NAG Toolbox

nag_matop_complex_gen_matrix_log (f01fj)

1 Purpose

nag_matop_complex_gen_matrix_log (f01fj) computes the principal matrix logarithm, $\log(A)$, of a complex n by n matrix A, with no eigenvalues on the closed negative real line.

2 Syntax

```
[a, ifail] = nag_matop_complex_gen_matrix_log(a, 'n', n)
[a, ifail] = f01fj(a, 'n', n)
```

3 Description

Any nonsingular matrix A has infinitely many logarithms. For a matrix with no eigenvalues on the closed negative real line, the principal logarithm is the unique logarithm whose spectrum lies in the strip $\{z : -\pi < \mathrm{Im}(z) < \pi\}$. If A is nonsingular but has eigenvalues on the negative real line, the principal logarithm is not defined, but nag_matop_complex_gen_matrix_log (f01fj) will return a non-principal logarithm.

$\log(A)$ is computed using the inverse scaling and squaring algorithm for the matrix logarithm described in Al–Mohy and Higham (2011).

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

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

5 Parameters

5.1 Compulsory Input Parameters

1: **a**($lda, :$) – COMPLEX (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**.

n, the order of the matrix A.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **a**($lda, :$) – COMPLEX (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, $\log(A)$, unless **ifail** $= 2$, in which case a non-principal logarithm is returned.

2: **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 $= 2$ (*warning*)

A was found to have eigenvalues on the negative real line. The principal logarithm is not defined in this case, so a non-principal logarithm was returned.

ifail $= 3$ (*warning*)

$\log(A)$ 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 has occurred. Please contact NAG.

ifail $= -1$

Constraint: $\mathbf{n} \geq 0$.

ifail $= -3$

Constraint: $lda \geq \mathbf{n}$.

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 $A^{\mathrm{H}}A = AA^{\mathrm{H}}$), the Schur decomposition is diagonal and the algorithm 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. See Al–Mohy and Higham (2011) and Section 9.4 of Higham (2008) for details and further discussion.

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.

If estimates of the condition number of the matrix logarithm are required then nag_matop_complex_gen_matrix_cond_log (f01kj) should be used.

8 Further Comments

The cost of the algorithm is $O(n^3)$ floating-point operations (see Al–Mohy and Higham (2011)). The complex allocatable memory required is approximately $3 \times n^2$.

If the Fréchet derivative of the matrix logarithm is required then nag_matop_complex_gen_matrix_frcht_log (f01kk) should be used.

nag_matop_real_gen_matrix_log (f01ej) can be used to find the principal logarithm of a real matrix.

9 Example

This example finds the principal matrix logarithm of the matrix

$$A = \begin{pmatrix} 1.0 + 2.0i & 0.0 + 1.0i & 1.0 + 0.0i & 3.0 + 2.0i \\ 0.0 + 3.0i & -2.0 + 0.0i & 0.0 + 0.0i & 1.0 + 0.0i \\ 1.0 + 0.0i & -2.0 + 0.0i & 3.0 + 2.0i & 0.0 + 3.0i \\ 2.0 + 0.0i & 0.0 + 1.0i & 0.0 + 1.0i & 2.0 + 3.0i \end{pmatrix}.$$

9.1 Program Text

```
function f01fj_example

fprintf('f01fj example results\n\n');

a =  [1.0+2.0i,  0.0+1.0i, 1.0+0.0i, 3.0+2.0i;
      0.0+3.0i, -2.0+0.0i, 0.0+0.0i, 1.0+0.0i;
      1.0+0.0i, -2.0+0.0i, 3.0+2.0i, 0.0+3.0i;
      2.0+0.0i,  0.0+1.0i, 0.0+1.0i, 2.0+3.0i];

% Compute log(a)
[loga, ifail] = f01fj(a);

disp('f(A) = log(A)');
disp(loga);
```

9.2 Program Results

```
f01fj example results

f(A) = log(A)
   1.0390 + 1.1672i   0.2859 + 0.3998i   0.0516 - 0.2562i   0.7586 - 0.4678i
  -2.7481 + 2.6187i   1.1898 - 2.2287i   0.1369 - 0.9128i   2.1771 - 1.0118i
  -0.8514 + 0.3927i  -0.2517 - 0.4791i   1.3839 + 0.2129i   1.1920 + 0.4240i
   1.1970 - 0.1242i  -0.6813 + 0.3969i   0.0051 + 0.3511i   0.7867 + 0.7502i
```