NAG Library Routine Document

F01JAF

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

F01JAF computes an estimate of the absolute condition number of a matrix function f at a real 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 F01JAF (FUN, N, A, LDA, CONDA, NORMA, NORMFA, IFAIL)
INTEGER             N, LDA, IFAIL
REAL (KIND=nag_wp) A(LDA,*), CONDA, NORMA, NORMFA
CHARACTER(*)        FUN
```

3 Description

The absolute condition number of f at A, $\mathrm{cond}_{\mathrm{abs}}(f, A)$ 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

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

where the vec operator stacks the columns of a matrix into one vector, so that $K(X)$ is $n^2 \times n^2$. F01JAF computes an estimate γ such that $\gamma \leq \|K(X)\|_1$, where $\|K(X)\|_1 \in \left[n^{-1}\|L(X)\|_1, n\|L(X)\|_1\right]$. The relative condition number can then be computed via

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

The algorithm used to find γ 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, $*$) – REAL (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 F01JAF 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 evaluating the matrix function $f(A)$. Please contact NAG.

IFAIL $= 2$

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

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

F01JAF uses the norm estimation routine F04YDF to estimate a quantity γ, where $\gamma \leq \|K(X)\|_1$ and $\|K(X)\|_1 \in \left[n^{-1}\|L(X)\|_1, n\|L(X)\|_1\right]$. For further details on the accuracy of norm estimation, see the documentation for F04YDF.

8 Parallelism and Performance

F01JAF 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.

F01JAF 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 matrix function is computed using one of three underlying matrix function routines:

if FUN = 'EXP', F01ECF is used;

if FUN = 'LOG', F01EJF is used;

else, F01EKF is used.

Approximately $6n^2$ of real allocatable memory is required by the routine, in addition to the memory used by these underlying matrix function routines.

If only $f(A)$ is required, without an estimate of the condition number, then it is far more efficient to use the appropriate matrix function routine listed above.

F01KAF can be used to find the condition number of the exponential, logarithm, sine, cosine, sinh or cosh matrix functions at a complex matrix.

10 Example

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

$$A = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & -1 & 0 & 2 \\ 1 & 0 & 3 & 1 \\ 1 & 2 & 0 & 3 \end{pmatrix}.$$

10.1 Program Text

```
    Program f01jafe

!     F01JAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f01jaf, nag_wp, x02ajf, x04caf
!     .. 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 ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'F01JAF 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 x04caf('G','N',n,n,a,lda,'A',ifail)
```

```
!     Find absolute condition number estimate
      ifail = 0
      Call f01jaf(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,F7.2)
99998 Format (/,1X,A,/,1X,A)

    End Program f01jafe
```

10.2 Program Data

```
F01JAF Example Program Data

4       SINH              :Value of N and FUN

2.0    1.0    3.0    1.0
3.0   -1.0    0.0    2.0
1.0    0.0    3.0    1.0
1.0    2.0    0.0    3.0  :End of matrix A
```

10.3 Program Results

```
 F01JAF Example Program Results

 A
             1          2          3          4
 1      2.0000     1.0000     3.0000     1.0000
 2      3.0000    -1.0000     0.0000     2.0000
 3      1.0000     0.0000     3.0000     1.0000
 4      1.0000     2.0000     0.0000     3.0000

 F(A) = SINH(A)
 Estimated absolute condition number is:  204.45
 Estimated relative condition number is:    7.90
```