NAG Toolbox for MATLAB

f08wp

1 Purpose

f08wp computes for a pair of n by n complex nonsymmetric matrices (A, B) the generalized eigenvalues and, optionally, the left and/or right generalized eigenvectors using the QZ algorithm.

Optionally it also computes a balancing transformation to improve the conditioning of the eigenvalues and eigenvectors, reciprocal condition numbers for the eigenvalues, and reciprocal condition numbers for the right eigenvectors).

2 Syntax

```
[a, b, alpha, beta, vl, vr, ilo, ihi, lscale, rscale, abnrm, bbnrm,
rconde, rcondv, info] = f08wp(balanc, jobvl, jobvr, sense, a, b, 'n', n)
```

3 Description

A generalized eigenvalue for a pair of matrices (A, B) is a scalar λ or a ratio $\alpha/\beta = \lambda$, such that $A - \lambda B$ is singular. It is usually represented as the pair (α, β), as there is a reasonable interpretation for $\beta = 0$, and even for both being zero.

The right generalized eigenvector v_j corresponding to the generalized eigenvalue λ_j of (A, B) satisfies

$$Av_j = \lambda_j Bv_j.$$

The left generalized eigenvector u_j corresponding to the generalized eigenvalues λ_j of (A, B) satisfies

$$u_j^{\mathrm{H}} A = \lambda_j u_j^{\mathrm{H}} B,$$

where u_j^{H} is the conjugate-transpose of u_j.

All the eigenvalues and, if required, all the eigenvectors of the complex generalized eigenproblem $Ax = \lambda Bx$ where A and B are complex, square matrices, are determined using the QZ algorithm. The complex QZ algorithm consists of three stages:

1. A is reduced to upper Hessenberg form (with real, nonnegative subdiagonal elements) and at the same time B is reduced to upper triangular form.
2. A is further reduced to triangular form while the triangular form of B is maintained and the diagonal elements of B are made real and nonnegative. This is the generalized Schur form of the pair (A, B).

   This function does not actually produce the eigenvalues λ_j, but instead returns α_j and β_j such that

   $$\lambda_j = \alpha_j/\beta_j, \qquad j = 1, 2, \ldots, n.$$

   The division by β_j becomes your responsibility, since β_j may be zero, indicating an infinite eigenvalue.
3. If the eigenvectors are required they are obtained from the triangular matrices and then transferred back into the original co-ordinate system.

For details of the balancing option, see Section 3 of the document for f08wv.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D 1999 *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia URL: http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Wilkinson J H 1979 Kronecker's canonical form and the *QZ* algorithm *Linear Algebra Appl.* **28** 285–303

5 Parameters

5.1 Compulsory Input Parameters

1: **balanc – string**

Specifies the balance option to be performed.

balanc = 'N'

Do not diagonally scale or permute.

balanc = 'P'

Permute only.

balanc = 'S'

Scale only.

balanc = 'B'

Both permute and scale.

Computed reciprocal condition numbers will be for the matrices after permuting and/or balancing. Permuting does not change condition numbers (in exact arithmetic), but balancing does. In the absence of other information, **balanc** = 'B' is recommended.

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

2: **jobvl – string**

If **jobvl** = 'N', do not compute the left generalized eigenvectors.

If **jobvl** = 'V', compute the left generalized eigenvectors.

Constraint: **jobvl** = 'N' or 'V'.

3: **jobvr – string**

If **jobvr** = 'N', do not compute the right generalized eigenvectors.

If **jobvr** = 'V', compute the right generalized eigenvectors.

Constraint: **jobvr** = 'N' or 'V'.

4: **sense – string**

Determines which reciprocal condition numbers are computed.

sense = 'N'

None are computed.

sense = 'E'

Computed for eigenvalues only.

sense = 'V'

Computed for eigenvectors only.

sense = 'B'

Computed for eigenvalues and eigenvectors.

Constraint: **sense** = 'N', 'E', 'V' or 'B'.

5: **a(lda,$*$) – complex array**

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

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

The matrix A in the pair (A, B).

6: **b(ldb,$*$) – complex array**

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

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

The matrix B in the pair (A, B).

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **a** The second dimension of the array **b**.

n, the order of the matrix pencil (A, B).

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldvl, ldvr, work, lwork, rwork, iwork, bwork

5.4 Output Parameters

1: **a(lda,$*$) – complex array**

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

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

a has been overwritten. If **jobvl** $=$ 'V' or **jobvr** $=$ 'V' or both, then A contains the first part of the Schur form of the 'balanced' versions of the input A and B.

2: **b(ldb,$*$) – complex array**

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

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

b has been overwritten.

3: **alpha($*$) – complex array**

Note: the dimension of the array **alpha** must be at least $\max(1, \mathbf{n})$.

See the description of **beta**.

4: **beta($*$) – complex array**

Note: the dimension of the array **beta** must be at least $\max(1, \mathbf{n})$.

alpha$(j)/$**beta**(j), for $j = 1, \ldots, \mathbf{n}$, will be the generalized eigenvalues.

Note: the quotients **alpha**$(j)/$**beta**(j) may easily overflow or underflow, and **beta**(j) may even be zero. Thus, you should avoid naively computing the ratio α_j / β_j. However, $\max|\alpha_j|$ will always be less than and usually comparable with $\|\mathbf{a}\|_2$ in magnitude, and $\max|\beta_j|$ will always be less than and usually comparable with $\|\mathbf{b}\|_2$.

5: **vl(ldvl,$*$) – complex array**

The first dimension, **ldvl**, of the array **vl** must satisfy

if **jobvl** = 'V', **ldvl** $\geq \max(1, \mathbf{n})$;
ldvl ≥ 1 otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **jobvl** = 'V', and at least 1 otherwise

If **jobvl** = 'V', the left generalized eigenvectors u_j are stored one after another in the columns of **vl**, in the same order as the corresponding eigenvalues. Each eigenvector will be scaled so the largest component will have $|\text{real part}| + |\text{imag. part}| = 1$.

If **jobvl** = 'N', **vl** is not referenced.

6: **vr(ldvr,$*$) – complex array**

The first dimension, **ldvr**, of the array **vr** must satisfy

if **jobvr** = 'V', **ldvr** $\geq \max(1, \mathbf{n})$;
ldvr ≥ 1 otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **jobvr** = 'V', and at least 1 otherwise

If **jobvr** = 'V', the right generalized eigenvectors v_j are stored one after another in the columns of **vr**, in the same order as the corresponding eigenvalues. Each eigenvector will be scaled so the largest component will have $|\text{real part}| + |\text{imag. part}| = 1$.

If **jobvr** = 'N', **vr** is not referenced.

7: **ilo – int32 scalar**
8: **ihi – int32 scalar**

ilo and **ihi** are integer values such that $\mathbf{a}(i,j) = 0$ and $\mathbf{b}(i,j) = 0$ if $i > j$ and $j = 1, \ldots, \mathbf{ilo} - 1$ or $i = \mathbf{ihi} + 1, \ldots, \mathbf{n}$.

If **balanc** = 'N' or 'S', **ilo** $= 1$ and **ihi** $= \mathbf{n}$.

9: **lscale($*$) – double array**

Note: 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 A and B.

If pl_j is the index of the row interchanged with row j, and dl_j is the scaling factor applied to row j, then:

$\mathbf{lscale}(j) = pl_j$, for $j = 1, \ldots, \mathbf{ilo} - 1$;

$\mathbf{lscale} = dl_j$, for $j = \mathbf{ilo}, \ldots, \mathbf{ihi}$;

$\mathbf{lscale} = pl_j$, for $j = \mathbf{ihi} + 1, \ldots, \mathbf{n}$.

The order in which the interchanges are made is $\mathbf{n}$ to $\mathbf{ihi} + 1$, then 1 to $\mathbf{ilo} - 1$.

10: **rscale($*$) – double array**

Note: 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 A and B.

If pr_j is the index of the column interchanged with column j, and dr_j is the scaling factor applied to column j, then:

$\mathbf{rscale}(j) = pr_j$, for $j = 1, \ldots, \mathbf{ilo} - 1$;
if $\mathbf{rscale} = dr_j$, for $j = \mathbf{ilo}, \ldots, \mathbf{ihi}$;
if $\mathbf{rscale} = pr_j$, for $j = \mathbf{ihi} + 1, \ldots, \mathbf{n}$.

The order in which the interchanges are made is **n** to **ihi** $+ 1$, then 1 to **ilo** $- 1$.

11: **abnrm – double scalar**

The 1-norm of the balanced matrix A.

12: **bbnrm – double scalar**

The 1-norm of the balanced matrix B.

13: **rconde**$(*)$ **– double array**

Note: the dimension of the array **rconde** must be at least $\max(1, \mathbf{n})$.

If **sense** $=$ 'E' or 'B', the reciprocal condition numbers of the eigenvalues, stored in consecutive elements of the array.

If **sense** $=$ 'N' or 'V', **rconde** is not referenced.

14: **rcondv**$(*)$ **– double array**

Note: the dimension of the array **rcondv** must be at least $\max(1, \mathbf{n})$.

If **sense** $=$ 'V' or 'B', the estimated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elements of the array.

If **sense** $=$ 'N' or 'E', **rcondv** is not referenced.

15: **info – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

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

1: **balanc**, 2: **jobvl**, 3: **jobvr**, 4: **sense**, 5: **n**, 6: **a**, 7: **lda**, 8: **b**, 9: **ldb**, 10: **alpha**, 11: **beta**, 12: **vl**, 13: **ldvl**, 14: **vr**, 15: **ldvr**, 16: **ilo**, 17: **ihi**, 18: **lscale**, 19: **rscale**, 20: **abnrm**, 21: **bbnrm**, 22: **rconde**, 23: **rcondv**, 24: **work**, 25: **lwork**, 26: **rwork**, 27: **iwork**, 28: **bwork**, 29: **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.

info $= 1$ to N

The QZ iteration failed. No eigenvectors have been calculated, but **alpha**(j) and **beta**(j) should be correct for $j = \mathbf{info} + 1, \dots, \mathbf{n}$.

info $= N + 1$

Unexpected error returned from f08xs.

info $= N + 2$

Error returned from f08yx.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrices $(A + E)$ and $(B + F)$, where

$$\|(E, F)\|_F = O(\epsilon)\|(A, B)\|_F,$$

and ϵ is the ***machine precision***.

An approximate error bound on the chordal distance between the ith computed generalized eigenvalue w and the corresponding exact eigenvalue λ is

$$\epsilon \times \|\mathbf{abnrm}, \mathbf{bbnrm}\|_2 / \mathbf{rconde}(i).$$

An approximate error bound for the angle between the ith computed eigenvector $\mathbf{vl}(i)$ or $\mathbf{vr}(i)$ is given by

$$\epsilon \times \|\mathbf{abnrm}, \mathbf{bbnrm}\|_2 / \mathbf{rcondv}(i).$$

For further explanation of the reciprocal condition numbers **rconde** and **rcondv**, see Section 4.11 of Anderson *et al.* 1999.

Note: interpretation of results obtained with the *QZ* algorithm often requires a clear understanding of the effects of small changes in the original data. These effects are reviewed in Wilkinson 1979, in relation to the significance of small values of α_j and β_j. It should be noted that if α_j and β_j are **both** small for any j, it may be that no reliance can be placed on **any** of the computed eigenvalues $\lambda_i = \alpha_i / \beta_i$. You are recommended to study Wilkinson 1979 and, if in difficulty, to seek expert advice on determining the sensitivity of the eigenvalues to perturbations in the data.

8 Further Comments

The total number of floating-point operations is proportional to n^3.

The real analogue of this function is f08wb.

9 Example

```
balanc = 'Balance';
jobvl = 'No vectors (left)';
jobvr = 'Vectors (right)';
sense = 'Both reciprocal condition numbers';
a = [complex(-21.1, -22.5), complex(53.5, -50.5), complex(-34.5, +127.5),
complex(7.5, +0.5);
      complex(-0.46, -7.78), complex(-3.5, -37.5), complex(-15.5, +58.5),
complex(-10.5, -1.5);
       complex(4.3, -5.5), complex(39.7, -17.1), complex(-68.5, +12.5),
complex(-7.5, -3.5);
        complex(5.5, +4.4), complex(14.4, +43.3), complex(-32.5, -46),
complex(-19, -32.5)];
b = [complex(1, -5), complex(1.6, +1.2), complex(-3, +0), complex(0, -1);
     complex(0.8, -0.6), complex(3, -5), complex(-4, +3), complex(-2.4, -
3.2);
     complex(1, +0), complex(2.4, +1.8), complex(-4, -5), complex(0, -3);
      complex(0, +1), complex(-1.8, +2.4), complex(0, -4), complex(4, -
5)];
[aOut, bOut, alpha, beta, vl, vr, ilo, ihi, lscale, rscale, abnrm, bbnrm,
rconde, rcondv, info] = ...
    f08wp(balanc, jobvl, jobvr, sense, a, b)
```

```
aOut =
    0.9609 - 0.3203i   -3.6446 - 4.8774i   -7.7988 - 3.2459i   -2.6719 -
6.9098i
         0                  0.4333 - 1.0833i   -2.0687 - 2.2227i   -1.7528 -
3.6886i
         0                   0                  0.5974 - 1.7923i    2.4089 -
1.0332i
         0                   0                   0                  2.1795 -
2.7244i
```

```
bOut =
    0.3203                 0.5272 - 0.7921i  -0.1835 - 0.6366i   1.8524 -
1.9614i
         0                0.2167              0.1155 - 0.2971i   0.6997 -
0.9676i
         0                     0             0.1991              0.7279 +
0.4512i
       0                  0                  0              0.5449
alpha =
   0.9609 - 0.3203i
   0.4333 - 1.0833i
   0.5974 - 1.7923i
   2.1795 - 2.7244i
beta =
    0.3203
    0.2167
    0.1991
    0.5449
vl =
  4.2440e-314 - 2.4370e-54i
vr =
   -0.7326 - 0.2674i  -0.5202 + 0.4798i  -0.3614 + 0.6386i  -0.3660 +
0.6340i
   -0.1493 + 0.0451i  -0.0007 + 0.0040i   0.0188 + 0.1455i   0.0010 +
0.0081i
   -0.1307 + 0.0851i  -0.0327 + 0.0302i  -0.1455 + 0.0188i   0.0122 -
0.0211i
   -0.0851 - 0.1307i  -0.0302 - 0.0327i  -0.0188 - 0.1455i  -0.0986 -
0.0569i
ilo =
           1
ihi =
           4
lscale =
    0.1000
    0.1000
    1.0000
    0.1000
rscale =
    1.0000
    0.1000
    0.1000
    1.0000
abnrm =
   13.8534
bbnrm =
    4.1403
rconde =
    0.1340
    0.3756
    0.5108
    0.6195
rcondv =
    0.1290
    0.0584
    0.0515
    0.0626
info =
           0
```