NAG Library Routine Document

F08KCF (DGELSD)

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

F08KCF (DGELSD) computes the minimum norm solution to a real linear least squares problem

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

2 Specification

```
SUBROUTINE F08KCF (M, N, NRHS, A, LDA, B, LDB, S, RCOND, RANK, WORK, LWORK,     &
                   IWORK, INFO)
INTEGER            M, N, NRHS, LDA, LDB, RANK, LWORK, IWORK(*), INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), S(*), RCOND, WORK(max(1,LWORK))
```

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

3 Description

F08KCF (DGELSD) uses the singular value decomposition (SVD) of A, where A is a real 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 problem is solved in three steps:

1. reduce the coefficient matrix A to bidiagonal form with Householder transformations, reducing the original problem into a 'bidiagonal least squares problem' (BLS);
2. solve the BLS using a divide-and-conquer approach;
3. apply back all the Householder transformations to solve the original least squares problem.

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 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: $NRHS \geq 0$.

4: A(LDA,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: the m by n coefficient matrix A.

On exit: 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 F08KCF (DGELSD) is called.

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

6: B(LDB,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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

On exit: B is overwritten by the n by r solution matrix X. If $m \geq n$ and $RANK = n$, the residual sum of squares for the solution in the ith column is given by the sum of squares 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 F08KCF (DGELSD) is called.

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

8: S($*$) – REAL (KIND=nag_wp) 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 in decreasing order.

9: RCOND – REAL (KIND=nag_wp) *Input*

On entry: used to determine the effective rank of A. Singular values $S(i) \leq RCOND \times S(1)$ are treated as zero. If $RCOND < 0$, ***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 $RCOND \times S(1)$.

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

On exit: if $INFO = 0$, $WORK(1)$ contains the minimum value of LWORK required for optimal performance.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08KCF (DGELSD) is called.

The exact minimum amount of workspace needed depends on M, N and NRHS. As long as LWORK is at least

$$12r + 2r \times smlsiz + 8r \times nlvl + r \times \mathrm{NRHS} + (smlsiz + 1)^2,$$

where $smlsiz$ is equal to the maximum size of the subproblems at the bottom of the computation tree (usually about 25), $nlvl = \max(0, \mathrm{int}(\log_2(\min(\mathrm{M},\mathrm{N})/(smlsiz + 1))) + 1)$ and $r = \min(\mathrm{M},\mathrm{N})$, the code will execute correctly.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array and the minimum size of the IWORK array, and returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK is issued.

Suggested value: for optimal performance, LWORK should generally be larger than the minimum required as set out above. Consider increasing LWORK by at least $nb \times \min(\mathrm{M},\mathrm{N})$, where nb is the optimal ***block size***.

Constraint:
LWORK $\geq 12r + 2r \times smlsiz + 8r \times nlvl + r \times \mathrm{NRHS} + (smlsiz + 1)^2$ or LWORK $= -1$.

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

Note: the dimension of the array IWORK must be at least $\max(1, liwork)$, where $liwork$ is at least $\max(1, 3 \times \min(\mathrm{M},\mathrm{N}) \times nlvl + 11 \times \min(\mathrm{M},\mathrm{N}))$.

On exit: if INFO $= 0$, IWORK(1) returns the minimum $liwork$.

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

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 complex analogue of this routine is F08KQF (ZGELSD).

9 Example

This example solves the linear least squares problem

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

for the solution, x, of minimum norm, where

$$A = \begin{pmatrix} -0.09 & -1.56 & -1.48 & -1.09 & 0.08 & -1.59 \\ 0.14 & 0.20 & -0.43 & 0.84 & 0.55 & -0.72 \\ -0.46 & 0.29 & 0.89 & 0.77 & -1.13 & 1.06 \\ 0.68 & 1.09 & -0.71 & 2.11 & 0.14 & 1.24 \\ 1.29 & 0.51 & -0.96 & -1.27 & 1.74 & 0.34 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 7.4 \\ 4.3 \\ -8.1 \\ 1.8 \\ 8.7 \end{pmatrix}.$$

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

```
    Program f08kcfe

!     F08KCF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dgelsd, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rcond
      Integer                          :: i, info, lda, liwork, lwork, m, n,   &
                                          rank
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:), s(:), work(:)
      Real (Kind=nag_wp)               :: lw(1)
      Integer, Allocatable             :: iwork(:)
      Integer                          :: liw(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: nint
!     .. Executable Statements ..
      Write (nout,*) 'F08KCF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      lda = m
      Allocate (a(lda,n),b(n),s(m))

!     Read A and B from data file

      Read (nin,*)(a(i,1:n),i=1,m)
      Read (nin,*) b(1:m)

!     Choose RCOND to reflect the relative accuracy of the input
!     data

      rcond = 0.01_nag_wp

!     Call f08kcf/dgelsd in workspace query mode.
      lwork = -1
!     The NAG name equivalent of dgelsd is f08kcf
      Call dgelsd(m,n,1,a,lda,b,n,s,rcond,rank,lw,lwork,liw,info)
      lwork = nint(lw(1))
      liwork = liw(1)
      Allocate (work(lwork),iwork(liwork))

!     Now Solve the least squares problem min( norm2(b - Ax) ) for the
!     x of minimum norm.
      Call dgelsd(m,n,1,a,lda,b,n,s,rcond,rank,work,lwork,iwork,info)
```

```
      If (info==0) Then

!       Print solution

        Write (nout,*) 'Least squares solution'
        Write (nout,99999) b(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,99999) s(1:m)
      Else
        Write (nout,*) 'The SVD algorithm failed to converge'
      End If

99999 Format (1X,7F11.4)
99998 Format (3X,1P,E11.2)
99997 Format (1X,I6)
    End Program f08kcfe
```

9.2 Program Data

```
F08KCF Example Program Data

  5      6                                :Values of M and N

 -0.09  -1.56  -1.48  -1.09   0.08  -1.59
  0.14   0.20  -0.43   0.84   0.55  -0.72
 -0.46   0.29   0.89   0.77  -1.13   1.06
  0.68   1.09  -0.71   2.11   0.14   1.24
  1.29   0.51  -0.96  -1.27   1.74   0.34 :End of matrix A

  7.4
  4.3
 -8.1
  1.8
  8.7                                     :End of vector b
```

9.3 Program Results

```
 F08KCF Example Program Results

 Least squares solution
      1.5938    -0.1180    -3.1501     0.1554     2.5529    -1.6730

 Tolerance used to estimate the rank of A
      1.00E-02
 Estimated rank of A
      4

 Singular values of A
      3.9997     2.9962     2.0001     0.9988     0.0025
```