

## NAG Toolbox

### nag\_lapack\_zgseequ (f07at)

#### 1 Purpose

nag\_lapack\_zgseequ (f07at) computes real diagonal scaling matrices  $D_R$  and  $D_C$  intended to equilibrate a complex  $m$  by  $n$  matrix  $A$  and reduce its condition number.

#### 2 Syntax

```
[r, c, rowcnd, colcnd, amax, info] = nag_lapack_zgseequ(a, 'm', m, 'n', n)
[r, c, rowcnd, colcnd, amax, info] = f07at(a, 'm', m, 'n', n)
```

#### 3 Description

nag\_lapack\_zgseequ (f07at) 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  $\delta$  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: **a**(lda,:) – COMPLEX (KIND=nag\_wp) array  
 The first dimension of the array **a** must be at least  $\max(1, \mathbf{m})$ .  
 The second dimension of the array **a** must be at least  $\max(1, \mathbf{n})$ .  
 The matrix  $A$  whose scaling factors are to be computed.

##### 5.2 Optional Input Parameters

- 1: **m** – INTEGER  
*Default:* the first dimension of the array **a**.  
 $m$ , the number of rows of the matrix  $A$ .  
*Constraint:*  $\mathbf{m} \geq 0$ .
- 2: **n** – INTEGER  
*Default:* the second dimension of the array **a**.  
 $n$ , the number of columns of the matrix  $A$ .  
*Constraint:*  $\mathbf{n} \geq 0$ .

### 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  $\mathbf{r}(i)$  to the largest value of  $\mathbf{r}(i)$ . If **rowcnd**  $\geq$  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  $\mathbf{c}(i)$  to the largest value of  $\mathbf{c}(i)$ . If **colcnd**  $\geq$  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\_dgeequ (f07af).

## 9 Example

This example equilibrates the general matrix  $A$  given by

$$A = \begin{pmatrix} -1.34 + 2.55i & (0.28 + 3.17i) \times 10^{10} & -6.39 - 2.20i \\ -1.70 - 1.41i & (3.31 - 0.15i) \times 10^{10} & -0.15 + 1.34i \\ (2.41 + 0.39i) \times 10^{-10} & -0.56 + 1.47i & (-0.83 - 0.69i) \times 10^{-10} \end{pmatrix}.$$

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

## 9.1 Program Text

```
function f07at_example

fprintf('f07at example results\n\n');

a = [ -1.34      + 2.55i,      2.80e9 + 3.17e10i, -6.39      - 2.2i;
      -1.70      - 1.41i,      3.31e10 - 1.50e9i, -0.15      + 1.34i;
      2.41e-10 + 3.9e-11i, -0.56      + 1.47i,      -8.3e-11 - 6.9e-11i];

% Compute row and column scaling factors
thresh = 0.1;
[r, c, rowcnd, colcnd, amax, info] = ...
    f07at(a);

% Print rowcnd, colcnd, amax and the scale factors
format shorte;
fprintf('rowcnd = %8.1e colcnd = %8.1e amax = %8.1e\n', rowcnd, colcnd, amax);
fprintf('\nRow scale factors:\n');
disp(r');
fprintf('Column scale factors:\n');
disp(c');

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

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

format short
fprintf('Scaled Matrix:\n');
disp(as);
```

## 9.2 Program Results

```
f07at example results

rowcnd =  5.9e-11 colcnd =  1.4e-10 amax =  3.5e+10

Row scale factors:
  2.8986e-11  2.8902e-11  4.9261e-01

Column scale factors:
  7.2500e+09  1.0000e+00  4.0163e+09

Scaled Matrix:
-0.2816 + 0.5359i   0.0812 + 0.9188i  -0.7439 - 0.2561i
-0.3562 - 0.2954i   0.9566 - 0.0434i  -0.0174 + 0.1555i
 0.8607 + 0.1393i  -0.2759 + 0.7241i  -0.1642 - 0.1365i
```

---