NAG Toolbox for MATLAB

f08xp

1 Purpose

f08xp computes the generalized eigenvalues, the generalized Schur form (S, T) and, optionally, the left and/or right generalized Schur vectors for a pair of n by n complex nonsymmetric matrices (A, B).

Estimates of condition numbers for selected generalized eigenvalue clusters and Schur vectors are also computed.

2 Syntax

```
[a, b, sdim, alpha, beta, vsl, vsr, rconde, rcondv, info] =
f08xp(jobvsl, jobvsr, sort, selctg, sense, a, b, 'n', n)
```

3 Description

The generalized Schur factorization for a pair of complex matrices (A, B) is given by

$$A = QSZ^{\mathrm{H}}, \qquad B = QTZ^{\mathrm{H}},$$

where Q and Z are unitary, T and S are upper triangular. The generalized eigenvalues, λ, of (A, B) are computed from the diagonals of T and S and satisfy

$$Az = \lambda Bz,$$

where z is the corresponding generalized eigenvector. λ is actually returned as the pair (α, β) such that

$$\lambda = \alpha/\beta$$

since β, or even both α and β can be zero. The columns of Q and Z are the left and right generalized Schur vectors of (A, B).

Optionally, f08xn can order the generalized eigenvalues on the diagonals of (S, T) so that selected eigenvalues are at the top left. The leading columns of Q and Z then form an orthonormal basis for the corresponding eigenspaces, the deflating subspaces.

f08xn computes T to have real nonnegative diagonal entries. The generalized Schur factorization, before reordering, is computed by the QZ algorithm.

The reciprocals of the condition estimates, the reciprocal values of the left and right projection norms, are returned in **rconde**(1) and **rconde**(2) respectively, for the selected generalized eigenvalues, together with reciprocal condition estimates for the corresponding left and right deflating subspaces, in **rcondv**(1) and **rcondv**(2). See Section 4.11 of Anderson *et al.* 1999 for further information.

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

5 Parameters

5.1 Compulsory Input Parameters

1: **jobvsl – string**

If **jobvsl** = 'N', do not compute the left Schur vectors.

If **jobvsl** = 'V', compute the left Schur vectors.

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

2: **jobvsr – string**

If **jobvsr** = 'N', do not compute the right Schur vectors.

If **jobvsr** = 'V', compute the right Schur vectors.

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

3: **sort – string**

Specifies whether or not to order the eigenvalues on the diagonal of the generalized Schur form.

sort = 'N'

Eigenvalues are not ordered.

sort = 'S'

Eigenvalues are ordered (see user-supplied logical function **selctg**).

Constraint: **sort** = 'N' or 'S'.

4: **selctg – string containing name of m-file**

If **sort** = 'S', **selctg** is used to select generalized eigenvalues to the top left of the generalized Schur form.

If **sort** = 'N', **selctg** is not referenced and f08xp may be called with the string `'f08xnz'`.

Its specification is:

```
[result] = selctg(a, b)
```

Input Parameters

1: **a – complex scalar**

2: **b – complex scalar**

An eigenvalue $\mathbf{a}(j)/\mathbf{b}(j)$ is selected if $\mathbf{selctg}(\mathbf{a}(j), \mathbf{b}(j))$ is true.

Note that in the ill-conditioned case, a selected generalized eigenvalue may no longer satisfy $\mathbf{selctg}(\mathbf{a}(j), \mathbf{b}(j)) = \mathbf{true}$ after ordering. **info** is set to $\mathbf{n} + 2$ in this case. (See **info** below).

Output Parameters

1: **result – logical scalar**

The result of the function.

5: **sense – string**

Determines which reciprocal condition numbers are computed.

sense = 'N'

None are computed.

sense = 'E'

Computed for average of selected eigenvalues only.

sense = 'V'

Computed for selected deflating subspaces only.

sense = 'B'

Computed for both.

If **sense** = 'E', 'V' or 'B', **sort** must equal 'S'.

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

6: **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 first of the pair of matrices, A.

7: **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 second of the pair of matrices, B.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The first dimension of the arrays **a**, **b** and the second dimension of the arrays **a**, **b**. (An error is raised if these dimensions are not equal.)

n, the order of the matrices A and B.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldvsl, ldvsr, work, lwork, rwork, iwork, liwork, 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 by its generalized Schur form S.

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 by its generalized Schur form T.

3: **sdim – int32 scalar**

If **sort** = 'N', **sdim** = 0.

If **sort** = 'S', **sdim** = number of eigenvalues (after sorting) for which user-supplied logical function **selctg** is **true**.

4: **alpha**($*$) – **complex array**

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

See the description of **beta**.

5: **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. **alpha**(j) and **beta**(j), $j = 1, \ldots, \mathbf{n}$ are the diagonals of the complex Schur form (S, T). **beta**(j) will be nonnegative real.

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/β. However, **alpha** will always be less than and usually comparable with $\|\mathbf{a}\|$ in magnitude, and **beta** will always be less than and usually comparable with $\|\mathbf{b}\|$.

6: **vsl**(**ldvsl**,$*$) – **complex array**

The first dimension, **ldvsl**, of the array **vsl** must satisfy

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

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

If **jobvsl** = 'V', **vsl** will contain the left Schur vectors, Q.

If **jobvsl** = 'N', **vsl** is not referenced.

7: **vsr**(**ldvsr**,$*$) – **complex array**

The first dimension, **ldvsr**, of the array **vsr** must satisfy

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

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

If **jobvsr** = 'V', **vsr** will contain the right Schur vectors, Z.

If **jobvsr** = 'N', **vsr** is not referenced.

8: **rconde**(**2**) – **double array**

If **sense** = 'E' or 'B', **rconde**(1) and **rconde**(2) contain the reciprocal condition numbers for the average of the selected eigenvalues.

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

9: **rcondv**(**2**) – **double array**

If **sense** = 'V' or 'B', **rcondv**(1) and **rcondv**(2) contain the reciprocal condition numbers for the selected deflating subspaces.

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

10: **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: **jobvsl**, 2: **jobvsr**, 3: **sort**, 4: **selctg**, 5: **sense**, 6: **n**, 7: **a**, 8: **lda**, 9: **b**, 10: **ldb**, 11: **sdim**, 12: **alpha**, 13: **beta**, 14: **vsl**, 15: **ldvsl**, 16: **vsr**, 17: **ldvsr**, 18: **rconde**, 19: **rcondv**, 20: **work**, 21: **lwork**, 22: **rwork**, 23: **iwork**, 24: **liwork**, 25: **bwork**, 26: **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. (A, B) are not in Schur form, but **alpha**(j) and **beta**(j) should be correct for $j =$ **info** $+ 1, \ldots,$ **n**.

info $= N + 1$

Unexpected error returned from f08xs.

info $= N + 2$

After reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the generalized Schur form no longer satisfy **selctg** $=$ **true**. This could also be caused by underflow due to scaling.

info $= N + 3$

The eigenvalues could not be reordered because some eigenvalues were too close to separate (the problem is very ill-conditioned).

7 Accuracy

The computed generalized Schur factorization satisfies

$$A + E = QSZ^{\mathrm{T}}, \qquad B + F = QTZ^{\mathrm{T}},$$

where

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

and ϵ is the ***machine precision***. See Section 4.11 of Anderson *et al.* 1999 for further details.

8 Further Comments

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

The real analogue of this function is f08xb.

9 Example

```
f08xp_selctg.m
```

```
function [result] = selctg(a, b)

  if (abs(a) < 6*abs(b))
    result = true;
  else
```

```
    result = false;
  end
```

```
jobvsl = 'Vectors (left)';
jobvsr = 'Vectors (right)';
sort = 'Sort';
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, sdim, alpha, beta, vsl, vsr, rconde, rcondv, info] = ...
    f08xp(jobvsl, jobvsr, sort, 'f08xp_selctg', sense, a, b)
```

```
aOut =
   1.0e+02 *
    0.1070 - 0.2674i   -0.7269 - 0.1571i   -1.2235 - 0.1408i    0.9900 -
0.3874i
         0              0.1101 - 0.0367i    0.0422 + 0.3157i   -0.1903 -
0.3856i
         0                   0              0.2104 - 0.6313i    0.1256 +
0.3220i
         0                   0                   0              0.2187 -
0.2734i
bOut =
    5.3490             -0.1267 - 1.0111i   -1.1949 - 3.2556i    4.4160 +
1.9056i
         0             3.6701              -1.9413 + 2.2087i    2.8993 -
6.1707i
         0                   0             7.0147              -2.6653 +
4.8354i
        0                  0                  0             5.4681
sdim =
           2
alpha =
  10.6980 -26.7450i
  11.0103 - 3.6701i
  21.0441 -63.1323i
  21.8722 -27.3403i
beta =
    5.3490
    3.6701
    7.0147
    5.4681
vsl =
   -0.3733 + 0.8687i    0.2117 - 0.1177i   -0.2156 + 0.0104i    0.0144 -
0.0212i
   -0.1606 + 0.0762i   -0.7130 - 0.5203i    0.1568 - 0.3985i   -0.0087 -
0.0767i
   -0.1864 + 0.0164i   -0.2349 + 0.0826i    0.2003 + 0.6054i   -0.1464 -
0.6892i
   -0.0137 - 0.1978i    0.0473 - 0.3131i   -0.5982 + 0.0746i   -0.7049 -
0.0133i
vsr =
  -0.9697 - 0.2276i   0.0340 + 0.0612i   0.0530 - 0.0126i        0
   -0.0052 + 0.0023i    0.0189 - 0.6299i    0.7066 + 0.3218i   -0.0000 -
0.0000i
   -0.0610 - 0.0143i   -0.2882 - 0.4647i   -0.4385 + 0.0694i    0.7034 -
```

```
0.0728i
    0.0143 - 0.0610i    0.4647 - 0.2882i   -0.0694 - 0.4385i   -0.0728 -
0.7034i
rconde =
    0.1205
    0.1625
rcondv =
    0.4770
    0.4690
info =
           0
```