NAG Toolbox for MATLAB

f07as

1 Purpose

f07as solves a complex system of linear equations with multiple right-hand sides,

$$AX = B, \quad A^{\mathrm{T}}X = B \quad \text{or} \quad A^{\mathrm{H}}X = B,$$

where A has been factorized by f07ar.

2 Syntax

```
[b, info] = f07as(trans, a, ipiv, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07as is used to solve a complex system of linear equations $AX = B$, $A^{\mathrm{T}}X = B$ or $A^{\mathrm{H}}X = B$, the function must be preceded by a call to f07ar which computes the LU factorization of A as $A = PLU$. The solution is computed by forward and backward substitution.

If **trans** = 'N', the solution is computed by solving $PLY = B$ and then $UX = Y$.

If **trans** = 'T', the solution is computed by solving $U^{\mathrm{T}}Y = B$ and then $L^{\mathrm{T}}P^{\mathrm{T}}X = Y$.

If **trans** = 'C', the solution is computed by solving $U^{\mathrm{H}}Y = B$ and then $L^{\mathrm{H}}P^{\mathrm{T}}X = Y$.

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: **trans – string**

Indicates the form of the equations.

trans = 'N'

$AX = B$ is solved for X.

trans = 'T'

$A^{\mathrm{T}}X = B$ is solved for X.

trans = 'C'

$A^{\mathrm{H}}X = B$ is solved for X.

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

2: **a(lda,$*$) – complex array**

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

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

The LU factorization of A, as returned by f07ar.

3: **ipiv**($*$) **– int32 array**

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

The pivot indices, as returned by f07ar.

4: **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.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **a** The dimension of the array **ipiv**.

n, the order of the matrix A.

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

2: **nrhs_p – int32 scalar**

Default: The second dimension of the array **b**.

r, the number of right-hand sides.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb

5.4 Output Parameters

1: **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 solution matrix X.

2: **info – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

$\mathbf{info} = -i$

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

1: **trans**, 2: **n**, 3: **nrhs_p**, 4: **a**, 5: **lda**, 6: **ipiv**, 7: **b**, 8: **ldb**, 9: **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

For each right-hand side vector b, the computed solution x is the exact solution of a perturbed system of equations $(A+E)x=b$, where

$$|E| \leq c(n)\epsilon P|L||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***.

If $\hat{x}$ is the true solution, then the computed solution x satisfies a forward error bound of the form

$$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty} \leq c(n)\,\mathrm{cond}(A,x)\epsilon$$

where $\mathrm{cond}(A,x) = \||A^{-1}||A||x|\|_\infty/\|x\|_\infty \leq \mathrm{cond}(A) = \||A^{-1}||A|\|_\infty \leq \kappa_\infty(A)$.

Note that $\mathrm{cond}(A,x)$ can be much smaller than $\mathrm{cond}(A)$, and $\mathrm{cond}(A^{\mathrm{H}})$ (which is the same as $\mathrm{cond}(A^{\mathrm{T}})$) can be much larger (or smaller) than $\mathrm{cond}(A)$.

Forward and backward error bounds can be computed by calling f07av, and an estimate for $\kappa_\infty(A)$ can be obtained by calling f07au with **norm_p** = 'I'.

8 Further Comments

The total number of real floating-point operations is approximately $8n^2r$.

This function may be followed by a call to f07av to refine the solution and return an error estimate.

The real analogue of this function is f07ae.

9 Example

```
trans = 'N';
a = [complex(-3.29, -2.39), complex(-1.91, +4.42), complex(-0.14, -1.35), complex(1.72, +1.35);
                        complex(0.2376120269469406,    +0.2559596521570857), complex(4.895180634002975, ...
     -0.711362223485444), complex(-0.462279846639494, +1.696610587680362), ...
      complex(1.226852844063328, +0.6189731619114427);
             complex(-0.1019520808892006,  -0.7010135339437114),  complex(-0.6691496431943212, ...
                  +0.3688698547971655),    complex(-5.141410913810232,    -1.129969734660953), ...
      complex(0.9982579915199449, +0.3850152275763778);
             complex(-0.5358546703595748,  +0.2707272529359829),  complex(-0.2040217794587079, ...
                     +0.8601180679984047),     complex(0.008233047063940025, +0.1210636819910635), ...
      complex(0.1482391810748419, -0.1252239986075848)];
ipiv = [int32(3);
      int32(2);
      int32(3);
      int32(4)];
b = [complex(26.26, +51.78), complex(31.32, -6.7);
      complex(6.43, -8.68), complex(15.86, -1.42);
      complex(-5.75, +25.31), complex(-2.15, +30.19);
      complex(1.16, +2.57), complex(-2.56, +7.55)];
[bOut, info] = f07as(trans, a, ipiv, b)
```

```
bOut =
   1.0000 + 1.0000i  -1.0000 - 2.0000i
   2.0000 - 3.0000i   5.0000 + 1.0000i
  -4.0000 - 5.0000i  -3.0000 + 4.0000i
   0.0000 + 6.0000i   2.0000 - 3.0000i
```

```
info =

          0
```