,*) 'F01RGF 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 f01rgf(m,n,a,lda,theta,ifail)

      Write (nout,*)
      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','F7.4',                          &
        'Matrix A after factorization (R is in left-hand upper triangle)','N', &
        dummy,'N',dummy,80,0,ifail)

99999 Format (1X,4(' (',F7.4,',',F8.4,')',:))
    End Program f01rgfe
```

10.2 Program Data

```
F01RGF Example Program Data
  3     4                                              : m, n
( 2.4, 0.0 ) ( 0.8, 0.8 ) (-1.4, 0.6 ) ( 3.0,-1.0 )
(  0 ,  0  ) ( 1.6, 0.0 ) ( 0.8, 0.3 ) ( 0.4, 0.5 )
(  0 ,  0  ) (  0 ,  0  ) ( 1.0, 0.0 ) ( 2.0,-1.0 )   : a
```

10.3 Program Results

```
F01RGF Example Program Results


RQ factorization of A

Vector THETA
 ( 1.2924, -0.0000) ( 1.3861, -0.0000) ( 1.1867, -0.0000)

Matrix A after factorization (R is in left-hand upper triangle)
   (-3.5808, 0.0000) ( 0.2533,-0.9059) (-2.2862,-0.6532) ( 0.5120, 0.2601)
   ( 0.0000, 0.0000) (-1.7369, 0.0000) (-0.4491,-0.6940) (-0.2544,-0.1187)
   ( 0.0000, 0.0000) ( 0.0000, 0.0000) (-2.4495, 0.0000) ( 0.6880, 0.3440)
```