

NAG Toolbox for MATLAB

f08zt

1 Purpose

f08zt computes a generalized RQ factorization of a complex matrix pair (A, B) , where A is an m by n matrix and B is a p by n matrix.

2 Syntax

```
[a, taua, b, taub, info] = f08zt(a, b, 'm', m, 'p', p, 'n', n)
```

3 Description

f08zt forms the generalized RQ factorization of an m by n matrix A and a p by n matrix B

$$A = RQ, \quad B = ZTQ,$$

where Q is an n by n unitary matrix, Z is a p by p unitary matrix and R and T are of the form

$$R = \begin{cases} m \begin{pmatrix} n-m & m \\ 0 & R_{12} \end{pmatrix}; & \text{if } m \leq n, \\ m-n \begin{pmatrix} n \\ R_{11} \\ n \end{pmatrix} \begin{pmatrix} R_{21} \end{pmatrix}; & \text{if } m > n, \end{cases}$$

with R_{12} or R_{21} upper triangular,

$$T = \begin{cases} p-n \begin{pmatrix} n \\ T_{11} \\ 0 \end{pmatrix}; & \text{if } p \geq n, \\ p \begin{pmatrix} p & n-p \\ T_{11} & T_{12} \end{pmatrix}; & \text{if } p < n, \end{cases}$$

with T_{11} upper triangular.

In particular, if B is square and nonsingular, the generalized RQ factorization of A and B implicitly gives the RQ factorization of AB^{-1} as

$$AB^{-1} = (RT^{-1})Z^H.$$

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>

Anderson E, Bai Z and Dongarra J 1992 Generalized QR factorization and its applications *Linear Algebra Appl. (Volume 162–164)* 243–271

Hammarling S 1987 The numerical solution of the general Gauss-Markov linear model *Mathematics in Signal Processing* (ed T S Durrani, J B Abbiss, J E Hudson, R N Madan, J G McWhirter, and T A Moore) 441–456 Oxford University Press

Paige C C 1990 Some aspects of generalized QR factorizations . *In Reliable Numerical Computation* (ed M G Cox and S Hammarling) 73–91 Oxford University Press

5 Parameters

5.1 Compulsory Input Parameters

1: **a(lda,*) – complex array**

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

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

The m by n matrix A .

2: **b(ldb,*) – complex array**

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

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

The p by n matrix B .

5.2 Optional Input Parameters

1: **m – int32 scalar**

Default: The first dimension of the array **a**.

m , the number of rows of the matrix A .

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

2: **p – int32 scalar**

Default: The first dimension of the array **b**.

p , the number of rows of the matrix B .

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

3: **n – int32 scalar**

Default: The second dimension of the array **a**.

n , the number of columns of the matrices A and B .

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, work, lwork

5.4 Output Parameters

1: **a(lda,*) – complex array**

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

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

If $m \leq n$, the upper triangle of the subarray **a**(1 : m , $n - m + 1 : n$) contains the m by m upper triangular matrix R_{12} .

If $m \geq n$, the elements on and above the $(m - n)$ th subdiagonal contain the m by n upper trapezoidal matrix R ; the remaining elements, with the array **tauau**, represent the unitary matrix Q as a product of $\min(m, n)$ elementary reflectors (see Section 3.2.6 in the F08 Chapter Introduction).

2: **taua(*)** – complex array

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

The scalar factors of the elementary reflectors which represent the unitary matrix Q .

3: **b(ldb,*)** – complex array

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

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

The elements on and above the diagonal of the array contain the $\min(p, n)$ by n upper trapezoidal matrix T (T is upper triangular if $p \geq n$); the elements below the diagonal, with the array **taub**, represent the unitary matrix Z as a product of elementary reflectors (see Section 3.2.6 in the F08 Chapter Introduction).

4: **taub(*)** – complex array

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

The scalar factors of the elementary reflectors which represent the unitary matrix Z .

5: **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: **m**, 2: **p**, 3: **n**, 4: **a**, 5: **lda**, 6: **taua**, 7: **b**, 8: **ldb**, 9: **taub**, 10: **work**, 11: **lwork**, 12: **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.

7 Accuracy

The computed generalized RQ factorization is the exact factorization for nearby matrices $(A + E)$ and $(B + F)$, where

$$\|E\|_2 = O\epsilon\|A\|_2 \quad \text{and} \quad \|F\|_2 = O\epsilon\|B\|_2,$$

and ϵ is the *machine precision*.

8 Further Comments

The unitary matrices Q and Z may be formed explicitly by calls to f08cw and f08at respectively. f08cx may be used to multiply Q by another matrix and f08au may be used to multiply Z by another matrix.

The real analogue of this function is f08zf.

9 Example

```
a = [complex(1, +0), complex(0, +0), complex(-1, +0), complex(0, +0);
      complex(0, +0), complex(1, +0), complex(0, +0), complex(-1, +0)];
b = [complex(0.96, -0.8100000000000001), complex(-0.03, +0.96), complex(-
0.91, +2.06), complex(-0.05, +0.41);
```

```

        complex(-0.98, +1.98), complex(-1.2, +0.19), complex(-0.66, +0.42),
        ...
        complex(-0.8100000000000001, +0.5600000000000001);
        complex(0.62, -0.46), complex(1.01, +0.02), complex(0.63, -0.17),
        complex(-1.11, +0.6);
        complex(0.37, +0.38), complex(0.19, -0.54), complex(-0.98, -0.36),
        complex(0.22, -0.2);
        complex(0.83, +0.51), complex(0.2, +0.01), complex(-0.17, -0.46),
        complex(1.47, +1.59);
        complex(1.08, -0.28), complex(0.2, -0.12), complex(-
        0.0700000000000001, +1.23), complex(0.26, +0.26)];
[aOut, taua, bOut, taub, info] = f08zt(a, b)

```

```

aOut =
    -0.4142         0     1.4142         0
         0    -0.4142         0     1.4142
taua =
    1.7071
    1.7071
bOut =
    -2.7118          -1.4390 - 1.0315i   -0.1054 + 1.3176i   -0.3924 -
0.1950i
    -0.2024 + 0.6829i   -1.8583          -0.9453 + 0.1928i   1.4355 +
0.2631i
    0.2443 - 0.2408i   -0.4279 + 0.1339i    2.9079          -0.2395 +
0.1886i
    -0.1408 + 0.0504i   0.2817 - 0.1483i   -0.4394 - 0.1145i   -1.5759
    0.1577 - 0.0379i    0.4242 + 0.4187i   -0.0867 - 0.7048i   -0.0589 +
0.0550i
    0.3069 + 0.1458i    0.0940 - 0.3436i   -0.1277 + 0.1658i    0.1632 -
0.4069i
taub =
    1.0130 + 0.3259i
    1.1208 - 0.0047i
    1.0674 - 0.2785i
    1.6275 - 0.2580i
info =
         0

```