NAG Toolbox for MATLAB

f07bt

1 Purpose

f07bt computes diagonal scaling matrices D_R and D_C intended to equilibrate a complex m by n band matrix A of band width $(k_l + k_u + 1)$, and reduce its condition number.

2 Syntax

```
[r, c, rowcnd, colcnd, amax, info] = f07bt(m, kl, ku, ab, 'n', n)
```

3 Description

f07bt computes the diagonal scaling matrices. The diagonal scaling matrices are chosen to try to make the elements of largest absolute value in each row and column of the matrix B given by

$$B = D_R A D_C$$

have absolute value 1. The diagonal elements of D_R and D_C are restricted to lie in the safe range $(\delta, 1/\delta)$, where δ is the value returned by function x02am. Use of these scaling factors is not guaranteed to reduce the condition number of A but works well in practice.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **m – int32 scalar**

m, the number of rows of the matrix A.

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

2: **kl – int32 scalar**

k_l, the number of subdiagonals of the matrix A.

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

3: **ku – int32 scalar**

k_u, the number of superdiagonals of the matrix A.

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

4: **ab**(**ldab**,$*$) **– complex array**

The first dimension of the array **ab** must be at least $\mathbf{kl} + \mathbf{ku} + 1$

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

The m by n band matrix A whose scaling factors are to be computed.

The matrix is stored in rows 1 to $k_l + k_u + 1$, more precisely, the element A_{ij} must be stored in

$$\mathbf{ab}(k_u + 1 + i - j, j) \qquad \text{for } \max(1, j - k_u) \leq i \leq \min(m, j + k_l).$$

See Section 8 for further details.

If **fact** = 'F' and **equed** $\neq$ 'N', AB must have been equilibrated by the scaling factors in **r** and/or **c**.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the number of columns of the matrix A.

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

5.3 Input Parameters Omitted from the MATLAB Interface

ldab

5.4 Output Parameters

1: **r**$(*)$ **– double array**

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

If **info** $= 0$ or **info** $> \mathbf{m}$, **r** contains the row scale factors, the diagonal elements of D_R. The elements of **r** will be positive.

2: **c**$(*)$ **– double array**

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

If **info** $= 0$, **c** contains the column scale factors, the diagonal elements of D_C. The elements of **c** will be positive.

3: **rowcnd – double scalar**

If **info** $= 0$ or **info** $> \mathbf{m}$, **rowcnd** contains the ratio of the smallest value of $\mathbf{r}(i)$ to the largest value of $\mathbf{r}(i)$. If **rowcnd** ≥ 0.1 and **amax** is neither too large nor too small, it is not worth scaling by D_R.

4: **colcnd – double scalar**

If **info** $= 0$, **colcnd** contains the ratio of the smallest value of $\mathbf{c}(i)$ to the largest value of $\mathbf{c}(i)$.

If **colcnd** ≥ 0.1, it is not worth scaling by D_C.

5: **amax – double scalar**

$\max |a_{ij}|$. If **amax** is very close to overflow or underflow, the matrix A should be scaled.

6: **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: **m**, 2: **n**, 3: **kl**, 4: **ku**, 5: **ab**, 6: **ldab**, 7: **r**, 8: **c**, 9: **rowcnd**, 10: **colcnd**, 11: **amax**, 12: **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.

info > 0 and **info** $\leq M$

If **info** $= i$, the ith row of A is exactly zero.

info $>$ M

If **info** $= i$, the $(i - \mathbf{m})$th column of A is exactly zero.

7 Accuracy

The computed scale factors will be close to the exact scale factors.

8 Further Comments

The real analogue of this function is f07bf.

9 Example

```
m = int32(4);
kl = int32(1);
ku = int32(2);
ab = [complex(0, 0), complex(0, 0), complex(0.97, -2.84), complex(0.59, -
0.48);
      complex(0, 0), complex(-2.05e-10, -8.5e-11), complex(-3.99, +4.01),
complex(33300000000, -10400000000);
        complex(-1.65, +2.26), complex(-1.48e-10, -1.75e-10), complex(-
10600000000, +19400000000), complex(-0.46, -1.72);
        complex(0, +6.3), complex(-0.77, +2.83), complex(4.48, -1.09),
complex(0, +0)];
[r, c, rowcnd, colcnd, amax, info] = f07bt(m, kl, ku, ab)
```

```
r =
    0.2558
    0.1250
    0.0000
    0.1795
c =
   1.0e+10 *
    0.0000
    1.2139
    0.0000
    0.0000
rowcnd =
   8.9474e-11
colcnd =
   8.2380e-11
amax =
   4.3700e+10
info =
           0
```