NAG Toolbox

nag_lapack_zgbequ (f07bt)

1 Purpose

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

3 Description

nag_lapack_zgbequ (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 nag_machine_real_safe (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** – INTEGER

m, the number of rows of the matrix A.

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

2: **kl** – INTEGER

k_l, the number of subdiagonals of the matrix A.

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

3: **ku** – INTEGER

k_u, the number of superdiagonals of the matrix A.

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

4: **ab**$(ldab,:)$ – COMPLEX (KIND=nag_wp) array

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

The second dimension of the array **ab** 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

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

See Section 9 in nag_lapack_zgbsv (f07bn) for further details.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the number of columns of the matrix A.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **r**(**m**) – REAL (KIND=nag_wp) array

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

2: **c**(**n**) – REAL (KIND=nag_wp) array

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

3: **rowcnd** – REAL (KIND=nag_wp)

If **info** $= 0$ or **info** $>$ **m**, **rowcnd** contains the ratio of the smallest value of **r**(i) to the largest value of **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** – REAL (KIND=nag_wp)

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

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

5: **amax** – REAL (KIND=nag_wp)

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

6: **info** – INTEGER

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

6 Error Indicators and Warnings

info < 0

If **info** $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

info > 0 and **info** $\leq$ **m** (*warning*)

Row $\langle$*value*$\rangle$ of A is exactly zero.

info $>$ **m** (*warning*)

Column $\langle$*value*$\rangle$ 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 nag_lapack_dgbequ (f07bf).

9 Example

This example equilibrates the complex band matrix A given by

$$A = \begin{pmatrix} -1.65 + 2.26i & (-2.05 - 0.85i) \times 10^{-10} & 0.97 - 2.84i & 0 \\ 0.00 + 6.30i & (-1.48 - 1.75i) \times 10^{-10} & -3.99 + 4.01i & 0.59 - 0.48i \\ 0 & -0.77 + 2.83i & (-1.06 + 1.94i) \times 10^{10} & (3.33 - 1.04i) \times 10^{10} \\ 0 & 0 & 0.48 - 1.09i & -0.46 - 1.72i \end{pmatrix}.$$

Details of the scaling factors, and the scaled matrix are output.

9.1 Program Text

```
     function f07bt_example

fprintf('f07bt example results\n\n');

m  = nag_int(4);
kl = nag_int(1);
ku = nag_int(2);
abr = [ 0     0         0.97     0.59   ;
        0    -2.05e-10 -2.99     3.33e10;
       -1.65 -1.48e-10 -1.06e10 -0.46;
        0    -0.77      4.48     0];
abi = [ 0     0        -2.84    -0.48;
        0    -8.50e-11  3.01    -1.04e10;
        2.26 -1.75e-10  1.94e10 -1.72;
        6.3   2.83     -1.09     0      ];
ab = abr + i*abi;

[r, c, rowcnd, colcnd, amax, info] = ...
  f07bt( m, kl, ku, ab);

fprintf('rowcnd = %8.1e colcnd = %8.1e amax = %8.1e\n\n', ...
        rowcnd, colcnd, amax);
fprintf('Row scale factors:\n');
fprintf('%12.2e',r)
fprintf('\nColumn scale factors:\n');
fprintf('%12.2e',c)
fprintf('\n\n');

% Compute values close to underflow and overflow
small = x02am/(x02aj*double(x02bh));
big = 1/small;
thresh = 0.1;

% Convert ab to full representation a
[a, ab, ifail] = f01zd( ...
                       'u', kl, ku, complex(zeros(m, m)), ab);

if (rowcnd >= thresh) && (amax >= small) && (amax <= big)
  if colcnd<thresh
    % Just column scale A
    as = a*diag(c);
  end
elseif colcnd>=thresh
  % Just row scale A
  as = diag(r)*a;
else
```

```
  % Row and column scale A
  as = diag(r)*a*diag(c);
end

mtitle = 'Scaled Matrix RAC:';
[ifail] = x04da( ...
                 'G', 'N', as, mtitle);
```

9.2 Program Results

```
     f07bt example results

rowcnd =  8.9e-11 colcnd =  8.2e-11 amax =  4.4e+10

Row scale factors:
    2.56e-01    1.59e-01    2.29e-11    1.80e-01
Column scale factors:
    1.00e+00    1.21e+10    1.00e+00    1.00e+00

 Scaled Matrix RAC:
          1       2       3       4
 1  -0.4220 -0.6364  0.2481  0.0000
     0.5780 -0.2639 -0.7263  0.0000

 2   0.0000 -0.2852 -0.4746  0.0937
     1.0000 -0.3372  0.4778 -0.0762

 3   0.0000 -0.2139 -0.2426  0.7620
     0.0000  0.7861  0.4439 -0.2380

 4   0.0000  0.0000  0.8043 -0.0826
     0.0000  0.0000 -0.1957 -0.3088
```