NAG Toolbox for MATLAB

f08zs

1 Purpose

f08zs computes a generalized QR factorization of a complex matrix pair (A, B), where A is an n by m matrix and B is an n by p matrix.

2 Syntax

```
[a, taua, b, taub, info] = f08zs(a, b, 'n', n, 'm', m, 'p', p)
```

3 Description

f08zs forms the generalized QR factorization of an n by m matrix A and an n by p matrix B

$$A = QR, \qquad B = QTZ,$$

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

$$R = \begin{cases} \begin{array}{c} \\ m \\ n-m \end{array} \begin{array}{c} m \\ \begin{pmatrix} R_{11} \\ 0 \end{pmatrix} \end{array}, & \text{if } n \geq m; \\ \begin{array}{c} \\ n \end{array} \begin{array}{c} \begin{array}{cc} n & m-n \end{array} \\ \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} \end{array}, & \text{if } n < m, \end{cases}.$$

with R_{11} upper triangular,

$$T = \begin{cases} \begin{array}{c} \\ n \end{array} \begin{array}{c} \begin{array}{cc} p-n & n \end{array} \\ \begin{pmatrix} 0 & T_{12} \end{pmatrix} \end{array}, & \text{if } n \leq p, \\ \begin{array}{c} \\ n-p \\ p \end{array} \begin{array}{c} p \\ \begin{pmatrix} T_{11} \\ T_{21} \end{pmatrix} \end{array}, & \text{if } n > p, \end{cases}.$$

with T_{12} or T_{21} upper triangular.

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

$$B^{-1}A = Z^{\mathrm{H}}\left(T^{-1}R\right).$$

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

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

The n by m matrix A.

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

The n by p matrix B.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The first dimension of the arrays **a**, **b**. (An error is raised if these dimensions are not equal.)

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

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

2: **m – int32 scalar**

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

m, the number of columns of the matrix A.

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

3: **p – int32 scalar**

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

p, the number of columns of the matrix B.

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

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

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

The elements on and above the diagonal of the array contain the $\min(n,m)$ by m upper trapezoidal matrix R (R is upper triangular if $n \geq m$); the elements below the diagonal, with the array **taua**, represent the unitary matrix Q as a product of $\min(n,m)$ 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 $\max(1,\min(\mathbf{n},\mathbf{m}))$.

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

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

If $n \leq p$, the upper triangle of the subarray $\mathbf{b}(1:n,p-n+1:p)$ contains the n by n upper triangular matrix T_{12}.

If $n > p$, the elements on and above the $(n-p)$th subdiagonal contain the n by p upper trapezoidal matrix T; the remaining elements, 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 $\max(1,\min(\mathbf{n},\mathbf{p}))$.

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: **n**, 2: **m**, 3: **p**, 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 QR factorization is the exact factorization for nearby matrices $(A+E)$ and $(B+F)$, where

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

and ϵ is the ***machine precision***.

8 Further Comments

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

The real analogue of this function is f08ze.

9 Example

```
a = [complex(0.96, -0.8100000000000001), complex(-0.03, +0.96), complex(-
0.91, +2.06);
     complex(-0.98, +1.98), complex(-1.2, +0.19), complex(-0.66, +0.42);
     complex(0.62, -0.46), complex(1.01, +0.02), complex(0.63, -0.17);
               complex(1.08,   -0.28),   complex(0.2,   -0.12),   complex(-
0.07000000000000001, +1.23)];
b = [complex(0.5, -1), complex(0, +0), complex(0, +0), complex(0, +0);
     complex(0, +0), complex(1, -2), complex(0, +0), complex(0, +0);
```

```
     complex(0, +0), complex(0, +0), complex(2, -3), complex(0, +0);
     complex(0, +0), complex(0, +0), complex(0, +0), complex(5, -4)];
[aOut, taua, bOut, taub, info] = f08zs(a, b)
```

```
aOut =
  -2.8809             -0.5597 - 1.2116i   0.3523 - 1.2597i
  -0.3484 + 0.4420i   1.3009              1.5520 - 0.4961i
   0.1787 - 0.0821i  -0.5245 - 0.0361i   1.8059
   0.2839 - 0.0130i  -0.2338 + 0.4152i  -0.3539 - 0.7964i
taua =
   1.3332 - 0.2812i
   1.0427 - 0.5476i
   1.0736 - 0.2603i
bOut =
    1.8936              -1.3320 + 0.3960i    0.1213 - 1.4943i    1.4096 +
0.7140i
    0.3756 - 0.1350i    1.9049              -2.7958 - 0.0662i    0.4497 -
0.1602i
    0.1262 + 0.0131i    0.1989 + 0.0903i    4.1872              -1.5248 +
1.9624i
  -0.0015 + 0.0595i   0.1014 + 0.1408i   0.3901 + 0.3498i  -3.8214
taub =
   1.9245 + 0.3813i
   1.7199 - 0.0951i
   1.5174 + 0.7418i
   1.2841 + 0.5605i
info =
           0
```