NAG Library Routine Document

F08ANF (ZGELS)

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

F08ANF (ZGELS) solves linear least-squares problems of the form

$$\min_x \|b - Ax\|_2 \qquad \text{or} \qquad \min_x \|b - A^{\mathrm{H}}x\|_2,$$

where A is an m by n complex matrix of full rank, using a QR or LQ factorization of A.

2 Specification

```
SUBROUTINE F08ANF(TRANS, M, N, NRHS, A, LDA, B, LDB, WORK, LWORK, INFO)
INTEGER           M, N, NRHS, LDA, LDB, LWORK, INFO
complex*16        A(LDA,*), B(LDB,*), WORK(max(1,LWORK))
CHARACTER*1       TRANS
```

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

3 Description

The following options are provided:

1. If TRANS = 'N' and $m \geq n$: find the least-squares solution of an overdetermined system, i.e., solve the least-squares problem

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

2. If TRANS = 'N' and $m < n$: find the minimum norm solution of an underdetermined system $Ax = b$.
3. If TRANS = 'C' and $m \geq n$: find the minimum norm solution of an undetermined system $A^{\mathrm{H}}x = b$.
4. If TRANS = 'C' and $m < n$: find the least-squares solution of an overdetermined system, i.e., solve the least-squares problem

$$\min_x \|b - A^{\mathrm{H}}x\|_2.$$

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.

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, Philadelphiahttp://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: TRANS – CHARACTER*1 *Input*

On entry: if TRANS = 'N', the linear system involves A.

If TRANS = 'C', the linear system involves A^{H}.

Constraint: TRANS = 'N' or 'C'.

2: M – INTEGER *Input*

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

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

3: N – INTEGER *Input*

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

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

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

5: A(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: if $\mathrm{M} \geq \mathrm{N}$, A is overwritten by details of its QR factorization as returned by F08ASF (ZGEQRF).

If $\mathrm{M} < \mathrm{N}$, A is overwritten by details of its LQ factorization as returned by F08AVF (ZGELQF).

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08ANF (ZGELS) is called.

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

7: B(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 matrix B of right-hand side vectors, stored in columns; B is m by r if TRANS = 'N', or n by r if TRANS = 'C'.

On exit: B is overwritten by the solution vectors, x, stored in columns:

if TRANS = 'N' and $m \geq n$, or TRANS = 'C' and $m < n$, elements 1 to $\min(m, n)$ in each column of B contain the least squares solution vectors; the residual sum of squares for the solution is given by the sum of squares of the modulus of elements $(\min(m, n) + 1)$ to $\max(m, n)$ in that column;

otherwise, elements 1 to $\max(m, n)$ in each column of B contain the minimum norm solution vectors.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08ANF (ZGELS) is called.

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

9: WORK($\max(1, \mathrm{LWORK})$) – ***complex*16*** array *Workspace*

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

10: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08ANF (ZGELS) is called.

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.

Suggested value: for optimal performance, $\mathrm{LWORK} \geq \min(\mathrm{M},\mathrm{N}) + \max(1,\mathrm{M},\mathrm{N},\mathrm{NRHS}) \times nb$, where nb is the optimal ***block size***.

Constraint: $\mathrm{LWORK} \geq \min(\mathrm{M},\mathrm{N}) + \max(1,\mathrm{M},\mathrm{N},\mathrm{NRHS})$ or $\mathrm{LWORK} = -1$.

11: 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

If INFO $= i$, diagonal element i of the triangular factor of A is zero, so that A does not have full rank; the least squares solution could not be computed.

7 Accuracy

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

8 Further Comments

The total number of floating-point operations required to factorize A is approximately $\frac{8}{3}n^2(3m-n)$ if $m \geq n$ and $\frac{8}{3}m^2(3n-m)$ otherwise. Following the factorization the solution for a single vector x requires $O\left(\min\left(m^2,n^2\right)\right)$ operations.

The real analogue of this routine is F08AAF (DGELS).

9 Example

This example solves the linear least-squares problem

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

where

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

and

$$b = \begin{pmatrix} -2.09 + 1.93i \\ 3.34 - 3.53i \\ -4.94 - 2.04i \\ 0.17 + 4.23i \\ -5.19 + 3.63i \\ 0.98 + 2.53i \end{pmatrix}.$$

The square root of the residual sum of squares is also output.

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

```
*     F08ANF 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=NMAX+NB*MMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION RNORM
      INTEGER          I, INFO, J, M, N
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(MMAX), WORK(LWORK)
*     .. External Functions ..
      DOUBLE PRECISION DZNRM2
      EXTERNAL         DZNRM2
*     .. External Subroutines ..
      EXTERNAL         ZGELS
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08ANF 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)
*
*        Solve the least squares problem min( norm2(b - Ax) ) for x
*
         CALL ZGELS('No transpose',M,N,1,A,LDA,B,M,WORK,LWORK,INFO)
*
*        Print solution
*
         WRITE (NOUT,*) 'Least squares solution'
         WRITE (NOUT,99999) (B(I),I=1,N)
*
*        Compute and print estimate of the square root of the residual
*        sum of squares
*
         RNORM = DZNRM2(M-N,B(N+1),1)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Square root of the residual sum of squares'
         WRITE (NOUT,99998) RNORM
      ELSE
         WRITE (NOUT,*) 'MMAX and/or NMAX too small, and/or M.LT.N'
```

```
      END IF
*
99999 FORMAT (4(' (',F7.4,',',F7.4,')',:))
99998 FORMAT (1X,1P,E10.2)
      END
```

9.2 Program Data

```
F08ANF Example Program Data

   6            4                               :Values of M, N and NRHS

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

 (-2.09, 1.93)
 ( 3.34,-3.53)
 (-4.94,-2.04)
 ( 0.17, 4.23)
 (-5.19, 3.63)
 ( 0.98, 2.53)                                          :End of vector b
```

9.3 Program Results

```
 F08ANF Example Program Results

 Least squares solution
 (-0.5044,-1.2179) (-2.4281, 2.8574) ( 1.4872,-2.1955) ( 0.4537, 2.6904)

 Square root of the residual sum of squares
   6.88E-02
```