

NAG Toolbox

nag_matop_real_gen_matrix_cond_log (f01jj)

1 Purpose

nag_matop_real_gen_matrix_cond_log (f01jj) computes an estimate of the relative condition number $\kappa_{\log}(A)$ of the logarithm of a real n by n matrix A , in the 1-norm. The principal matrix logarithm $\log(A)$ is also returned.

2 Syntax

```
[a, condla, ifail] = nag_matop_real_gen_matrix_cond_log(a, 'n', n)
[a, condla, ifail] = f01jj(a, '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 $\{z : -\pi < \text{Im}(z) < \pi\}$.

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 $\log(A)$.

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

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

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

To obtain the estimate of $\kappa_{\log}(A)$, nag_matop_real_gen_matrix_cond_log (f01jj) first estimates $\|L(A)\|$ by computing an estimate γ of a quantity $K \in [n^{-1}\|L(A)\|_1, n\|L(A)\|_1]$, such that $\gamma \leq K$.

The algorithms used to compute $\kappa_{\log}(A)$ and $\log(A)$ are based on a Schur decomposition, the inverse scaling and squaring method and Padé approximants. Further details can be found in Al-Mohy and Higham (2011) and Al-Mohy *et al.* (2012).

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 .

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n , the order of the matrix A .

Constraint: $n \geq 0$.

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: **condla** – REAL (KIND=nag_wp)

With **ifail** = 0 or 3, an estimate of the relative condition number of the matrix logarithm, $\kappa_{\log}(A)$. Alternatively, if **ifail** = 4, contains the absolute condition number of the matrix logarithm.

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_cond_log (f01kj)` 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

The relative condition number is infinite. The absolute condition number was returned instead.

ifail = 5

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

ifail = -1

Constraint: $n \geq 0$.

ifail = -3

Constraint: $lda \geq n$.

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

`nag_matop_real_gen_matrix_cond_log` (f01jj) uses the norm estimation function `nag_linsys_real_gen_norm_rcomm` (f04yd) to produce an estimate γ of a quantity $K \in [n^{-1}\|L(A)\|_1, n\|L(A)\|_1]$, such that $\gamma \leq K$. For further details on the accuracy of norm estimation, see the documentation for `nag_linsys_real_gen_norm_rcomm` (f04yd).

For a normal matrix A (for which $A^T A = A A^T$), 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 $\log(A)$ 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) and Section 11.2 of Higham (2008) for details and further discussion.

8 Further Comments

`nag_matop_real_gen_matrix_cond_std` (f01ja) uses a similar algorithm to `nag_matop_real_gen_matrix_cond_log` (f01jj) to compute an estimate of the *absolute* condition number (which is related to the relative condition number by a factor of $\|A\|/\|\log(A)\|$). However, the required Fréchet derivatives are computed in a more efficient and stable manner by `nag_matop_real_gen_matrix_cond_log` (f01jj) and so its use is recommended over `nag_matop_real_gen_matrix_cond_std` (f01ja).

The amount of real allocatable memory required by the algorithm is typically of the order $10n^2$.

The cost of the algorithm is $O(n^3)$ floating-point operations; see Al-Mohy *et al.* (2012).

If the matrix logarithm alone is required, without an estimate of the condition number, then `nag_matop_real_gen_matrix_log` (f01ej) should be used. If the Fréchet derivative of the matrix logarithm is required then `nag_matop_real_gen_matrix_frcht_log` (f01jk) should be used. If A has negative real eigenvalues then `nag_matop_complex_gen_matrix_cond_log` (f01kj) can be used to return a complex, non-principal matrix logarithm and its condition number.

9 Example

This example estimates the relative condition number of the matrix logarithm $\log(A)$, where

$$A = \begin{pmatrix} 4 & -1 & 0 & 1 \\ 2 & 5 & -2 & 2 \\ 1 & 1 & 3 & -1 \\ 2 & 0 & 2 & 8 \end{pmatrix}.$$

9.1 Program Text

```
function f01jj_example
fprintf('f01jj example results\n\n');
% Logarithm and conditioning of matrix A
a = [4, -1, 0, 1;
     2, 5, -2, 2;
     1, 1, 3, -1;
     2, 0, 2, 8];
% Compute log(a)
[loga, condla, ifail] = f01jj(a);
% Display results
disp('Log(A):');
disp(loga);
fprintf('Estimated condition number is: %6.2f\n', condla);
```

9.2 Program Results

```
f01jj example results
Log(A):
 1.4081   -0.2051   -0.1071    0.1904
 0.4396    1.7096   -0.5147    0.2226
 0.2560    0.2613    1.2485   -0.2413
 0.3030   -0.0107    0.3834    2.0891
Estimated condition number is:  5.50
```
