NAG Fortran Library Routine Document

F08ZAF (DGGLSE)

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

F08ZAF (DGGLSE) solves a real linear equality-constrained least-squares problem.

2 Specification

```
SUBROUTINE F08ZAF (M, N, P, A, LDA, B, LDB, C, D, X, WORK, LWORK, INFO)
INTEGER             M, N, P, LDA, LDB, LWORK, INFO
double precision    A(LDA,*), B(LDB,*), C(*), D(*), X(*), WORK(*)
```

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

3 Description

F08ZAF (DGGLSE) solves the real linear equality-constrained least-squares (LSE) problem

$$\underset{x}{\text{minimize}} \ \|c - Ax\|_2 \quad \text{subject to} \quad Bx = d$$

where A is an m by n matrix, B is a p by n matrix, c is an m element vector and d is a p element vector. It is assumed that $p \leq n \leq m+p$, $\text{rank}(B) = p$ and $\text{rank}(E) = n$, where $E = \begin{pmatrix} A \\ B \end{pmatrix}$. These conditions ensure that the LSE problem has a unique solution, which is obtained using a generalized RQ factorization of the matrices B and A.

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

Anderson E, Bai Z and Dongarra J (1991) Generalized *QR* factorization and its applications *LAPACK Working Note No. 31* University of Tennessee, Knoxville

Eldèn L (1980) Perturbation theory for the least-squares problem with linear equality constraints *SIAM J. Numer. Anal.* **17** 338–350

5 Parameters

1: M – INTEGER *Input*

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

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

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrices A and B.

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

3: P – INTEGER *Input*

On entry: p, the number of rows of the matrix B.

Constraint: $0 \leq \mathrm{P} \leq \mathrm{N} \leq \mathrm{M} + \mathrm{P}$.

4: A(LDA,*) – ***double precision*** 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: A is overwritten.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08ZAF (DGGLSE) is called.

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

6: B(LDB,*) – ***double precision*** array *Input/Output*

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

On entry: the p by n matrix B.

On exit: B is overwritten.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08ZAF (DGGLSE) is called.

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

8: C(*) – ***double precision*** array *Input/Output*

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

On entry: the right-hand side vector c for the least-squares part of the LSE problem.

On exit: the residual sum of squares for the solution vector x is given by the sum of squares of elements $\mathrm{C}(\mathrm{N} - \mathrm{P} + 1), \mathrm{C}(\mathrm{N} - \mathrm{P} + 2), \ldots, \mathrm{C}(\mathrm{M})$, provided $m + p > n$; the remaining elements are overwritten.

9: D(*) – ***double precision*** array *Input/Output*

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

On entry: the right-hand side vector d for the equality constraints.

On exit: D is overwritten.

10: X(*) – ***double precision*** array *Output*

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

On exit: the solution vector x of the LSE problem.

11: WORK(*) – ***double precision*** array *Workspace*

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

On exit: if $\mathrm{IFAIL} = 0$, $\mathrm{WORK}(1)$ contains the minimum value of LWORK required for optimum performance.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the subprogram from which F08ZAF (DGGLSE) is called unless $\mathrm{LWORK} = -1$, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK (using the formula given below).

Suggested value: for optimum performance LWORK should be at least $\mathrm{P} + \min(\mathrm{M}, \mathrm{N}) + \max(\mathrm{M}, \mathrm{N}) \times nb$, where nb is the ***block size***.

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

13: INFO – INTEGER *Output*

On exit: $\mathrm{INFO} = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\mathrm{INFO} < 0$

If $\mathrm{INFO} = -i$, the ith argument had an illegal value.

7 Accuracy

For an error analysis, see Anderson *et al.* (1991) and Eldèn (1980). See also Section 4.6 of Anderson *et al.* (1999).

8 Further Comments

When $m \geq n = p$, the total number of floating-point operations is approximately $\frac{2}{3}n^2(6m + n)$; if $p \ll n$, the number reduces to approximately $\frac{2}{3}n^2(3m - n)$.

E04NCF/E04NCA may also be used to solve LSE problems. It differs from F08ZAF (DGGLSE) in that it uses an iterative (rather than direct) method, and that it allows general upper and lower bounds to be specified for the variables x and the linear constraints Bx.

9 Example

To solve the least-squares problem

$$\underset{x}{\text{minimize}}\ \|c - Ax\|_2 \quad \text{subject to} \quad Bx = d$$

where

$$c = \begin{pmatrix} -1.50 \\ -2.14 \\ 1.23 \\ -0.54 \\ -1.68 \\ 0.82 \end{pmatrix},$$

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 & 0.25 \\ -1.93 & 1.08 & -0.31 & -2.14 \\ 2.30 & 0.24 & 0.40 & -0.35 \\ -1.93 & 0.64 & -0.66 & 0.08 \\ 0.15 & 0.30 & 0.15 & -2.13 \\ -0.02 & 1.03 & -1.43 & 0.50 \end{pmatrix},$$

$$B = \begin{pmatrix} 1.0 & 0 & -1.0 & 0 \\ 0 & 1.0 & 0 & -1.0 \end{pmatrix}$$

and

$$d = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The constraints $Bx = d$ correspond to $x_1 = x_3$ and $x_2 = x_4$.

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.

```
*     F08ZAF Example Program Text
*     Mark 17 Release. NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NB, NMAX, PMAX
      PARAMETER        (MMAX=10,NB=64,NMAX=10,PMAX=10)
      INTEGER          LDA, LDB, LWORK
      PARAMETER        (LDA=MMAX,LDB=PMAX,LWORK=PMAX+NMAX+NB*(MMAX+NMAX)
     +                 )
*     .. Local Scalars ..
      DOUBLE PRECISION RNORM
      INTEGER          I, INFO, J, M, N, P
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(LDB,NMAX), C(MMAX), D(PMAX),
     +                 WORK(LWORK), X(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION DNRM2
      EXTERNAL         DNRM2
*     .. External Subroutines ..
      EXTERNAL         DGGLSE
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08ZAF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N, P
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. P.LE.PMAX) THEN
*
*        Read A, B, C and D from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) ((B(I,J),J=1,N),I=1,P)
         READ (NIN,*) (C(I),I=1,M)
         READ (NIN,*) (D(I),I=1,P)
*
*        Solve the equality-constrained least-squares problem
*
*        minimize ||c - A*x|| (in the 2-norm) subject to B*x = D
*
         CALL DGGLSE(M,N,P,A,LDA,B,LDB,C,D,X,WORK,LWORK,INFO)
*
*        Print least-squares solution
*
         WRITE (NOUT,*) 'Constrained least-squares solution'
         WRITE (NOUT,99999) (X(I),I=1,N)
*
*        Compute the square root of the residual sum of squares
*
         RNORM = DNRM2(M-N+P,C(N-P+1),1)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Square root of the residual sum of squares'
         WRITE (NOUT,99998) RNORM
      ELSE
```

```
         WRITE (NOUT,*)
     +     'One or more of MMAX, NMAX and PMAX is too small'
      END IF
      STOP
*
99999 FORMAT (1X,7F11.4)
99998 FORMAT (3X,1P,E11.2)
      END
```

9.2 Program Data

```
F08ZAF Example Program Data

  6      4      2           :Values of M, N and P

 -0.57  -1.28  -0.39   0.25
 -1.93   1.08  -0.31  -2.14
  2.30   0.24   0.40  -0.35
 -1.93   0.64  -0.66   0.08
  0.15   0.30   0.15  -2.13
 -0.02   1.03  -1.43   0.50 :End of matrix A

  1.00   0.00  -1.00   0.00
  0.00   1.00   0.00  -1.00 :End of matrix B

 -1.50
 -2.14
  1.23
 -0.54
 -1.68
  0.82                      :End of vector c

  0.00
  0.00                      :End of vector d
```

9.3 Program Results

```
 F08ZAF Example Program Results

 Constrained least-squares solution
      0.4890     0.9975     0.4890     0.9975

 Square root of the residual sum of squares
      2.51E-02
```