NAG Fortran Library Routine Document

F08KSF (CGEBRD/ZGEBRD)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F08KSF (CGEBRD/ZGEBRD) reduces a complex m by n matrix to bidiagonal form.

2 Specification

```
SUBROUTINE F08KSF(M, N, A, LDA, D, E, TAUQ, TAUP, WORK, LWORK, INFO)
ENTRY      cgebrd (M, N, A, LDA, D, E, TAUQ, TAUP, WORK, LWORK, INFO)
INTEGER           M, N, LDA, LWORK, INFO
real              D(*), E(*)
complex           A(LDA,*), TAUQ(*), TAUP(*), WORK(*)
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine reduces a complex m by n matrix A to real bidiagonal form B by a unitary transformation: $A = QBP^H$, where Q and P^H are unitary matrices of order m and n respectively.

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

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

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

$$A = Q(\,B_1 \quad 0\,)P^H = QB_1P_1^H,$$

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). Routines are provided to work with Q and P in this representation (see Section 8).

4 References

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

5 Parameters

1: M – INTEGER *Input*

On entry: m, the number of rows of the matrix A.

Constraint: $M \geq 0$.

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

Constraint: $N \geq 0$.

3: A(LDA,*) – ***complex*** array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1, \mathrm{N})$.

On entry: the m by n matrix A.

On exit: if $m \geq n$, the diagonal and first super-diagonal are overwritten by the upper bidiagonal matrix B, elements below the diagonal are overwritten by details of the unitary matrix Q and elements above the first super-diagonal are overwritten by details of the unitary matrix P.

If $m < n$, the diagonal and first sub-diagonal are overwritten by the lower bidiagonal matrix B, elements below the first sub-diagonal are overwritten by details of the unitary matrix Q and elements above the diagonal are overwritten by details of the unitary matrix P.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08KSF (CGEBRD/ZGEBRD) is called.

Constraint: $\mathrm{LDA} \geq \max(1, \mathrm{M})$.

5: D($*$) – ***real*** array *Output*

Note: the dimension of the array D must be at least $\max(1, \min(\mathrm{M}, \mathrm{N}))$.

On exit: the diagonal elements of the bidiagonal matrix B.

6: E($*$) – ***real*** array *Output*

Note: the dimension of the array E must be at least $\max(1, \min(\mathrm{M}, \mathrm{N}) - 1)$.

On exit: the off-diagonal elements of the bidiagonal matrix B.

7: TAUQ($*$) – ***complex*** array *Output*

Note: the dimension of the array TAUQ must be at least $\max(1, \min(\mathrm{M}, \mathrm{N}))$.

On exit: further details of the unitary matrix Q.

8: TAUP($*$) – ***complex*** array *Output*

Note: the dimension of the array TAUP must be at least $\max(1, \min(\mathrm{M}, \mathrm{N}))$.

On exit: further details of the unitary matrix P.

9: WORK($*$) – ***complex*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, \mathrm{LWORK})$.

On exit: if INFO $= 0$, the real part of WORK(1) contains the minimum value of LWORK required for optimum performance.

10: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08KSF (CGEBRD/ZGEBRD) is called, unless LWORK $= -1$, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK (using the formula given below).

Suggested value: for optimum performance LWORK should be at least $(\mathrm{M} + \mathrm{N}) \times nb$, where nb is the ***blocksize***.

Constraint: $\mathrm{LWORK} \geq \max(1, \mathrm{M}, \mathrm{N})$ or LWORK $= -1$.

11: INFO – INTEGER *Output*

On exit: INFO $= 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

The computed bidiagonal form B satisfies $QBP^H = A + E$, where

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

$c(n)$ 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 $m \ge n$ or $16m^2(3n - m)/3$ if $m < n$.

If $m \gg n$, it can be more efficient to first call F08ASF (CGEQRF/ZGEQRF) to perform a QR factorization of A, and then to call F08KSF (CGEBRD/ZGEBRD) 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 F08AVF (CGELQF/ZGELQF) to perform an LQ factorization of A, and then to call F08KSF (CGEBRD/ZGEBRD) 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, this routine may be followed by calls to F08KTF (CUNGBR/ZUNGBR):

to form the m by m unitary matrix Q

```
CALL CUNGBR ('Q',M,M,N,A,LDA,TAUQ,WORK,LWORK,INFO)
```

but note that the second dimension of the array A must be at least M, which may be larger than was required by F08KSF (CGEBRD/ZGEBRD);

to form the n by n unitary matrix P^H

```
CALL CUNGBR ('P',N,N,M,A,LDA,TAUP,WORK,LWORK,INFO)
```

but note that the first dimension of the array A, specified by the parameter LDA, must be at least N, which may be larger than was required by F08KSF (CGEBRD/ZGEBRD).

To apply Q or P to a complex rectangular matrix C, this routine may be followed by a call to F08KUF (CUNMBR/ZUNMBR).

The real analogue of this routine is F08KSF (CGEBRD/ZGEBRD).

9 Example

To reduce 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

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08KSF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, LDA, LWORK
      PARAMETER        (MMAX=8,NMAX=8,LDA=MMAX,LWORK=64*(MMAX+NMAX))
*     .. Local Scalars ..
      INTEGER          I, INFO, J, M, N
*     .. Local Arrays ..
      complex          A(LDA,NMAX), TAUP(NMAX), TAUQ(NMAX), WORK(LWORK)
      real             D(NMAX), E(NMAX-1)
*     .. External Subroutines ..
      EXTERNAL         cgebrd
*     .. Intrinsic Functions ..
      INTRINSIC        MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08KSF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N
      IF (M.LE.MMAX .AND. N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
*
*        Reduce A to bidiagonal form
*
         CALL cgebrd(M,N,A,LDA,D,E,TAUQ,TAUP,WORK,LWORK,INFO)
*
*        Print bidiagonal form
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Diagonal'
         WRITE (NOUT,99999) (D(I),I=1,MIN(M,N))
         IF (M.GE.N) THEN
            WRITE (NOUT,*) 'Super-diagonal'
         ELSE
            WRITE (NOUT,*) 'Sub-diagonal'
         END IF
         WRITE (NOUT,99999) (E(I),I=1,MIN(M,N)-1)
      END IF
      STOP
*
99999 FORMAT (1X,8F9.4)
      END
```

9.2 Program Data

```
F08KSF Example Program Data
  6  4                                                      :Values of M and N
 ( 0.96,-0.81) (-0.03, 0.96) (-0.91, 2.06) (-0.05, 0.41)
 (-0.98, 1.98) (-1.20, 0.19) (-0.66, 0.42) (-0.81, 0.56)
 ( 0.62,-0.46) ( 1.01, 0.02) ( 0.63,-0.17) (-1.11, 0.60)
 (-0.37, 0.38) ( 0.19,-0.54) (-0.98,-0.36) ( 0.22,-0.20)
 ( 0.83, 0.51) ( 0.20, 0.01) (-0.17,-0.46) ( 1.47, 1.59)
 ( 1.08,-0.28) ( 0.20,-0.12) (-0.07, 1.23) ( 0.26, 0.26)   :End of matrix A
```

9.3 Program Results

```
F08KSF Example Program Results

Diagonal
  -3.0870   2.0660   1.8731   2.0022
Super-diagonal
   2.1126   1.2628  -1.6126
```