NAG 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/or right singular vectors, by using a divide-and-conquer method.

2 Specification

```
SUBROUTINE F08KDF (JOBZ, M, N, A, LDA, S, U, LDU, VT, LDVT, WORK, LWORK,        &
                   IWORK, INFO)

INTEGER            M, N, LDA, LDU, LDVT, LWORK, IWORK(8*min(M,N)), INFO
REAL (KIND=nag_wp) A(LDA,*), S(min(M,N)), U(LDU,*), VT(LDVT,*),                 &
                   WORK(max(1,LWORK))
CHARACTER(1)       JOBZ
```

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

3 Description

The SVD is written as

$$A = U\Sigma V^{\mathrm{T}},$$

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 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.

JOBZ = 'A'
: All m columns of U and all n rows of V^{T} are returned in the arrays U and VT.

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.

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.

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 matrix A.

Constraint: M $\geq$ 0.

3: N – INTEGER *Input*

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

Constraint: N $\geq$ 0.

4: A(LDA,$*$) – REAL (KIND=nag_wp) 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 column-wise) if M $\geq$ N; A is overwritten with the first m rows of V^{T} (the right singular vectors, stored row-wise) 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: LDA $\geq \max(1, \mathrm{M})$.

6: S($\min(\mathrm{M}, \mathrm{N})$) – REAL (KIND=nag_wp) array *Output*

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

7: U(LDU,$*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array U must be at least $\max(1, \mathrm{M})$ if JOBZ = 'A' or JOBZ = 'O' and M $<$ N, $\max(1, \min(\mathrm{M}, \mathrm{N}))$ if JOBZ = 'S', and at least 1 otherwise.

On exit: If JOBZ = 'A' or JOBZ = 'O' and M $<$ 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 column-wise).

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

9: VT(LDVT,$*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array VT must be at least $\max(1,\mathrm{N})$ if JOBZ = 'A' or 'S' or JOBZ = 'O' and $\mathrm{M} \geq \mathrm{N}$, and at least 1 otherwise.

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 row-wise).

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', $\mathrm{LDVT} \geq \max(1,\min(\mathrm{M},\mathrm{N}))$;
otherwise $\mathrm{LDVT} \geq 1$.

11: WORK($\max(1,\mathrm{LWORK})$) – REAL (KIND=nag_wp) array *Workspace*

12: LWORK – INTEGER *Input*

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

If $\mathrm{LWORK} = -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Suggested value: for optimal 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***.

Constraints:

if JOBZ = 'N', $\mathrm{LWORK} \geq 3 \times \min(\mathrm{M},\mathrm{N}) + \max(1,\max(\mathrm{M},\mathrm{N}),7 \times \min(\mathrm{M},\mathrm{N}))$;
if JOBZ = 'O', $\mathrm{LWORK} \geq 3 \times \min(\mathrm{M},\mathrm{N}) +$
$\max(1,\max(\mathrm{M},\mathrm{N}),5 \times \min(\mathrm{M},\mathrm{N}) \times \min(\mathrm{M},\mathrm{N}) + 4 \times \min(\mathrm{M},\mathrm{N}))$;
if JOBZ = 'S' or 'A', $\mathrm{LWORK} \geq 3 \times \min(\mathrm{M},\mathrm{N}) +$
$\max(1,\max(\mathrm{M},\mathrm{N}),4 \times \min(\mathrm{M},\mathrm{N}) \times \min(\mathrm{M},\mathrm{N}) + 4 \times \min(\mathrm{M},\mathrm{N}))$;
otherwise $\mathrm{LWORK} \geq 1$.

13: IWORK($8 \times \min(\mathrm{M},\mathrm{N})$) – INTEGER array *Workspace*

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 $\mathrm{INFO} = -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

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

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

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

This example finds 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

```
    Program f08kdfe

!     F08KDF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: ddisna, dgesdd, nag_wp, x02ajf, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: eps, serrbd
      Integer                          :: i, ifail, info, lda, ldu, lwork, m, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), rcondu(:), rcondv(:), s(:),  &
                                          u(:,:), uerrbd(:), verrbd(:), work(:)
      Real (Kind=nag_wp)               :: dummy(1,1)
      Integer, Allocatable             :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, min, nint
!     .. Executable Statements ..
      Write (nout,*) 'F08KDF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      lda = m
      ldu = m
      Allocate (a(lda,n),rcondu(m),rcondv(m),s(m),u(ldu,m),uerrbd(m), &
        verrbd(m),iwork(8*min(m,n)))
```

```
!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of dgesdd is f08kdf
      Call dgesdd('Overwrite A by tranpose(V)',m,n,a,lda,s,u,ldu,dummy,1, &
        dummy,lwork,iwork,info)

!     Make sure that there is enough workspace for blocksize nb.
      lwork = max((5*m+9)*m+n+nb*(m+n),nint(dummy(1,1)))
      Allocate (work(lwork))


!     Read the m by n matrix A from data file

      Read (nin,*)(a(i,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)

!     The NAG name equivalent of dgesdd is f08kdf
      Call dgesdd('Overwrite A by tranpose(V)',m,n,a,lda,s,u,ldu,dummy,1,work, &
        lwork,iwork,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Singular values'
        Write (nout,99999) s(1:m)
        Flush (nout)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04caf('General',' ',m,m,u,ldu,'Left singular vectors',ifail)

        Write (nout,*)
        Flush (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 i = 1, m
          uerrbd(i) = serrbd/rcondu(i)
          verrbd(i) = serrbd/rcondv(i)
        End Do

!       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(1:m)
```

```
        Write (nout,*)
        Write (nout,*) 'Error estimates for the right singular vectors'
        Write (nout,99998) verrbd(1:m)
      Else
        Write (nout,99997) 'Failure in DGESDD. INFO =', info
      End If

99999 Format (3X,(8F8.4))
99998 Format (4X,1P,6E11.1)
99997 Format (1X,A,I4)
    End Program f08kdfe
```

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
```