G02DGF – NAG Fortran Library Routine Document

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

G02DGF calculates the estimates of the parameters of a general linear regression model for a new dependent variable after a call to G02DAF.

2 Specification

```
 SUBROUTINE G02DGF(WEIGHT, N, WT, RSS, IP, IRANK, COV, Q, LDQ, SVD,
1                  P, Y, B, SE, RES, WK, IFAIL)
 INTEGER           N, IP, IRANK, LDQ, IFAIL
 real              WT(*), RSS, COV(IP*(IP+1)/2), Q(LDQ,IP+1),
1                  P(IP*IP+2*IP), Y(N), B(IP), SE(IP), RES(N),
2                  WK(5*(IP-1)+IP*IP)
 LOGICAL           SVD
 CHARACTER*1       WEIGHT
```

3 Description

G02DGF uses the results given by G02DAF to fit the same set of independent variables to a new dependent variable.

G02DAF computes a QR decomposition of the matrix of p independent variables and also, if the model is not of full rank, a singular value decomposition (SVD). These results can be used to compute estimates of the parameters for a general linear model with a new dependent variable. The QR decomposition leads to the formation of an upper triangular p by p matrix R and an n by n orthogonal matrix Q. In addition the vector $c = Q^T y$ (or $Q^T W^{1/2} y$) is computed. For a new dependent variable, y_{new}, G02DGF computes a new value of $c = Q^T y_{\text{new}}$ or $Q^T W^{1/2} y_{\text{new}}$.

If R is of full rank, then the least-squares parameter estimates, $\hat{\beta}$, are the solution to: $R\hat{\beta} = c_1$, where c_1 is the first p elements of c.

If R is not of full rank, then G02DAF will have computed a SVD of R,

$$R = Q_* \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} P^T,$$

where D is a k by k diagonal matrix with non-zero diagonal elements, k being the rank of R and Q_* and P are p by p orthogonal matrices. This gives the solution

$$\hat{\beta} = P_1 D^{-1} Q_{*_1}^T c_1$$

P_1 being the first k columns of P, i.e., $P = (P_1 P_0)$ and Q_{*_1} being the first k columns of Q_*. Details of the SVD, are made available by G02DAF in the form of the matrix P^*:

$$P^* = \begin{pmatrix} D^{-1} P_1^T \\ P_0^T \end{pmatrix}.$$

The matrix Q_* is made available through the workspace of G02DAF.

In addition to parameter estimates, the new residuals are computed and the variance-covariance matrix of the parameter estimates are found by scaling the variance-covariance matrix for the original regression.

4 References

[1] Golub G H and van Loan C F (1996) *Matrix Computations* Johns Hopkins University Press (3rd Edition), Baltimore

[2] Hammarling S (1985) The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20 (3)** 2–25

[3] Searle S R (1971) *Linear Models* Wiley

5 Parameters

1: WEIGHT — CHARACTER*1 *Input*

On entry: indicates if weights are to be used.

If WEIGHT = 'U' (Unweighted), least-squares estimation is used.

If WEIGHT = 'W' (Weighted), weighted least-squares is used and weights must be supplied in the array WT.

Constraint: WEIGHT = 'U' or 'W'.

2: N — INTEGER *Input*

On entry: the number of observations, n.

Constraint: N $\geq$ IP.

3: WT($*$) — ***real*** array *Input*

On entry: if WEIGHT = 'W', then WT must contain the weights to be used in the weighted regression.

If WT(i) = 0.0, then the ith observation is not included in the model, in which case the effective number of observations is the number of observations with non-zero weights.

If WEIGHT = 'U', then WT is not referenced and the effective number of observations is n.

Constraint: if WEIGHT = 'W', WT(i) $\geq$ 0.0, for $i = 1, 2, \ldots, n$.

4: RSS — ***real*** *Input/Output*

On entry: the residual sum of squares for the original dependent variable.

On exit: the residual sum of squares for the new dependent variable.

Constraint: RSS $>$ 0.0.

5: IP — INTEGER *Input*

On entry: the number of independent variables (including the mean if fitted), p.

Constraint: $1 \leq$ IP $\leq$ N.

6: IRANK — INTEGER *Input*

On entry: the rank of the independent variables, as given by G02DAF.

Constraints: IRANK $>$ 0, and if SVD = .FALSE., then IRANK = IP, else IRANK $\leq$ IP.

7: COV(IP$*$(IP+1)/2) — ***real*** array *Input/Output*

On entry: the covariance matrix of the parameter estimates as given by G02DAF.

On exit: the upper triangular part of the variance-covariance matrix of the IP parameter estimates given in B. They are stored packed by column, i.e., the covariance between the parameter estimate given in B(i) and the parameter estimate given in B(j), $j \geq i$, is stored in COV($j \times (j-1)/2 + i$).

8: Q(LDQ,IP+1) — ***real*** array *Input/Output*

On entry: the results of the QR decomposition as returned by G02DAF.

On exit: the first column of Q contains the new values of c, the remainder of Q will be unchanged.

9: LDQ — INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which G02DGF is called.

Constraint: LDQ $\geq$ N.

10: SVD — LOGICAL *Input*

On entry: indicates if a singular value decomposition was used by G02DAF.

If SVD = .TRUE., a singular value decomposition was used by G02DAF.
If SVD = .FALSE., a singular value decomposition was not used by G02DAF.

11: P(IP∗IP+2∗IP) — ***real*** array *Input*

On entry: details of the QR decomposition and SVD, if used, as returned in array P by G02DAF.

If SVD = .FALSE., only the first IP elements of P are used, these contain the zeta values for the QR decomposition (see F08AEF for details).
If SVD = .TRUE., the first IP elements of P contain the zeta values for the QR decomposition (see F08AEF for details) and the next IP × IP + IP elements of P contain details of the singular value decomposition.

12: Y(N) — ***real*** array *Input*

On entry: the new dependent variable, y_{new}.

13: B(IP) — ***real*** array *Output*

On exit: the least-squares estimates of the parameters of the regression model, $\hat{\beta}$.

14: SE(IP) — ***real*** array *Output*

On exit: the standard error of the estimates of the parameters.

15: RES(N) — ***real*** array *Output*

On exit: the residuals for the new regression model.

16: WK(5∗(IP−1)+IP∗IP) — ***real*** array *Input*

On entry: if SVD = .TRUE., WK must be unaltered from the previous call to G02DAF or G02DGF.

If SVD = .FALSE., WK is used as workspace.

17: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors detected by the routine:

IFAIL = 1

On entry, IP < 1,
or N < IP,
or IRANK $\leq$ 0,
or SVD = .FALSE. and IRANK $\neq$ IP,

or SVD = .TRUE. and IRANK > IP,

or LDQ < N,

or RSS ≤ 0.0,

or WEIGHT ≠ 'U' or 'W'.

IFAIL = 2

On entry, WEIGHT = 'W' or 'V' and a value of WT < 0.0.

7 Accuracy

The same accuracy as G02DAF is obtained.

8 Further Comments

The values of the leverages, h_i, are unaltered by a change in the dependent variable so a call to G02FAF can be made using the value of H from G02DAF.

9 Example

A data set consisting of 12 observations with four independent variables and two dependent variables are read in. A model with all four independent variables is fitted to the first dependent variable by G02DAF and the results printed. The model is then fitted to the second dependent variable by G02DGF and those results printed.

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.

```
*     G02DGF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=5,NMAX=12)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             RSS, TOL
      INTEGER          I, IDF, IFAIL, IP, IRANK, J, M, N
      LOGICAL          SVD
      CHARACTER        MEAN, WEIGHT
*     .. Local Arrays ..
      real             B(MMAX), COV(MMAX*(MMAX+1)/2), H(NMAX),
     +                 NEWY(NMAX), P(MMAX*(MMAX+2)), Q(NMAX,MMAX+1),
     +                 RES(NMAX), SE(MMAX), WK(5*(MMAX-1)+MMAX*MMAX),
     +                 WT(NMAX), XM(NMAX,MMAX), Y(NMAX)
      INTEGER          ISX(MMAX)
*     .. External Subroutines ..
      EXTERNAL         G02DAF, G02DGF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G02DGF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, M, WEIGHT, MEAN
      IF (N.LE.NMAX .AND. M.LT.MMAX) THEN
```

```
         IF (WEIGHT.EQ.'W' .OR. WEIGHT.EQ.'w') THEN
            DO 20 I = 1, N
               READ (NIN,*) (XM(I,J),J=1,M), Y(I), WT(I), NEWY(I)
   20       CONTINUE
         ELSE
            DO 40 I = 1, N
               READ (NIN,*) (XM(I,J),J=1,M), Y(I), NEWY(I)
   40       CONTINUE
         END IF
         READ (NIN,*) (ISX(J),J=1,M), IP
*        Set tolerance
         TOL = 0.00001e0
         IFAIL = 0
*
*        Fit initial model using G02DAF
         CALL G02DAF(MEAN,WEIGHT,N,XM,NMAX,M,ISX,IP,Y,WT,RSS,IDF,B,SE,
     +               COV,RES,H,Q,NMAX,SVD,IRANK,P,TOL,WK,IFAIL)
*
         WRITE (NOUT,*) 'Results from G02DAF'
         WRITE (NOUT,*)
         IF (SVD) THEN
            WRITE (NOUT,*) 'Model not of full rank'
            WRITE (NOUT,*)
         END IF
         WRITE (NOUT,99999) 'Residual sum of squares = ', RSS
         WRITE (NOUT,99998) 'Degrees of freedom = ', IDF
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Variable   Parameter estimate   Standard error'
         WRITE (NOUT,*)
         DO 60 J = 1, IP
            WRITE (NOUT,99997) J, B(J), SE(J)
   60    CONTINUE
         IFAIL = 0
*
         CALL G02DGF(WEIGHT,N,WT,RSS,IP,IRANK,COV,Q,NMAX,SVD,P,NEWY,B,
     +               SE,RES,WK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Results for second y-variable using G02DGF'
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Residual sum of squares = ', RSS
         WRITE (NOUT,99998) 'Degrees of freedom = ', IDF
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Variable   Parameter estimate   Standard error'
         WRITE (NOUT,*)
         DO 80 J = 1, IP
            WRITE (NOUT,99997) J, B(J), SE(J)
   80    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,A,e12.4)
99998 FORMAT (1X,A,I4)
99997 FORMAT (1X,I6,2e20.4)
      END
```

9.2 Program Data

```
G02DGF Example Program Data
 12 4 'U' 'M'
1.0 0.0 0.0 0.0 33.63 63.0
0.0 0.0 0.0 1.0 39.62 69.0
0.0 1.0 0.0 0.0 38.18 68.0
0.0 0.0 1.0 0.0 41.46 71.0
0.0 0.0 0.0 1.0 38.02 68.0
0.0 1.0 0.0 0.0 35.83 65.0
0.0 0.0 0.0 1.0 35.99 65.0
1.0 0.0 0.0 0.0 36.58 66.0
0.0 0.0 1.0 0.0 42.92 72.0
1.0 0.0 0.0 0.0 37.80 67.0
0.0 0.0 1.0 0.0 40.43 70.0
0.0 1.0 0.0 0.0 37.89 67.0
 1   1   1   1   5
```

9.3 Program Results

```
 G02DGF Example Program Results
 Results from G02DAF

 Model not of full rank

 Residual sum of squares =   0.2223E+02
 Degrees of freedom =    8

 Variable   Parameter estimate   Standard error

      1          0.3056E+02          0.3849E+00
      2          0.5447E+01          0.8390E+00
      3          0.6743E+01          0.8390E+00
      4          0.1105E+02          0.8390E+00
      5          0.7320E+01          0.8390E+00

 Results for second y-variable using G02DGF

 Residual sum of squares =   0.2400E+02
 Degrees of freedom =    8

 Variable   Parameter estimate   Standard error

      1          0.5407E+02          0.4000E+00
      2          0.1127E+02          0.8718E+00
      3          0.1260E+02          0.8718E+00
      4          0.1693E+02          0.8718E+00
      5          0.1327E+02          0.8718E+00
```