

# NAG Toolbox for MATLAB

## f08yx

### 1 Purpose

f08yx computes some or all of the right and/or left generalized eigenvectors of a pair of complex upper triangular matrices  $(A, B)$ .

### 2 Syntax

```
[vl, vr, m, info] = f08yx(side, howmny, select, a, b, vl, vr, mm, 'n', n)
```

### 3 Description

f08yx computes some or all of the right and/or left generalized eigenvectors of the matrix pair  $(A, B)$  which is assumed to be in upper triangular form. If the matrix pair  $(A, B)$  is not upper triangular then the function f08xs should be called before invoking f08yx.

The right generalized eigenvector  $x$  and the left generalized eigenvector  $y$  of  $(A, B)$  corresponding to a generalized eigenvalue  $\lambda$  are defined by

$$(A - \lambda B)x = 0$$

and

$$y^H(A - \lambda B) = 0.$$

If a generalized eigenvalue is determined as 0/0, which is due to zero diagonal elements at the same locations in both  $A$  and  $B$ , a unit vector is returned as the corresponding eigenvector.

Note that the generalized eigenvalues are computed using f08xs but f08yx does not explicitly require the generalized eigenvalues to compute eigenvectors. The ordering of the eigenvectors is based on the ordering of the eigenvalues as computed by f08yx.

If all eigenvectors are requested, the function may either return the matrices  $X$  and/or  $Y$  of right or left eigenvectors of  $(A, B)$ , or the products  $ZX$  and/or  $QY$ , where  $Z$  and  $Q$  are two matrices supplied by you. Usually,  $Q$  and  $Z$  are chosen as the unitary matrices returned by f08xs. Equivalently,  $Q$  and  $Z$  are the left and right Schur vectors of the matrix pair supplied to f08xs. In that case,  $QY$  and  $ZX$  are the left and right generalized eigenvectors, respectively, of the matrix pair supplied to f08xs.

### 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

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

Moler C B and Stewart G W 1973 An algorithm for generalized matrix eigenproblems *SIAM J. Numer. Anal.* **10** 241–256

Stewart G W and Sun J-G 1990 *Matrix Perturbation Theory* Academic Press, London

## 5 Parameters

### 5.1 Compulsory Input Parameters

1: **side** – **string**

Specifies the required sets of generalized eigenvectors.

**side** = 'R'

Only right eigenvectors are computed.

**side** = 'L'

Only left eigenvectors are computed.

**side** = 'B'

Both left and right eigenvectors are computed.

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

2: **howmny** – **string**

Specifies further details of the required generalized eigenvectors.

**howmny** = 'A'

All right and/or left eigenvectors are computed.

**howmny** = 'B'

All right and/or left eigenvectors are computed; they are backtransformed using the input matrices supplied in arrays **vr** and/or **vl**.

**howmny** = 'S'

Selected right and/or left eigenvectors, defined by the array **select**, are computed.

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

3: **select**(\*) – **logical array**

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

Specifies the eigenvectors to be computed if **howmny** = 'S'. To select the generalized eigenvector corresponding to the  $j$ th generalized eigenvalue, the  $j$ th element of **select** should be set to **true**.

*Constraint:* **select**( $j$ ) = **true** or **false**, for  $j = 1, 2, \dots, n$ .

4: **a**(lda,\*) – **complex array**

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

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

The matrix  $A$  must be in upper triangular form. Usually, this is the matrix  $A$  returned by f08xs.

5: **b**(ldb,\*) – **complex array**

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

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

The matrix  $B$  must be in upper triangular form with nonnegative real diagonal elements. Usually, this is the matrix  $B$  returned by f08xs

6: **vl(ldvl,\*) – complex array**

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

if **side** = 'L' or 'B', **ldvl**  $\geq \max(1, n)$ ;  
 if **side** = 'R', **ldvl**  $\geq 1$ .

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

If **howmny** = 'B' and **side** = 'L' or 'B', **vl** must be initialized to an  $n$  by  $n$  matrix  $Q$ . Usually, this is the unitary matrix  $Q$  of left Schur vectors returned by f08xs.

7: **vr(ldvr,\*) – complex array**

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

if **side** = 'R' or 'B', **ldvr**  $\geq \max(1, n)$ ;  
 if **side** = 'L', **ldvr**  $\geq 1$ .

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

If **howmny** = 'B' and **side** = 'R' or 'B', **vr** must be initialized to an  $n$  by  $n$  matrix  $Z$ . Usually, this is the unitary matrix  $Z$  of right Schur vectors returned by f08xe.

8: **mm – int32 scalar**

The number of columns in the arrays **vl** and/or **vr**.

*Constraints:*

if **howmny** = 'A' or 'B', **mm**  $\geq n$ ;  
 if **howmny** = 'S', **mm** must not be less than the number of requested eigenvectors.

**5.2 Optional Input Parameters**1: **n – int32 scalar**

*Default:* The first dimension of the arrays **vl**, **vr**. (An error is raised if these dimensions are not equal.)

$n$ , the order of the matrices  $A$  and  $B$ .

*Constraint:*  $n \geq 0$ .

**5.3 Input Parameters Omitted from the MATLAB Interface**

lda, ldb, ldvl, ldvr, work, rwork

**5.4 Output Parameters**1: **vl(ldvl,\*) – complex array**

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

if **side** = 'L' or 'B', **ldvl**  $\geq \max(1, n)$ ;  
 if **side** = 'R', **ldvl**  $\geq 1$ .

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

If **side** = 'L' or 'B', **vl** contains:

if **howmny** = 'A', the matrix  $Y$  of left eigenvectors of  $(A, B)$ ;

if **howmny** = 'B', the matrix  $QY$ ;

if **howmny** = 'S', the left eigenvectors of  $(A, B)$  specified by **select**, stored consecutively in the columns of the array **vl**, in the same order as their corresponding eigenvalues.

2: **vr(ldvr,\*)** – complex array

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

if **side** = 'R' or 'B',  $\text{ldvr} \geq \max(1, \mathbf{n})$ ;

if **side** = 'L',  $\text{ldvr} \geq 1$ .

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

If **side** = 'R' or 'B', **vr** contains:

if **howmny** = 'A', the matrix  $X$  of right eigenvectors of  $(A, B)$ ;

if **howmny** = 'B', the matrix  $ZX$ ;

if **howmny** = 'S', the right eigenvectors of  $(A, B)$  specified by **select**, stored consecutively in the columns of the array **vr**, in the same order as their corresponding eigenvalues.

3: **m** – int32 scalar

The number of columns in the arrays **vl** and/or **vr** actually used to store the eigenvectors. If **howmny** = 'A' or 'B', **m** is set to **n**. Each selected eigenvector occupies one column.

4: **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: **side**, 2: **howmny**, 3: **select**, 4: **n**, 5: **a**, 6: **lda**, 7: **b**, 8: **ldb**, 9: **vl**, 10: **ldvl**, 11: **vr**, 12: **ldvr**, 13: **mm**, 14: **m**, 15: **work**, 16: **rwork**, 17: **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

It is beyond the scope of this manual to summarize the accuracy of the solution of the generalized eigenvalue problem. Interested readers should consult Section 4.11 of the LAPACK Users' Guide (see Anderson *et al.* 1999) and Chapter 6 of Stewart and Sun 1990.

## 8 Further Comments

f08yx is the sixth step in the solution of the complex generalized eigenvalue problem and is usually called after f08xs.

The real analogue of this function is f08yk.

## 9 Example

```

side = 'B';
howmny = 'B';
select = [false];
a = [complex(-0.1438233530385224, +0.3741062911974998),
complex(1.619781251864666, ...
+0.5366170936785172), complex(-5.73564259051686, -3.142957175229123),
...
complex(0.8798344594474423, +2.121055303267389);
complex(0, +0), complex(0.2823856561928724, +0.5208777880043725),
complex(0.8621110088889029, ...
-2.167166334598099), complex(-3.106018675330553,
+1.841506903478952);
complex(0, +0), complex(0, +0), complex(1.855074244893335, -
0.137336971784532), ...
complex(-1.098834831815077, -0.9880107323733596);
complex(0, +0), complex(0, +0), complex(0, +0),
complex(0.07048820152838613, -0.1296769105942374)];
b = [complex(0.2263375141379972, +0), complex(-0.002717330036079723, -
0.7423473646116365), ...
complex(-3.468409696159831, +0.1539788635359557),
complex(0.7560522313684904, +1.232848952545645);
complex(0, +0), complex(0.5722925376877426, +0),
complex(0.5726907354403078, ...
-2.024908875044398), complex(-3.138311880819062,
+1.830164861889475);
complex(0, +0), complex(0, +0), complex(2.750626071820657, +0),
complex(-1.220622047233075, -1.233359591775969);
complex(0, +0), complex(0, +0), complex(0, +0),
complex(0.1539006001169547, +0)];
vr = [complex(-0.7090128348336845, -0.4614766052957729), complex(-
0.03302144109782951, ...
+0.09031754537464898), complex(-0.1628796424652455, -
0.4924480172799654), ...
complex(0.07738808120167298, -0.008290082415805335);
complex(-0.4837138711442771, +0.01888381070268117),
complex(0.1585276550385831, ...
+0.2160986717612117), complex(0.6753888500844341,
+0.4579080836762112), ...
complex(-0.1663170609273306, +0.0185300441734911);
complex(0.01301179598327496, +0.04484513137596133),
complex(0.07439715940099263, ...
+0.2069093549321275), complex(0.1013875542126264,
+0.09091777594277473), ...
complex(0.9636186134385214, -0.04865057630715362);
complex(0.1975761880057718, +0.09374269385944185), complex(-
0.03434567875120977, ...
+0.9324144078370313), complex(-0.0529966554546879, -
0.2092394246122108), ...
complex(-0.1815547581349474, -0.04525252686036998)];
vr = [complex(0.1125541434588206, -0.2530015045809291), complex(-
0.8454652086827854, ...
-0.0439376337233772), complex(0.3778872609370834, -
0.1354773959002216), ...
complex(-0.1007041124568257, +0.1878447328930133);
complex(-0.1679732170998382, +0.1870876653951373),
complex(0.3287962269251304, ...
+0.1147271208604701), complex(0.8540175661380118, -
0.1668164345930881), ...
complex(-0.220993168644864, -0.09747936182932303);
complex(0.6207048630041609, -0.6731519289895047),
complex(0.2900654196264747, ...
+0.1373850965514047), complex(0.08053884907529973, -
0.06748539251590967), ...
complex(-0.08875151592027926, -0.1991525172615823);
complex(-0.08915551057292456, +0.1172529562072554), complex(-
0.2275026764742463, ...

```

```

        +0.08496301506044719),    complex(-0.2211484305422954,    -
0.1473972671191666), ...
        complex(-0.5421617631561396, -0.744818700993698)];
mm = int32(4);
[vlOut, vrOut, m, info] = f08yx(side, howmny, select, a, b, vl, vr, mm)

vlOut =
    -0.2725 - 0.1776i    0.0474 + 0.0490i    -0.1146 - 0.1935i    0.0765 -
0.0082i
     0.2762 + 0.0441i   -0.1435 - 0.0529i     0.3578 + 0.2103i   -0.1643 +
0.0183i
    -0.9540 - 0.0460i    0.8642 + 0.1358i   -0.6773 - 0.3227i    0.9519 -
0.0481i
     0.1276 - 0.0192i   -0.1641 + 0.0312i     0.0941 + 0.0343i   -0.1794 -
0.0447i
vrOut =
     0.0870 - 0.1955i    0.0550 + 0.0318i   -0.5392 - 0.2697i    0.0467 -
0.0597i
    -0.1298 + 0.1446i   -0.1060 - 0.0705i     0.6027 + 0.1760i   -0.0801 +
0.0956i
     0.4797 - 0.5203i    0.6392 + 0.3608i   -0.7265 - 0.2735i    0.5616 -
0.4384i
    -0.0689 + 0.0906i   -0.1388 - 0.0303i   -0.0422 - 0.0065i   -0.1293 +
0.0420i
m =
      4
info =
      0

```