NAG Toolbox

nag_lapack_zggbal (f08wv)

1 Purpose

nag_lapack_zggbal (f08wv) balances a pair of complex square matrices (A, B) of order n. Balancing usually improves the accuracy of computed generalized eigenvalues and eigenvectors.

2 Syntax

```
[a, b, ilo, ihi, lscale, rscale, info] = nag_lapack_zggbal(job, a, b, 'n', n)
[a, b, ilo, ihi, lscale, rscale, info] = f08wv(job, a, b, 'n', n)
```

3 Description

Balancing may reduce the 1-norm of the matrices and improve the accuracy of the computed eigenvalues and eigenvectors in the complex generalized eigenvalue problem

$$Ax = \lambda Bx.$$

nag_lapack_zggbal (f08wv) is usually the first step in the solution of the above generalized eigenvalue problem. Balancing is optional but it is highly recommended.

The term 'balancing' covers two steps, each of which involves similarity transformations on A and B. The function can perform either or both of these steps. Both steps are optional.

1. The function first attempts to permute A and B to block upper triangular form by a similarity transformation:

$$PAP^{\mathrm{T}} = F = \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ & F_{22} & F_{23} \\ & & F_{33} \end{pmatrix}$$

$$PBP^{\mathrm{T}} = G = \begin{pmatrix} G_{11} & G_{12} & G_{13} \\ & G_{22} & G_{23} \\ & & G_{33} \end{pmatrix}$$

   where P is a permutation matrix, F_{11}, F_{33}, G_{11} and G_{33} are upper triangular. Then the diagonal elements of the matrix pairs (F_{11}, G_{11}) and (F_{33}, G_{33}) are generalized eigenvalues of (A, B). The rest of the generalized eigenvalues are given by the matrix pair (F_{22}, G_{22}) which are in rows and columns i_{lo} to i_{hi}. Subsequent operations to compute the generalized eigenvalues of (A, B) need only be applied to the matrix pair (F_{22}, G_{22}); this can save a significant amount of work if $i_{\mathrm{lo}} > 1$ and $i_{\mathrm{hi}} < n$. If no suitable permutation exists (as is often the case), the function sets $i_{\mathrm{lo}} = 1$ and $i_{\mathrm{hi}} = n$.

2. The function applies a diagonal similarity transformation to (F, G), to make the rows and columns of (F_{22}, G_{22}) as close in norm as possible:

$$DF\hat{D} = \begin{pmatrix} I & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ & F_{22} & F_{23} \\ & & F_{33} \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \hat{D}_{22} & 0 \\ 0 & 0 & I \end{pmatrix}$$

$$DGD^{-1} = \begin{pmatrix} I & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} G_{11} & G_{12} & G_{13} \\ & G_{22} & G_{23} \\ & & G_{33} \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \hat{D}_{22} & 0 \\ 0 & 0 & I \end{pmatrix}$$

   This transformation usually improves the accuracy of computed generalized eigenvalues and eigenvectors.

4 References

Ward R C (1981) Balancing the generalized eigenvalue problem *SIAM J. Sci. Stat. Comp.* **2** 141–152

5 Parameters

5.1 Compulsory Input Parameters

1: **job** – CHARACTER(1)

Specifies the operations to be performed on matrices A and B.

job = 'N'
No balancing is done. Initialize **ilo** = 1, **ihi** = **n**, **lscale**$(i) = 1.0$ and **rscale**$(i) = 1.0$, for $i = 1, 2, \ldots, n$.

job = 'P'
Only permutations are used in balancing.

job = 'S'
Only scalings are are used in balancing.

job = 'B'
Both permutations and scalings are used in balancing.

Constraint: **job** = 'N', 'P', 'S' or 'B'.

2: **a**$(lda, :)$ – COMPLEX (KIND=nag_wp) array

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

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

The n by n matrix A.

3: **b**$(ldb, :)$ – COMPLEX (KIND=nag_wp) array

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

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

The n by n matrix B.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the arrays **a**, **b** and the second dimension of the arrays **a**, **b**.

n, the order of the matrices A and B.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **a**$(lda, :)$ – COMPLEX (KIND=nag_wp) array

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

The second dimension of the array **a** will be max$(1, \mathbf{n})$.

a stores the balanced matrix. If **job** = 'N', **a** is not referenced.

2: **b**$(ldb, :)$ – COMPLEX (KIND=nag_wp) array

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

The second dimension of the array **b** will be max$(1, \mathbf{n})$.

b stores the balanced matrix. If **job** = 'N', **b** is not referenced.

3: **ilo** – INTEGER

4: **ihi** – INTEGER

i_{lo} and i_{hi} are set such that $\mathbf{a}(i,j) = 0$ and $\mathbf{b}(i,j) = 0$ if $i > j$ and $1 \leq j < i_{\mathrm{lo}}$ or $i_{\mathrm{hi}} < i \leq n$.

If **job** = 'N' or 'S', $i_{\mathrm{lo}} = 1$ and $i_{\mathrm{hi}} = n$.

5: **lscale**(**n**) – REAL (KIND=nag_wp) array

Details of the permutations and scaling factors applied to the left side of the matrices A and B. If P_i is the index of the row interchanged with row i and d_i is the scaling factor applied to row i, then

lscale$(i) = P_i$, for $i = 1, 2, \ldots, i_{\mathrm{lo}} - 1$;

lscale$(i) = d_i$, for $i = i_{\mathrm{lo}}, \ldots, i_{\mathrm{hi}}$;

lscale$(i) = P_i$, for $i = i_{\mathrm{hi}} + 1, \ldots, n$.

The order in which the interchanges are made is n to $i_{\mathrm{hi}} + 1$, then 1 to $i_{\mathrm{lo}} - 1$.

6: **rscale**(**n**) – REAL (KIND=nag_wp) array

Details of the permutations and scaling factors applied to the right side of the matrices A and B.

If P_j is the index of the column interchanged with column j and $\hat{d}_j$ is the scaling factor applied to column j, then

rscale$(j) = P_j$, for $j = 1, 2, \ldots, i_{\mathrm{lo}} - 1$;

rscale$(j) = \hat{d}_j$, for $j = i_{\mathrm{lo}}, \ldots, i_{\mathrm{hi}}$;

rscale$(j) = P_j$, for $j = i_{\mathrm{hi}} + 1, \ldots, n$.

The order in which the interchanges are made is n to $i_{\mathrm{hi}} + 1$, then 1 to $i_{\mathrm{lo}} - 1$.

7: **info** – INTEGER

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

6 Error Indicators and Warnings

info $= -i$

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

1: **job**, 2: **n**, 3: **a**, 4: **lda**, 5: **b**, 6: **ldb**, 7: **ilo**, 8: **ihi**, 9: **lscale**, 10: **rscale**, 11: **work**, 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.

7 Accuracy

The errors are negligible, compared to those in subsequent computations.

8 Further Comments

nag_lapack_zggbal (f08wv) is usually the first step in computing the complex generalized eigenvalue problem but it is an optional step. The matrix B is reduced to the triangular form using the QR factorization function nag_lapack_zgeqrf (f08as) and the unitary transformation Q is applied to the matrix A by calling nag_lapack_zunmqr (f08au). This is followed by nag_lapack_zgghrd (f08ws) which reduces the matrix pair into the generalized Hessenberg form.

If the matrix pair (A, B) is balanced by this function, then any generalized eigenvectors computed subsequently are eigenvectors of the balanced matrix pair. In that case, to compute the generalized eigenvectors of the original matrix, nag_lapack_zggbak (f08ww) must be called.

The total number of floating-point operations is approximately proportional to n^2.

The real analogue of this function is nag_lapack_dggbal (f08wh).

9 Example

See Section 10 in nag_lapack_zhgeqz (f08xs) and nag_lapack_ztgevc (f08yx).

9.1 Program Text

```
function f08wv_example

fprintf('f08wv example results\n\n');

job = 'B';
a = [ 1 + 3i,   1 + 4i,   1 + 5i,    1 + 6i;
      2 + 2i,   4 + 3i,   8 + 4i,   16 + 5i;
      3 + 1i,   9 + 2i,  27 + 3i,   81 + 4i;
      4 + 0i,  16 + 1i,  64 + 2i,  256 + 3i];
b = [ 1,        2 + 1i,   3 + 2i,    4 + 3i;
      1 + 1i,   4 + 2i,   9 + 3i,   16 + 4i;
      1 + 2i,   8 + 3i,  27 + 4i,   64 + 5i;
      1 + 3i,  16 + 4i,  81 + 5i,  256 + 6i];

[abal, bbal, ilo, ihi, lscale, rscale, info] = ...
  f08wv(job, a, b);

disp('Balanced A:');
disp(abal);
disp('Balanced B:');
disp(bbal);
disp('Scaling factors:');
disp('     Left      Right');
disp([lscale rscale]);
```

9.2 Program Results

```
f08wv example results

Balanced A:
   1.0000 + 3.0000i   1.0000 + 4.0000i   0.1000 + 0.5000i   0.1000 + 0.6000i
   2.0000 + 2.0000i   4.0000 + 3.0000i   0.8000 + 0.4000i   1.6000 + 0.5000i
   0.3000 + 0.1000i   0.9000 + 0.2000i   0.2700 + 0.0300i   0.8100 + 0.0400i
   0.4000 + 0.0000i   1.6000 + 0.1000i   0.6400 + 0.0200i   2.5600 + 0.0300i

Balanced B:
   1.0000 + 0.0000i   2.0000 + 1.0000i   0.3000 + 0.2000i   0.4000 + 0.3000i
   1.0000 + 1.0000i   4.0000 + 2.0000i   0.9000 + 0.3000i   1.6000 + 0.4000i
   0.1000 + 0.2000i   0.8000 + 0.3000i   0.2700 + 0.0400i   0.6400 + 0.0500i
   0.1000 + 0.3000i   1.6000 + 0.4000i   0.8100 + 0.0500i   2.5600 + 0.0600i

Scaling factors:
     Left      Right
    1.0000    1.0000
    1.0000    1.0000
    0.1000    0.1000
    0.1000    0.1000
```