NAG Fortran Library Routine Document

G02HBF

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

G02HBF finds, for a real matrix X of full column rank, a lower triangular matrix A such that $(A^T A)^{-1}$ is proportional to a robust estimate of the covariance of the variables. G02HBF is intended for the calculation of weights of bounded influence regression using G02HDF.

2 Specification

```
      SUBROUTINE G02HBF(UCV, N, M, X, IX, A, Z, BL, BD, TOL, MAXIT, NITMON,
     1                  NIT, WK, IFAIL)
      INTEGER           N, M, IX, MAXIT, NITMON, NIT, IFAIL
      real              UCV, X(IX,M), A(M*(M+1)/2), Z(N), BL, BD, TOL,
     1                  WK(M*(M+1)/2)
      EXTERNAL          UCV
```

3 Description

In fitting the linear regression model

$$y = X\theta + \epsilon,$$

where y is a vector of length n of the dependent variable,
X is an n by m matrix of independent variables,
θ is a vector of length m of unknown parameters,
and ϵ is a vector of length n of unknown errors,

it may be desirable to bound the influence of rows of the X matrix. This can be achieved by calculating a weight for each observation. Several schemes for calculating weights have been proposed (see Hampel *et al.* (1986) and Marazzi (1987a)). As the different independent variables may be measured on different scales one group of proposed weights aims to bound a standardised measure of influence. To obtain such weights the matrix A has to be found such that

$$\frac{1}{n}\sum_{i=1}^{n} u(\|z_i\|_2) z_i z_i^T = I, \quad (I \quad \text{is the indentity matrix})$$

and

$$z_i = Ax_i,$$

where x_i is a vector of length m containing the elements of the ith row of X,
A is an m by m lower triangular matrix,
z_i is a vector of length m,
and u is a suitable function.

The weights for use with G02HDF may then be computed using

$$w_i = f(\|z_i\|_2)$$

for a suitable user function f.

G02HBF finds A using the iterative procedure

$$A_k = (S_k + I)A_{k-1},$$

where $S_k = (s_{jl})$, for $j, l = 1, 2, \ldots, m$ is a lower triangular matrix such that

$$s_{jl} = \begin{cases} -\min[\max(h_{jl}/n, -BL), \mathrm{BL}], & j > l \\ -\min[\max(\frac{1}{2}(h_{jj}/n - 1), -BD), \mathrm{BD}], & j = l \end{cases}$$

$$h_{jl} = \sum_{i=1}^{n} u(\|z_i\|_2) z_{ij} z_{il}$$

and BD and BL are suitable bounds.

In addition the values of $\|z_i\|_2$, for $i = 1, 2, \ldots, n$, are calculated.

G02HBF is based on routines in ROBETH; see Marazzi (1987a).

4 References

Hampel F R, Ronchetti E M, Rousseeuw P J and Stahel W A (1986) *Robust Statistics. The Approach Based on Influence Functions* Wiley

Huber P J (1981) *Robust Statistics* Wiley

Marazzi A (1987a) Weights for bounded influence regression in ROBETH *Cah. Rech. Doc. IUMSP, No. 3 ROB 3* Institut Universitaire de Médecine Sociale et Préventive, Lausanne

5 Parameters

1: UCV – ***real*** FUNCTION, supplied by the user. *External Procedure*

UCV must return the value of the function u for a given value of its argument. The value of u must be non-negative.

Its specification is:

```
real FUNCTION UCV(T)
real              T
```

1: T – ***real*** *Input*

On entry: the argument for which UCV must be evaluated.

UCV must be declared as EXTERNAL in the (sub)program from which G02HBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: N – INTEGER *Input*

On entry: the number, n, of observations.

Constraint: $\mathrm{N} > 1$.

3: M – INTEGER *Input*

On entry: the number, m, of independent variables.

Constraint: $1 \leq \mathrm{M} \leq \mathrm{N}$.

4: X(IX,M) – ***real*** array *Input*

On entry: the real matrix X, i.e., the independent variables. $\mathrm{X}(i,j)$ must contain the ijth element of X, for $i = 1, 2, \ldots n$, $j = 1, 2, \ldots, m$.

5: IX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G02HBF is called.

Constraint: $\mathrm{IX} \geq \mathrm{N}$.

6: A(M*(M+1)/2) – ***real*** array *Input/Output*

On entry: an initial estimate of the lower triangular real matrix A. Only the lower triangular elements must be given and these should be stored row-wise in the array.

The diagonal elements must be $\neq 0$, although in practice will usually be > 0. If the magnitudes of the columns of X are of the same order the identity matrix will often provide a suitable initial value for A. If the columns of X are of different magnitudes, the diagonal elements of the initial value of A should be approximately inversely proportional to the magnitude of the columns of X.

On exit: the lower triangular elements of the matrix A, stored row-wise.

7: Z(N) – ***real*** array *Output*

On exit: the value $\|z_i\|_2$, $i = 1, 2, \ldots, n$.

8: BL – ***real*** *Input*

On entry: the magnitude of the bound for the off-diagonal elements of S_k.

Suggested value: BL = 0.9.

Constraint: BL > 0.

9: BD – ***real*** *Input*

On entry: the magnitude of the bound for the diagonal elements of S_k.

Suggested value: BD = 0.9.

Constraint: BD > 0.

10: TOL – ***real*** *Input*

On entry: the relative precision for the final value of A. Iteration will stop when the maximum value of $|s_{jl}|$ is less than TOL.

Constraint: TOL > 0.0.

11: MAXIT – INTEGER *Input*

On entry: the maximum number of iterations that will be used during the calculation of A.

A value of MAXIT = 50 will often be adequate.

Constraint: MAXIT > 0.

12: NITMON – INTEGER *Input*

On entry: determines the amount of information that is printed on each iteration.

If NITMON > 0 then the value of A and the maximum value of $|s_{jl}|$ will be printed at the first and every NITMON iterations.

If NITMON $\leq$ 0 then no iteration monitoring is printed.

When printing occurs the output is directed to the current advisory message unit (see X04ABF).

13: NIT – INTEGER *Output*

On exit: the number of iterations performed.

14: WK(M*(M+1)/2) – ***real*** array *Workspace*

15: 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 ≤ 1,
or M < 1,
or N $<$ M,
or IX $<$ N.

IFAIL $= 2$

On entry, TOL ≤ 0,
or MAXIT ≤ 0,
or diagonal element of A $= 0$,
or BL ≤ 0,
or BD ≤ 0.

IFAIL $= 3$

Value returned by UCV < 0.

IFAIL $= 4$

The routine has failed to converge in MAXIT iterations.

7 Accuracy

On successful exit the accuracy of the results is related to the value of TOL; see Section 5.

8 Further Comments

The existence of A will depend upon the function u; (see Hampel *et al.* (1986) and Marazzi (1987a)), also if X is not of full rank a value of A will not be found. If the columns of X are almost linearly related then convergence will be slow.

9 Example

The example program reads in a matrix of real numbers and computes the Krasker–Welsch weights (see Marazzi (1987a)). The matrix A and the weights are then 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.

```
*     G02HBF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX
```

```
      PARAMETER        (NMAX=5,MMAX=3)
*     .. Local Scalars ..
      real             BD, BL, TOL
      INTEGER          I, IFAIL, IX, J, K, L1, L2, M, MAXIT, MM, N, NIT,
     +                 NITMON
*     .. Local Arrays ..
      real             A(MMAX*(MMAX+1)/2), WK(MMAX*(MMAX+1)/2),
     +                 X(NMAX,MMAX), Z(NMAX)
*     .. External Functions ..
      real             UCV
      EXTERNAL         UCV
*     .. External Subroutines ..
      EXTERNAL         G02HBF, X04ABF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G02HBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      CALL X04ABF(1,NOUT)
*     Read in the dimensions of X
      READ (NIN,*) N, M
      IF (N.GT.0 .AND. N.LE.NMAX .AND. M.GT.0 .AND. M.LE.MMAX) THEN
*        Read in the X matrix
         DO 20 I = 1, N
            READ (NIN,*) (X(I,J),J=1,M)
   20    CONTINUE
         IX = NMAX
*        Read in the initial value of A
         MM = (M+1)*M/2
         READ (NIN,*) (A(J),J=1,MM)
*        Set the values remaining parameters
         BL = 0.9e0
         BD = 0.9e0
         MAXIT = 50
         TOL = 0.5e-4
         IFAIL = 0
*        * Change NITMON to a positive value if monitoring information
*          is required *
         NITMON = 0
*
         CALL G02HBF(UCV,N,M,X,IX,A,Z,BL,BD,TOL,MAXIT,NITMON,NIT,WK,
     +               IFAIL)
*
         WRITE (NOUT,99999) 'G02HBF required ', NIT,
     +     ' iterations to converge'
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Matrix A'
         L2 = 0
         DO 40 J = 1, M
            L1 = L2 + 1
            L2 = L2 + J
            WRITE (NOUT,99998) (A(K),K=L1,L2)
   40    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Vector Z'
         DO 60 I = 1, N
            WRITE (NOUT,99998) Z(I)
   60    CONTINUE
*        Calculate Krasker-Welsch weights
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Vector of weights'
         DO 80 I = 1, N
            Z(I) = 1.0e0/Z(I)
            WRITE (NOUT,99998) Z(I)
   80    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,A,I4,A)
99998 FORMAT (1X,6F9.4)
      END
*
```

```
      real  FUNCTION UCV(T)
*     UCV function for Krasker-Welsch weights
*     .. Parameters ..
      real              UCVC
      PARAMETER         (UCVC=2.5e0)
*     .. Scalar Arguments ..
      real              T
*     .. Local Scalars ..
      real              PC, PD, Q, Q2
      INTEGER           IFAIL
*     .. External Functions ..
      real              S15ABF, X01AAF, X02AKF
      EXTERNAL          S15ABF, X01AAF, X02AKF
*     .. Intrinsic Functions ..
      INTRINSIC         EXP, LOG, SQRT
*     .. Executable Statements ..
      UCV = 1.0e0
      IF (T.NE.0.0e0) THEN
         Q = UCVC/T
         Q2 = Q*Q
         IFAIL = 0
         PC = S15ABF(Q,IFAIL)
         IF (Q2.LT.-LOG(X02AKF())) THEN
            PD = EXP(-Q2/2.0e0)/SQRT(X01AAF(0.0e0)*2.0e0)
         ELSE
            PD = 0.0e0
         END IF
         UCV = (2.0e0*PC-1.0e0)*(1.0e0-Q2) + Q2 - 2.0e0*Q*PD
      END IF
      RETURN
      END
```

9.2 Program Data

```
G02HBF Example Program Data

   5    3                  : N  M

 1.0 -1.0 -1.0             : X1  X2  X3
 1.0 -1.0  1.0
 1.0  1.0 -1.0
 1.0  1.0  1.0
 1.0  0.0  3.0             : End of X1 X2 and X3 values

 1.0 0.0 1.0 0.0 0.0 1.0   : A
```

9.3 Program Results

```
 G02HBF Example Program Results
 G02HBF required   16 iterations to converge

 Matrix A
    1.3208
   -0.0000   1.4518
   -0.5753  -0.0000   0.9340

 Vector Z
    2.4760
    1.9953
    2.4760
    1.9953
    2.5890

 Vector of weights
    0.4039
    0.5012
    0.4039
    0.5012
    0.3862
```