NAG Toolbox for MATLAB

f08bt

1 Purpose

f08bt computes the QR factorization, with column pivoting, of a complex m by n matrix.

2 Syntax

```
[a, jpvt, tau, info] = f08bt(a, jpvt, 'm', m, 'n', n)
```

3 Description

f08bt forms the QR factorization, with column pivoting, of an arbitrary rectangular complex m by n matrix.

If $m \geq n$, the factorization is given by:

$$AP = Q\begin{pmatrix} R \\ 0 \end{pmatrix},$$

where R is an n by n upper triangular matrix (with real diagonal elements), Q is an m by m unitary matrix and P is an n by n permutation matrix. It is sometimes more convenient to write the factorization as

$$AP = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}\begin{pmatrix} R \\ 0 \end{pmatrix},$$

which reduces to

$$AP = Q_1R,$$

where Q_1 consists of the first n columns of Q, and Q_2 the remaining $m - n$ columns.

If $m < n$, R is trapezoidal, and the factorization can be written

$$AP = Q\begin{pmatrix} R_1 & R_2 \end{pmatrix},$$

where R_1 is upper triangular and R_2 is rectangular.

The matrix Q is not formed explicitly but is represented as a product of $\min(m, n)$ elementary reflectors (see the F08 Chapter Introduction for details). Functions are provided to work with Q in this representation (see Section 8).

Note also that for any $k < n$, the information returned in the first k columns of the array **a** represents a QR factorization of the first k columns of the permuted matrix AP.

The function allows specified columns of A to be moved to the leading columns of AP at the start of the factorization and fixed there. The remaining columns are free to be interchanged so that at the ith stage the pivot column is chosen to be the column which maximizes the 2-norm of elements i to m over columns i to n.

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

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

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: **jpvt($*$) – int32 array**

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

If $\mathbf{jpvt}(j) \neq 0$, then the jth column of A is moved to the beginning of AP before the decomposition is computed and is fixed in place during the computation. Otherwise, the jth column of A is a free column (i.e., one which may be interchanged during the computation with any other free column).

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: **n – int32 scalar**

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

n, the number of columns of the matrix A.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, work, lwork, rwork

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 $m \geq n$, the elements below the diagonal are overwritten by details of the unitary matrix Q and the upper triangle contains the corresponding elements of the n by n upper triangular matrix R.

If $m < n$, the strictly lower triangular part contains details of the unitary matrix Q and the remaining elements are overwritten by the corresponding elements of the m by n upper trapezoidal matrix R.

The diagonal elements of R are real.

2: **jpvt($*$) – int32 array**

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

Details of the permutation matrix P. More precisely, if $\mathbf{jpvt}(j) = k$, then the kth column of A is moved to become the jth column of AP; in other words, the columns of AP are the columns of A in the order $\mathbf{jpvt}(1), \mathbf{jpvt}(2), \ldots, \mathbf{jpvt}(n)$.

3: **tau**$(*)$ – **complex array**

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

further details of the unitary matrix Q.

4: **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: **n**, 3: **a**, 4: **lda**, 5: **jpvt**, 6: **tau**, 7: **work**, 8: **lwork**, 9: **rwork**, 10: **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 factorization is the exact factorization of a nearby matrix $(A + E)$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***.

8 Further Comments

The total number of real floating-point operations is approximately $\frac{8}{3}n^2(3m - n)$ if $m \geq n$ or $\frac{8}{3}m^2(3n - m)$ if $m < n$.

To form the unitary matrix Q f08bt may be followed by a call to f08at:

```
[a, info] = f08at(a(:,1:m), tau);
```

but note that the second dimension of the array **a** must be at least **m**, which may be larger than was required by f08bt.

When $m \geq n$, it is often only the first n columns of Q that are required, and they may be formed by the call:

```
[a, info] = f08at(a, tau);
```

To apply Q to an arbitrary complex rectangular matrix C, f08bt may be followed by a call to f08au. For example,

```
[c, info] = f08au('Left','Conjugate Transpose', a(:,min(m,n)), tau, c);
```

forms $C = Q^{\mathrm{H}}C$, where C is m by p.

To compute a QR factorization without column pivoting, use f08as.

The real analogue of this function is f08bf.

9 Example

```
a = [complex(0.47, -0.34), complex(-0.4, +0.54), complex(0.6, +0.01),
complex(0.8, -1.02);
      complex(-0.32, -0.23), complex(-0.05, +0.2), complex(-0.26, -0.44),
```

```
complex(-0.43, +0.17);
       complex(0.35, -0.6), complex(-0.52, -0.34), complex(0.87, -0.11),
complex(-0.34, -0.09);
      complex(0.89, +0.71), complex(-0.45, -0.45), complex(-0.02, -0.57),
complex(1.14, -0.78);
       complex(-0.19, +0.06), complex(0.11, -0.85), complex(1.44, +0.8),
complex(0.07000000000000001, +1.14)];
jpvt = [int32(0);
     int32(0);
     int32(0);
     int32(0)];
[aOut, jpvtOut, tau, info] = f08bt(a, jpvt)
```

```
aOut =
  -2.2870               -0.7239 + 0.5605i   0.7570 + 0.4519i  -0.5563 -
1.0147i
  -0.1420 + 0.0082i   1.8950               -0.3620 - 0.4787i   0.1855 -
0.5741i
  -0.0906 - 0.0591i  -0.4734 + 0.1684i   0.9393               -0.5318 +
0.5694i
   0.4082 - 0.1178i   0.2585 + 0.0267i   0.0451 + 0.5815i  -0.0094
  -0.0896 + 0.3397i  -0.6872 + 0.0317i  -0.1171 - 0.0121i   0.8616 -
0.2123i
jpvtOut =
           4
           3
           2
           1
tau =
   1.3498 - 0.4460i
   1.0404 + 0.2791i
   1.4703 + 0.0998i
   1.1181 + 0.0300i
info =
           0
```