NAG Fortran Library Routine Document

F08KNF (ZGELSS)

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

F08KNF (ZGELSS) computes the minimum norm solution to a complex linear least-squares problem

$$\min_x \|b - Ax\|_2.$$

2 Specification

```
      SUBROUTINE F08KNF (M, N, NRHS, A, LDA, B, LDB, S, RCOND, RANK, WORK,
     1                   LWORK, RWORK, INFO)
      INTEGER            M, N, NRHS, LDA, LDB, RANK, LWORK, INFO
      double precision   S(*), RCOND, RWORK(*)
      complex*16         A(LDA,*), B(LDB,*), WORK(*)
```

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

3 Description

F08KNF (ZGELSS) uses the singular value decomposition (SVD) of A, where A is an m by n matrix which may be rank-deficient.

Several right-hand side vectors b and solution vectors x can be handled in a single call; they are stored as the columns of the m by r right-hand side matrix B and the n by r solution matrix X.

The effective rank of A is determined by treating as zero those singular values which are less than RCOND times the largest singular value.

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: 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: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides, i.e., the number of columns of the matrices B and X.

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

4: $\mathrm{A}(\mathrm{LDA},*)$ – ***complex*16*** 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: the first $\min(m, n)$ rows of A are overwritten with its right singular vectors, stored rowwise.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08KNF (ZGELSS) is called.

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

6: $\mathrm{B}(\mathrm{LDB},*)$ – ***complex*16*** array *Input/Output*

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

On entry: the m by r right-hand side matrix B.

On exit: is overwritten by the n by r solution matrix X. If $m \geq n$ and $\mathrm{RANK} = n$, the residual sum-of-squares for the solution in the ith column is given by the sum of squares of the modulus of elements $n+1, \ldots, m$ in that column.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08KNF (ZGELSS) is called.

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

8: $\mathrm{S}(*)$ – ***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 in decreasing order.

9: RCOND – ***double precision*** *Input*

On entry: used to determine the effective rank of A. Singular values $\mathrm{S}(i) \leq \mathrm{RCOND} \times \mathrm{S}(1)$ are treated as zero.

If $\mathrm{RCOND} < 0$, ***machine precision*** is used instead.

10: RANK – INTEGER *Output*

On exit: the effective rank of A, i.e., the number of singular values which are greater than $\mathrm{RCOND} \times \mathrm{S}(1)$.

11: $\mathrm{WORK}(*)$ – ***complex*16*** array *Workspace*

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

On exit: if $\mathrm{INFO} = 0$, $\mathrm{WORK}(1)$ returns the optimal LWORK.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08KNF (ZGELSS) 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 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.

Constraint: LWORK ≥ 1 and LWORK $\geq 2 \times \min(\mathrm{M}, \mathrm{N}) + \max(\mathrm{M}, \mathrm{N}, \mathrm{NRHS})$.

13: RWORK($*$) – ***double precision*** array *Workspace*

Note: the dimension of the array RWORK must be at least $\max(1, 5 \times \min(\mathrm{M}, \mathrm{N}))$.

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

The algorithm for computing the SVD failed to converge; if INFO $= i$, i off-diagonal elements of an intermediate bidiagonal form did not converge to zero.

7 Accuracy

See Section 4.5 of Anderson *et al.* (1999) for details.

8 Further Comments

The real analogue of this routine is F08KAF (DGELSS).

9 Example

To solve the linear least squares problem

$$\min_x \|b - Ax\|_2$$

for the solution, x, of minimum norm, where

$$A = \begin{pmatrix} 0.47 - 0.34i & -0.40 + 0.54i & 0.60 + 0.01i & 0.80 - 1.02i \\ -0.32 - 0.23i & -0.05 + 0.20i & -0.26 - 0.44i & -0.43 + 0.17i \\ 0.35 - 0.60i & -0.52 - 0.34i & 0.87 - 0.11i & -0.34 - 0.09i \\ 0.89 + 0.71i & -0.45 - 0.45i & -0.02 - 0.57i & 1.14 - 0.78i \\ -0.19 + 0.06i & 0.11 - 0.85i & 1.44 + 0.80i & 0.07 + 1.14i \end{pmatrix}$$

and

$$b = \begin{pmatrix} -1.08 - 2.59i \\ -2.61 - 1.49i \\ 3.13 - 3.61i \\ 7.33 - 8.01i \\ 9.12 + 7.63i \end{pmatrix}.$$

A tolerance of 0.01 is used to determine the effective rank of A.

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

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.

```
*     F08KNF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NB, NMAX
      PARAMETER        (MMAX=16,NB=64,NMAX=8)
      INTEGER          LDA, LWORK
      PARAMETER        (LDA=MMAX,LWORK=2*NMAX+NB*(MMAX+NMAX))
*     .. Local Scalars ..
      DOUBLE PRECISION RCOND, RNORM
      INTEGER          I, INFO, J, M, N, RANK
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(MMAX), WORK(LWORK)
      DOUBLE PRECISION RWORK(5*NMAX), S(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION DZNRM2
      EXTERNAL         DZNRM2
*     .. External Subroutines ..
      EXTERNAL         ZGELSS
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08KNF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. M.GE.N) THEN
*
*        Read A and B from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) (B(I),I=1,M)
*
*        Choose RCOND to reflect the relative accuracy of the input data
*
         RCOND = 0.01D0
*
*        Solve the least squares problem min( norm2(b - Ax) ) for the x
*        of minimum norm.
*
         CALL ZGELSS(M,N,1,A,LDA,B,M,S,RCOND,RANK,WORK,LWORK,RWORK,INFO)
*
         IF (INFO.EQ.0) THEN
*
*           Print solution
*
            WRITE (NOUT,*) 'Least squares solution'
            WRITE (NOUT,99999) (B(I),I=1,N)
*
*           Print the effective rank of A
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Tolerance used to estimate the rank of A'
            WRITE (NOUT,99998) RCOND
            WRITE (NOUT,*) 'Estimated rank of A'
            WRITE (NOUT,99997) RANK
*
*           Print singular values of A
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Singular values of A'
```

```
            WRITE (NOUT,99996) (S(I),I=1,N)
*
*           Compute and print estimate of the square root of the
*           residual sum of squares
*
            IF (RANK.EQ.N) THEN
               RNORM = DZNRM2(M-N,B(N+1),1)
               WRITE (NOUT,*)
               WRITE (NOUT,*)
     +           'Square root of the residual sum of squares'
               WRITE (NOUT,99998) RNORM
            END IF
         ELSE
            WRITE (NOUT,*) 'The SVD algorithm failed to converge'
         END IF
      ELSE
         WRITE (NOUT,*) 'MMAX and/or NMAX too small, and/or M.LT.N'
      END IF
      STOP
*
99999 FORMAT (4(' (',F7.4,',',F7.4,')',:))
99998 FORMAT (3X,1P,E11.2)
99997 FORMAT (1X,I6)
99996 FORMAT (1X,7F11.4)
      END
```

9.2 Program Data

```
F08KNF Example Program Data

   5           4                                       :Values of M and N

 ( 0.47,-0.34) (-0.40, 0.54) ( 0.60, 0.01) ( 0.80,-1.02)
 (-0.32,-0.23) (-0.05, 0.20) (-0.26,-0.44) (-0.43, 0.17)
 ( 0.35,-0.60) (-0.52,-0.34) ( 0.87,-0.11) (-0.34,-0.09)
 ( 0.89, 0.71) (-0.45,-0.45) (-0.02,-0.57) ( 1.14,-0.78)
 (-0.19, 0.06) ( 0.11,-0.85) ( 1.44, 0.80) ( 0.07, 1.14) :End of matrix A

 (-1.08,-2.59)
 (-2.61,-1.49)
 ( 3.13,-3.61)
 ( 7.33,-8.01)
 ( 9.12, 7.63)                                         :End of vector b
```

9.3 Program Results

```
 F08KNF Example Program Results

 Least squares solution
 ( 1.1673,-3.3222) ( 1.3480, 5.5028) ( 4.1762, 2.3434) ( 0.6465, 0.0105)

 Tolerance used to estimate the rank of A
      1.00E-02
 Estimated rank of A
      3

 Singular values of A
      2.9979     1.9983     1.0044     0.0064
```