# NAG Fortran Library Routine Document

# G03BCF

**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

G03BCF computes Procrustes rotations in which an orthogonal rotation is found so that a transformed matrix best matches a target matrix.

## 2 Specification

```
      SUBROUTINE G03BCF(STAND, PSCALE, N, M, X, LDX, Y, LDY, YHAT, R, LDR,
     1                  ALPHA, RSS, RES, WK, IFAIL)
      INTEGER           N, M, LDX, LDY, LDR, IFAIL
      real              X(LDX,M), Y(LDY,M), YHAT(LDY,M), R(LDR,M), ALPHA, RSS,
     1                  RES(N), WK(M*M+7*M)
      CHARACTER*1       STAND, PSCALE
```

## 3 Description

Let $X$ and $Y$ be $n$ by $m$ matrices. They can be considered as representing sets of $n$ points in an $m$-dimensional space. The $X$ matrix may be a matrix of loadings from say factor analysis or canonical variate analysis and the $Y$ matrix may be a postulated pattern matrix or the loadings from a different sample. The problem is to relate the two sets of points without disturbing the relationships between the points in each set. This can be achieved by translating, rotating and scaling the sets of points. The $Y$ matrix is considered as the target matrix and the $X$ matrix is rotated to match that matrix.

First the two sets of points are translated so that their centroids are at the origin to give $X_c$ and $Y_c$, i.e., the matrices will have zero column means. Then the rotation of the translated $X_c$ matrix which minimizes the sum of squared distances between corresponding points in the two sets is found. This is computed from the singular value decomposition of the matrix:

$$X_c^{\mathrm{T}} Y_c = UDV^{\mathrm{T}},$$

where $U$ and $V$ are orthogonal matrices and $D$ is a diagonal matrix. The matrix of rotations, $R$, is computed as:

$$R = UV^{\mathrm{T}}.$$

After rotation a scaling or dilation factor, $\alpha$, may be estimated by least-squares. Thus the final set of points that best match $Y_c$ is given by:

$$\hat{Y}_c = \alpha X_c R.$$

Before rotation both sets of points may be normalized to have unit sums of squares or the $X$ matrix may be normalized to have the same sum of squares as the $Y$ matrix. After rotation the results may be translated to the original $Y$ centroid.

The $i$th residual, $r_i$, is given by the distance between the point given in the $i$th row of $Y$ and the point given in the $i$th row of $\hat{Y}$. The residual sum of squares is also computed.

## 4 References

Krzanowski W J (1990) *Principles of Multivariate Analysis* Oxford University Press

Lawley D N and Maxwell A E (1971) *Factor Analysis as a Statistical Method* (2nd Edition) Butterworths

## 5 Parameters

1: STAND – CHARACTER*1 *Input*

*On entry*: indicates if translation/normalization is required.

If STAND = 'N' there is no translation or normalization.

If STAND = 'Z' there is translation to the origin (i.e., to zero).

If STAND = 'C' there is translation to origin and then to the $Y$ centroid after rotation.

If STAND = 'U' there is unit normalization.

If STAND = 'S' there is translation and normalization (i.e., there is standardization).

If STAND = 'M' there is translation and normalization to $Y$ scale, then translation to the $Y$ centroid after rotation (i.e., they are matched).

*Constraint*: STAND = 'N', 'Z', 'C', 'U', 'S' or 'M'.

2: PSCALE – CHARACTER*1 *Input*

*On entry*: indicates if least-squares scaling is to be applied after rotation.

If PSCALE = 'S' , then scaling is applied.

If PSCALE = 'U', then no scaling is applied.

*Constraint*: PSCALE = 'S' or 'U'.

3: N – INTEGER *Input*

*On entry*: the number of points, $n$.

*Constraint*: N $\geq$ M.

4: M – INTEGER *Input*

*On entry*: the number of dimensions, $m$.

*Constraint*: M $\geq$ 1.

5: X(LDX,M) – ***real*** array *Input/Output*

*On entry*: the matrix to be rotated, $X$.

*On exit*: if STAND = 'N', then X will be unchanged.

If STAND = 'Z', 'C', 'S' or 'M', then X will be translated to have zero column means.

If STAND = 'U' or 'S', then X will be scaled to have unit sum of squares.

If STAND = 'M', then X will be scaled to have the same sum of squares as Y.

6: LDX – INTEGER *Input*

*On entry*: the first dimension of the array X as declared in the (sub)program from which G03BCF is called.

*Constraint*: LDX $\geq$ N.

7: Y(LDY,M) – ***real*** array *Input/Output*

*On entry*: the target matrix, Y.

*On exit*: if STAND = 'N', then Y will be unchanged.

If STAND = 'Z' or 'S', then Y will be translated to have zero column means.

If STAND = 'U' or 'S', then Y will be scaled to have unit sum of squares.

If STAND = 'C' or 'M', then Y will be translated and then after rotation translated back. The output Y should be the same as the input Y except for rounding errors.

8: LDY – INTEGER *Input*

*On entry*: the first dimension of the arrays Y and YHAT as declared in the (sub)program from which G03BCF is called.

*Constraint*: LDY ≥ N.

9: YHAT(LDY,M) – ***real*** array *Output*

*On exit*: the fitted matrix, $\hat{Y}$.

10: R(LDR,M) – ***real*** array *Output*

*On exit*: the matrix of rotations, $R$, see Section 8.

11: LDR – INTEGER *Input*

*On entry*: the first dimension of the array R as declared in the (sub)program from which G03BCF is called.

*Constraint*: LDR ≥ M.

12: ALPHA – ***real*** *Output*

*On exit*: if PSCALE = 'S' the scaling factor, $\alpha$; otherwise ALPHA is not set.

13: RSS – ***real*** *Output*

*On exit*: the residual sum of squares.

14: RES(N) – ***real*** array *Output*

*On exit*: the residuals, $r_i$, for $i = 1, 2, \ldots n$.

15: WK(M*M+7*M) – ***real*** array *Workspace*

16: 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, N < M,
or M < 1,
or LDX < N,
or LDY < N,
or LDR < M,

or STAND $\neq$ 'N', 'Z', 'C', 'U', 'S' or 'M',
or PSCALE $\neq$ 'S' or 'U'.

IFAIL = 2

On entry, either X or Y contain only zero-points (possibly after translation) when normalization is to be applied.

IFAIL = 3

The $\hat{Y}$ matrix contains only zero-points when least-squares scaling is applied.

IFAIL = 4

The singular value decomposition has failed to converge. This is an unlikely error exit.

## 7 Accuracy

The accuracy of the calculation of the rotation matrix largely depends upon the singular value decomposition. See F02WEF for further details.

## 8 Further Comments

Note that if the matrix $X_c^{\mathrm{T}}Y$ is not of full rank, then the matrix of rotations, $R$, may not be unique even if there is a unique solution in terms of the rotated matrix, $\hat{Y}_c$. The matrix $R$ may also include reflections as well as pure rotations, see Krzanowski (1990).

If the column dimensions of the $X$ and $Y$ matrices are not equal, the smaller of the two should be supplemented by columns of zeros. Adding a column of zeros to both $X$ and $Y$ will have the effect of allowing reflections as well as rotations.

## 9 Example

Three points representing the vertices of a triangle in two dimensions are input. The points are translated and rotated to match the triangle given by (0,0), (1,0), (0,2) and scaling is applied after rotation. The target matrix and fitted matrix are printed along with additional information.

### 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.

```
*     G03BCF Example Program Text
*     Mark 15 Release. NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX
      PARAMETER        (NMAX=3,MMAX=2)
*     .. Local Scalars ..
      real             ALPHA, RSS
      INTEGER          I, IFAIL, J, M, N
      CHARACTER        SCALE, STAND
*     .. Local Arrays ..
      real             R(MMAX,MMAX), RES(NMAX), WK(MMAX*MMAX+7*MMAX),
     +                 X(NMAX,MMAX), Y(NMAX,MMAX), YHAT(NMAX,MMAX)
*     .. External Subroutines ..
      EXTERNAL         G03BCF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G03BCF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, M, STAND, SCALE
```

```
      IF (N.LE.NMAX .AND. M.LE.MMAX) THEN
         DO 20 I = 1, N
            READ (NIN,*) (X(I,J),J=1,M)
   20    CONTINUE
         DO 40 I = 1, N
            READ (NIN,*) (Y(I,J),J=1,M)
   40    CONTINUE
         IFAIL = 0
*
         CALL G03BCF(STAND,SCALE,N,M,X,NMAX,Y,NMAX,YHAT,R,MMAX,ALPHA,
     +               RSS,RES,WK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) '         Rotation Matrix'
         WRITE (NOUT,*)
         DO 60 I = 1, M
            WRITE (NOUT,99999) (R(I,J),J=1,M)
   60    CONTINUE
         IF (SCALE.EQ.'S' .OR. SCALE.EQ.'s') THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99998) ' Scale factor = ', ALPHA
         END IF
         WRITE (NOUT,*)
         WRITE (NOUT,*) '        Target Matrix'
         WRITE (NOUT,*)
         DO 80 I = 1, N
            WRITE (NOUT,99999) (Y(I,J),J=1,M)
   80    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) '        Fitted Matrix'
         WRITE (NOUT,*)
         DO 100 I = 1, N
            WRITE (NOUT,99999) (YHAT(I,J),J=1,M)
  100    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,99998) 'RSS = ', RSS
      END IF
      STOP
*
99999 FORMAT (6(2X,F7.3))
99998 FORMAT (1X,A,F10.3)
      END
```

### 9.2 Program Data

```
 G03BCF EXAMPLE PROGRAM DATA
 3 2 'c' 's'
 0.63 0.58
 1.36 0.39
 1.01 1.76
 0.0 0.0
 1.0 0.0
 0.0 2.0
```

## 9.3 Program Results

```
 G03BCF Example Program Results

          Rotation Matrix

    0.967    0.254
   -0.254    0.967

  Scale factor =      1.556

         Target Matrix

    0.000    0.000
    1.000    0.000
    0.000    2.000

         Fitted Matrix

   -0.093    0.024
    1.080    0.026
    0.013    1.950

 RSS =      0.019
```