NAG Toolbox

nag_lapack_zgebrd (f08ks)

1 Purpose

nag_lapack_zgebrd (f08ks) reduces a complex m by n matrix to bidiagonal form.

2 Syntax

```
[a, d, e, tauq, taup, info] = nag_lapack_zgebrd(a, 'm', m, 'n', n)
[a, d, e, tauq, taup, info] = f08ks(a, 'm', m, 'n', n)
```

3 Description

nag_lapack_zgebrd (f08ks) reduces a complex m by n matrix A to real bidiagonal form B by a unitary transformation: $A = QBP^{\mathrm{H}}$, where Q and P^{H} are unitary matrices of order m and n respectively.

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

$$A = Q\begin{pmatrix} B_1 \\ 0 \end{pmatrix}P^{\mathrm{H}} = Q_1 B_1 P^{\mathrm{H}},$$

where B_1 is a real n by n upper bidiagonal matrix and Q_1 consists of the first n columns of Q.

If $m < n$, the reduction is given by

$$A = Q\begin{pmatrix} B_1 & 0 \end{pmatrix}P^{\mathrm{H}} = QB_1P_1^{\mathrm{H}},$$

where B_1 is a real m by m lower bidiagonal matrix and P_1^{H} consists of the first m rows of P^{H}.

The unitary matrices Q and P are not formed explicitly but are represented as products of elementary reflectors (see the F08 Chapter Introduction for details). Functions are provided to work with Q and P in this representation (see Section 9).

4 References

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 (KIND=nag_wp) array

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

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

The m by n matrix A.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the first dimension of the array **a**.

m, the number of rows of the matrix A.

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

2: **n** – INTEGER

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

n, the number of columns of the matrix A.

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

5.3 Output Parameters

1: $\mathbf{a}(lda,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **a** will be $\max(1, \mathbf{m})$.

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

If $m \geq n$, the diagonal and first superdiagonal store the upper bidiagonal matrix B, elements below the diagonal store details of the unitary matrix Q and elements above the first superdiagonal store details of the unitary matrix P.

If $m < n$, the diagonal and first subdiagonal store the lower bidiagonal matrix B, elements below the first subdiagonal store details of the unitary matrix Q and elements above the diagonal store details of the unitary matrix P.

2: $\mathbf{d}(:)$ – REAL (KIND=nag_wp) array

The dimension of the array **d** will be $\max(1, \min(\mathbf{m}, \mathbf{n}))$

The diagonal elements of the bidiagonal matrix B.

3: $\mathbf{e}(:)$ – REAL (KIND=nag_wp) array

The dimension of the array **e** will be $\max(1, \min(\mathbf{m}, \mathbf{n}) - 1)$

The off-diagonal elements of the bidiagonal matrix B.

4: $\mathbf{tauq}(:)$ – COMPLEX (KIND=nag_wp) array

The dimension of the array **tauq** will be $\max(1, \min(\mathbf{m}, \mathbf{n}))$

Further details of the unitary matrix Q.

5: $\mathbf{taup}(:)$ – COMPLEX (KIND=nag_wp) array

The dimension of the array **taup** will be $\max(1, \min(\mathbf{m}, \mathbf{n}))$

Further details of the unitary matrix P.

6: **info** – INTEGER

$\mathbf{info} = 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

$\mathbf{info} = -i$

If $\mathbf{info} = -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **m**, 2: **n**, 3: **a**, 4: **lda**, 5: **d**, 6: **e**, 7: **tauq**, 8: **taup**, 9: **work**, 10: **lwork**, 11: **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 $QBP^{\mathrm{H}} = A + E$, where

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

$c(n)$ is a modestly increasing function of n, and ϵ 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.

8 Further Comments

The total number of real floating-point operations is approximately $16n^2(3m - n)/3$ if $m \geq n$ or $16m^2(3n - m)/3$ if $m < n$.

If $m \gg n$, it can be more efficient to first call nag_lapack_zgeqrf (f08as) to perform a QR factorization of A, and then to call nag_lapack_zgebrd (f08ks) to reduce the factor R to bidiagonal form. This requires approximately $8n^2(m + n)$ floating-point operations.

If $m \ll n$, it can be more efficient to first call nag_lapack_zgelqf (f08av) to perform an LQ factorization of A, and then to call nag_lapack_zgebrd (f08ks) to reduce the factor L to bidiagonal form. This requires approximately $8m^2(m + n)$ operations.

To form the unitary matrices P^{H} and/or Q nag_lapack_zgebrd (f08ks) may be followed by calls to nag_lapack_zungbr (f08kt):

to form the m by m unitary matrix Q

```
[a, info] = f08kt('Q', n, a, tauq);
```

but note that the second dimension of the array **a** must be at least **m**, which may be larger than was required by nag_lapack_zgebrd (f08ks);

to form the n by n unitary matrix P^{H}

```
[a, info] = f08kt('P', m, a, taup);
```

but note that the first dimension of the array **a**, specified by the argument *lda*, must be at least **n**, which may be larger than was required by nag_lapack_zgebrd (f08ks).

To apply Q or P to a complex rectangular matrix C, nag_lapack_zgebrd (f08ks) may be followed by a call to nag_lapack_zunmbr (f08ku).

The real analogue of this function is nag_lapack_dgebrd (f08ke).

9 Example

This example reduces the matrix A to bidiagonal form, where

$$A = \begin{pmatrix} 0.96 - 0.81i & -0.03 + 0.96i & -0.91 + 2.06i & -0.05 + 0.41i \\ -0.98 + 1.98i & -1.20 + 0.19i & -0.66 + 0.42i & -0.81 + 0.56i \\ 0.62 - 0.46i & 1.01 + 0.02i & 0.63 - 0.17i & -1.11 + 0.60i \\ -0.37 + 0.38i & 0.19 - 0.54i & -0.98 - 0.36i & 0.22 - 0.20i \\ 0.83 + 0.51i & 0.20 + 0.01i & -0.17 - 0.46i & 1.47 + 1.59i \\ 1.08 - 0.28i & 0.20 - 0.12i & -0.07 + 1.23i & 0.26 + 0.26i \end{pmatrix}.$$

9.1 Program Text

```
function f08ks_example

fprintf('f08ks example results\n\n');

a = [ 0.96 - 0.81i, -0.03 + 0.96i, -0.91 + 2.06i, -0.05 + 0.41i;
     -0.98 + 1.98i, -1.20 + 0.19i, -0.66 + 0.42i, -0.81 + 0.56i;
      0.62 - 0.46i,  1.01 + 0.02i,  0.63 - 0.17i, -1.11 + 0.60i;
     -0.37 + 0.38i,  0.19 - 0.54i, -0.98 - 0.36i,  0.22 - 0.20i;
```

```
      0.83 + 0.51i,  0.20 + 0.01i, -0.17 - 0.46i,  1.47 + 1.59i;
      1.08 - 0.28i,  0.20 - 0.12i, -0.07 + 1.23i,  0.26 + 0.26i];

% Reduce general complex matrix to real bidiagonal form A = ZBP^H
[B, d, e, tauq, taup, info] = f08ks(a);

fprintf(' Bidiagonal matrix B\n   Main diagonal  ');
fprintf(' %7.3f',d);
fprintf('\n   super-diagonal ');
fprintf(' %7.3f',e);
fprintf('\n');
```

9.2 Program Results

```
     f08ks example results

 Bidiagonal matrix B
   Main diagonal    -3.087   2.066   1.873   2.002
   super-diagonal    2.113   1.263  -1.613
```