NAG Toolbox for MATLAB

f08yg

1 Purpose

f08yg reorders the generalized Schur factorization of a matrix pair in real generalized Schur form, so that a selected cluster of eigenvalues appears in the leading elements, or blocks on the diagonal of the generalized Schur form. The function also, optionally, computes the reciprocal condition numbers of the cluster of eigenvalues and/or corresponding deflating subspaces.

2 Syntax

```
[a, b, alphar, alphai, beta, q, z, m, pl, pr, dif, info] = f08yg(ijob,
wantq, wantz, select, a, b, q, z, 'n', n)
```

3 Description

f08yg factorizes the generalized real n by n matrix pair (S, T) in real generalized Schur form, using an orthogonal equivalence transformation as

$$S = \hat{Q}\hat{S}\hat{Z}^{\mathrm{T}}, \qquad T = \hat{Q}\hat{T}\hat{Z}^{\mathrm{T}},$$

where $\left(\hat{S}, \hat{T}\right)$ are also in real generalized Schur form and have the selected eigenvalues as the leading diagonal elements, or diagonal blocks. The leading columns of Q and Z are the generalized Schur vectors corresponding to the selected eigenvalues and form orthonormal subspaces for the left and right eigenspaces (deflating subspaces) of the pair (S, T).

The pair (S, T) are in real generalized Schur form if S is block upper triangular with 1 by 1 and 2 by 2 diagonal blocks and T is upper triangular as returned, for example, by f08xa, or f08xe with **job** = 'S'. The diagonal elements, or blocks, define the generalized eigenvalues (α_i, β_i), for $i = 1, 2, \ldots, n$ of the pair (S, T). The eigenvalues are given by

$$\lambda_i = \alpha_i / \beta_i,$$

but are returned as the pair (α_i, β_i) in order to avoid possible overflow in computing λ_i. Optionally, the function returns reciprocals of condition number estimates for the selected eigenvalue cluster, p and q, the right and left projection norms, and of deflating subspaces, Dif_u and Dif_l. For more information see Sections 2.4.8 and 4.11 of Anderson *et al.* 1999.

If S and T are the result of a generalized Schur factorization of a matrix pair (A, B)

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

then, optionally, the matrices Q and Z can be updated as $Q\hat{Q}$ and $Z\hat{Z}$. Note that the condition numbers of the pair (S, T) are the same as those of the pair (A, B).

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

5 Parameters

5.1 Compulsory Input Parameters

1: **ijob – int32 scalar**

Specifies whether condition numbers are required for the cluster of eigenvalues (p and q) or the deflating subspaces (Dif_u and Dif_l).

ijob $= 0$

Only reorder with respect to **select**. No extras.

ijob $= 1$

Reciprocal of norms of 'projections' onto left and right eigenspaces with respect to the selected cluster (p and q).

ijob $= 2$

The upper bounds on Dif_u and Dif_l. F-norm-based estimate (**dif**$(1:2)$).

ijob $= 3$

Estimate of Dif_u and Dif_l. 1-norm-based estimate (**dif**$(1:2)$). About five times as expensive as **ijob** $= 2$.

ijob $= 4$

Compute **pl**, **pr** and **dif** as in **ijob** $= 0$, 1 and 2. Economic version to get it all.

ijob $= 5$

Compute **pl**, **pr** and **dif** as in **ijob** $= 0$, 1 and 3.

2: **wantq – logical scalar**

If **wantq** $=$ **true**, update the left transformation matrix Q.

If **wantq** $=$ **false**, do not update Q.

3: **wantz – logical scalar**

If **wantz** $=$ **true**, update the right transformation matrix Z.

If **wantz** $=$ **false**, do not update Z.

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

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

Specifies the eigenvalues in the selected cluster.

To select a real eigenvalue λ_j, **select**(j) must be set to **true**.

To select a complex conjugate pair of eigenvalues λ_j and λ_{j+1}, corresponding to a 2 by 2 diagonal block, either **select**(j) or **select**$(j+1)$ or both must be set to **true**; a complex conjugate pair of eigenvalues must be either both included in the cluster or both excluded.

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 matrix S in the pair (S, T).

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 matrix T, in the pair (S, T).

7: **q(ldq,$*$) – double array**

The first dimension, **ldq**, of the array **q** must satisfy

if **wantq** $=$ **true**, **ldq** $\geq \max(1, \mathbf{n})$;
ldq ≥ 1 otherwise.

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

If **wantq** $=$ **true**, the n by n matrix Q.

8: **z(ldz,$*$) – double array**

The first dimension, **ldz**, of the array **z** must satisfy

if **wantz** $=$ **true**, **ldz** $\geq \max(1, \mathbf{n})$;
ldz ≥ 1 otherwise.

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

If **wantz** $=$ **true**, the n by n matrix Z.

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 S and T.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldq, ldz, work, lwork, iwork, liwork

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 $\max(1, \mathbf{n})$

The updated matrix $\hat{S}$.

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 $\max(1, \mathbf{n})$

The updated matrix $\hat{T}$

3: **alphar($*$) – double array**

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

See the description of **beta**.

4: **alphai($*$) – double array**

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

See the description of **beta**.

5: **beta**($*$) – **double array**

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

alphar(j)/**beta**(j) and **alphai**(j)/**beta**(j) are the real and imaginary parts respectively of the jth eigenvalue, for $j = 1, \ldots, \mathbf{n}$.

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

Conjugate pairs of eigenvalues correspond to the 2 by 2 diagonal blocks of $\hat{S}$. These 2 by 2 blocks can be reduced by applying complex unitary transformations to $(\hat{S}, \hat{T})$ to obtain the complex Schur form $(\tilde{S}, \tilde{T})$, where $\tilde{S}$ is triangular (and complex). In this form **alphar** + i**alphai** and **beta** are the diagonals of $\tilde{S}$ and $\tilde{T}$ respectively.

6: **q**(**ldq**,$*$) – **double array**

The first dimension, **ldq**, of the array **q** must satisfy

if **wantq** = **true**, **ldq** $\geq \max(1, \mathbf{n})$;
ldq ≥ 1 otherwise.

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

If **wantq** = **true**, the updated matrix $Q\hat{Q}$.

If **wantq** = **false**, **q** is not referenced.

7: **z**(**ldz**,$*$) – **double array**

The first dimension, **ldz**, of the array **z** must satisfy

if **wantz** = **true**, **ldz** $\geq \max(1, \mathbf{n})$;
ldz ≥ 1 otherwise.

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

If **wantz** = **true**, the updated matrix $Z\hat{Z}$.

If **wantz** = **false**, **z** is not referenced.

8: **m** – **int32 scalar**

The dimension of the specified pair of left and right eigenspaces (deflating subspaces).

9: **pl** – **double scalar**
10: **pr** – **double scalar**

If **ijob** = 1, 4 or 5, **pl** and **pr** are lower bounds on the reciprocal of the norm of 'projections' p and q onto left and right eigenspaces with respect to the selected cluster. $0 <$ **pl**, **pr** ≤ 1.

If **m** = 0 or **m** = **n**, **pl** = **pr** = 1.

If **ijob** = 0, 2 or 3, **pl** and **pr** are not referenced.

11: **dif**($*$) – **double array**

Note: the dimension of the array **dif** must be at least 2.

If **ijob** ≥ 2, **dif**$(1:2)$ store the estimates of Dif_u and Dif_l.

If **ijob** = 2 or 4, **dif**$(1:2)$ are F-norm-based upper bounds on Dif_u and Dif_l.

If **ijob** = 3 or 5, **dif**$(1:2)$ are 1-norm-based estimates of Dif_u and Dif_l.

If **m** = 0 or n, **dif**$(1:2) = \|(A, B)\|_F$.

If **ijob** $= 0$ or 1, **dif** is not referenced.

12: **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: **ijob**, 2: **wantq**, 3: **wantz**, 4: **select**, 5: **n**, 6: **a**, 7: **lda**, 8: **b**, 9: **ldb**, 10: **alphar**, 11: **alphai**, 12: **beta**, 13: **q**, 14: **ldq**, 15: **z**, 16: **ldz**, 17: **m**, 18: **pl**, 19: **pr**, 20: **dif**, 21: **work**, 22: **lwork**, 23: **iwork**, 24: **liwork**, 25: **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 $= 1$

Reordering of (S, T) failed because the transformed matrix pair $\left(\hat{S}, \hat{T}\right)$ would be too far from generalized Schur form; the problem is very ill-conditioned. (S, T) may have been partially reordered. If requested, 0 is returned in **dif**$(1:2)$, **pl** and **pr**.

7 Accuracy

The computed generalized Schur form is nearly the exact generalized Schur form for nearby matrices $(S + E)$ and $(T + F)$, where

$$\|E\|_2 = O\,\epsilon\|S\|_2 \qquad \text{and} \qquad \|F\|_2 = O\,\epsilon\|T\|_2,$$

and ϵ is the ***machine precision***. See Section 4.11 of Anderson *et al.* 1999 for further details of error bounds for the generalized nonsymmetric eigenproblem, and for information on the condition numbers returned.

8 Further Comments

The complex analogue of this function is f08yu.

9 Example

```
ijob = int32(4);
wantq = true;
wantz = true;
select = [true;
     false;
     false;
     true];
a = [4, 1, 1, 2;
     0, 3, 4, 1;
     0, 1, 3, 1;
     0, 0, 0, 6];
b = [2, 1, 1, 3;
     0, 1, 2, 1;
     0, 0, 1, 1;
     0, 0, 0, 2];
q = [1, 0, 0, 0;
```

```
     0, 1, 0, 0;
     0, 0, 1, 0;
     0, 0, 0, 1];
z = [1, 0, 0, 0;
     0, 1, 0, 0;
     0, 0, 1, 0;
     0, 0, 0, 1];
[aOut, bOut, alphar, alphai, beta, qOut, zOut, m, pl, pr, dif, info] =
...
    f08yg(ijob, wantq, wantz, select, a, b, q, z)
```

```
aOut =
    4.0000    1.2247   -1.7055   -1.2615
         0    2.7386   -3.4009   -4.4423
         0         0    4.9328   -2.4277
         0         0   -0.9368   -1.7597
bOut =
    2.0000    1.6330   -1.9307   -2.1461
         0    0.9129   -1.4726   -1.7315
         0         0    2.2471         0
         0         0         0   -0.9750
alphar =
    4.0000
    2.7386
    1.3333
    1.3333
alphai =
         0
         0
    0.6667
   -0.6667
beta =
    2.0000
    0.9129
    0.6667
    0.6667
qOut =
    1.0000         0         0         0
         0    0.4472   -0.7715   -0.4526
         0    0.0000   -0.5060    0.8625
         0    0.8944    0.3857    0.2263
zOut =
    1.0000         0         0         0
         0    0.8165   -0.3433    0.4642
         0   -0.4082   -0.9118    0.0437
         0    0.4082   -0.2252   -0.8847
m =
           2
pl =
    0.3714
pr =
    0.6667
dif =
    0.2523
    0.2451
info =
           0
```