

## NAG Toolbox

### **nag\_lapack\_zgbbrd (f08ls)**

## 1 Purpose

nag\_lapack\_zgbbrd (f08ls) reduces a complex  $m$  by  $n$  band matrix to real upper bidiagonal form.

## 2 Syntax

```
[ab, d, e, pt, c, info] = nag_lapack_zgbbrd(vect, m, kl, ku, ab, c, 'n', n,
'ncc', ncc)
[ab, d, e, pt, c, info] = f08ls(vect, m, kl, ku, ab, c, 'n', n, 'ncc', ncc)
```

## 3 Description

nag\_lapack\_zgbbrd (f08ls) reduces a complex  $m$  by  $n$  band matrix to real upper bidiagonal form  $B$  by a unitary transformation:  $A = QBP^H$ . The unitary matrices  $Q$  and  $P^H$ , 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  $\tilde{C} = Q^H C$ .

The function uses a vectorizable form of the reduction.

## 4 References

None.

## 5 Parameters

### 5.1 Compulsory Input Parameters

1: **vect** – CHARACTER(1)

Indicates whether the matrices  $Q$  and/or  $P^H$  are generated.

**vect** = 'N'

Neither  $Q$  nor  $P^H$  is generated.

**vect** = 'Q'

$Q$  is generated.

**vect** = 'P'

$P^H$  is generated.

**vect** = 'B'

Both  $Q$  and  $P^H$  are generated.

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

2: **m** – INTEGER

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

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

3: **kl** – INTEGER

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

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

4: **ku** – INTEGER

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

*Constraint:*  $\mathbf{ku} \geq 0$ .

5: **ab**( $ldab, :)$  – COMPLEX (KIND=nag\_wp) array

The first dimension of the array **ab** must be at least  $\mathbf{kl} + \mathbf{ku} + 1$ .

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

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

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

$$\mathbf{ab}(k_u + 1 + i - j, j) \quad \text{for } \max(1, j - k_u) \leq i \leq \min(m, j + k_l).$$

6: **c**( $ldc, :)$  – COMPLEX (KIND=nag\_wp) array

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

$$\begin{aligned} &\text{if } \mathbf{ncc} > 0, \quad ldc \geq \max(1, \mathbf{m}); \\ &\text{if } \mathbf{ncc} = 0, \quad ldc \geq 1. \end{aligned}$$

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

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

## 5.2 Optional Input Parameters

1: **n** – INTEGER

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

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

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

2: **ncc** – INTEGER

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

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

*Constraint:*  $\mathbf{ncc} \geq 0$ .

## 5.3 Output Parameters

1: **ab**( $ldab, :)$  – COMPLEX (KIND=nag\_wp) array

The first dimension of the array **ab** will be  $\mathbf{kl} + \mathbf{ku} + 1$ .

The second dimension of the array **ab** will be  $\max(1, \mathbf{n})$ .

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

2: **d**( $\min(\mathbf{m}, \mathbf{n})$ ) – REAL (KIND=nag\_wp) array

The diagonal elements of the bidiagonal matrix  $B$ .

3: **e**( $\min(\mathbf{m}, \mathbf{n}) - 1$ ) – REAL (KIND=nag\_wp) array

The superdiagonal elements of the bidiagonal matrix  $B$ .

4: **q**(*ldq*,:) – COMPLEX (KIND=nag\_wp) array

The first dimension, *ldq*, of the array **q** will be

if **vect** = 'Q' or 'B', *ldq* = max(1, **m**);  
otherwise *ldq* = 1.

The second dimension of the array **q** will be max(1, **m**) if **vect** = 'Q' or 'B' and 1 otherwise.

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

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

5: **pt**(*ldpt*,:) – COMPLEX (KIND=nag\_wp) array

The first dimension, *ldpt*, of the array **pt** will be

if **vect** = 'P' or 'B', *ldpt* = max(1, **n**);  
otherwise *ldpt* = 1.

The second dimension of the array **pt** will be max(1, **n**) if **vect** = 'P' or 'B' and 1 otherwise.

The *n* by *n* unitary matrix *P*<sup>H</sup>, if **vect** = 'P' or 'B'. If **vect** = 'N' or 'Q', **pt** is not referenced.

6: **c**(*ldc*,:) – COMPLEX (KIND=nag\_wp) array

The first dimension, *ldc*, of the array **c** will be

if **ncc** > 0, *ldc* = max(1, **m**);  
if **ncc** = 0, *ldc* = 1.

The second dimension of the array **c** will be max(1, **ncc**).

**c** stores *Q*<sup>H</sup>*C*. If **ncc** = 0, **c** is not referenced.

7: **info** – INTEGER

**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: **rwork**, 19: **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*<sup>H</sup> = *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 unitary matrix by a matrix *F* such that

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

A similar statement holds for the computed matrix *P*<sup>H</sup>.

## 8 Further Comments

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

$20n^2k$ , if **vect** = 'N' and **ncc** = 0, and

$10n^2n_C(k - 1)/k$ , if  $C$  is updated, and

$10n^3(k - 1)/k$ , if either  $Q$  or  $P^H$  is generated (double this if both),

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

The real analogue of this function is nag\_lapack\_dgbbrd (f08le).

## 9 Example

This example reduces the matrix  $A$  to upper bidiagonal form, where

$$A = \begin{pmatrix} 0.96 - 0.81i & -0.03 + 0.96i & 0.00 + 0.00i & 0.00 + 0.00i \\ -0.98 + 1.98i & -1.20 + 0.19i & -0.66 + 0.42i & 0.00 + 0.00i \\ 0.62 - 0.46i & 1.01 + 0.02i & 0.63 - 0.17i & -1.11 + 0.60i \\ 0.00 + 0.00i & 0.19 - 0.54i & -0.98 - 0.36i & 0.22 - 0.20i \\ 0.00 + 0.00i & 0.00 + 0.00i & -0.17 - 0.46i & 1.47 + 1.59i \\ 0.00 + 0.00i & 0.00 + 0.00i & 0.00 + 0.00i & 0.26 + 0.26i \end{pmatrix}.$$

### 9.1 Program Text

```
function f08ls_example

fprintf('f08ls example results\n\n');

% Banded complex matrix A stored in banded format
m = nag_int(6);
n = nag_int(4);
k1 = nag_int(2);
ku = nag_int(1);
ab = [ 0.00 + 0.00i, -0.03 + 0.96i, -0.66 + 0.42i, -1.11 + 0.60i;
       0.96 - 0.81i, -1.20 + 0.19i, 0.63 - 0.17i, 0.22 - 0.20i;
      -0.98 + 1.98i, 1.01 + 0.02i, -0.98 - 0.36i, 1.47 + 1.59i;
       0.62 - 0.46i, 0.19 - 0.54i, -0.17 - 0.46i, 0.26 + 0.26i];

% Reduce A to bidiagonal form
c = [];
vect = 'No Q or PT';
[~, d, e, ~, ~, ~, info] = f08ls( ...
                           vect, m, k1, ku, ab, c);

fprintf('Diagonal:\n');
fprintf(' %8.4f',d);
fprintf('\nOff-diagonal (absolute values):\n');
fprintf(' %8.4f',abs(e));
fprintf('\n');
```

### 9.2 Program Results

```
f08ls example results
```

```
Diagonal:
 2.6560   1.7501   2.0607   0.8658
Off-diagonal (absolute values):
 1.7033   1.2800   0.1467
```

---