# NAG Toolbox for MATLAB

# g02dk

## 1 Purpose

g02dk calculates the estimates of the parameters of a general linear regression model for given constraints from the singular value decomposition results.

## 2 Syntax

```
[b, se, cov, ifail] = g02dk(p, c, b, rss, idf, 'ip', ip, 'iconst', iconst)
```

## 3 Description

g02dk 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 g02da or g02dd.

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

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

as described by g02da and g02dd.

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

$$C^{\mathrm{T}}\beta = 0$$

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

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

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

$$AP_1D^{-2}P_1^{\mathrm{T}}A^{\mathrm{T}},$$

provided $\left(C^{\mathrm{T}}P_0\right)^{-1}$ 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

## 5.1 Compulsory Input Parameters

1: **p**(**ip** × **ip** + **2** × **ip**) – **double array**

As returned by g02da and g02dd.

2: **c**(**ldc**,**iconst**) – **double array**

**ldc**, the first dimension of the array, must be at least **ip**.

The **iconst** constraints stored by column, i.e., the *i*th constraint is stored in the *i*th column of **c**.

3: **b**(**ip**) – **double array**

The parameter estimates computed by using the singular value decomposition, $\hat{\beta}_{\mathrm{svd}}$.

4: **rss** – **double scalar**

The residual sum of squares as returned by g02da or g02dd.

*Constraint*: **rss** $> 0.0$.

5: **idf** – **int32 scalar**

The degrees of freedom associated with the residual sum of squares as returned by g02da or g02dd.

*Constraint*: **idf** $> 0$.

## 5.2 Optional Input Parameters

1: **ip** – **int32 scalar**

*Default*: The dimension of the arrays **p**, **ip**, **se**, **cov**. (An error is raised if these dimensions are not equal.)

$p$, the number of terms in the linear model.

*Constraint*: **ip** $\geq 1$.

2: **iconst** – **int32 scalar**

*Default*: The dimension of the array **c**.

the number of constraints to be imposed on the parameters, $n_{\mathrm{c}}$.

*Constraint*: $0 <$ **iconst** $<$ **ip**.

## 5.3 Input Parameters Omitted from the MATLAB Interface

ldc, wk

## 5.4 Output Parameters

1: **b**(**ip**) – **double array**

The parameter estimates of the parameters with the constraints imposed, $\hat{\beta}_{\mathrm{c}}$.

2: **se**(**ip**) – **double array**

The standard error of the parameter estimates in **b**.

3: **cov**((**ip** × (**ip** + **1**)/**2**)) **– double array**

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

4: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the function:

**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

g02dk is intended for use in situations in which dummy (0 − −1) variables have been used such as in the analysis of designed experiments when you do not wish to change the parameters of the model to give a full rank model. The function is not intended for situations in which the relationships between the independent variables are only approximate.

## 9 Example

```
p = [1.135198279858991;
     1.13078348304906;
     1.234006904805161;
     1.228003039942231;
     1.263332232672042;
     3.872983346207417;
     1.732050807568878;
     1.732050807568878;
     1.732050807568877;
     1.64780261914309e-16;
     0.2309401076758503;
     8.35089227579159e-18;
     -1.238223776129407e-17;
     7.900875770685859e-17;
     -0.447213595499958;
     0.0577350269189626;
     0.2871305767251709;
     0.2022129403845741;
     -0.3559016137228031;
     0.447213595499958;
```

```
     0.05773502691896256;
     0.03726461792122844;
     0.271286303307133;
     0.4183480487454616;
     0.447213595499958;
     0.05773502691896257;
     0.1515575558927682;
     -0.4677758485215264;
     0.0906424999201606;
     0.447213595499958;
     0.05773502691896258;
     -0.4759527505391677;
     -0.005723395170180857;
     -0.153088934942819;
     0.4472135954999578];
c = [0;
     1;
     1;
     1;
     1];
b = [30.55666666666667;
     5.446666666666673;
     6.743333333333336;
     11.04666666666667;
     7.32];
rss = 22.22679999999987;
idf = int32(8);
[bOut, se, cov, ifail] = g02dk(p, c, b, rss, idf)
```

```
bOut =
   38.1958
   -2.1925
   -0.8958
    3.4075
   -0.3192
se =
    0.4812
    0.8334
    0.8334
    0.8334
    0.8334
cov =
    0.2315
   -0.0000
    0.6946
    0.0000
   -0.2315
    0.6946
    0.0000
   -0.2315
   -0.2315
    0.6946
    0.0000
   -0.2315
   -0.2315
   -0.2315
    0.6946
ifail =
           0
```