

NAG Fortran Library Routine Document

G02DKF

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

G02DKF calculates the estimates of the parameters of a general linear regression model for given constraints from the singular value decomposition results.

2 Specification

```

SUBROUTINE G02DKF(IP, ICONST, P, C, LDC, B, RSS, IDF, SE, COV, WK,
1          IFAIL)
  INTEGER      IP, ICONST, LDC, IDF, IFAIL
  real       P(IP*IP+2*IP), C(LDC, ICONST), B(IP), RSS, SE(IP),
1          COV((IP*(IP+1)/2)),
2          WK(2*IP*IP+IP*ICONST+2*ICONST*ICONST+4*ICONST)

```

3 Description

This routine computes the estimates given a set of linear constraints for a general linear regression model which is not of full rank. It is intended for use after a call to G02DAF or G02DDF.

In the case of a model not of full rank the routines use a singular value decomposition (SVD) to find the parameter estimates, $\hat{\beta}_{\text{svd}}$, and their variance-covariance matrix. Details of the SVD are made available in the form of the matrix P^* :

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

as described by G02DAF and G02DDF.

Alternative solutions can be formed by imposing constraints on the parameters. If there are p parameters and the rank of the model is k , then $n_c = p - k$ constraints will have to be imposed to obtain a unique solution.

Let C be a p by n_c matrix of constraints, such that

$$C^T \beta = 0$$

then the new parameter estimates $\hat{\beta}_c$ are given by

$$\begin{aligned} \hat{\beta}_c &= A \hat{\beta}_{\text{svd}} \\ &= (I - P_0(C^T P_0)^{-1}) \hat{\beta}_{\text{svd}}, \end{aligned}$$

where I is the identity matrix, and the variance-covariance matrix is given by

$$A P_1 D^{-2} P_1^T A^T,$$

provided $(C^T P_0)^{-1}$ exists.

4 References

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

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

Searle S R (1971) *Linear Models* Wiley

5 Parameters

- 1: IP – INTEGER *Input*
On entry: the number of terms in the linear model, p .
Constraint: $IP \geq 1$.

- 2: ICONST – INTEGER *Input*
On entry: the number of constraints to be imposed on the parameters, n_c .
Constraint: $0 < ICONST < IP$.

- 3: P(IP*IP+2*IP) – *real* array *Input*
On entry: P as returned by G02DAF and G02DDF.

- 4: C(LDC,ICONST) – *real* array *Input*
On entry: the ICONST constraints stored by column, i.e., the i th constraint is stored in the i th column of C.

- 5: LDC – INTEGER *Input*
On entry: the first dimension of the array C as declared in the (sub)program from which G02DKF is called.
Constraint: $LDC \geq IP$.

- 6: B(IP) – *real* array *Input/Output*
On entry: the parameter estimates computed by using the singular value decomposition, $\hat{\beta}_{\text{svd}}$.
On exit: the parameter estimates of the parameters with the constraints imposed, $\hat{\beta}_c$.

- 7: RSS – *real* *Input*
On entry: the residual sum of squares as returned by G02DAF or G02DDF.
Constraint: $RSS > 0.0$.

- 8: IDF – INTEGER *Input*
On entry: the degrees of freedom associated with the residual sum of squares as returned by G02DAF or G02DDF.
Constraint: $IDF > 0$.

- 9: SE(IP) – *real* array *Output*
On entry: the standard error of the parameter estimates in B.

10: COV((IP*(IP+1)/2)) – *real* array *Output*

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($j \times (j - 1)/2 + i$).

11: WK(2*IP*IP+IP*ICONST+2*ICONST*ICONST+4*ICONST) – *real* array *Workspace*

Note that a simple upper bound for the size of the workspace is $5 \times IP \times IP$.

12: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

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

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

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 or warnings detected by the routine:

IFAIL = 1

On entry, IP < 1,
or ICONST ≤ 0,
or ICONST ≥ IP,
or LDC < IP,
or RSS ≤ 0.0,
or IDF ≤ 0.

IFAIL = 2

C does not give a model of full rank.

7 Accuracy

It should be noted that due to rounding errors a parameter that should be zero when the constraints have been imposed may be returned as a value of order *machine precision*.

8 Further Comments

This routine is intended for use in situations in which dummy (0–1) variables have been used such as in the analysis of designed experiments when the user does not wish to change the parameters of the model to give a full rank model. The routine is not intended for situations in which the relationships between the independent variables are only approximate.

9 Example

Data from an experiment with four treatments and three observations per treatment are read in. A model, including the mean term, is fitted by G02DAF and the results printed. The constraint that the sum of treatment effect is zero is then read in and the parameter estimates with this constraint imposed are computed by G02DKF and 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.

```

*      G02DKF 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, ICONST, IDF, IFAIL, IP, IRANK, J, M, N
LOGICAL          SVD
CHARACTER       MEAN, WEIGHT
*      .. Local Arrays ..
real          B(MMAX), C(MMAX,MMAX), COV((MMAX*MMAX+MMAX)/2),
+             H(NMAX), P(MMAX*(MMAX+2)), Q(NMAX,MMAX+1),
+             RES(NMAX), SE(MMAX), WK(4*MMAX*MMAX+5*(MMAX-1)),
+             WT(NMAX), X(NMAX,MMAX), Y(NMAX)
INTEGER          ISX(MMAX)
*      .. External Subroutines ..
EXTERNAL        GO2DAF, G02DKF
*      .. Executable Statements ..
WRITE (NOUT,*) 'G02DKF Example Program Results'
*      Skip heading in data file
READ (NIN,*)
READ (NIN,*) N, M, WEIGHT, MEAN
WRITE (NOUT,*)
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,*) (X(I,J),J=1,M), Y(I), WT(I)
20    CONTINUE
  ELSE
    DO 40 I = 1, N
      READ (NIN,*) (X(I,J),J=1,M), Y(I)
40    CONTINUE
  END IF
  READ (NIN,*) (ISX(J),J=1,M), IP
*      Set tolerance
  TOL = 0.00001e0
  IFAIL = 0
*
*      Find initial estimates using G02DAF
  CALL G02DAF(MEAN,WEIGHT,N,X,NMAX,M,ISX,IP,Y,WT,RSS,IDF,B,SE,
+           COV,RES,H,Q,NMAX,SVD,IRANK,P,TOL,WK,IFAIL)
*
  WRITE (NOUT,*) 'Estimates from G02DAF'
  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 60 J = 1, IP
    WRITE (NOUT,99997) J, B(J), SE(J)
60  CONTINUE
*      Input constraints and call G02DKF
  ICONST = IP - IRANK
  DO 80 I = 1, IP
    READ (NIN,*) (C(I,J),J=1,ICONST)
80  CONTINUE
  IFAIL = 0
*
  CALL G02DKF(IP,ICONST,P,C,MMAX,B,RSS,IDF,SE,COV,WK,IFAIL)
*

```

```

        WRITE (NOUT,*)
        WRITE (NOUT,*) 'Estimates from G02DKF using constraints'
        WRITE (NOUT,*)
        WRITE (NOUT,*)
+       'Variable      Parameter estimate      Standard error'
        WRITE (NOUT,*)
        DO 100 J = 1, IP
            WRITE (NOUT,99997) J, B(J), SE(J)
100     CONTINUE
        END IF
        STOP
*
99999 FORMAT (1X,A,e13.4)
99998 FORMAT (1X,A,I4)
99997 FORMAT (1X,I6,2e20.4)
        END

```

9.2 Program Data

```

G02DKF Example Program Data
 12 4 'U' 'M'
1.0 0.0 0.0 0.0 33.63
0.0 0.0 0.0 1.0 39.62
0.0 1.0 0.0 0.0 38.18
0.0 0.0 1.0 0.0 41.46
0.0 0.0 0.0 1.0 38.02
0.0 1.0 0.0 0.0 35.83
0.0 0.0 0.0 1.0 35.99
1.0 0.0 0.0 0.0 36.58
0.0 0.0 1.0 0.0 42.92
1.0 0.0 0.0 0.0 37.80
0.0 0.0 1.0 0.0 40.43
0.0 1.0 0.0 0.0 37.89
 1  1  1  1  5
0.0
1.0
1.0
1.0
1.0

```

9.3 Program Results

G02DKF Example Program Results

Estimates from G02DAF

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

Estimates from G02DKF using constraints

Variable	Parameter estimate	Standard error
1	0.3820E+02	0.4812E+00
2	-0.2192E+01	0.8334E+00
3	-0.8958E+00	0.8334E+00
4	0.3408E+01	0.8334E+00
5	-0.3192E+00	0.8334E+00