

## NAG Toolbox

### nag\_matop\_real\_gen\_matrix\_cond\_exp (f01jg)

#### 1 Purpose

nag\_matop\_real\_gen\_matrix\_cond\_exp (f01jg) computes an estimate of the relative condition number  $\kappa_{\text{exp}}(A)$  of the exponential of a real  $n$  by  $n$  matrix  $A$ , in the 1-norm. The matrix exponential  $e^A$  is also returned.

#### 2 Syntax

```
[a, condea, ifail] = nag_matop_real_gen_matrix_cond_exp(a, 'n', n)
[a, condea, ifail] = f01jg(a, 'n', n)
```

#### 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

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

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

To obtain the estimate of  $\kappa_{\text{exp}}(A)$ , nag\_matop\_real\_gen\_matrix\_cond\_exp (f01jg) first estimates  $\|L(A)\|$  by computing an estimate  $\gamma$  of a quantity  $K \in [n^{-1}\|L(A)\|_1, n\|L(A)\|_1]$ , such that  $\gamma \leq K$ .

The algorithms used to compute  $\kappa_{\text{exp}}(A)$  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 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**.

$n$ , 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  $n$  matrix exponential  $e^A$ .

2: **condea** – REAL (KIND=nag\_wp)

An estimate of the relative condition number of the matrix exponential  $\kappa_{\text{exp}}(A)$ .

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

The linear equations to be solved for the Padé approximant are singular; it is likely that this function 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

*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_exp` (f01jg) uses the norm estimation function `nag_linsys_real_gen_norm_rcomm` (f04yd) to produce an estimate  $\gamma$  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 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 Further Comments

`nag_matop_real_gen_matrix_cond_std` (f01ja) uses a similar algorithm to `nag_matop_real_gen_matrix_cond_exp` (f01jg) 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 `nag_matop_real_gen_matrix_cond_exp` (f01jg) and so its use is recommended over `nag_matop_real_gen_matrix_cond_std` (f01ja).

The cost of the algorithm is  $O(n^3)$  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 `nag_matop_real_gen_matrix_exp` (f01ec) should be used. If the Fréchet derivative of the matrix exponential is required then `nag_matop_real_gen_matrix_frcht_exp` (f01jh) 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).

## 9 Example

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

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

### 9.1 Program Text

```
function f01jg_example
fprintf('f01jg example results\n\n');

% Exponential and conditioning of matrix A
a = [2, 2, 1, 2;
     3, 1, 4, 0;
     2, 3, 1, 2;
     0, 1, 3, 3];
```

```
% Compute exp(a)
[expa, condea, ifail] = f01jg(a);

% Display results
disp('exp(A):');
disp(expa);

fprintf('Estimated condition number is: %6.2f\n', condea);
```

## **9.2 Program Results**

f01jg example results

```
exp(A):
 404.4441  412.6036  496.7221  398.3043
 474.4388  482.8457  579.1310  460.6474
 466.9764  477.2769  574.3994  458.3804
 407.7005  420.8935  510.1939  410.4808
```

```
Estimated condition number is: 9.40
```

---