NAG Toolbox

nag_lapack_zggbak (f08ww)

1 Purpose

nag_lapack_zggbak (f08ww) 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_ztgevc (f08yx). It is necessary to call this function only if the optional balancing function nag_lapack_zggbal (f08wv) was previously called to balance the matrix pair (A, B).

2 Syntax

```
[v, info] = nag_lapack_zggbak(job, side, ilo, ihi, lscale, rscale, v, 'n', n, 'm', m)
[v, info] = f08ww(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_zggbal (f08wv) then nag_lapack_zggbak (f08ww) backtransforms the eigenvector solution given by nag_lapack_ztgevc (f08yx). 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_zggbal (f08wv).

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 backtransformation 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 identical to the 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_zggbal (f08wv).

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_zggbal (f08wv).

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_zggbal (f08wv).

7: $\mathbf{v}(ldv, :)$ – COMPLEX (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_zggbal (f08wv).

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: $\mathbf{v}(ldv, :)$ – COMPLEX (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

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.

8 Further Comments

The number of operations is proportional to n^2.

The real analogue of this function is nag_lapack_dggbak (f08wj).

9 Example

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

9.1 Program Text

```
function f08ww_example

fprintf('f08ww example results\n\n');

% Generalized eigenvalues of matrix pair (A,B) , where
a = [ 1.0+3.0i  1.0+4.0i  1.0+5.0i   1.0+6.0i;
      2.0+2.0i  4.0+3.0i  8.0+4.0i  16.0+5.0i;
      3.0+1.0i  9.0+2.0i 27.0+3.0i  81.0+4.0i;
      4.0+0.0i 16.0+1.0i 64.0+2.0i 256.0+3.0i];
b = [ 1.0+0.0i  2.0+1.0i  3.0+2.0i   4.0+3.0i;
      1.0+1.0i  4.0+2.0i  9.0+3.0i  16.0+4.0i;
      1.0+2.0i  8.0+3.0i 27.0+4.0i  64.0+5.0i;
      1.0+3.0i 16.0+4.0i 81.0+5.0i 256.0+6.0i];

% Balance matrix pair
job = 'B';
[a, b, ilo, ihi, lscale, rscale, info] = ...
  f08wv(job, a, b);
bbal = b(ilo:ihi,ilo:ihi);
abal = a(ilo:ihi,ilo:ihi);

% QR factorize balanced B
[QR, tau, info] = f08as(bbal);

% Perform C = Q^H*A
side = 'Left';
trans = 'Conjugate transpose';
[c, info] = f08au( ...
    side, trans, QR, tau, abal);

% Generalized Hessenberg form (C,R) -> (H,T)
compq = 'Vectors Q';
compz = 'Vectors Z';

% Form Q explicitly and let Z = I.
[q, info] = f08at(QR, tau);
z = complex(eye(4));
```

```
jlo = nag_int(1);
jhi = nag_int(ihi-ilo+1);
[H, T, q, z, info] = ...
  f08ws( ...
  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';
ilo = nag_int(1);
ihi = nag_int(4);
[HS, TS, alpha, beta, q, z, info] = ...
  f08xs( ...
  job, compq, compz, jlo, jhi, H, T, q, z);

disp('Generalized eigenvalues of (A,B):');
disp(alpha./beta);

% Obtain scaled eigenvectors from Schur form
side = 'Both sides';
howmny = 'Backtransformed using Q and Z';
select = [false];
[q, z, m, info] = f08yx( ...
  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] = f08ww( ...
     job, side, jlo, jhi, lscale, rscale, q);
side = 'Right';
[VR, info] = f08ww( ...
     job, side, jlo, jhi, lscale, rscale, z);

disp('Left Eigenvectors');
disp(VL);

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

9.2 Program Results

```
     f08ww example results

Generalized eigenvalues of (A,B):
  -0.6354 + 1.6529i
   0.4580 - 0.8426i
   0.4934 + 0.9102i
   0.6744 - 0.0499i

Left Eigenvectors
  -0.1725 - 0.2037i  -0.0406 - 0.0402i   0.0560 + 0.0022i   0.2371 + 0.1695i
   0.2068 + 0.0997i   0.0877 + 0.0860i  -0.1153 + 0.0499i  -0.5341 - 0.0621i
  -0.0739 - 0.0261i  -0.0475 - 0.0525i   0.0591 - 0.0409i   0.0972 + 0.0028i
   0.0105 + 0.0015i   0.0058 + 0.0124i  -0.0080 + 0.0112i  -0.0129 + 0.0009i

Right Eigenvectors
   0.1437 - 0.1675i  -0.0889 + 0.0018i   0.0693 - 0.0167i   0.7361 + 0.0355i
  -0.1710 + 0.1045i   0.1462 + 0.0020i  -0.1406 + 0.0248i  -0.7608 + 0.1006i
   0.0628 - 0.0372i  -0.0815 - 0.0185i   0.0798 - 0.0202i   0.0947 - 0.0053i
  -0.0095 + 0.0069i   0.0134 + 0.0086i  -0.0136 + 0.0083i   0.0050 - 0.0014i
```