NAG Toolbox

nag_matop_real_gen_matrix_cond_sqrt (f01jd)

1 Purpose

nag_matop_real_gen_matrix_cond_sqrt (f01jd) computes an estimate of the relative condition number $\kappa_{A^{1/2}}$ and a bound on the relative residual, in the Frobenius norm, for the square root of a real n by n matrix A. The principal square root, $A^{1/2}$, of A is also returned.

2 Syntax

```
[a, alpha, condsa, ifail] = nag_matop_real_gen_matrix_cond_sqrt(a, 'n', n)
[a, alpha, condsa, ifail] = f01jd(a, 'n', n)
```

3 Description

For a matrix with no eigenvalues on the closed negative real line, the principal matrix square root, $A^{1/2}$, of A is the unique square root with eigenvalues in the right half-plane.

The Fréchet derivative of a matrix function $A^{1/2}$ in the direction of the matrix E is the linear function mapping E to $L(A,E)$ such that

$$(A+E)^{1/2} - A^{1/2} - L(A,E) = o(\|A\|).$$

The absolute condition number is given by the norm of the Fréchet derivative which is defined by

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

The Fréchet derivative is linear in E and can therefore be written as

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

where the vec operator stacks the columns of a matrix into one vector, so that $K(A)$ is $n^2 \times n^2$.

nag_matop_real_gen_matrix_cond_sqrt (f01jd) uses Algorithm 3.20 from Higham (2008) to compute an estimate γ such that $\gamma \leq \|K(X)\|_F$. The quantity of γ provides a good approximation to $\|L(A)\|_F$. The relative condition number, $\kappa_{A^{1/2}}$, is then computed via

$$\kappa_{A^{1/2}} = \frac{\|L(A)\|_F \|A\|_F}{\|A^{1/2}\|_F}.$$

$\kappa_{A^{1/2}}$ is returned in the argument **condsa**.

$A^{1/2}$ is computed using the algorithm described in Higham (1987). This is a real arithmetic version of the algorithm of BjÎrck and Hammarling (1983). In addition, a blocking scheme described in Deadman *et al.* (2013) is used.

The computed quantity α is a measure of the stability of the relative residual (see Section 7). It is computed via

$$\alpha = \frac{\|A^{1/2}\|_F^2}{\|A\|_F}.$$

4 References

BjÎrck Đ and Hammarling S (1983) A Schur method for the square root of a matrix *Linear Algebra Appl.* **52/53** 127–140

Deadman E, Higham N J and Ralha R (2013) Blocked Schur Algorithms for Computing the Matrix Square Root *Applied Parallel and Scientific Computing: 11th International Conference, (PARA 2012, Helsinki, Finland)* P. Manninen and P. Úster, Eds *Lecture Notes in Computer Science* **7782** 171–181 Springer–Verlag

Higham N J (1987) Computing real square roots of a real matrix *Linear Algebra Appl.* **88/89** 405–430

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** ≥ 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**.

Contains, if **ifail** $= 0$, the n by n principal matrix square root, $A^{1/2}$. Alternatively, if **ifail** $= 1$, contains an n by n non-principal square root of A.

2: **alpha** – REAL (KIND=nag_wp)

An estimate of the stability of the relative residual for the computed principal (if **ifail** $= 0$) or non-principal (if **ifail** $= 1$) matrix square root, α.

3: **condsa** – REAL (KIND=nag_wp)

An estimate of the relative condition number, in the Frobenius norm, of the principal (if **ifail** $= 0$) or non-principal (if **ifail** $= 1$) matrix square root at A, $\kappa_{A^{1/2}}$.

4: **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 has a semisimple vanishing eigenvalue. A non-principal square root was returned.

ifail $= 2$

A has a defective vanishing eigenvalue. The square root and condition number cannot be found in this case.

ifail $= 3$

A has a negative real eigenvalue. The principal square root is not defined. nag_matop_complex_gen_matrix_cond_sqrt (f01kd) can be used to return a complex, non-principal square root.

ifail $= 4$

An error occurred when computing the matrix square root. Consequently, **alpha** and **condsa** could not be computed. It is likely that the function was called incorrectly.

ifail $= 5$

An error occurred when computing the condition number. The matrix square root was still returned but you should use nag_matop_real_gen_matrix_sqrt (f01en) to check if it is the principal matrix square root.

ifail $= -1$

Constraint: **n** ≥ 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

If the computed square root is $\tilde{X}$, then the relative residual

$$\frac{\left\|A - \tilde{X}^2\right\|_F}{\|A\|_F},$$

is bounded approximately by $n\alpha\epsilon$, where ϵ is ***machine precision***. The relative error in $\tilde{X}$ is bounded approximately by $n\alpha\kappa_{A^{1/2}}\epsilon$.

8 Further Comments

Approximately $3 \times n^2$ of real allocatable memory is required by the function.

The cost of computing the matrix square root is $85n^3/3$ floating-point operations. The cost of computing the condition number depends on how fast the algorithm converges. It typically takes over twice as long as computing the matrix square root.

If condition estimates are not required then it is more efficient to use nag_matop_real_gen_matrix_sqrt (f01en) to obtain the matrix square root alone. Condition estimates for the square root of a complex matrix can be obtained via nag_matop_complex_gen_matrix_cond_sqrt (f01kd).

9 Example

This example estimates the matrix square root and condition number of the matrix

$$A = \begin{pmatrix} -5 & 2 & -1 & 1 \\ -2 & -3 & 19 & 27 \\ -9 & 0 & 15 & 24 \\ 7 & 8 & 11 & 16 \end{pmatrix}.$$

9.1 Program Text

```
function f01jd_example

fprintf('f01jd example results\n\n');

% Principal square root and conditioning of matrix A

a = [ -5  2 -1  1;
      -2 -3 19 27;
      -9  0 15 24;
       7  8 11 16];

[as, alpha, condsa, ifail] = f01jd(a);

disp('Square root of A:');
disp(as);

fprintf('\nEstimated relative condition number is       : %6.2f\n', condsa);

fprintf('Condition number for the relative residual is: %6.2f\n', alpha)
```

9.2 Program Results

```
f01jd example results

Square root of A:
    1.0000    2.0000   -1.0000   -1.0000
   -3.0000    1.0000    2.0000    4.0000
   -2.0000    3.0000    1.0000    2.0000
    2.0000   -1.0000    3.0000    4.0000

Estimated relative condition number is       :  77.10
Condition number for the relative residual is:   1.70
```