# NAG Toolbox for MATLAB

# f08le

## 1 Purpose

f08le reduces a real $m$ by $n$ band matrix to upper bidiagonal form.

## 2 Syntax

```
[ab, d, e, q, pt, c, info] = f08le(vect, m, kl, ku, ab, c, 'n', n, 'ncc', ncc)
```

## 3 Description

f08le reduces a real $m$ by $n$ band matrix to upper bidiagonal form $B$ by an orthogonal transformation: $A = QBP^{\mathrm{T}}$. The orthogonal matrices $Q$ and $P^{\mathrm{T}}$, of order $m$ and $n$ respectively, are determined as a product of Givens rotation matrices, and may be formed explicitly by the function if required. A matrix $C$ may also be updated to give $\tilde{C} = Q^{\mathrm{T}}C$.

The function uses a vectorizable form of the reduction.

## 4 References

None.

## 5 Parameters

### 5.1 Compulsory Input Parameters

1: **vect – string**

Indicates whether the matrices $Q$ and/or $P^{\mathrm{T}}$ are generated.

**vect** = 'N'

Neither $Q$ nor $P^{\mathrm{T}}$ is generated.

**vect** = 'Q'

$Q$ is generated.

**vect** = 'P'

$P^{\mathrm{T}}$ is generated.

**vect** = 'B'

Both $Q$ and $P^{\mathrm{T}}$ are generated.

*Constraint*: **vect** = 'N', 'Q', 'P' or 'B'.

2: **m – int32 scalar**

$m$, the number of rows of the matrix $A$.

*Constraint*: **m** $\geq 0$.

3: **kl – int32 scalar**

The number of subdiagonals, $k_l$, within the band of $A$.

*Constraint*: **kl** $\geq 0$.

4: **ku – int32 scalar**

The number of superdiagonals, $k_u$, within the band of $A$.

*Constraint*: **ku** $\geq 0$.

5: **ab**(**ldab**,$*$) **– double array**

The first dimension of the array **ab** must be at least **kl** + **ku** + 1

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

The original $m$ by $n$ band matrix $A$.

The matrix is stored in rows 1 to $++1$, more precisely, the element $A_{ij}$ must be stored in

$$\mathbf{ab}(+1 + i - j, j) \qquad \text{for } \max(1j-) \leq i \leq \min(mj+).$$

6: **c**(**ldc**,$*$) **– double array**

The first dimension, **ldc**, of the array **c** must satisfy

if **ncc** $> 0$, **ldc** $\geq \max(1, \mathbf{m})$;
if **ncc** $= 0$, **ldc** $\geq 1$.

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

An $m$ by $n_C$ matrix $C$.

## 5.2 Optional Input Parameters

1: **n – int32 scalar**

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

$n$, the number of columns of the matrix $A$.

*Constraint*: **n** $\geq 0$.

2: **ncc – int32 scalar**

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

$n_C$, the number of columns of the matrix $C$.

*Constraint*: **ncc** $\geq 0$.

## 5.3 Input Parameters Omitted from the MATLAB Interface

ldab, ldq, ldpt, ldc, work

## 5.4 Output Parameters

1: **ab**(**ldab**,$*$) **– double array**

The first dimension of the array **ab** must be at least **kl** + **ku** + 1

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

**ab** contains values generated during the reduction.

2: **d**($*$) **– double array**

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

The diagonal elements of the bidiagonal matrix $B$.

3: **e**$(*)$ **– double array**

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

The superdiagonal elements of the bidiagonal matrix $B$.

4: **q**(**ldq**,$*$) **– double array**

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

if **vect** = 'Q' or 'B', **ldq** $\geq \max(1, \mathbf{m})$;
**ldq** $\geq 1$ otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{m})$ if **vect** = 'Q' or 'B', and at least 1 otherwise

If **vect** = 'Q' or 'B', contains the $m$ by $m$ orthogonal matrix $Q$.

If **vect** = 'N' or 'P', **q** is not referenced.

5: **pt**(**ldpt**,$*$) **– double array**

The first dimension, **ldpt**, of the array **pt** must satisfy

if **vect** = 'P' or 'B', **ldpt** $\geq \max(1, \mathbf{n})$;
**ldpt** $\geq 1$ otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **vect** = 'P' or 'B', and at least 1 otherwise

The $n$ by $n$ orthogonal matrix $P^{\mathrm{T}}$, if **vect** = 'P' or 'B'. If **vect** = 'N' or 'Q', **pt** is not referenced.

6: **c**(**ldc**,$*$) **– double array**

The first dimension, **ldc**, of the array **c** must satisfy

if **ncc** $> 0$, **ldc** $\geq \max(1, \mathbf{m})$;
if **ncc** $= 0$, **ldc** $\geq 1$.

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

**c** contains $Q^{\mathrm{T}}C$. If **ncc** $= 0$, **c** is not referenced.

7: **info – int32 scalar**

**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: **vect**, 2: **m**, 3: **n**, 4: **ncc**, 5: **kl**, 6: **ku**, 7: **ab**, 8: **ldab**, 9: **d**, 10: **e**, 11: **q**, 12: **ldq**, 13: **pt**, 14: **ldpt**, 15: **c**, 16: **ldc**, 17: **work**, 18: **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 bidiagonal form $B$ satisfies $QBP^{\mathrm{T}} = A + E$, where

$$\|E\|_2 \leq c(n)\epsilon\|A\|_2,$$

$c(n)$ is a modestly increasing function of $n$, and $\epsilon$ is the ***machine precision***.

The elements of $B$ themselves may be sensitive to small perturbations in $A$ or to rounding errors in the computation, but this does not affect the stability of the singular values and vectors.

The computed matrix $Q$ differs from an exactly orthogonal matrix by a matrix $F$ such that

$$\|F\|_2 = O(\epsilon).$$

A similar statement holds for the computed matrix $P^{\mathrm{T}}$.

## 8 Further Comments

The total number of real floating-point operations is approximately the sum of:

$6n^2k$, if **vect** = 'N' and **ncc** = 0, and

$3n^2n_C(k-1)/k$, if $C$ is updated, and

$3n^3(k-1)/k$, if either $Q$ or $P^{\mathrm{T}}$ is generated (double this if both),

where $k = k_l + k_u$, assuming $n \gg k$. For this section we assumed that $m = n$.

The complex analogue of this function is f08ls.

## 9 Example

```
vect = 'N';
m = int32(6);
kl = int32(2);
ku = int32(1);
ab = [0, -1.28, -0.31, -0.35;
     -0.57, 1.08, 0.4, 0.08;
     -1.93, 0.24, -0.66, -2.13;
     2.3, 0.64, 0.15, 0.5];
c = [];
[abOut, d, e, q, pt, cOut, info] = f08le(vect, m, kl, ku, ab, c)
```

```
abOut =
         0    0.6206   -1.2353   -1.1240
    3.0561    1.5259    0.9690    1.5685
    3.0025    1.3713    0.9687   -1.0654
    2.3000    1.3371    0.9687    0.0371
d =
    3.0561
    1.5259
    0.9690
    1.5685
e =
    0.6206
   -1.2353
   -1.1240
q =
     0
pt =
     0
cOut =
info =
           0
```