NAG Fortran Library Routine Document

F08KDF (DGESDD)

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

F08KDF (DGESDD) computes the singular value decomposition (SVD) of a real m by n matrix A, optionally computing the left and right singular vectors. If singular vectors are desired, it uses a divide-and-conquer algorithm.

2 Specification

```
      SUBROUTINE F08KDF (JOBZ, M, N, A, LDA, S, U, LDU, VT, LDVT, WORK, LWORK,
     1                   IWORK, INFO)
      INTEGER            M, N, LDA, LDU, LDVT, LWORK, IWORK(*), INFO
      double precision   A(LDA,*), S(*), U(LDU,*), VT(LDVT,*), WORK(*)
      CHARACTER*1        JOBZ
```

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

3 Description

The SVD is written as

$$A = U\Sigma V^T,$$

where Σ is an m by n matrix which is zero except for its $\min(m,n)$ diagonal elements, U is an m by m orthogonal matrix, and V is an n by n orthogonal matrix. The diagonal elements of Σ are the singular values of A; they are real and non-negative, and are returned in descending order. The first $\min(m,n)$ columns of U and V are the left and right singular vectors of A.

Note that the routine returns V^T, not V.

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: JOBZ – CHARACTER*1 *Input*

On entry: specifies options for computing all or part of the matrix U:

if JOBZ = 'A', all m columns of U and all n rows of V^T are returned in the arrays U and VT;
if JOBZ = 'S', the first $\min(m,n)$ columns of U and the first $\min(m,n)$ rows of V^T are returned in the arrays U and VT;
if JOBZ = 'O', if $m \geq n$, the first n columns of U are overwritten on the array A and all rows of V^T are returned in the array VT;

otherwise, all columns of U are returned in the array U and the first m rows of V^T are overwritten in the array VT;

if JOBZ = 'N', no columns of U or rows of V^T are computed.

Constraint: JOBZ = 'A', 'S', 'O' or 'N'.

2: M – INTEGER *Input*

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

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

3: N – INTEGER *Input*

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

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

4: A(LDA,$*$) – ***double precision*** 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 JOBZ = 'O', A is overwritten with the first n columns of U (the left singular vectors, stored columnwise) if $m \geq n$; A is overwritten with the first m rows of V^T (the right singular vectors, stored rowwise) otherwise.

If JOBZ $\neq$ 'O', the contents of A are destroyed.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08KDF (DGESDD) is called.

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

6: S($*$) – ***double precision*** array *Output*

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

On exit: the singular values of A, sorted so that $\mathrm{S}(i) \geq \mathrm{S}(i+1)$.

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

Note: the second dimension of the array U must be at least $\max(1, ucol)$, where $ucol$ is the number of columns of U requested.

On exit: $ucol = \mathrm{M}$ if JOBZ = 'A' or JOBZ = 'O' and $\mathrm{M} < \mathrm{N}$; $ucol = \min(\mathrm{M}, \mathrm{N})$ if JOBZ = 'S'.

If JOBZ = 'A' or JOBZ = 'O' and $\mathrm{M} < \mathrm{N}$, U contains the m by m orthogonal matrix U.

If JOBZ = 'S', U contains the first $\min(m, n)$ columns of U (the left singular vectors, stored columnwise).

If JOBZ = 'O' and $\mathrm{M} \geq \mathrm{N}$, or JOBZ = 'N', U is not referenced.

8: LDU – INTEGER *Input*

On entry: the first dimension of the array U as declared in the (sub)program from which F08KDF (DGESDD) is called.

Constraints:

if JOBZ = 'S' or 'A' or JOBZ = 'O' and $\mathrm{M} < \mathrm{N}$, $\mathrm{LDU} \geq \max(1, \mathrm{M})$;
$\mathrm{LDU} \geq 1$ otherwise.

9: 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 JOBZ = 'A' or JOBZ = 'O' and $\mathrm{M} \geq \mathrm{N}$, VT contains the n by n orthogonal matrix V^T.

If JOBZ = 'S', VT contains the first $\min(m, n)$ rows of V^T (the right singular vectors, stored rowwise).

If JOBZ = 'O' and $\mathrm{M} < \mathrm{N}$, or JOBZ = 'N', VT is not referenced.

10: LDVT – INTEGER *Input*

On entry: the first dimension of the array VT as declared in the (sub)program from which F08KDF (DGESDD) is called.

Constraints:

if JOBZ = 'A' or JOBZ = 'O' and $\mathrm{M} \geq \mathrm{N}$, $\mathrm{LDVT} \geq \max(1, \mathrm{N})$;
if JOBZ = 'S', $\mathrm{LDVT} \geq \max(1, \min(\mathrm{M}, \mathrm{N}))$;
$\mathrm{LDVT} \geq 1$ otherwise.

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

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

On exit: if INFO = 0, WORK(1) returns the optimal LWORK.

If $\mathrm{INFO} > 0$, $\mathrm{WORK}(2 : \min(\mathrm{M}, \mathrm{N}))$ contains the unconverged super-diagonal elements of an upper bidiagonal matrix B whose diagonal is in S (not necessarily sorted). B satisfies $A = UBV^T$, so it has the same singular values as A, and singular vectors related by U and V^T.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08KDF (DGESDD) is called.

For good performance, LWORK should generally be larger. Consider increasing LWORK by at least $nb \times \min(\mathrm{M}, \mathrm{N})$, where nb is the optimal block size.

If $\mathrm{LWORK} = -1$ but other input arguments are legal, WORK(1) returns the optimal LWORK.

Constraints:

if JOBZ = 'N', $\mathrm{LWORK} \geq 3 \times \min(\mathrm{M}, \mathrm{N}) + \max(1, \max(\mathrm{M}, \mathrm{N}), 6 \times \min(\mathrm{M}, \mathrm{N}))$;
if JOBZ = 'O',
$\mathrm{LWORK} \geq 3 \times \min(\mathrm{M}, \mathrm{N}) \times \min(\mathrm{M}, \mathrm{N}) +$
$\max(1, \max(\mathrm{M}, \mathrm{N}), 5 \times \min(\mathrm{M}, \mathrm{N}) \times \min(\mathrm{M}, \mathrm{N}) + 4 \times \min(\mathrm{M}, \mathrm{N}))$;
if JOBZ = 'S' or 'A',
$\mathrm{LWORK} \geq 3 \times \min(\mathrm{M}, \mathrm{N}) \times \min(\mathrm{M}, \mathrm{N}) +$
$\max(1, \max(\mathrm{M}, \mathrm{N}), 4 \times \min(\mathrm{M}, \mathrm{N}) \times \min(\mathrm{M}, \mathrm{N}) + 4 \times \min(\mathrm{M}, \mathrm{N}))$;
$\mathrm{LWORK} \geq 1$ otherwise.

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

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

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 argument had an illegal value.

INFO > 0

F08KDF (DGESDD) did not converge, the updating process failed.

7 Accuracy

The computed singular value decomposition is nearly the exact singular value decomposition for a nearby matrix $(A + E)$, where

$$\|E\|_2 = \mathrm{O}(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***. In addition, the computed singular vectors are nearly orthogonal to working precision. See Section 4.9 of Anderson *et al.* (1999) for further details.

8 Further Comments

The total number of floating point operations is approximately proportional to mn^2 when $m > n$ and m^2n otherwise.

The singular values are returned in descending order.

The complex analogue of this routine is F08KPF (ZGESVD).

9 Example

To find the singular values and left and right singular vectors of the 4 by 6 matrix

$$A = \begin{pmatrix} 2.27 & 0.28 & -0.48 & 1.07 & -2.35 & 0.62 \\ -1.54 & -1.67 & -3.09 & 1.22 & 2.93 & -7.39 \\ 1.15 & 0.94 & 0.99 & 0.79 & -1.45 & 1.03 \\ -1.94 & -0.78 & -0.21 & 0.63 & 2.30 & -2.57 \end{pmatrix},$$

together with approximate error bounds for the computed singular values and vectors.

The example program for F08KBF (DGESVD) illustrates finding a singular value decomposition for the case $m \geq n$.

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.

```
*     F08KDF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NB, NMAX
      PARAMETER        (MMAX=8,NB=64,NMAX=10)
      INTEGER          LDA, LDU, LWORK
      PARAMETER        (LDA=MMAX,LDU=MMAX,LWORK=(5*MMAX+9)
     +                 *MMAX+NMAX+NB*(MMAX+NMAX))
*     .. Local Scalars ..
      DOUBLE PRECISION EPS, SERRBD
      INTEGER          I, IFAIL, INFO, J, LWKOPT, M, N
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), DUMMY(1,1), RCONDU(MMAX),
```

```
     +                 RCONDV(MMAX), S(MMAX), U(LDU,MMAX), UERRBD(MMAX),
     +                 VERRBD(MMAX), WORK(LWORK)
      INTEGER          IWORK(8*MMAX)
*     .. External Functions ..
      DOUBLE PRECISION X02AJF
      EXTERNAL         X02AJF
*     .. External Subroutines ..
      EXTERNAL         DDISNA, DGESDD, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08KDF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N
      IF (M.LE.MMAX .AND. N.LE.NMAX) THEN
*
*        Read the m by n matrix A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
*
*        Compute the singular values and left and right singular vectors
*        of A (A = U*S*(V**T), m.le.n)
*
         CALL DGESDD('Overwrite A by tranpose(V)',M,N,A,LDA,S,U,LDU,
     +               DUMMY,1,WORK,LWORK,IWORK,INFO)
         LWKOPT = WORK(1)
*
         IF (INFO.EQ.0) THEN
*
*           Print solution
*
            WRITE (NOUT,*) 'Singular values'
            WRITE (NOUT,99999) (S(J),J=1,M)
*
            IFAIL = 0
            CALL X04CAF('General',' ',M,M,U,LDU,'Left singular vectors',
     +                  IFAIL)
            WRITE (NOUT,*)
            CALL X04CAF('General',' ',M,N,A,LDA,
     +                  'Right singular vectors by row '//
     +                  '(first m rows of V**T)',IFAIL)
*
*           Get the machine precision, EPS and compute the approximate
*           error bound for the computed singular values.  Note that for
*           the 2-norm, S(1) = norm(A)
*
            EPS = X02AJF()
            SERRBD = EPS*S(1)
*
*           Call DDISNA (F08FLF) to estimate reciprocal condition
*           numbers for the singular vectors
*
            CALL DDISNA('Left',M,N,S,RCONDU,INFO)
            CALL DDISNA('Right',M,N,S,RCONDV,INFO)
*
*           Compute the error estimates for the singular vectors
*
            DO 20 I = 1, M
               UERRBD(I) = SERRBD/RCONDU(I)
               VERRBD(I) = SERRBD/RCONDV(I)
   20       CONTINUE
*
*           Print the approximate error bounds for the singular values
*           and vectors
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Error estimate for the singular values'
            WRITE (NOUT,99998) SERRBD
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'Error estimates for the left singular vectors'
```

```
            WRITE (NOUT,99998) (UERRBD(I),I=1,M)
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'Error estimates for the right singular vectors'
            WRITE (NOUT,99998) (VERRBD(I),I=1,M)
         ELSE
            WRITE (NOUT,99997) 'Failure in DGESDD. INFO =', INFO
         END IF
*
*        Print workspace information
*
         IF (LWORK.LT.LWKOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99996) 'Optimum workspace required = ', LWKOPT,
     +        'Workspace provided         = ', LWORK
         END IF
      ELSE
         WRITE (NOUT,*) 'MMAX and/or NMAX too small'
      END IF
      STOP
*
99999 FORMAT (3X,(8F8.4))
99998 FORMAT (4X,1P,6E11.1)
99997 FORMAT (1X,A,I4)
99996 FORMAT (1X,A,I5,/1X,A,I5)
      END
```

9.2 Program Data

```
F08KDF Example Program Data

   4      6                                :Values of M and N

   2.27   0.28  -0.48   1.07  -2.35   0.62
  -1.54  -1.67  -3.09   1.22   2.93  -7.39
   1.15   0.94   0.99   0.79  -1.45   1.03
  -1.94  -0.78  -0.21   0.63   2.30  -2.57  :End of matrix A
```

9.3 Program Results

```
 F08KDF Example Program Results

 Singular values
     9.9966  3.6831  1.3569  0.5000
 Left singular vectors
          1       2       3       4
 1  -0.1921  0.8030 -0.0041  0.5642
 2   0.8794  0.3926  0.0752 -0.2587
 3  -0.2140  0.2980 -0.7827 -0.5027
 4   0.3795 -0.3351 -0.6178  0.6017

 Right singular vectors by row (first m rows of V**T)
          1       2       3       4       5       6
 1  -0.2774 -0.2020 -0.2918  0.0938  0.4213 -0.7816
 2   0.6003  0.0301 -0.3348  0.3699 -0.5266 -0.3353
 3   0.1277 -0.2805 -0.6453 -0.6781 -0.0413  0.1645
 4  -0.1323 -0.7034 -0.1906  0.5399  0.0575  0.3957

 Error estimate for the singular values
        1.1E-15

 Error estimates for the left singular vectors
        1.8E-16    4.8E-16    1.3E-15    1.3E-15

 Error estimates for the right singular vectors
        1.8E-16    4.8E-16    1.3E-15    2.2E-15
```