NAG Library Routine Document

F01JKF

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

F01JKF computes the Fréchet derivative $L(A,E)$ of the matrix logarithm of the real n by n matrix A applied to the real n by n matrix E. The principal matrix logarithm $\log(A)$ is also returned.

2 Specification

```
SUBROUTINE F01JKF (N, A, LDA, E, LDE, IFAIL)
INTEGER            N, LDA, LDE, IFAIL
REAL (KIND=nag_wp) A(LDA,*), E(LDE,*)
```

3 Description

For a matrix with no eigenvalues on the closed negative real line, the principal matrix logarithm $\log(A)$ is the unique logarithm whose spectrum lies in the strip $\{z : -\pi < \mathrm{Im}(z) < \pi\}$.

The Fréchet derivative of the matrix logarithm of A is the unique linear mapping $E \mapsto L(A,E)$ such that for any matrix E

$$\log(A+E) - \log(A) - L(A,E) = o(\|E\|).$$

The derivative describes the first order effect of perturbations in A on the logarithm $\log(A)$.

F01JKF uses the algorithm of Al–Mohy *et al.* (2012) to compute $\log(A)$ and $L(A,E)$. The principal matrix logarithm $\log(A)$ is computed using a Schur decomposition, a Padé approximant and the inverse scaling and squaring method. The Padé approximant is then differentiated in order to obtain the Fréchet derivative $L(A,E)$.

4 References

Al–Mohy A H and Higham N J (2011) Improved inverse scaling and squaring algorithms for the matrix logarithm *SIAM J. Sci. Comput.* **34(4)** C152–C169

Al–Mohy A H, Higham N J and Relton S D (2012) Computing the Fréchet derivative of the matrix logarithm and estimating the condition number *SIAM J. Sci. Comput.* **35(4)** C394–C410

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

5 Arguments

1: N – INTEGER *Input*

On entry: n, the order of the matrix A.

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

2: 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 principal matrix logarithm, $\log(A)$.

3: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F01JKF is called.

Constraint: LDA $\geq$ N.

4: E(LDE, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array E must be at least N.

On entry: the n by n matrix E

On exit: the Fréchet derivative $L(A, E)$

5: LDE – INTEGER *Input*

On entry: the first dimension of the array E as declared in the (sub)program from which F01JKF is called.

Constraint: LDE $\geq$ N.

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: 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 $= 1$

A is singular so the logarithm cannot be computed.

IFAIL $= 2$

A has eigenvalues on the negative real line. The principal logarithm is not defined in this case; F01KKF can be used to return a complex, non-principal log.

IFAIL $= 3$

$\log(A)$ has been computed using an IEEE double precision Padé approximant, although the arithmetic precision is higher than IEEE double precision.

IFAIL $= 4$

An unexpected internal error occurred. This failure should not occur and suggests that the routine has been called incorrectly.

IFAIL $= -1$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 0.

IFAIL $= -3$

On entry, LDA $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LDA $\geq$ N.

IFAIL $= -5$

On entry, LDE $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LDE $\geq$ 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

For a normal matrix A (for which $A^{\mathrm{T}}A = AA^{\mathrm{T}}$), the Schur decomposition is diagonal and the computation of the matrix logarithm reduces to evaluating the logarithm of the eigenvalues of A and then constructing $\log(A)$ using the Schur vectors. This should give a very accurate result. In general, however, no error bounds are available for the algorithm. The sensitivity of the computation of $\log(A)$ and $L(A, E)$ is worst when A has an eigenvalue of very small modulus or has a complex conjugate pair of eigenvalues lying close to the negative real axis. See Al–Mohy and Higham (2011), Al–Mohy *et al.* (2012) and Section 11.2 of Higham (2008) for details and further discussion.

8 Parallelism and Performance

F01JKF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F01JKF 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 cost of the algorithm is $O(n^3)$ floating-point operations. The real allocatable memory required is approximately $5n^2$; see Al–Mohy *et al.* (2012) for further details.

If the matrix logarithm alone is required, without the Fréchet derivative, then F01EJF should be used. If the condition number of the matrix logarithm is required then F01JJF should be used. If A has negative real eigenvalues then F01KKF can be used to return a complex, non-principal matrix logarithm and its Fréchet derivative $L(A, E)$.

10 Example

This example finds the principal matrix logarithm $\log(A)$ and the Fréchet derivative $L(A,E)$, where

$$A = \begin{pmatrix} 4 & 2 & 0 & 2 \\ 3 & 3 & 1 & 1 \\ 3 & 2 & 1 & 0 \\ 3 & 3 & 1 & 2 \end{pmatrix} \quad \text{and} \quad E = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 0 & 0 & 3 & 1 \\ 1 & 2 & 1 & 2 \\ 1 & 3 & 1 & 1 \end{pmatrix}.$$

10.1 Program Text

```
    Program f01jkfe

!     F01JKF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f01jkf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lda, lde, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), e(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'F01JKF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      lde = n
      Allocate (a(lda,n))
      Allocate (e(lde,n))
!     Read A from data file
      Read (nin,*)(a(i,1:n),i=1,n)
!     Read E from data file
      Read (nin,*)(e(i,1:n),i=1,n)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0

!     Find log( A ) and L_log(A,E)
      Call f01jkf(n,a,lda,e,lde,ifail)

!     Print solution
      Call x04caf('General',' ',n,n,a,lda,'Log(A)',ifail)
      Write (nout,*)
      Call x04caf('General',' ',n,n,e,lde,'L_log(A,E)',ifail)

    End Program f01jkfe
```

10.2 Program Data

```
F01JKF Example Program Data

  4                        :Value N

  4.0   2.0   0.0   2.0
  3.0   3.0   1.0   1.0
  3.0   2.0   1.0   0.0
  3.0   3.0   1.0   2.0  :End of matrix A

  1.0   2.0   2.0   2.0
  0.0   0.0   3.0   1.0
  1.0   2.0   1.0   2.0
  1.0   3.0   1.0   1.0  :End of matrix E
```

10.3 Program Results

```
F01JKF Example Program Results

Log(A)
             1          2          3          4
1       1.1165     0.5296    -0.4079     0.6962
2       0.6996     0.2025     0.8192     0.4745
3       1.3114     1.5867    -0.1433    -1.1720
4       0.5272     1.2856     0.4055     0.2106

L_log(A,E)
             1          2          3          4
1      -0.1211     0.1974     0.1463     0.8268
2      -1.2615    -4.1260     3.4035     2.4651
3       1.2387     5.7968    -3.6489    -2.7203
4       0.6231     3.7059    -1.9334    -1.8540
```