NAG Library Routine Document

F01JGF

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

F01JGF computes an estimate of the relative condition number $\kappa_{\exp}(A)$ of the exponential of a real n by n matrix A, in the 1-norm. The matrix exponential e^A is also returned.

2 Specification

```
SUBROUTINE F01JGF (N, A, LDA, CONDEA, IFAIL)
INTEGER            N, LDA, IFAIL
REAL (KIND=nag_wp) A(LDA,*), CONDEA
```

3 Description

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

$$e^{A+E} - e^A - L(A, E) = o(\|E\|).$$

The derivative describes the first-order effect of perturbations in A on the exponential e^A.

The relative condition number of the matrix exponential can be defined by

$$\kappa_{\exp}(A) = \frac{\|L(A)\|\|A\|}{\|\exp(A)\|},$$

where $\|L(A)\|$ is the norm of the Fréchet derivative of the matrix exponential at A.

To obtain the estimate of $\kappa_{\exp}(A)$, F01JGF first estimates $\|L(A)\|$ by computing an estimate γ of a quantity $K \in \left[n^{-1}\|L(A)\|_1, n\|L(A)\|_1\right]$, such that $\gamma \leq K$.

The algorithms used to compute $\kappa_{\exp}(A)$ are detailed in the Al–Mohy and Higham (2009a) and Al–Mohy and Higham (2009b).

The matrix exponential e^A is computed using a Padé approximant and the scaling and squaring method. The Padé approximant is differentiated to obtain the Fréchet derivatives $L(A, E)$ which are used to estimate the condition number.

4 References

Al–Mohy A H and Higham N J (2009a) A new scaling and squaring algorithm for the matrix exponential *SIAM J. Matrix Anal.* **31(3)** 970–989

Al–Mohy A H and Higham N J (2009b) Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation *SIAM J. Matrix Anal. Appl.* **30(4)** 1639–1657

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

Moler C B and Van Loan C F (2003) Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later *SIAM Rev.* **45** 3–49

5 Arguments

1: N – INTEGER *Input*

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

Constraint: N ≥ 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 matrix exponential e^A.

3: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F01JGF is called.

Constraint: LDA $\geq$ N.

4: CONDEA – REAL (KIND=nag_wp) *Output*

On exit: an estimate of the relative condition number of the matrix exponential $\kappa_{\exp}(A)$.

5: 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 $= 1$

The linear equations to be solved for the Padé approximant are singular; it is likely that this routine has been called incorrectly.

IFAIL $= 2$

e^A has been computed using an IEEE double precision Padé approximant, although the arithmetic precision is higher than IEEE double precision.

IFAIL $= 3$

An unexpected internal error has occurred. Please contact NAG.

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 $= -99$

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

F01JGF uses the norm estimation routine F04YDF to produce an estimate γ of a quantity $K \in \left[n^{-1}\|L(A)\|_1, n\|L(A)\|_1\right]$, such that $\gamma \leq K$. For further details on the accuracy of norm estimation, see the documentation for F04YDF.

For a normal matrix A (for which $A^{\mathrm{T}}A = AA^{\mathrm{T}}$) the computed matrix, e^A, is guaranteed to be close to the exact matrix, that is, the method is forward stable. No such guarantee can be given for non-normal matrices. See Section 10.3 of Higham (2008) for details and further discussion.

For further discussion of the condition of the matrix exponential see Section 10.2 of Higham (2008).

8 Parallelism and Performance

F01JGF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F01JGF 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

F01JAF uses a similar algorithm to F01JGF to compute an estimate of the *absolute* condition number (which is related to the relative condition number by a factor of $\|A\|/\|\exp(A)\|$). However, the required Fréchet derivatives are computed in a more efficient and stable manner by F01JGF and so its use is recommended over F01JAF.

The cost of the algorithm is $O\left(n^3\right)$ and the real allocatable memory required is approximately $15n^2$; see Al–Mohy and Higham (2009a) and Al–Mohy and Higham (2009b) for further details.

If the matrix exponential alone is required, without an estimate of the condition number, then F01ECF should be used. If the Fréchet derivative of the matrix exponential is required then F01JHF should be used.

As well as the excellent book Higham (2008), the classic reference for the computation of the matrix exponential is Moler and Van Loan (2003).

10 Example

This example estimates the relative condition number of the matrix exponential e^A, where

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

10.1 Program Text

```
    Program f01jgfe

!     F01JGF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f01jgf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: condea
      Integer                          :: i, ifail, lda, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'F01JGF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      Allocate (a(lda,n))
!     Read A from data file
      Read (nin,*)(a(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 exp( A ) and the condition estimate
      Call f01jgf(n,a,lda,condea,ifail)

!     Print solution
      Call x04caf('General',' ',n,n,a,lda,'Exp(A)',ifail)

      Write (nout,*)
      Write (nout,99999) 'Estimated condition number is: ', condea
99999 Format (1X,A,F6.2)
    End Program f01jgfe
```

10.2 Program Data

```
F01JGF Example Program Data

  4                       : N

  2.0   2.0   1.0   2.0
  3.0   1.0   4.0   0.0
  2.0   3.0   1.0   2.0
  0.0   1.0   3.0   3.0   : A
```

10.3 Program Results

```
 F01JGF Example Program Results

 Exp(A)
             1          2          3          4
 1    404.4441   412.6036   496.7221   398.3043
 2    474.4388   482.8457   579.1310   460.6474
 3    466.9764   477.2769   574.3994   458.3804
 4    407.7005   420.8935   510.1939   410.4808

 Estimated condition number is:   9.40
```