NAG Toolbox for MATLAB

f07uv

1 Purpose

f07uv returns error bounds for the solution of a complex triangular system of linear equations with multiple right-hand sides, $AX = B$, $A^{\mathrm{T}}X = B$ or $A^{\mathrm{H}}X = B$, using packed storage.

2 Syntax

```
[ferr, berr, info] = f07uv(uplo, trans, diag, ap, b, x, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07uv returns the backward errors and estimated bounds on the forward errors for the solution of a complex triangular system of linear equations with multiple right-hand sides $AX = B$, $A^{\mathrm{T}}X = B$ or $A^{\mathrm{H}}X = B$, using packed storage. The function handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of f07uv in terms of a single right-hand side b and solution x.

Given a computed solution x, the function computes the *component-wise backward error* β. This is the size of the smallest relative perturbation in each element of A and b such that x is the exact solution of a perturbed system

$$(A + \delta A)x = b + \delta b$$
$$|\delta a_{ij}| \leq \beta |a_{ij}| \quad \text{and} \quad |\delta b_i| \leq \beta |b_i|.$$

Then the function estimates a bound for the *component-wise forward error* in the computed solution, defined by:

$$\max_i |x_i - \hat{x}_i| / \max_i |x_i|$$

where $\hat{x}$ is the true solution.

For details of the method, see the F07 Chapter Introduction.

4 References

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

5.1 Compulsory Input Parameters

1: **uplo – string**

Indicates whether A is upper or lower triangular.

uplo = 'U'

A is upper triangular.

uplo = 'L'

A is lower triangular.

Constraint: **uplo** = 'U' or 'L'.

2: **trans – string**

Indicates the form of the equations

trans = 'N'

The equations are of the form $AX = B$.

trans = 'T'

The equations are of the form $A^{\mathrm{T}}X = B$.

trans = 'C'

The equations are of the form $A^{\mathrm{H}}X = B$.

Constraint: **trans** = 'N', 'T' or 'C'.

3: **diag – string**

Indicates whether A is a nonunit or unit triangular matrix.

diag = 'N'

A is a nonunit triangular matrix.

diag = 'U'

A is a unit triangular matrix; the diagonal elements are not referenced and are assumed to be 1.

Constraint: **diag** = 'N' or 'U'.

4: **ap**$(*)$ **– complex array**

Note: the dimension of the array **ap** must be at least $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$.

The n by n triangular matrix A, packed by columns.

More precisely,

if **uplo** = 'U', the upper triangle of A must be stored with element A_{ij} in **ap**$(i+j(j-1)/2)$ for $i \leq j$;
if **uplo** = 'L', the lower triangle of A must be stored with element A_{ij} in **ap**$(i+(2n-j)(j-1)/2)$ for $i \geq j$.

If **diag** = 'U', the diagonal elements of A are assumed to be 1, and are not referenced; the same storage scheme is used whether **diag** = 'N' or 'U'.

5: **b**(**ldb,***) **– complex array**

The first dimension of the array **b** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{nrhs\_p})$

The n by r right-hand side matrix B.

6: **x**(**ldx,***) **– complex array**

The first dimension of the array **x** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{nrhs\_p})$

The n by r solution matrix X, as returned by f07us.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the order of the matrix A.

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

2: **nrhs_p – int32 scalar**

Default: The second dimension of the arrays **b**, **x**. (An error is raised if these dimensions are not equal.)

r, the number of right-hand sides.

Constraint: $\mathbf{nrhs\_p} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldb, ldx, work, rwork

5.4 Output Parameters

1: **ferr**$(*)$ **– double array**

Note: the dimension of the array **ferr** must be at least $\max(1, \mathbf{nrhs\_p})$.

ferr(j) contains an estimated error bound for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

2: **berr**$(*)$ **– double array**

Note: the dimension of the array **berr** must be at least $\max(1, \mathbf{nrhs\_p})$.

berr(j) contains the component-wise backward error bound β for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

3: **info – int32 scalar**

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **uplo**, 2: **trans**, 3: **diag**, 4: **n**, 5: **nrhs_p**, 6: **ap**, 7: **b**, 8: **ldb**, 9: **x**, 10: **ldx**, 11: **ferr**, 12: **berr**, 13: **work**, 14: **rwork**, 15: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

7 Accuracy

The bounds returned in **ferr** are not rigorous, because they are estimated, not computed exactly; but in practice they almost always overestimate the actual error.

8 Further Comments

A call to f07uv, for each right-hand side, involves solving a number of systems of linear equations of the form $Ax = b$ or $A^{\mathrm{H}}x = b$; the number is usually 5 and never more than 11. Each solution involves approximately $4n^2$ real floating-point operations.

The real analogue of this function is f07uh.

9 Example

```
uplo = 'L';
trans = 'N';
diag = 'N';
ap = [complex(4.78, +4.56);
     complex(2, -0.3);
     complex(2.89, -1.34);
     complex(-1.89, +1.15);
     complex(-4.11, +1.25);
     complex(2.36, -4.25);
     complex(0.04, -3.69);
     complex(4.15, +0.8);
     complex(-0.02, +0.46);
     complex(0.33, -0.26)];
b = [complex(-14.78, -32.36), complex(-18.02, +28.46);
     complex(2.98, -2.14), complex(14.22, +15.42);
     complex(-20.96, +17.06), complex(5.62, +35.89);
     complex(9.539999999999999, +9.91), complex(-16.46, -1.73)];
x = [complex(-5, -2), complex(0.9999999999999998, +5);
     complex(-3, -1), complex(-2, -2);
     complex(2, +1.000000000000001), complex(2.999999999999999, +4);
          complex(3.999999999999997,  +3),  complex(3.999999999999993,  -
3.000000000000003)];
[ferr, berr, info] = f07uv(uplo, trans, diag, ap, b, x)
```

```
ferr =
   1.0e-13 *
    0.3105
    0.3355
berr =
   1.0e-16 *
    0.6524
    0.7925
info =
           0
```