NAG Toolbox

nag_lapack_dggbak (f08wj)

1 Purpose

nag_lapack_dggbak (f08wj) forms the right or left eigenvectors of the real generalized eigenvalue problem $Ax = \lambda Bx$, by backward transformation on the computed eigenvectors given by nag_lapack_dtgevc (f08yk). It is necessary to call this function only if the optional balancing function nag_lapack_dggbal (f08wh) was previously called to balance the matrix pair (A, B).

2 Syntax

```
[v, info] = nag_lapack_dggbak(job, side, ilo, ihi, lscale, rscale, v, 'n', n,
'm', m)
```

```
[v, info] = f08wj(job, side, ilo, ihi, lscale, rscale, v, 'n', n, 'm', m)
```

3 Description

If the matrix pair has been previously balanced using the function nag_lapack_dggbal (f08wh) then nag_lapack_dggbak (f08wj) backtransforms the eigenvector solution given by nag_lapack_dtgevc (f08yk). This is usually the sixth and last step in the solution of the generalized eigenvalue problem.

For a description of balancing, see the document for nag_lapack_dggbal (f08wh).

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 backward transformation step required.

job = 'N'
No transformations are done.

job = 'P'
Only do backward transformations based on permutations.

job = 'S'
Only do backward transformations based on scaling.

job = 'B'
Do backward transformations for both permutations and scaling.

Note: this must be the same argument **job** as supplied to nag_lapack_dggbal (f08wh).

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

2: **side** – CHARACTER(1)

Indicates whether left or right eigenvectors are to be transformed.

side = 'L'
The left eigenvectors are transformed.

side = 'R'
The right eigenvectors are transformed.

Constraint: **side** = 'L' or 'R'.

3: **ilo** – INTEGER
4: **ihi** – INTEGER

i_{lo} and i_{hi} as determined by a previous call to nag_lapack_dggbal (f08wh).

Constraints:

if $\mathbf{n} > 0$, $1 \leq \mathbf{ilo} \leq \mathbf{ihi} \leq \mathbf{n}$;
if $\mathbf{n} = 0$, $\mathbf{ilo} = 1$ and $\mathbf{ihi} = 0$.

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

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

Details of the permutations and scaling factors applied to the left side of the matrices A and B, as returned by a previous call to nag_lapack_dggbal (f08wh).

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

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

Details of the permutations and scaling factors applied to the right side of the matrices A and B, as returned by a previous call to nag_lapack_dggbal (f08wh).

7: $\mathbf{v}(ldv, :)$ – REAL (KIND=nag_wp) array

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

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

The matrix of right or left eigenvectors, as returned by nag_lapack_dggbal (f08wh).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **v**.

n, the order of the matrices A and B of the generalized eigenvalue problem.

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

2: **m** – INTEGER

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

m, the required number of left or right eigenvectors.

Constraint: $0 \leq \mathbf{m} \leq \mathbf{n}$.

5.3 Output Parameters

1: $\mathbf{v}(ldv, :)$ – REAL (KIND=nag_wp) array

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

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

The transformed right or left eigenvectors.

2: **info** – INTEGER

$\mathbf{info} = 0$ 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: **side**, 3: **n**, 4: **ilo**, 5: **ihi**, 6: **lscale**, 7: **rscale**, 8: **m**, 9: **v**, 10: **ldv**, 11: **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 with the previous computations.

8 Further Comments

The number of operations is proportional to n^2.

The complex analogue of this function is nag_lapack_zggbak (f08ww).

9 Example

See Section 10 in nag_lapack_dhgeqz (f08xe) and nag_lapack_dtgevc (f08yk).

9.1 Program Text

```
     function f08wj_example

fprintf('f08wj example results\n\n');

n = 5;
a = [ 1.0   1.0    1.0    1.0     1.0;
      2.0   4.0    8.0   16.0    32.0;
      3.0   9.0   27.0   81.0   243.0;
      4.0  16.0   64.0  256.0  1024.0;
      5.0  25.0  125.0  625.0  3125.0];
b = a';

%' Balance A and B
job = 'B';
[a, b, ilo, ihi, lscale, rscale, info] = ...
  f08wh(job, a, b);

bbal = b(ilo:ihi,ilo:ihi);
abal = a(ilo:ihi,ilo:ihi);

% QR factorize balanced B
[QR, tau, info] = f08ae(bbal);

% Perform C = Q^T*A
side = 'Left';
trans = 'Transpose';
[c, info] = f08ag( ...
    side, trans, QR, tau, abal);

% Generalized Hessenberg form (C,R) -> (H,T)
% Form Q explicitly and let Z = I.
[q, info] = f08af(QR, tau);

z         = eye(n);
jlo       = nag_int(1);
jhi       = nag_int(ihi-ilo+1);
compq     = 'Vectors Q';
compz     = 'Vectors Z';
```

```
[H, T, q, z, info] = ...
  f08we( ...
         compq, compz, jlo, jhi, c, QR, q, z);

% Find eigenvalues of generalized Hessenberg form
%    = eigenvalues of (A,B).
% and return Schur form for computing eigenvectors
job = ’Schur form’;
[HS, TS, alphar, alphai, beta, q, z, info] = ...
  f08xe( ...
  job, compq, compz, jlo, jhi, H, T, q, z);

disp(’Generalized eigenvalues of (A,B):’);
w = complex(alphar+i*alphai);
disp(w./beta);

% Obtain scaled eigenvectors from Schur form
side = ’Both sides’;
howmny = ’Backtransformed using Q and Z’;
select = [false];
[q, z, m, info] = f08yk( ...
  side, howmny, select, HS, TS, q, z, jhi);

% rescale to obtain left and right eigenvectors of (A,B)
job  = ’Back scale’;
side = ’Left’;
[VL, info] = f08wj( ...
     job, side, jlo, jhi, lscale, rscale, q);
side = ’Right’;
[VR, info] = f08wj( ...
     job, side, jlo, jhi, lscale, rscale, z);

disp(’Left Eigenvectors’);
disp(VL);

disp(’Right Eigenvectors’);
disp(VR);
```

9.2 Program Results

```
     f08wj example results

Generalized eigenvalues of (A,B):
  -2.4367 + 0.0000i
   0.6069 + 0.7948i
   0.6069 - 0.7948i
   1.0000 + 0.0000i
  -0.4104 + 0.0000i

Left Eigenvectors
   -0.0695   -0.2092   -0.0053   -0.0741    0.0494
    0.1361    0.1635    0.1137    0.1354   -0.1061
   -0.1000   -0.0463   -0.0537   -0.1000    0.1000
    0.0319    0.0059    0.0148    0.0265   -0.0438
   -0.0036   -0.0002   -0.0021   -0.0037    0.0070

Right Eigenvectors
   -0.0494   -0.2077    0.0257   -0.0741   -0.0695
    0.1061    0.1785    0.0883    0.1354    0.1361
   -0.1000   -0.0537   -0.0463   -0.1000   -0.1000
    0.0438    0.0080    0.0138    0.0265    0.0319
   -0.0070   -0.0006   -0.0021   -0.0037   -0.0036
```