

NAG Toolbox for MATLAB

f08ze

1 Purpose

f08ze computes a generalized QR factorization of a real 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] = f08ze(a, b, 'n', n, 'm', m, 'p', p)
```

3 Description

f08ze forms the generalized QR factorization of an n by m matrix A and an n by p matrix B

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

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

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

with R_{11} upper triangular,

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

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

$$B^{-1}A = Z^T(T^{-1}R).$$

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,*) – 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 $\max(1, \mathbf{m})$

The n by m matrix A .

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

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,*) – 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 $\max(1, \mathbf{m})$

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 orthogonal matrix Q as a product of $\min(n, m)$ elementary reflectors (see Section 3.2.6 in the F08 Chapter Introduction).

2: **taua(*) – double 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 orthogonal matrix Q .

3: **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 $\max(1, \mathbf{p})$

If $n \leq p$, the upper triangle of the subarray **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 orthogonal matrix Z as a product of elementary reflectors (see Section 3.2.6 in the F08 Chapter Introduction).

4: **taub(*)** – double 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 orthogonal 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

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

and ϵ is the *machine precision*.

8 Further Comments

The orthogonal matrices Q and Z may be formed explicitly by calls to f08af and f08cj respectively. f08ag may be used to multiply Q by another matrix and f08ck may be used to multiply Z by another matrix.

The complex analogue of this function is f08zs.

9 Example

```
a = [-0.57, -1.28, -0.39;  
      -1.93, 1.08, -0.31;  
      2.3, 0.24, -0.4;  
      -0.02, 1.03, -1.43];  
b = [0.5, 0, 0, 0;  
      0, 1, 0, 0;  
      0, 0, 2, 0;  
      0, 0, 0, 5];
```

```
[aOut, taua, bOut, taub, info] = f08ze(a, b)
```

```
aOut =  
    3.0562    -0.2694   -0.0232  
    0.5322   -1.9623    0.7189  
   -0.6343   -0.1120    1.3913  
    0.0055    0.2893    0.5605  
taua =  
    1.1865  
    1.8245  
    1.5218  
bOut =  
   -1.2131    0.2669   -0.8691    0.6131  
   -0.5707    0.7105    1.8876    1.8248  
    0.1229    0.2076    2.4417    2.9277  
   -0.0373   -0.1437   -0.2765   -2.3759  
taub =  
         0  
    1.5086  
    1.8900  
    1.8207  
info =  
         0
```
