NAG Toolbox

nag_matop_real_gen_matrix_frcht_log (f01jk)

1 Purpose

nag_matop_real_gen_matrix_frcht_log (f01jk) 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 Syntax

```
[a, e, ifail] = nag_matop_real_gen_matrix_frcht_log(a, e, 'n', n)
[a, e, ifail] = f01jk(a, e, 'n', n)
```

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 $\log(A)$.

nag_matop_real_gen_matrix_frcht_log (f01jk) 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 *MIMS EPrint* **2012.72**

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

5 Parameters

5.1 Compulsory Input Parameters

1: **a**$(lda,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** must be at least **n**.

The second dimension of the array **a** must be at least **n**.

The n by n matrix A.

2: **e**$(lde,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **e** must be at least **n**.

The second dimension of the array **e** must be at least **n**.

The n by n matrix E

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the arrays **a**, **e** and the second dimension of the arrays **a**, **e**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix A.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **a**$(lda, :)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** will be **n**.

The second dimension of the array **a** will be **n**.

The n by n principal matrix logarithm, $\log(A)$.

2: **e**$(lde, :)$ – REAL (KIND=nag_wp) array

The first dimension of the array **e** will be **n**.

The second dimension of the array **e** will be **n**.

The Fréchet derivative $L(A, E)$

3: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

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; nag_matop_complex_gen_matrix_frcht_log (f01kk) 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 function has been called incorrectly.

ifail $= -1$

Constraint: **n** ≥ 0.

ifail $= -3$

Constraint: $lda \geq$ **n**.

ifail $= -5$

Constraint: $lde \geq$ **n**.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

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 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 nag_matop_real_gen_matrix_log (f01ej) should be used. If the condition number of the matrix logarithm is required then nag_matop_real_gen_matrix_cond_log (f01jj) should be used. If A has negative real eigenvalues then nag_matop_complex_gen_matrix_frcht_log (f01kk) can be used to return a complex, non-principal matrix logarithm and its Fréchet derivative $L(A,E)$.

9 Example

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

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

9.1 Program Text

```
function f01jk_example

fprintf('f01jk example results\n\n');

% Logarithm of matrix A and Frechet derivative of log(A)E.

a = [ 4 2 0 2;
      3 3 1 1;
      3 2 1 0;
      3 3 1 2];

e = [ 1 2 2 2;
      0 0 3 1;
      1 2 1 2;
      1 3 1 1];
```

```
[loga, lae, ifail] = f01jk(a,e);

[ifail] = x04ca('General', ' ', loga, 'log(A):');
disp(' ');
[ifail] = x04ca('General', ' ', lae, 'L_log(A,E):');
```

9.2 Program Results

```
     f01jk example 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
```