NAG Toolbox for MATLAB

f08xa

1 Purpose

f08xa computes the generalized eigenvalues, the generalized real Schur form (S, T) and, optionally, the left and/or right generalized Schur vectors for a pair of n by n real nonsymmetric matrices (A, B).

2 Syntax

```
[a, b, sdim, alphar, alphai, beta, vsl, vsr, info] = f08xa(jobvsl,
jobvsr, sort, selctg, a, b, 'n', n)
```

3 Description

The generalized real Schur factorization of (A, B) is given by

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

where Q and Z are orthogonal, T is upper triangular and S is upper quasi-triangular with 1 by 1 and 2 by 2 diagonal blocks. The generalized eigenvalues, λ, of (A, B) are computed from the diagonals of S and T 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, f08xa 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.

f08xa computes T to have nonnegative diagonal elements, and the 2 by 2 blocks of S correspond to complex conjugate pairs of generalized eigenvalues. The generalized Schur factorization, before reordering, is computed by the QZ algorithm.

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 f08xa may be called with the string `'f08xaz'`.

Its specification is:

```
[result] = selctg(ar, ai, b)
```

Input Parameters

1: **ar – double scalar**
2: **ai – double scalar**
3: **b – double scalar**

An eigenvalue $\left(\mathbf{ar}(j) + \sqrt{-1} \times \mathbf{ai}(j)\right)/\mathbf{b}(j)$ is selected if $\mathbf{selctg}(\mathbf{ar}(j), \mathbf{ai}(j), \mathbf{b}(j))$ is **true**.
If either one of a complex conjugate pair is selected, then both complex generalized eigenvalues are selected.

Note that in the ill-conditioned case, a selected complex generalized eigenvalue may no longer satisfy $\mathbf{selctg}(\mathbf{ar}(j), \mathbf{ai}(j), \mathbf{b}(j)) =$ **true** after ordering. **info** = Np2 in this case.

Output Parameters

1: **result – logical scalar**

The result of the function.

5: **a(lda,*) – double 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.

6: **b(ldb,*) – double 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.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldvsl, ldvsr, work, lwork, bwork

5.4 Output Parameters

1: **a(lda,$*$) – double 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,$*$) – double 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**. (Complex conjugate pairs for which **selctg** is **true** for either eigenvalue count as 2.)

4: **alphar($*$) – double array**

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

See the description of **beta**.

5: **alphai($*$) – double array**

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

See the description of **beta**.

6: **beta($*$) – double array**

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

$(\mathbf{alphar}(j) + \mathbf{alphai}(j) \times i)/\mathbf{beta}(j)$, for $j = 1, \dots, \mathbf{n}$, will be the generalized eigenvalues. $\mathbf{alphar}(j) + \mathbf{alphai}(j) \times i$, and $\mathbf{beta}(j)$, for $j = 1, \dots, \mathbf{n}$, are the diagonals of the complex Schur form (S, T) that would result if the 2 by 2 diagonal blocks of the real Schur form of (A, B) were further reduced to triangular form using 2 by 2 complex unitary transformations.

If $\mathbf{alphai}(j)$ is zero, then the jth eigenvalue is real; if positive, then the jth and $(j+1)$st eigenvalues are a complex conjugate pair, with $\mathbf{alphai}(j+1)$ negative.

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

7: **vsl(ldvsl,$*$) – double 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.

8: **vsr(ldvsr,$*$) – double 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.

9: **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: **n**, 6: **a**, 7: **lda**, 8: **b**, 9: **ldb**, 10: **sdim**, 11: **alphar**, 12: **alphai**, 13: **beta**, 14: **vsl**, 15: **ldvsl**, 16: **vsr**, 17: **ldvsr**, 18: **work**, 19: **lwork**, 20: **bwork**, 21: **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 **alphar**(j), **alphai**(j), and **beta**(j) should be correct for $j = \mathbf{info} + 1, \ldots, \mathbf{n}$.

info $= N + 1$

Unexpected error returned from f08xe.

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 complex analogue of this function is f08xn.

9 Example

```
f08xa_selctg.m

function [result] = selctg(ar, ai, b)
  if (ai == 0)
    result = true;
  else
    result = false;
  end
```

```
jobvsl = 'Vectors (left)';
jobvsr = 'Vectors (right)';
sort = 'Sort';
a = [3.9, 12.5, -34.5, -0.5;
     4.3, 21.5, -47.5, 7.5;
     4.3, 21.5, -43.5, 3.5;
     4.4, 26, -46, 6];
b = [1, 2, -3, 1;
     1, 3, -5, 4;
     1, 3, -4, 3;
     1, 3, -4, 4];
[aOut, bOut, sdim, alphar, alphai, beta, vsl, vsr, info] = ...
    f08xa(jobvsl, jobvsr, sort, 'f08xa_selctg', a, b)
```

```
aOut =
    3.8009  -69.4505   50.3135  -43.2884
         0    9.2033   -0.2001    5.9881
         0         0    1.4279    4.4453
         0         0    0.9019   -1.1962
bOut =
    1.9005  -10.2285    0.8658   -5.2134
         0    2.3008    0.7915    0.4262
         0         0    0.8101         0
         0         0         0   -0.2823
sdim =
           2
alphar =
    3.8009
    9.2033
    0.8571
    0.8571
alphai =
         0
         0
    1.1429
```

```
   -1.1429
beta =
    1.9005
    2.3008
    0.2857
    0.2857
vsl =
    0.4642    0.7886    0.2915   -0.2786
    0.5002   -0.5986    0.5638   -0.2713
    0.5002    0.0154   -0.0107    0.8657
    0.5331   -0.1395   -0.7727   -0.3151
vsr =
    0.9961   -0.0014    0.0887   -0.0026
    0.0057   -0.0404   -0.0938   -0.9948
    0.0626    0.7194   -0.6908    0.0363
    0.0626   -0.6934   -0.7114    0.0956
info =
           0
```