NAG Fortran Library Routine Document

F08MDF (DBDSDC)

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

F08MDF (DBDSDC) computes the singular values and, optionally, the left and right singular vectors of a real n by n (upper or lower) bidiagonal matrix B.

2 Specification

```
      SUBROUTINE F08MDF (UPLO, COMPQ, N, D, E, U, LDU, VT, LDVT, Q, IQ, WORK,
     1                   IWORK, INFO)
      INTEGER            N, LDU, LDVT, IQ(*), IWORK(*), INFO
      double precision   D(*), E(*), U(LDU,*), VT(LDVT,*), Q(*), WORK(*)
      CHARACTER*1        UPLO, COMPQ
```

The routine may be called by its LAPACK name ***dbdsdc***.

3 Description

F08MDF (DBDSDC) computes the singular value decomposition (SVD) of the (upper or lower) bidiagonal matrix B as

$$B = USV^{\mathrm{T}},$$

where S is a diagonal matrix with non-negative diagonal elements $s_{ii} = s_i$, such that

$$s_1 \geq s_2 \geq \ldots \geq s_n \geq 0,$$

and U and V are orthogonal matrices. The diagonal elements of S are the singular values of B and the columns of U and V are respectively the corresponding left and right singular vectors of B.

When only singular values are required the routine uses the QR algorithm, but when singular vectors are required a divide and conquer method is used. The singular values can optionally be returned in compact form, although currently no routine is available to apply U or V when stored in compact form.

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

1: UPLO – CHARACTER*1 *Input*

On entry: indicates whether B is upper or lower bidiagonal.

UPLO = 'U'

B is upper bidiagonal.

UPLO = 'L'

B is lower bidiagonal.

2: COMPQ – CHARACTER*1 *Input*

On entry: specifies whether singular vectors are to be computed.

COMPQ = 'N'

Compute singular values only.

COMPQ = 'P'

Compute singular values and compute singular vectors in compact form.

COMPQ = 'I'

Compute singular values and singular vectors.

3: N – INTEGER *Input*

On entry: n, the order of the matrix B.

Constraint: $\mathrm{N} \geq 0$.

4: D($*$) – ***double precision*** array *Input/Output*

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

On entry: the n diagonal elements of the bidiagonal matrix B.

On exit: if INFO = 0, the singular values of B.

5: E($*$) – ***double precision*** array *Input/Output*

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

On entry: the $(n - 1)$ off-diagonal elements of the bidiagonal matrix B.

On exit: the contents of E are destroyed.

6: U(LDU,$*$) – ***double precision*** array *Output*

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

On exit: if COMPQ = 'I', then if INFO = 0, U contains the left singular vectors of the bidiagonal matrix B.

If COMPQ $\neq$ 'I', U is not referenced.

7: LDU – INTEGER *Input*

On entry: the first dimension of the array U as declared in the (sub)program from which F08MDF (DBDSDC) is called.

Constraints:

if COMPQ = 'I', LDU $\geq \max(1, \mathrm{N})$;
LDU ≥ 1 otherwise.

8: VT(LDVT,$*$) – ***double precision*** array *Output*

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

On exit: if COMPQ = 'I', then if INFO = 0, the rows of VT contain the right singular vectors of the bidiagonal matrix B.

If COMPQ $\neq$ 'I', VT is not referenced.

9: LDVT – INTEGER *Input*

On entry: the first dimension of the array VT as declared in the (sub)program from which F08MDF (DBDSDC) is called.

Constraints:

if COMPQ = 'I', LDVT $\geq$ max(1, N);
LDVT $\geq$ 1 otherwise.

10: Q($*$) – ***double precision*** array *Output*

Note: the dimension of the array Q must be at least max$(1, LDQ)$, where LDQ is defined below.

On exit: if COMPQ = 'P', then if INFO = 0, Q and IQ contain the left and right singular vectors in a compact form, requiring $O(\mathrm{N}\log\mathrm{N})$ space instead of $2 \times \mathrm{N}^2$. In particular, Q contains all the real data in $LDQ = \mathrm{N} \times (11 + 2 \times smlsiz + 8 \times \mathrm{int}(\log_2(\mathrm{N}/(smlsiz + 1))))$ words of memory, where $smlsiz$ is equal to the maximum size of the subproblems at the bottom of the computation tree (usually about 25).

If COMPQ $\neq$ 'P', Q is not referenced.

11: IQ($*$) – INTEGER array *Output*

Note: the dimension of the array IQ must be at least max$(1, LDIQ)$, where $LDIQ$ is defined below.

On exit: if COMPQ = 'P', then if INFO = 0, Q and IQ contain the left and right singular vectors in a compact form, requiring $O(\mathrm{N}\log\mathrm{N})$ space instead of $2 \times \mathrm{N}^2$. In particular, IQ contains all integer data in $LDIQ = \mathrm{N} \times (3 + 3 \times \mathrm{int}(\log_2(\mathrm{N}/(smlsiz + 1))))$ words of memory, where $smlsiz$ is equal to the maximum size of the subproblems at the bottom of the computation tree (usually about 25).

If COMPQ $\neq$ 'P', IQ is not referenced.

12: WORK($*$) – ***double precision*** array *Workspace*

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

Constraints:

if COMPQ = 'N', $LWORK \geq$ max$(1, 4 \times \mathrm{N})$;
if COMPQ = 'P', $LWORK \geq$ max$(1, 6 \times \mathrm{N})$;
if COMPQ = 'I', $LWORK \geq \max\left(1, 3 \times \mathrm{N}^2 + 4 \times \mathrm{N}\right)$.

13: IWORK($*$) – INTEGER array *Workspace*

Note: the dimension of the array IWORK must be at least max$(1, 8 \times \mathrm{N})$.

14: 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.

INFO > 0

The algorithm failed to compute a singular value. The update process of divide-and-conquer failed.

7 Accuracy

Each computed singular value of B is accurate to nearly full relative precision, no matter how tiny the singular value. The ith computed singular value, $\hat{s}_i$, satisfies the bound

$$|\hat{s}_i - s_i| \leq p(n)\epsilon s_i$$

where ϵ is the ***machine precision*** and $p(n)$ is a modest function of n.

For bounds on the computed singular values, see Anderson *et al.* (1999) (Section 4.9.1) . See also F08FLF (DDISNA).

8 Further Comments

If only singular values are required, the total number of floating point operations is approximately proportional to n^2. When singular vectors are required the number of operations is bounded above by approximately the same number of operations as F08MEF (DBDSQR), but for large matrices F08MDF (DBDSDC) is usually much faster.

There is no complex analogue of F08MDF (DBDSDC).

9 Example

This example computes the singular value decomposition of the upper bidiagonal matrix

$$B = \begin{pmatrix} 3.62 & 1.26 & 0 & 0 \\ 0 & -2.41 & -1.53 & 0 \\ 0 & 0 & 1.92 & 1.19 \\ 0 & 0 & 0 & -1.43 \end{pmatrix}.$$

9.1 Program Text

```
*     F08MDF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=10)
      INTEGER          LDU, LDVT
      PARAMETER        (LDU=NMAX,LDVT=NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, N
*     .. Local Arrays ..
      DOUBLE PRECISION D(NMAX), E(NMAX-1), Q(1), U(LDU,NMAX),
     +                 VT(LDVT,NMAX), WORK(NMAX*(3*NMAX+2))
      INTEGER          IQ(1), IWORK(7*NMAX)
*     .. External Subroutines ..
      EXTERNAL         DBDSDC, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08MDF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read the bidiagonal matrix B from data file, first
*        the diagonal elements, and then the off diagonal elements
*
         READ (NIN,*) (D(I),I=1,N)
         READ (NIN,*) (E(I),I=1,N-1)
*
*        Calculate the singular values and left and right singular
*        vectors of B.
*
         CALL DBDSDC('Upper','I',N,D,E,U,LDU,VT,LDVT,Q,IQ,WORK,IWORK,
     +               INFO)
*
         IF (INFO.EQ.0) THEN
*
```

```
*           Print singular values and vectors
*
            IFAIL = 0
            CALL X04CAF('General',' ',N,N,U,LDU,
     +                  'Left singular vectors, by column',IFAIL)
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Singular values'
            WRITE (NOUT,99999) (D(I),I=1,N)
*
            WRITE (NOUT,*)
            IFAIL = 0
            CALL X04CAF('General',' ',N,N,VT,LDVT,
     +                  'Right singular vectors, by row',IFAIL)
*
         ELSE
            WRITE (NOUT,99998)
     +        'Failure to compute a singular value, INFO = ', INFO
         END IF
      ELSE
         WRITE (NOUT,*) 'NMAX too small'
      END IF
      STOP
*
99999 FORMAT ((3X,8F8.4))
99998 FORMAT (1X,A,I10)
      END
```

9.2 Program Data

```
F08MDF Example Program Data

  4                         :Value of N

  3.62  -2.41   1.92  -1.43  :End of diagonal elements
  1.26  -1.53   1.19         :End of off-diagonal elements
```

9.3 Program Results

```
 F08MDF Example Program Results

 Left singular vectors, by column
          1       2       3       4
 1   0.9129  0.3740  0.1556  0.0512
 2  -0.3935  0.7005  0.5489  0.2307
 3   0.1081 -0.5904  0.6173  0.5086
 4  -0.0132  0.1444 -0.5417  0.8280

 Singular values
     4.0001  3.0006  1.9960  0.9998

 Right singular vectors, by row
          1       2       3       4
 1   0.8261  0.5246  0.2024  0.0369
 2   0.4512 -0.4056 -0.7350 -0.3030
 3   0.2823 -0.5644  0.1731  0.7561
 4   0.1852 -0.4916  0.6236 -0.5789
```