NAG Toolbox for MATLAB

f08yt

1 Purpose

f08yt reorders the generalized Schur factorization of a complex matrix pair in generalized Schur form.

2 Syntax

```
[a, b, q, z, ilst, info] = f08yt(wantq, wantz, a, b, q, z, ifst, ilst, 'n', n)
```

3 Description

f08yt reorders the generalized complex n by n matrix pair (S,T) in generalized Schur form, so that the diagonal element of (S,T) with row index i_1 is moved to row i_2, using a unitary equivalence transformation. That is, S and T are factorized as

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

where $\left(\hat{S},\hat{T}\right)$ are also in generalized Schur form.

The pair (S,T) are in generalized Schur form if S and T are upper triangular as returned, for example, by f08xn, or f08xs with **job** = 'S'.

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

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

then, optionally, the matrices Q and Z can be updated as $Q\hat{Q}$ and $Z\hat{Z}$.

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: **wantq – logical scalar**

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

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

2: **wantz – logical scalar**

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

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

3: **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 matrix S in the pair (S,T).

4: **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 matrix T, in the pair (S,T).

5: **q**(**ldq**,$*$) – **complex 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**, the unitary matrix Q.

6: **z**(**ldz**,$*$) – **complex 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**, the unitary matrix Z.

7: **ifst** – **int32 scalar**

8: **ilst** – **int32 scalar**

The indices i_1 and i_2 that specify the reordering of the diagonal elements of (S,T). The element with row index **ifst** is moved to row **ilst**, by a sequence of swapping between adjacent diagonal elements.

Constraint: $1 \leq \mathbf{ifst} \leq \mathbf{n}, 1 \leq \mathbf{ilst} \leq \mathbf{n}$.

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

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

The updated matrix $\hat{S}$.

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

The updated matrix $\hat{T}$

3: **q**(**ldq**,∗) – **complex 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**, the updated matrix $Q\hat{Q}$.

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

4: **z**(**ldz**,∗) – **complex 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**, the updated matrix $Z\hat{Z}$.

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

5: **ilst – int32 scalar**

ilst points to the row in its final position.

6: **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: **wantq**, 2: **wantz**, 3: **n**, 4: **a**, 5: **lda**, 6: **b**, 7: **ldb**, 8: **q**, 9: **ldq**, 10: **z**, 11: **ldz**, 12: **ifst**, 13: **ilst**, 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.

info = 1

The transformed matrix pair $(\hat{S}, \hat{T})$ would be too far from generalized Schur form; the problem is ill-conditioned. (S, T) may have been partially reordered, and **ilst** points to the first row of the current position of the block being moved.

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.

8 Further Comments

The real analogue of this function is f08yf.

9 Example

```
wantq = false;
wantz = false;
a = [complex(4, +4), complex(1, +1), complex(1, +1), complex(2, -1);
     complex(0, +0), complex(2, +1), complex(1, +1), complex(1, +1);
     complex(0, +0), complex(0, +0), complex(2, -1), complex(1, +1);
     complex(0, +0), complex(0, +0), complex(0, +0), complex(6, -2)];
b = [complex(2, +0), complex(1, +1), complex(1, +1), complex(3, -1);
     complex(0, +0), complex(1, +0), complex(2, +1), complex(1, +1);
     complex(0, +0), complex(0, +0), complex(1, +0), complex(1, +1);
     complex(0, +0), complex(0, +0), complex(0, +0), complex(2, +0)];
q = complex(zeros(1, 4));
z = complex(zeros(1, 4));
ifst = int32(1);
ilst = int32(4);
[aOut, bOut, qOut, zOut, ilstOut, info] = f08yt(wantq, wantz, a, b, q, z,
ifst, ilst)
```

```
aOut =
    3.7081 + 3.7081i   -2.0834 - 0.5688i    2.6374 + 1.0772i    0.2845 +
0.7991i
         0                1.6097 + 1.5656i   -0.0634 + 1.9234i   -0.0301 +
0.9720i
         0                     0              4.7029 - 2.1187i    1.1379 -
3.1199i
         0                     0                   0              2.3085 -
1.8289i
bOut =
    2.2249 + 0.7416i   -1.1631 + 1.5347i    2.2608 + 2.0851i    1.1094 -
0.3205i
         0                0.3308 + 0.9482i    0.3919 + 1.8172i   -0.6305 +
1.6053i
         0                     0              1.6227 - 0.1653i    0.9966 -
0.9074i
         0                     0                   0              0.1199 -
1.0343i
qOut =
     0     0     0     0
zOut =
     0     0     0     0
ilstOut =
           3
info =
           0
```