

# NAG Library Routine Document

## F01KAF

**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

F01KAF computes an estimate of the absolute condition number of a matrix function  $f$  of a complex  $n$  by  $n$  matrix  $A$  in the 1-norm, where  $f$  is either the exponential, logarithm, sine, cosine, hyperbolic sine (sinh) or hyperbolic cosine (cosh). The evaluation of the matrix function,  $f(A)$ , is also returned.

### 2 Specification

```
SUBROUTINE F01KAF (FUN, N, A, LDA, CONDA, NORMA, NORMFA, IFAIL)
INTEGER          N, LDA, IFAIL
REAL (KIND=nag_wp) CONDA, NORMA, NORMFA
COMPLEX (KIND=nag_wp) A(LDA,*)
CHARACTER(*)     FUN
```

### 3 Description

The absolute condition number of  $f$  at  $A$ ,  $\text{cond}_{\text{abs}}(f, A)$  is given by the norm of the Fréchet derivative of  $f$ ,  $L(A)$ , which is defined by

$$\|L(X)\| := \max_{E \neq 0} \frac{\|L(X, E)\|}{\|E\|},$$

where  $L(X, E)$  is the Fréchet derivative in the direction  $E$ .  $L(X, E)$  is linear in  $E$  and can therefore be written as

$$\text{vec}(L(X, E)) = K(X)\text{vec}(E),$$

where the  $\text{vec}$  operator stacks the columns of a matrix into one vector, so that  $K(X)$  is  $n^2 \times n^2$ . F01KAF computes an estimate  $\gamma$  such that  $\gamma \leq \|K(X)\|_1$ , where  $\|K(X)\|_1 \in [n^{-1}\|L(X)\|_1, n\|L(X)\|_1]$ . The relative condition number can then be computed via

$$\text{cond}_{\text{rel}}(f, A) = \frac{\text{cond}_{\text{abs}}(f, A)\|A\|_1}{\|f(A)\|_1}.$$

The algorithm used to find  $\gamma$  is detailed in Section 3.4 of Higham (2008).

### 4 References

Higham N J (2008) *Functions of Matrices: Theory and Computation* SIAM, Philadelphia, PA, USA

### 5 Arguments

1: FUN – CHARACTER(\*) *Input*

*On entry:* indicates which matrix function will be used.

FUN = 'EXP'

The matrix exponential,  $e^A$ , will be used.

FUN = 'SIN'

The matrix sine,  $\sin(A)$ , will be used.

FUN = 'COS'

The matrix cosine,  $\cos(A)$ , will be used.

FUN = 'SINH'

The hyperbolic matrix sine,  $\sinh(A)$ , will be used.

FUN = 'COSH'

The hyperbolic matrix cosine,  $\cosh(A)$ , will be used.

FUN = 'LOG'

The matrix logarithm,  $\log(A)$ , will be used.

*Constraint:* FUN = 'EXP', 'SIN', 'COS', 'SINH', 'COSH' or 'LOG'.

- 2: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 3: A(LDA,\*) – COMPLEX (KIND=nag\_wp) array *Input/Output*  
**Note:** the second dimension of the array  $A$  must be at least  $N$ .  
*On entry:* the  $n$  by  $n$  matrix  $A$ .  
*On exit:* the  $n$  by  $n$  matrix,  $f(A)$ .
- 4: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array  $A$  as declared in the (sub)program from which F01KAF is called.  
*Constraint:*  $LDA \geq N$ .
- 5: CONDA – REAL (KIND=nag\_wp) *Output*  
*On exit:* an estimate of the absolute condition number of  $f$  at  $A$ .
- 6: NORMA – REAL (KIND=nag\_wp) *Output*  
*On exit:* the 1-norm of  $A$ .
- 7: NORMFA – REAL (KIND=nag\_wp) *Output*  
*On exit:* the 1-norm of  $f(A)$ .
- 8: 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:

$IFAIL = 1$

An internal error occurred when estimating the norm of the Fréchet derivative of  $f$  at  $A$ . Please contact NAG.

$IFAIL = 2$

An internal error occurred when evaluating the matrix function  $f(A)$ . You can investigate further by calling  $F01FCF$ ,  $F01FJF$  or  $F01FKF$  with the matrix  $A$ .

$IFAIL = -1$

On entry,  $FUN = \langle value \rangle$  was an illegal value.

$IFAIL = -2$

On entry,  $N < 0$ .  
Input argument number  $\langle value \rangle$  is invalid.

$IFAIL = -4$

On entry, argument  $LDA$  is invalid.  
Constraint:  $LDA \geq N$ .

$IFAIL = -99$

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

$F01KAF$  uses the norm estimation routine  $F04ZDF$  to estimate a quantity  $\gamma$ , where  $\gamma \leq \|K(X)\|_1$  and  $\|K(X)\|_1 \in [n^{-1}\|L(X)\|_1, n\|L(X)\|_1]$ . For further details on the accuracy of norm estimation, see the documentation for  $F04ZDF$ .

## 8 Parallelism and Performance

$F01KAF$  is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library. In these implementations, this routine may make calls to the user-supplied functions from within an OpenMP parallel region. Thus OpenMP directives within the user functions can only be used if you are compiling the user-supplied function and linking the executable in accordance with the instructions in the Users' Note for your implementation.

F01KAF 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

Approximately  $6n^2$  of complex allocatable memory is required by the routine, in addition to the memory used by the underlying matrix function routines F01FCF, F01FJF or F01FKF.

F01KAF returns the matrix function  $f(A)$ . This is computed using F01FCF if FUN = 'EXP', F01FJF if FUN = 'LOG' and F01FKF otherwise. If only  $f(A)$  is required, without an estimate of the condition number, then it is far more efficient to use F01FCF, F01FJF or F01FKF directly.

F01JAF can be used to find the condition number of the exponential, logarithm, sine, cosine, sinh or cosh at a real matrix.

## 10 Example

This example estimates the absolute and relative condition numbers of the matrix sinh function for

$$A = \begin{pmatrix} 0.0 + 1.0i & -1.0 + 0.0i & 1.0 + 0.0i & 2.0 + 0.0i \\ 2.0 + 1.0i & 0.0 - 1.0i & 0.0 + 0.0i & 1.0 + 0.0i \\ 0.0 + 1.0i & 0.0 + 0.0i & 1.0 + 1.0i & 0.0 + 2.0i \\ 1.0 + 0.0i & 2.0 + 0.0i & -2.0 + 3.0i & 0.0 + 1.0i \end{pmatrix}.$$

### 10.1 Program Text

```

Program f01kafe
!
!   F01KAF Example Program Text
!
!   Mark 26 Release. NAG Copyright 2016.
!
!   .. Use Statements ..
!   Use nag_library, Only: f01kaf, nag_wp, x02ajf, x04daf
!   .. Implicit None Statement ..
!   Implicit None
!   .. Parameters ..
!   Integer, Parameter          :: nin = 5, nout = 6
!   .. Local Scalars ..
!   Real (Kind=nag_wp)         :: conda, cond_rel, eps, norma, normfa
!   Integer                     :: i, ifail, lda, n
!   Character (4)               :: fun
!   .. Local Arrays ..
!   Complex (Kind=nag_wp), Allocatable :: a(:,,:)
!   .. Executable Statements ..
!   Write (nout,*) 'F01KAF Example Program Results'
!   Write (nout,*)
!   Flush (nout)
!
!   Skip heading in data file
!   Read (nin,*)
!   Read (nin,*) n, fun
!
!   lda = n
!   Allocate (a(lda,n))
!
!   Read A from data file
!   Read (nin,*)(a(i,1:n),i=1,n)
!
!   Display A
!   ifail = 0
!   Call x04daf('G','N',n,n,a,lda,'A',ifail)
!
!   Find absolute condition number estimate
!   ifail = 0

```

```

Call f01kaf(fun,n,a,lda,conda,norma,normfa,ifail)

If (ifail==0) Then
!   Print solution
   Write (nout,*)
   Write (nout,*) 'F(A) = ', fun, '(A)'
   Write (nout,99999) 'Estimated absolute condition number is: ', conda

!   Find relative condition number estimate
   eps = x02ajf()
   If (normfa>eps) Then
     cond_rel = conda*norma/normfa
     Write (nout,99999) 'Estimated relative condition number is: ',      &
       cond_rel
   Else
     Write (nout,99998) 'The estimated norm of f(A) is effectively zero', &
       'and so the relative condition number is undefined.'
   End If
End If

99999 Format (1X,A,F6.2)
99998 Format (/ ,1X,A,/,1X,A)

End Program f01kafe

```

## 10.2 Program Data

F01KAF Example Program Data

```

4           SINH                               :Values of N and FUN

(0.0, 1.0)  (-1.0, 0.0)   ( 1.0, 0.0)   (2.0, 0.0)
(2.0, 1.0)  ( 0.0,-1.0)   ( 0.0, 0.0)   (1.0, 0.0)
(0.0, 1.0)  ( 0.0, 0.0)   ( 1.0, 1.0)   (0.0, 2.0)
(1.0, 0.0)  ( 2.0, 0.0)   (-2.0, 3.0)   (0.0, 1.0)      :End of matrix A

```

## 10.3 Program Results

F01KAF Example Program Results

```

A
      1      2      3      4
1     0.0000  -1.0000  1.0000  2.0000
     1.0000   0.0000  0.0000  0.0000

2     2.0000   0.0000  0.0000  1.0000
     1.0000  -1.0000  0.0000  0.0000

3     0.0000   0.0000  1.0000  0.0000
     1.0000   0.0000  1.0000  2.0000

4     1.0000   2.0000  -2.0000  0.0000
     0.0000   0.0000   3.0000  1.0000

```

```

F(A) = SINH(A)
Estimated absolute condition number is: 7.33
Estimated relative condition number is: 4.94

```

---