

NAG Toolbox

nag_lapack_ztgexc (f08yt)

1 Purpose

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

2 Syntax

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

3 Description

nag_lapack_ztgexc (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

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

where (\hat{S}, \hat{T}) 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 nag_lapack_zgges (f08xn), or nag_lapack_zhgeqz (f08xs) with **job** = 'S'.

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

$$A = QSZ^H, \quad B = QTZ^H$$

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 <http://www.netlib.org/lapack/lug>

5 Parameters

5.1 Compulsory Input Parameters

1: **wantq** – LOGICAL

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

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

2: **wantz** – LOGICAL

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

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

3: **a**(*lda*, :) – COMPLEX (KIND=nag_wp) array

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

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

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

4: **b**($ldb, :)$ – COMPLEX (KIND=nag_wp) array

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

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

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

5: **q**($ldq, :)$ – COMPLEX (KIND=nag_wp) array

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

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

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

If **wantq** = true, the unitary matrix Q .

6: **z**($ldz, :)$ – COMPLEX (KIND=nag_wp) array

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

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

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

If **wantz** = true, the unitary matrix Z .

7: **ifst** – INTEGER

8: **ilst** – INTEGER

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

5.2 Optional Input Parameters

1: **n** – INTEGER

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 Output Parameters

1: **a**($lda, :)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **a** will be $\max(1, \mathbf{n})$.

The second dimension of the array **a** will be $\max(1, \mathbf{n})$.

The updated matrix \hat{S} .

- 2: **b**(*ldb*,:) – COMPLEX (KIND=nag_wp) array
 The first dimension of the array **b** will be max(1, **n**).
 The second dimension of the array **b** will be max(1, **n**).
 The updated matrix \hat{T}
- 3: **q**(*ldq*,:) – COMPLEX (KIND=nag_wp) array
 The first dimension, *ldq*, of the array **q** will be
 if **wantq** = true, *ldq* = max(1, **n**);
 otherwise *ldq* = 1.
 The second dimension of the array **q** will be max(1, **n**) if **wantq** = true and 1 otherwise.
 If **wantq** = true, the updated matrix $Q\hat{Q}$.
 If **wantq** = false, **q** is not referenced.
- 4: **z**(*ldz*,:) – COMPLEX (KIND=nag_wp) array
 The first dimension, *ldz*, of the array **z** will be
 if **wantz** = true, *ldz* = max(1, **n**);
 otherwise *ldz* = 1.
 The second dimension of the array **z** will be max(1, **n**) if **wantz** = true and 1 otherwise.
 If **wantz** = true, the updated matrix $Z\hat{Z}$.
 If **wantz** = false, **z** is not referenced.
- 5: **ilst** – INTEGER
 `ilst` points to the row in its final position.
- 6: **info** – INTEGER
 info = 0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

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

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

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 nag_lapack_dtgexc (f08yf).

9 Example

This example exchanges rows 4 and 1 of the matrix pair (S, T) , where

$$S = \begin{pmatrix} 4.0 + 4.0i & 1.0 + 1.0i & 1.0 + 1.0i & 2.0 - 1.0i \\ 0 & 2.0 + 1.0i & 1.0 + 1.0i & 1.0 + 1.0i \\ 0 & 0 & 2.0 - 1.0i & 1.0 + 1.0i \\ 0 & 0 & 0 & 6.0 - 2.0i \end{pmatrix}$$

and

$$T = \begin{pmatrix} 2.0 & 1.0 + 1.0i & 1.0 + 1.0i & 3.0 - 1.0i \\ 0 & 1.0 & 2.0 + 1.0i & 1.0 + 1.0i \\ 0 & 0 & 1.0 & 1.0 + 1.0i \\ 0 & 0 & 0 & 2.0 \end{pmatrix}.$$

9.1 Program Text

```
function f08yt_example

fprintf('f08yt example results\n\n');

% exchanges rows 4 and 1 of the matrix pair S,T, where
s = [ 4 + 4i, 1 + 1i, 1 + 1i, 2 - 1i;
      0 + 0i, 2 + 1i, 1 + 1i, 1 + 1i;
      0 + 0i, 0 + 0i, 2 - 1i, 1 + 1i;
      0 + 0i, 0 + 0i, 0 + 0i, 6 - 2i];
t = [ 2, 1 + 1i, 1 + 1i, 3 - 1i;
      0 + 0i, 1 + 0i, 2 + 1i, 1 + 1i;
      0 + 0i, 0 + 0i, 1 + 0i, 1 + 1i;
      0 + 0i, 0 + 0i, 0 + 0i, 2 + 0i];

wantq = false;
wantz = false;
q = complex(zeros(1, 4));
z = complex(zeros(1, 4));
ifst = nag_int(1);
ilst = nag_int(4);
[s, t, q, z, ilst, info] = ...
    f08yt( ...
        wantq, wantz, s, t, q, z, ifst, ilst);

disp('Reordered Schur matrix S');
disp(s);
disp('Reordered Schur matrix T');
disp(t);
```

9.2 Program Results

f08yt example results

```
Reordered Schur matrix S
 3.7081 + 3.7081i -2.0834 - 0.5688i  2.6374 + 1.0772i  0.2845 + 0.7991i
 0.0000 + 0.0000i  1.6097 + 1.5656i -0.0634 + 1.9234i -0.0301 + 0.9720i
 0.0000 + 0.0000i  0.0000 + 0.0000i  4.7029 - 2.1187i  1.1379 - 3.1199i
 0.0000 + 0.0000i  0.0000 + 0.0000i  0.0000 + 0.0000i  2.3085 - 1.8289i
```

```
Reordered Schur matrix T
 2.2249 + 0.7416i -1.1631 + 1.5347i  2.2608 + 2.0851i  1.1094 - 0.3205i
 0.0000 + 0.0000i  0.3308 + 0.9482i  0.3919 + 1.8172i -0.6305 + 1.6053i
 0.0000 + 0.0000i  0.0000 + 0.0000i  1.6227 - 0.1653i  0.9966 - 0.9074i
 0.0000 + 0.0000i  0.0000 + 0.0000i  0.0000 + 0.0000i  0.1199 - 1.0343i
```
