NAG Toolbox for MATLAB

f08qk

1 Purpose

f08qk computes selected left and/or right eigenvectors of a real upper quasi-triangular matrix.

2 Syntax

```
[select, vl, vr, m, info] = f08qk(job, howmny, select, t, vl, vr, mm,
'n', n)
```

3 Description

f08qk computes left and/or right eigenvectors of a real upper quasi-triangular matrix T in canonical Schur form. Such a matrix arises from the Schur factorization of a real general matrix, as computed by f08pe, for example.

The right eigenvector x, and the left eigenvector y, corresponding to an eigenvalue λ, are defined by:

$$Tx = \lambda x \qquad \text{and} \qquad y^{\mathrm{H}}T = \lambda y^{\mathrm{H}}\left(\text{or } T^{\mathrm{T}}y = \bar{\lambda}y\right).$$

Note that even though T is real, λ, x and y may be complex. If x is an eigenvector corresponding to a complex eigenvalue λ, then the complex conjugate vector $\bar{x}$ is the eigenvector corresponding to the complex conjugate eigenvalue $\bar{\lambda}$.

The function can compute the eigenvectors corresponding to selected eigenvalues, or it can compute all the eigenvectors. In the latter case the eigenvectors may optionally be pre-multiplied by an input matrix Q. Normally Q is an orthogonal matrix from the Schur factorization of a matrix A as $A = QTQ^{\mathrm{T}}$; if x is a (left or right) eigenvector of T, then Qx is an eigenvector of A.

The eigenvectors are computed by forward or backward substitution. They are scaled so that, for a real eigenvector x, $\max(|x_i|) = 1$, and for a complex eigenvector, $\max(|\mathrm{Re}(x_i)| + |\mathrm{Im}(x_i)|) = 1$.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **job – string**

Indicates whether left and/or right eigenvectors are to be computed.

job = 'R'

Only right eigenvectors are computed.

job = 'L'

Only left eigenvectors are computed.

job = 'B'

Both left and right eigenvectors are computed.

Constraint: **job** = 'R', 'L' or 'B'.

2: **howmny – string**

Indicates how many eigenvectors are to be computed.

howmny = 'A'

All eigenvectors (as specified by **job**) are computed.

howmny = 'B' or 'O'

All eigenvectors (as specified by **job**) are computed and then pre-multiplied by the matrix Q (which is overwritten).

howmny = 'S'

Selected eigenvectors (as specified by **job** and **select**) are computed.

Constraint: **howmny** = 'A', 'B', 'O' or 'S'.

3: **select**($*$) **– logical array**

Note: the dimension of the array **select** must be at least $\max(1, \mathbf{n})$ if **howmny** = 'S', and at least 1 otherwise.

Specifies which eigenvectors are to be computed if **howmny** = 'S'. To obtain the real eigenvector corresponding to the real eigenvalue λ_j, **select**(j) must be set **true**. To select the complex eigenvector corresponding to a complex conjugate pair of eigenvalues λ_j and λ_{j+1}, **select**(j) and/or **select**($j+1$) must be set **true**; the eigenvector corresponding to the **first** eigenvalue in the pair is computed.

4: **t**(**ldt**,$*$) **– double array**

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

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

The n by n upper quasi-triangular matrix T in canonical Schur form, as returned by f08pe.

5: **vl**(**ldvl**,$*$) **– double array**

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

if **job** = 'L' or 'B', **ldvl** $\geq \max(1, \mathbf{n})$;
if **job** = 'R', **ldvl** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** = 'L' or 'B' and at least 1 if **job** = 'R'

If **howmny** = 'O' or 'B' and **job** = 'L' or 'B', **vl** must contain an n by n matrix Q (usually the matrix of Schur vectors returned by f08pe).

If **howmny** = 'A' or 'S', **vl** need not be set.

6: **vr**(**ldvr**,$*$) **– double array**

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

if **job** = 'R' or 'B', **ldvr** $\geq \max(1, \mathbf{n})$;
if **job** = 'L', **ldvr** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** = 'R' or 'B' and at least 1 if **job** = 'L'

If **howmny** = 'O' or 'B' and **job** = 'R' or 'B', **vr** must contain an n by n matrix Q (usually the matrix of Schur vectors returned by f08pe).

If **howmny** = 'A' or 'S', **vr** need not be set.

7: **mm – int32 scalar**

the number of columns in the arrays **vl** and/or **vr**. The precise number of columns required , m, is n if **howmny** = 'A', 'O' or 'B'; if **howmny** = 'S', m is obtained by counting 1 for each selected real eigenvector and 2 for each selected complex eigenvector (see **select**), in which case $0 \leq m \leq n$.

Constraint: **mm** $\geq m$.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **t**.

n, the order of the matrix T.

Constraint: **n** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

ldt, ldvl, ldvr, work

5.4 Output Parameters

1: **select**$(*)$ **– logical array**

Note: the dimension of the array **select** must be at least $\max(1, \mathbf{n})$ if **howmny** = 'S', and at least 1 otherwise.

If a complex eigenvector was selected as specified above, then **select**(j) is set to **true** and **select**$(j+1)$ to **false**.

If **howmny** = 'A', 'O' or 'B', **select** is not referenced.

2: **vl**(**ldvl**,$*$) **– double array**

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

if **job** = 'L' or 'B', **ldvl** $\geq \max(1, \mathbf{n})$;
if **job** = 'R', **ldvl** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** = 'L' or 'B' and at least 1 if **job** = 'R'

If **job** = 'L' or 'B', **vl** contains the computed left eigenvectors (as specified by **howmny** and **select**). The eigenvectors are stored consecutively in the columns of the array, in the same order as their eigenvalues. Corresponding to each real eigenvalue is a real eigenvector, occupying one column. Corresponding to each complex conjugate pair of eigenvalues, is a complex eigenvector occupying two columns; the first column holds the real part and the second column holds the imaginary part.

If **job** = 'R', **vl** is not referenced.

3: **vr**(**ldvr**,$*$) **– double array**

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

if **job** = 'R' or 'B', **ldvr** $\geq \max(1, \mathbf{n})$;
if **job** = 'L', **ldvr** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** = 'R' or 'B' and at least 1 if **job** = 'L'

If **job** = 'R' or 'B', **vr** contains the computed right eigenvectors (as specified by **howmny** and **select**). The eigenvectors are stored consecutively in the columns of the array, in the same order as their eigenvalues. Corresponding to each real eigenvalue is a real eigenvector, occupying one column. Corresponding to each complex conjugate pair of eigenvalues, is a complex eigenvector occupying two columns; the first column holds the real part and the second column holds the imaginary part.

If **job** = 'L', **vr** is not referenced.

4: **m – int32 scalar**

m, the number of columns of **vl** and/or **vr** actually used to store the computed eigenvectors. If **howmny** = 'A', 'O' or 'B', **m** is set to n.

5: **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: **job**, 2: **howmny**, 3: **select**, 4: **n**, 5: **t**, 6: **ldt**, 7: **vl**, 8: **ldvl**, 9: **vr**, 10: **ldvr**, 11: **mm**, 12: **m**, 13: **work**, 14: **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

If x_i is an exact right eigenvector, and $\tilde{x}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{x}_i, x_i)$ between them is bounded as follows:

$$\theta(\tilde{x}_i, x_i) \leq \frac{c(n)\epsilon\|T\|_2}{sep_i}$$

where sep_i is the reciprocal condition number of x_i.

The condition number sep_i may be computed by calling f08ql.

8 Further Comments

For a description of canonical Schur form, see the document for f08pe.

The complex analogue of this function is f08qx.

9 Example

```
job = 'Right';
howmny = 'Backtransform';
select = [false];
t = [-0.4, 1.102642886197179, -10.75740652411425, -12.80868391133386;
     0, -4.020781350205953, -0.9760128758133151, 1.791715817556312;
     0, 0, 3.013557194374264, 0.506368981440549;
     0, 0, 0, 7.007224155831688];
vl = [0];
vr = [1, 0, 0, 0;
     0, 0.7058854989680637, -0.6936097630075586, 0.1436355074736275;
     0, -0.3465942790431883, -0.5150714469853064, -0.7839475813056603;
     0, -0.6177362447526779, -0.5035939844369282, 0.603982641106347];
mm = int32(4);
```

```
[selectOut, vlOut, vrOut, m, info] = f08qk(job, howmny, select, t, vl,
vr, mm)
```

```
selectOut =
     0
vlOut =
     0
vrOut =
    1.0000   -0.4314   -1.0000   -1.0000
         0    1.0000   -0.2477    0.0859
         0   -0.4910   -0.1461   -0.4769
         0   -0.8751   -0.1307    0.2362
m =
           4
info =
           0
```