

NAG Fortran Library Routine Document

F08KRF (ZGESDD)

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

F08KRF (ZGESDD) computes the singular value decomposition (SVD) of a complex m by n matrix A , optionally computing the left and/or right singular vectors, by using divide-and-conquer method.

2 Specification

```

SUBROUTINE F08KRF (JOBZ, M, N, A, LDA, S, U, LDU, VT, LDVT, WORK, LWORK,
1                RWORK, IWORK, INFO)
    INTEGER          M, N, LDA, LDU, LDVT, LWORK, IWORK(*), INFO
    double precision S(*), RWORK(*)
    complex*16      A(LDA,*), U(LDU,*), VT(LDVT,*), WORK(*)
    CHARACTER*1     JOBZ
  
```

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

3 Description

The SVD is written as

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

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 unitary matrix, and V is an n by n unitary 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^H , 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^H 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^H 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^H are returned in the array VT ;

otherwise, all columns of U are returned in the array U and the first m rows of V^H are overwritten in the array VT ;
 if $JOBZ = 'N'$, no columns of U or rows of V^H 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: $M \geq 0$.
- 3: N – INTEGER *Input*
On entry: n , the number of columns of the input matrix A .
Constraint: $N \geq 0$.
- 4: $A(LDA,*)$ – **complex*16** array *Input/Output*
Note: the second dimension of the array A must be at least $\max(1, N)$.
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^H (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 F08KRF (ZGESDD) is called.
Constraint: $LDA \geq \max(1, M)$.
- 6: $S(*)$ – **double precision** array *Output*
Note: the dimension of the array S must be at least $\max(1, \min(M, N))$.
On exit: the singular values of A , sorted so that $S(i) \geq S(i + 1)$.
- 7: $U(LDU,*)$ – **complex*16** 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: $ucol = M$ if $JOBZ = 'A'$ or $JOBZ = 'O'$ and $M < N$; $ucol = \min(M, N)$ if $JOBZ = 'S'$.
 If $JOBZ = 'A'$ or $JOBZ = 'O'$ and $M < N$, U contains the m by m unitary 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 $M \geq N$, 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 F08KRF (ZGESDD) is called.
Constraints:
 if $JOBZ = 'S'$ or $'A'$ or $JOBZ = 'O'$ and $M < N$, $LDU \geq \max(1, M)$;
 $LDU \geq 1$ otherwise.

- 9: VT(LDVT,*) – **complex*16** array *Output*
Note: the second dimension of the array VT must be at least $\max(1, N)$.
On exit: if JOBZ = 'A' or JOBZ = 'O' and $M \geq N$, VT contains the n by n unitary matrix V^H .
 If JOBZ = 'S', VT contains the first $\min(m, n)$ rows of V^H (the right singular vectors, stored rowwise).
 If JOBZ = 'O' and $M < N$, 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 F08KRF (ZGESDD) is called.
Constraints:
 if JOBZ = 'A' or JOBZ = 'O' and $M \geq N$, $LDVT \geq \max(1, N)$;
 if JOBZ = 'S', $LDVT \geq \max(1, \min(M, N))$;
 $LDVT \geq 1$ otherwise.
- 11: WORK(*) – **complex*16** array *Workspace*
Note: the dimension of the array WORK must be at least $\max(1, LWORK)$.
On exit: if INFO = 0, WORK(1) returns the optimal LWORK.
- 12: LWORK – INTEGER *Input*
On entry: the dimension of the array WORK as declared in the (sub)program from which F08KRF (ZGESDD) is called.
 For good performance, LWORK should generally be larger. Consider increasing LWORK by at least $nb \times \min(M, N)$, where nb is the optimal block size.
 If LWORK = -1 but other input arguments are legal, WORK(1) returns the optimal LWORK.
Constraints:
 if JOBZ = 'N', $LWORK \geq 2 \times \min(M, N) + \max(1, M, N)$;
 if JOBZ = 'O', $LWORK \geq 2 \times \min(M, N) \times \min(M, N) + 2 \times \min(M, N) + \max(1, M, N)$;
 if JOBZ = 'S' or 'A',
 $LWORK \geq \min(M, N) \times \min(M, N) + 2 \times \min(M, N) + \max(1, M, N)$;
 $LWORK \geq 1$ otherwise.
- 13: RWORK(*) – **double precision** array *Workspace*
Note: the dimension of the array RWORK must be at least $\max(1, lrwork)$.
 If JOBZ = 'N', $lrwork \geq 7 \times \min(M, N)$.
 Otherwise, $lrwork \geq 5 \times \min(M, N) \times \min(M, N) + 5 \times \min(M, N)$.
- 14: IWORK(*) – INTEGER array *Workspace*
Note: the dimension of the array IWORK must be at least $\max(1, 8 \times \min(M, N))$.
- 15: 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 i th argument had an illegal value.

INFO > 0

The updating process of F08KRF (ZGESDD) did not converge.

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 real analogue of this routine is F08KDF (DGESDD).

9 Example

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

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

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

The example program for F08KPF (ZGESVD) 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.

```
*      F08KRF 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=(2*MMAX+2)
+                    *MMAX+NMAX+NB*(MMAX+NMAX))
*      .. Local Scalars ..
      DOUBLE PRECISION EPS, SERRBD
      INTEGER          I, IFAIL, INFO, J, LWKOPT, M, N
*      .. Local Arrays ..
      COMPLEX *16      A(LDA, NMAX), DUMMY(1, 1), U(LDU, MMAX), WORK(LWORK)
```

```

      DOUBLE PRECISION RCONDU(MMAX), RCONDV(MMAX),
+       RWORK((5*MMAX+7)*MMAX), S(MMAX), UERRBD(MMAX),
+       VERRBD(MMAX)
      INTEGER          IWORK(8*MMAX)
*   .. External Functions ..
      DOUBLE PRECISION X02AJF
      EXTERNAL          X02AJF
*   .. External Subroutines ..
      EXTERNAL          DDISNA, X04DAF, ZGESDD
*   .. Executable Statements ..
      WRITE (NOUT,*) 'F08KRF 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**H), m.le.n)
*
      CALL ZGESDD('Overwrite A by V**H',M,N,A,LDA,S,U,LDU,DUMMY,1,
+       WORK,LWORK,RWORK,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 X04DAF('General',' ',M,M,U,LDU,'Left singular vectors',
+       IFAIL)
*       WRITE (NOUT,*)
*       CALL X04DAF('General',' ',M,N,A,LDA,
+       'Conjugates of right singular vectors by row '//
+       '(first m rows of V**H)',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 ZGESDD. 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

F08KRF Example Program Data

```

      4              6                      :Values of M and N

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

```

9.3 Program Results

F08KRF Example Program Results

```

Singular values
  3.9994  3.0003  1.9944  0.9995
Left singular vectors
      1      2      3      4
1  0.6971  0.2403 -0.5123 -0.4403
   0.0000  0.0000  0.0000  0.0000

2  0.0867  0.0725 -0.3030  0.5294
   -0.3548  0.2336  0.1735 -0.6361

3  -0.0560 -0.2477  0.0678 -0.3027
   -0.5400  0.5291 -0.5162  0.0346

4  0.1878  0.7026  0.4418  0.1667
   -0.2253 -0.2177 -0.3864 -0.0258

Conjugates of right singular vectors by row (first m rows of V**H)
      1      2      3      4      5      6
1  0.5634 -0.1205  0.0816 -0.1441  0.2487  0.3758
   0.0016 -0.6108  0.1613 -0.1532 -0.0926  0.0793

2  -0.2687 -0.2909 -0.1660  0.1984  0.6253 -0.0307
   0.2749 -0.1085 -0.3885  0.1737 -0.3304  0.0816

```

3 0.2451 0.4329 -0.4667 -0.0034 0.2643 0.1266
-0.4657 0.1758 -0.3821 -0.1555 0.0194 -0.1747

4 0.3787 -0.0182 -0.0800 0.2608 0.1002 -0.4175
-0.2987 0.0437 0.2276 0.5382 -0.0140 0.4058

Error estimate for the singular values
4.4E-16

Error estimates for the left singular vectors
4.4E-16 4.4E-16 4.5E-16 4.5E-16

Error estimates for the right singular vectors
4.4E-16 4.4E-16 4.5E-16 4.5E-16
