

# NAG Library Routine Document

## F01RKF

**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

F01RKF returns the first  $\ell$  rows of the  $n$  by  $n$  unitary matrix  $P^H$ , where  $P$  is given as the product of Householder transformation matrices.

This routine is intended for use following F01RJF.

### 2 Specification

```
SUBROUTINE F01RKF (WHERE, M, N, NROWP, A, LDA, THETA, WORK, IFAIL)
  INTEGER          M, N, NROWP, LDA, IFAIL
  COMPLEX (KIND=nag_wp) A(LDA,*), THETA(*), WORK(max(M-1,NROWP-M,1))
  CHARACTER(1)     WHERE
```

### 3 Description

$P$  is assumed to be given by

$$P = P_m P_{m-1} \cdots P_1,$$

where

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

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

$\gamma_k$  is a scalar for which  $\text{Re}(\gamma_k) = 1.0$ ,  $\zeta_k$  is a real scalar,  $w_k$  is a  $(k-1)$  element vector and  $z_k$  is an  $(n-m)$  element vector.  $w_k$  must be supplied in the  $k$ th row of  $A$  in elements  $A(k,1), \dots, A(k,k-1)$ .  $z_k$  must be supplied in the  $k$ th row of  $A$  in elements  $A(k,m+1), \dots, A(k,n)$  and  $\theta_k$ , given by

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

must be supplied either in  $A(k,k)$  or in  $\text{THETA}(k)$ , depending upon the argument  $\text{WHERE}$ .

### 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: **WHERE** – CHARACTER(1) *Input*  
*On entry:* indicates where the elements of  $\theta$  are to be found.  
**WHERE** = 'I' (In A)  
 The elements of  $\theta$  are in A.

WHERE = 'S' (Separate)

The elements of  $\theta$  are separate from A, in THETA.

*Constraint:* WHERE = 'I' or 'S'.

- 2: M – INTEGER *Input*  
*On entry:*  $m$ , the number of rows of the matrix  $A$ .  
*Constraint:*  $M \geq 0$ .
  
- 3: N – INTEGER *Input*  
*On entry:*  $n$ , the number of columns of the matrix  $A$ .  
*Constraint:*  $N \geq M$ .
  
- 4: NROWP – INTEGER *Input*  
*On entry:*  $\ell$ , the required number of rows of  $P$ .  
 If NROWP = 0, an immediate return is effected.  
*Constraint:*  $0 \leq \text{NROWP} \leq N$ .
  
- 5: A(LDA,\*) – COMPLEX (KIND=nag\_wp) array *Input/Output*  
**Note:** the second dimension of the array A must be at least  $\max(1, N)$ .  
*On entry:* the leading  $m$  by  $m$  strictly lower triangular part of the array A, and the  $m$  by  $(n - m)$  rectangular part of A with top left-hand corner at element A(1, M + 1) must contain details of the matrix  $P$ . In addition, if WHERE = 'I', the diagonal elements of A must contain the elements of  $\theta$ .  
*On exit:* the first NROWP rows of the array A are overwritten by the first NROWP rows of the  $n$  by  $n$  unitary matrix  $P^H$ .
  
- 6: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array A as declared in the (sub)program from which F01RKF is called.  
*Constraint:*  $\text{LDA} \geq \max(1, M, \text{NROWP})$ .
  
- 7: THETA(\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the dimension of the array THETA must be at least  $\max(1, M)$  if WHERE = 'S', and at least 1 otherwise.  
*On entry:* if WHERE = 'S', the array THETA must contain the elements of  $\theta$ . If  $\text{THETA}(k) = 0.0$ ,  $P_k$  is assumed to be  $I$ , if  $\text{THETA}(k) = \alpha$  and  $\text{Re}(\alpha) < 0.0$ ,  $P_k$  is assumed to be of the form
 
$$P_k = \begin{pmatrix} I & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & I \end{pmatrix},$$
 otherwise  $\text{THETA}(k)$  is assumed to contain  $\theta_k$  given by
 
$$\theta_k = (\zeta_k, \text{Im}(\gamma_k)).$$
  
 If WHERE = 'I', the array THETA is not referenced.
  
- 8: WORK(max(M – 1, NROWP – M, 1)) – COMPLEX (KIND=nag\_wp) array *Workspace*

## 9: 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:* 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, WHERE  $\neq$  'T' or 'S',  
 or  $M < 0$ ,  
 or  $N < M$ ,  
 or  $NROWP < 0$  or  $NROWP > N$ ,  
 or  $LDA < \max(1, M, NROWP)$ .

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 matrix  $P$  satisfies the relation

$$P = Q + E,$$

where  $Q$  is an exactly unitary matrix and

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

where  $\epsilon$  the *machine precision* (see X02AJF),  $c$  is a modest function of  $n$ , and  $\|\cdot\|$  denotes the spectral (two) norm. See also Section 7 in F01RJF.

## 8 Parallelism and Performance

F01RKf 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

$$\frac{8}{3}n[(3n-m)(2\ell-m)-m(\ell-m)], \quad \text{if } \ell \geq m, \text{ and}$$

$$\frac{8}{3}\ell^2(3n-\ell), \quad \text{if } \ell < m.$$

## 10 Example

This example obtains the 5 by 5 unitary matrix  $P$  following the  $RQ$  factorization of the 3 by 5 matrix  $A$  given by

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

### 10.1 Program Text

```

Program f01rkfe

!      F01RKf Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
      Use nag_library, Only: f01rjf, f01rkf, nag_wp
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
      Integer                     :: i, ifail, lda, ldph, m, n, nrowp
!      .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,,:), ph(:,,:), theta(:), work(:)
!      .. Intrinsic Procedures ..
      Intrinsic                   :: conjg
!      .. Executable Statements ..
      Write (nout,*) 'F01RKf Example Program Results'
      Write (nout,*)
!      Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      lda = m
      ldph = n
      Allocate (a(lda,n),ph(ldph,n),theta(n),work(n))
      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)

!      Copy array A into PH and form the n by n matrix conjg(P')
      ph(1:m,1:n) = a(1:m,1:n)
      nrowp = n

      ifail = 0
      Call f01rkf('Separate',m,n,nrowp,ph,ldph,theta,work,ifail)

```

```

      Write (nout,*) 'Matrix P'
      Write (nout,99999)(conjg(ph(1:nrowp,i)),i=1,n)

99999 Format (5(' (',F6.3,',',F6.3,')',:))
      End Program f01rkfe

```

## 10.2 Program Data

F01RKf Example Program Data

```

      3      5
( 0.00,-0.50) ( 0.40,-0.30) ( 0.40, 0.00) ( 0.30, 0.40) ( 0.00, 0.30) : m, n
(-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

```

## 10.3 Program Results

F01RKf Example Program Results

```

Matrix P
(-0.197, 0.197) ( 0.164,-0.492) ( 0.277,-0.277) ( 0.364, 0.321) ( 0.012, 0.514)
( 0.039, 0.276) (-0.295,-0.426) (-0.055,-0.388) (-0.475, 0.098) (-0.419,-0.299)
( 0.315,-0.158) ( 0.452,-0.320) (-0.499,-0.000) (-0.276,-0.305) (-0.034, 0.387)
( 0.197,-0.591) (-0.047,-0.331) ( 0.000, 0.000) ( 0.512,-0.047) (-0.361,-0.324)
(-0.118,-0.565) ( 0.033, 0.208) (-0.000,-0.666) (-0.229, 0.207) ( 0.290, 0.025)

```

---