# NAG Library Routine Document

# F08ZBF (DGGGLM)

**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

F08ZBF (DGGGLM) solves a real general Gauss–Markov linear (least squares) model problem.

## 2 Specification

```
SUBROUTINE F08ZBF (M, N, P, A, LDA, B, LDB, D, X, Y, WORK, LWORK, INFO)

INTEGER             M, N, P, LDA, LDB, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), D(M), X(N), Y(P),              &
                    WORK(max(1,LWORK))
```

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

## 3 Description

F08ZBF (DGGGLM) solves the real general Gauss–Markov linear model (GLM) problem

$$\underset{x}{\text{minimize}}\|y\|_2 \qquad \text{subject to} \qquad d = Ax + By$$

where $A$ is an $m$ by $n$ matrix, $B$ is an $m$ by $p$ matrix and $d$ is an $m$ element vector. It is assumed that $n \leq m \leq n+p$, $\text{rank}(A) = n$ and $\text{rank}(E) = m$, where $E = \begin{pmatrix} A & B \end{pmatrix}$. Under these assumptions, the problem has a unique solution $x$ and a minimal 2-norm solution $y$, which is obtained using a generalized $QR$ factorization of the matrices $A$ and $B$.

In particular, if the matrix $B$ is square and nonsingular, then the GLM problem is equivalent to the weighted linear least squares problem

$$\underset{x}{\text{minimize}}\left\|B^{-1}(d - Ax)\right\|_2.$$

## 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 (1992) Generalized *QR* factorization and its applications *Linear Algebra Appl. (Volume 162–164)* 243–271


## 5 Parameters

1: M – INTEGER *Input*

*On entry*: $m$, the number of rows of the matrices $A$ and $B$.

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

2: N – INTEGER *Input*

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

*Constraint*: $0 \leq \text{N} \leq \text{M}$.

3: P – INTEGER *Input*

*On entry*: $p$, the number of columns of the matrix $B$.

*Constraint*: $\mathrm{P} \geq \mathrm{M} - \mathrm{N}$.

4: A(LDA,$*$) – REAL (KIND=nag_wp) 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 F08ZBF (DGGGLM) is called.

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

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

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

*On entry*: the $m$ by $p$ 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 F08ZBF (DGGGLM) is called.

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

8: D(M) – REAL (KIND=nag_wp) array *Input/Output*

*On entry*: the left-hand side vector $d$ of the GLM equation.

*On exit*: D is overwritten.

9: X(N) – REAL (KIND=nag_wp) array *Output*

*On exit*: the solution vector $x$ of the GLM problem.

10: Y(P) – REAL (KIND=nag_wp) array *Output*

*On exit*: the solution vector $y$ of the GLM problem.

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

*On exit*: if $\mathrm{INFO} = 0$, $\mathrm{WORK}(1)$ 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 F08ZBF (DGGGLM) is called.

If $\mathrm{LWORK} = -1$, 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, $\mathrm{LWORK} \geq \mathrm{N} + \min(\mathrm{M}, \mathrm{P}) + \max(\mathrm{M}, \mathrm{P}) \times nb$, where *nb* is the optimal ***block size***.

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

13: 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 $= 1$

The upper triangular factor $R$ associated with $A$ in the generalized $RQ$ factorization of the pair $(A, B)$ is singular, so that $\text{rank}(A) < m$; the least squares solution could not be computed.

INFO $= 2$

The bottom $(N - M)$ by $(N - M)$ part of the upper trapezoidal factor $T$ associated with $B$ in the generalized $QR$ factorization of the pair $(A, B)$ is singular, so that $\text{rank}\begin{pmatrix} A & B \end{pmatrix} < N$; the least squares solutions could not be computed.

# 7 Accuracy

For an error analysis, see Anderson *et al.* (1992). See also Section 4.6 of Anderson *et al.* (1999).

# 8 Further Comments

When $p = m \geq n$, the total number of floating point operations is approximately $\frac{2}{3}\left(2m^3 - n^3\right) + 4nm^2$; when $p = m = n$, the total number of floating point operations is approximately $\frac{14}{3}m^3$.

# 9 Example

This example solves the weighted least squares problem

$$\underset{x}{\text{minimize}} \left\| B^{-1}(d - Ax) \right\|_2,$$

where

$$B = \begin{pmatrix} 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 2.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 5.0 \end{pmatrix}, \quad d = \begin{pmatrix} 1.32 \\ -4.00 \\ 5.52 \\ 3.24 \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} -0.57 & -1.28 & -0.39 \\ -1.93 & 1.08 & -0.31 \\ 2.30 & 0.24 & -0.40 \\ -0.02 & 1.03 & -1.43 \end{pmatrix}.$$

## 9.1 Program Text

```
    Program f08zbfe

!     F08ZBF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dggglm, dnrm2, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rnorm
```

```
      Integer                          :: i, info, lda, ldb, lwork, m, n, p
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:,:), d(:), work(:), x(:), &
                                          y(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08ZBF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n, p
      lda = m
      ldb = m
      lwork = n + m + nb*(m+p)
      Allocate (a(lda,n),b(ldb,p),d(m),work(lwork),x(n),y(p))

!     Read A, B and D from data file

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

!     Solve the weighted least squares problem

!     minimize ||inv(B)*(d - A*x)|| (in the 2-norm)

!     The NAG name equivalent of dggglm is f08zbf
      Call dggglm(m,n,p,a,lda,b,ldb,d,x,y,work,lwork,info)

!     Print least squares solution, x

      Write (nout,*) 'Weighted least-squares solution'
      Write (nout,99999) x(1:n)

!     Print residual vector y = inv(B)*(d - A*x)

      Write (nout,*)
      Write (nout,*) 'Residual vector'
      Write (nout,99998) y(1:p)

!     Compute and print the square root of the residual sum of
!     squares

!     The NAG name equivalent of dnrm2 is f06ejf
      rnorm = dnrm2(p,y,1)

      Write (nout,*)
      Write (nout,*) 'Square root of the residual sum of squares'
      Write (nout,99998) rnorm

99999 Format (1X,7F11.4)
99998 Format (3X,1P,7E11.2)
    End Program f08zbfe
```

## 9.2 Program Data

```
F08ZBF Example Program Data

  4      3      4            :Values of M, N and P

 -0.57  -1.28  -0.39
 -1.93   1.08  -0.31
  2.30   0.24  -0.40
 -0.02   1.03  -1.43         :End of matrix A

  0.50   0.00   0.00   0.00
  0.00   1.00   0.00   0.00
  0.00   0.00   2.00   0.00
  0.00   0.00   0.00   5.00 :End of matrix B
```

```
  1.32
 -4.00
  5.52
  3.24                    :End of vector d
```

### 9.3 Program Results

```
F08ZBF Example Program Results

Weighted least-squares solution
     1.9889    -1.0058    -2.9911

Residual vector
    -6.37E-04  -2.45E-03  -4.72E-03   7.70E-03

Square root of the residual sum of squares
     9.38E-03
```