NAG Toolbox

nag_matop_complex_gen_matrix_cond_std (f01ka)

1 Purpose

nag_matop_complex_gen_matrix_cond_std (f01ka) computes an estimate of the absolute condition number of a matrix function f of a complex n by n matrix A in the 1-norm, where f is either the exponential, logarithm, sine, cosine, hyperbolic sine (sinh) or hyperbolic cosine (cosh). The evaluation of the matrix function, $f(A)$, is also returned.

2 Syntax

```
[a, conda, norma, normfa, ifail] = nag_matop_complex_gen_matrix_cond_std(fun, a,
'n', n)
```

```
[a, conda, norma, normfa, ifail] = f01ka(fun, a, 'n', n)
```

3 Description

The absolute condition number of f at A, $\mathrm{cond}_{\mathrm{abs}}(f,A)$ is given by the norm of the Fréchet derivative of f, $L(A)$, which is defined by

$$\|L(X)\| := \max_{E\neq 0}\frac{\|L(X,E)\|}{\|E\|},$$

where $L(X,E)$ is the Fréchet derivative in the direction E. $L(X,E)$ is linear in E and can therefore be written as

$$\mathrm{vec}(L(X,E)) = K(X)\mathrm{vec}(E),$$

where the vec operator stacks the columns of a matrix into one vector, so that $K(X)$ is $n^2 \times n^2$. nag_matop_complex_gen_matrix_cond_std (f01ka) computes an estimate γ such that $\gamma \leq \|K(X)\|_1$, where $\|K(X)\|_1 \in \left[n^{-1}\|L(X)\|_1, n\|L(X)\|_1\right]$. The relative condition number can then be computed via

$$\mathrm{cond}_{\mathrm{rel}}(f,A) = \frac{\mathrm{cond}_{\mathrm{abs}}(f,A)\|A\|_1}{\|f(A)\|_1}.$$

The algorithm used to find γ is detailed in Section 3.4 of Higham (2008).

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **fun** – CHARACTER(*)

Indicates which matrix function will be used.

fun = 'exp'
: The matrix exponential, e^A, will be used.

fun = 'sin'
: The matrix sine, $\sin(A)$, will be used.

fun = 'cos'
: The matrix cosine, $\cos(A)$, will be used.

fun = 'sinh'
The hyperbolic matrix sine, $\sinh(A)$, will be used.

fun = 'cosh'
The hyperbolic matrix cosine, $\cosh(A)$, will be used.

fun = 'log'
The matrix logarithm, $\log(A)$, will be used.

Constraint: **fun** = 'exp', 'sin', 'cos', 'sinh', 'cosh' or 'log'.

2: **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 matrix, $f(A)$.

2: **conda** – REAL (KIND=nag_wp)

An estimate of the absolute condition number of f at A.

3: **norma** – REAL (KIND=nag_wp)

The 1-norm of A.

4: **normfa** – REAL (KIND=nag_wp)

The 1-norm of $f(A)$.

5: **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 $= 1$

An internal error occurred when estimating the norm of the Fréchet derivative of f at A. Please contact NAG.

ifail $= 2$

An internal error occurred when evaluating the matrix function $f(A)$. You can investigate further by calling nag_matop_complex_gen_matrix_exp (f01fc), nag_matop_complex_gen_matrix_log (f01fj) or nag_matop_complex_gen_matrix_fun_std (f01fk) with the matrix A.

ifail $= -1$

On entry, **fun** $= \langle value \rangle$ was an illegal value.

ifail $= -2$

On entry, **n** < 0.
Input argument number $\langle value \rangle$ is invalid.

ifail $= -4$

On entry, argument lda is invalid.
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_complex_gen_matrix_cond_std (f01ka) uses the norm estimation function nag_linsys_complex_gen_norm_rcomm (f04zd) to estimate a quantity γ, where $\gamma \leq \|K(X)\|_1$ and $\|K(X)\|_1 \in \left[n^{-1}\|L(X)\|_1, n\|L(X)\|_1\right]$. For further details on the accuracy of norm estimation, see the documentation for nag_linsys_complex_gen_norm_rcomm (f04zd).

8 Further Comments

Approximately $6n^2$ of complex allocatable memory is required by the routine, in addition to the memory used by the underlying matrix function routines nag_matop_complex_gen_matrix_exp (f01fc), nag_matop_complex_gen_matrix_log (f01fj) or nag_matop_complex_gen_matrix_fun_std (f01fk).

nag_matop_complex_gen_matrix_cond_std (f01ka) returns the matrix function $f(A)$. This is computed using nag_matop_complex_gen_matrix_exp (f01fc) if **fun** = 'exp', nag_matop_complex_gen_matrix_log (f01fj) if **fun** = 'log' and nag_matop_complex_gen_matrix_fun_std (f01fk) otherwise. If only $f(A)$ is required, without an estimate of the condition number, then it is far more efficient to use nag_matop_complex_gen_matrix_exp (f01fc), nag_matop_complex_gen_matrix_log (f01fj) or nag_matop_complex_gen_matrix_fun_std (f01fk) directly.

nag_matop_real_gen_matrix_cond_std (f01ja) can be used to find the condition number of the exponential, logarithm, sine, cosine, sinh or cosh at a real matrix.

9 Example

This example estimates the absolute and relative condition numbers of the matrix sinh function for

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

9.1 Program Text

```
function f01ka_example

fprintf('f01ka example results\n\n');

a = [0+1i, -1+0i,  1+0i,  2+0i;
     2+1i,  0-1i,  0+0i,  1+0i;
     0+1i,  0+0i,  1+1i,  0+2i;
     1+0i,  2+0i, -2+3i,  0+1i];
fun = 'sinh';

% Find absolute condition number estimate
[a, conda, norma, normfa, ifail] = ...
f01ka(fun, a);

fprintf('\nf(A) = %s(A)\n', fun);
fprintf('Estimated absolute condition number is: %7.2f\n', conda);

%  Find relative condition number estimate
eps = x02aj;
if normfa > eps
   cond_rel = conda*norma/normfa;
   fprintf('Estimated relative condition number is: %7.2f\n', cond_rel);
else
  fprintf('The estimated norm of f(A) is effectively zero;\n');
  fprintf('the relative condition number is therefore undefined.\n');
end
```

9.2 Program Results

```
f01ka example results

f(A) = sinh(A)
Estimated absolute condition number is:    7.33
Estimated relative condition number is:    4.94
```