G02HDF – 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

G02HDF performs bounded influence regression (M-estimates) using an iterative weighted least-squares algorithm.

2 Specification

```
      SUBROUTINE G02HDF(CHI, PSI, PSIPO, BETA, INDW, ISIGMA, N, M, X,
     1                  IX, Y, WGT, THETA, K,SIGMA, RS, TOL, EPS, MAXIT,
     2                  NITMON, NIT, WK, IFAIL)
      INTEGER           INDW, ISIGMA, N, M, IX, K, MAXIT, NITMON, NIT,
     1                  IFAIL
      real              CHI, PSI, PSIPO, BETA, X(IX,M), Y(N), WGT(N),
     1                  THETA(M), SIGMA, RS(N), TOL, EPS, WK((M+4)*N)
      EXTERNAL          CHI, PSI
```

3 Description

For the linear regression model

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

where y is a vector of length n of the dependent variable,

X is a n by m matrix of independent variables of column rank k,

θ is a vector of length m of unknown parameters,

and ϵ is a vector of length n of unknown errors with var $(\epsilon_i) = \sigma^2$:

G02HDF calculates the M-estimates given by the solution, $\hat{\theta}$, to the equation

$$\sum_{i=1}^{n} \psi(r_i/(\sigma w_i))w_i x_{ij} = 0, \quad j = 1,2,...,m \tag{1}$$

where r_i is the ith residual i.e., the ith element of the vector $r = y - X\hat{\theta}$,

ψ is a suitable weight function,

w_i are suitable weights such as those that can be calculated by using output from G02HBF,

and σ may be estimated at each iteration by the median absolute deviation of the residuals:

$\hat{\sigma} = \operatorname*{med}_i[|r_i|]/\beta_1$

or as the solution to

$$\sum_{i=1}^{n} \chi(r_i/(\hat{\sigma} w_i))w_i^2 = (n-k)\beta_2$$

for suitable weight function χ, where β_1 and β_2 are constants, chosen so that the estimator of σ is asymptotically unbiased if the errors, ϵ_i, have a Normal distribution. Alternatively σ may be held at a constant value.

The above describes the Schweppe type regression. If the w_i are assumed to equal 1 for all i, then Huber type regression is obtained. A third type, due to Mallows, replaces (1) by

$$\sum_{i=1}^{n} \psi(r_i/\sigma)w_i x_{ij} = 0, \quad j = 1,2,\ldots,m$$

This may be obtained by use of the transformations

$$\begin{aligned} w_i^* &\leftarrow \sqrt{w_i} \\ y_i^* &\leftarrow y_i\sqrt{w_i} \\ x_{ij}^* &\leftarrow x_{ij}\sqrt{w_i}, \quad j = 1, 2, \ldots, m \end{aligned}$$

(see Marazzi [3]).

The calculation of the estimates of θ can be formulated as an iterative weighted least-squares problem with a diagonal weight matrix G given by

$$G_{ii} = \begin{cases} \dfrac{\psi(r_i/(\sigma w_i)),}{(r_i/(\sigma w_i))} & r_i \neq 0 \\ \psi'(0), & r_i = 0. \end{cases}$$

The value of θ at each iteration is given by the weighted least-squares regression of y on X. This is carried out by first transforming the y and X by

$$\begin{aligned} \tilde{y}_i &= y_i\sqrt{G_{ii}} \\ \tilde{x}_{ij} &= x_{ij}\sqrt{G_{ii}}, \quad j = 1, 2, \ldots, m \end{aligned}$$

and then using F04JGF. If X is of full column rank then an orthogonal-triangular (QR) decomposition is used; if not, a singular value decomposition is used.

Observations with zero or negative weights are not included in the solution.

Note. There is no explicit provision in the routine for a constant term in the regression model. However, the addition of a dummy variable whose value is 1.0 for all observations will produce a value of $\hat{\theta}$ corresponding to the usual constant term.

G02HDF is based on routines in ROBETH, see Marazzi [3].

4 References

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

[2] Huber P J (1981) *Robust Statistics* Wiley

[3] Marazzi A (1987) Subroutines for robust and bounded influence regression in ROBETH *Cah. Rech. Doc. IUMSP, No. 3 ROB 2* Institut Universitaire de Médecine Sociale et Préventive, Lausanne

5 Parameters

1: CHI — ***real*** FUNCTION, supplied by the user. *External Procedure*

If ISIGMA > 0, CHI must return the value of the weight function χ for a given value of its argument. The value of χ must be non-negative.

Its specification is:

```
real FUNCTION CHI(T)
real              T
```

1: T — ***real*** *Input*

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

If ISIGMA $\leq$ 0, the actual argument CHI may be the dummy routine G02HDZ. (G02HDZ is included in the NAG Fortran Library and so need not be supplied by the user. Its name may be implementation-dependent: see the Users' Note for your implementation). CHI must be declared as EXTERNAL in the (sub)program from which G02HDF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: PSI — ***real*** FUNCTION, supplied by the user. *External Procedure*

PSI must return the value of the weight function ψ for a given value of its argument.

Its specification is:

```
real FUNCTION PSI(T)
real              T
```

1: T — ***real*** *Input*

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

PSI must be declared as EXTERNAL in the (sub)program from which G02HDF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

3: PSIP0 — ***real*** *Input*

On entry: the value of $\psi'(0)$.

4: BETA — ***real*** *Input*

On entry: if ISIGMA < 0, BETA must specify the value of β_1.

For Huber and Schweppe type regressions, β_1 is the 75th percentile of the standard Normal distribution (see G01FAF). For Mallows type regression β_1 is the solution to

$$\frac{1}{n}\sum_{i=1}^{n}\Phi(\beta_1/\sqrt{w_i}) = 0.75$$

where Φ is the standard Normal cumulative distribution function (see S15ABF).

If ISIGMA > 0, BETA must specify the value of β_2.

$$\beta_2 = \int_{-\infty}^{\infty}\chi(z)\phi(z)\,dz, \qquad \text{in Huber case;}$$

$$\beta_2 = \frac{1}{n}\sum_{i=1}^{n}w_i\int_{-\infty}^{\infty}\chi(z)\phi(z)\,dz, \qquad \text{in Mallows case;}$$

$$\beta_2 = \frac{1}{n}\sum_{i=1}^{n}w_i^2\int_{-\infty}^{\infty}\chi(z/w_i)\phi(z)\,dz, \qquad \text{in Schweppe case;}$$

where ϕ is the standard normal density i.e., $\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right)$.

If ISIGMA $= 0$, BETA is not referenced.

Constraint: if ISIGMA $\neq 0$, BETA > 0.0.

5: INDW — INTEGER *Input*

On entry: determines the type of regression to be performed.

INDW $= 0$, Huber type regression.
INDW < 0, Mallows type regression.
INDW > 0, Schweppe type regression.

6: ISIGMA — INTEGER *Input*

On entry: determines how σ is to be estimated.

ISIGMA < 0, σ is estimated by median absolute deviation of residuals.
ISIGMA $= 0$, σ is held constant at its initial value.
ISIGMA > 0, σ is estimated using the χ function.

7: N — INTEGER *Input*

On entry: the number n, of observations.

Constraint: N > 1.

8: M — INTEGER *Input*

On entry: the number m, of independent variables.

Constraint: $1 \leq$ M $<$ N.

9: X(IX,M) — ***real*** array *Input/Output*

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

If INDW < 0, then during calculations the elements of X will be transformed as described in Section 3. Before exit the inverse transformation will be applied. As a result there may be slight differences between the input X and the output X.

On exit: unchanged, except as described above.

10: IX — INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G02HDF is called.

Constraint: IX $\geq$ N.

11: Y(N) — ***real*** array *Input/Output*

On entry: the data values of the dependent variable.

Y(i) must contain the value of y for the ith observation, for $i = 1, 2, \ldots, n$.

If INDW < 0, then during calculations the elements of Y will be transformed as described in Section 3. Before exit the inverse transformation will be applied. As a result there may be slight differences between the input Y and the output Y.

On exit: unchanged, except as described above.

12: WGT(N) — ***real*** array *Input/Output*

On entry: the weight for the ith observation, for $i = 1, 2, \ldots, n$.

If INDW < 0, then during calculations elements of WGT will be transformed as described in Section 3. Before exit the inverse transformation will be applied. As a result there may be slight differences between the input WGT and the output WGT.

If WGT(i) ≤ 0, then the ith observation is not included in the analysis.

If INDW $= 0$, WGT is not referenced.

On exit: unchanged, except as described above.

13: THETA(M) — ***real*** array *Input/Output*

On entry: starting values of the parameter vector θ. These may be obtained from least-squares regression. Alternatively if ISIGMA < 0 and SIGMA $= 1$ or if ISIGMA > 0 and SIGMA approximately equals the standard deviation of the dependent variable, y, then THETA(i) $= 0.0$, for $i = 1, 2, \ldots, m$ may provide reasonable starting values.

On exit: the M-estimate of θ_i, for $i = 1, 2, \ldots, m$.

14: K — INTEGER *Output*

On exit: the column rank of the matrix X.

15: SIGMA — ***real*** *Input/Output*

On entry: a starting value for the estimation of σ. SIGMA should be approximately the standard deviation of the residuals from the model evaluated at the value of θ given by THETA on entry.

Constraint: SIGMA > 0.0.

On exit: the final estimate of σ if ISIGMA $\neq 0$ or the value assigned on entry if ISIGMA $= 0$.

16: RS(N) — ***real*** array *Output*

On exit: the residuals from the model evaluated at final value of THETA i.e., RS contains the vector $(y - X\hat{\theta})$.

17: TOL — ***real*** *Input*

On entry: the relative precision for the final estimates. Convergence is assumed when both the relative change in the value of SIGMA and the relative change in the value of each element of THETA are less than TOL.

It is advisable for TOL to be greater than 100****machine precision***.

Constraint: TOL > 0.0.

18: EPS — ***real*** *Input*

On entry: a relative tolerance to be used to determine the rank of X. See F04JGF for further details.

If EPS < ***machine precision*** or EPS > 1.0 then ***machine precision*** will be used in place of TOL.

A reasonable value for EPS is 5.0×10^{-6} where this value is possible.

19: MAXIT — INTEGER *Input*

On entry: the maximum number of iterations that should be used during the estimation.

A value of MAXIT $= 50$ should be adequate for most uses.

Constraint: MAXIT > 0.

20: NITMON — INTEGER *Input*

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

If NITMON ≤ 0 no information is printed.
If NITMON > 0 then on the first and every NITMON iterations the values of SIGMA, THETA and the change in THETA during the iteration are printed.

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

21: NIT — INTEGER *Output*

On exit: the number of iterations that were used during the estimation.

22: WK((M+4)∗N) — ***real*** array *Workspace*

23: 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 or gives a warning (see Section 6).

For this routine, because the values of output parameters may be useful even if IFAIL ≠ 0 on exit, users are recommended to set IFAIL to −1 before entry. **It is then 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 specified by the routine:

IFAIL = 1

On entry, N ≤ 1,
or M < 1,
or N ≤ M,
or IX < N.

IFAIL = 2

On entry, BETA ≤ 0.0, and ISIGMA ≠ 0,
or SIGMA ≤ 0.0.

IFAIL = 3

On entry, TOL ≤ 0.0,
or MAXIT ≤ 0.

IFAIL = 4

A value returned by the CHI function is negative.

IFAIL = 5

During iterations a value of SIGMA ≤ 0 was encountered.

IFAIL = 6

A failure occurred in F04JGF. This is an extremely unlikely error. If it occurs, please consult NAG.

IFAIL = 7

The weighted least-squares equations are not of full rank. This may be due to the X matrix not being of full rank, in which case the results will be valid. It may also occur if some of the G_{ii} values become very small or zero, see Section 8. The rank of the equations is given by K. If the matrix just fails the test for non-singularity then the result IFAIL = 7 and K = M is possible (see F04JGF).

IFAIL = 8

The routine has failed to converge in MAXIT iterations.

IFAIL = 9

Having removed cases with zero weight, the value of N − K ≤ 0, i.e., no degree of freedom for error. This error will only occur if ISIGMA > 0.

7 Accuracy

The accuracy of the results is controlled by TOL. For the accuracy of the weighted least-squares see F04JGF.

8 Further Comments

In cases when ISIGMA ≥ 0 it is important for the value of SIGMA to be of a reasonable magnitude. Too small a value may cause too many of the winsorised residuals, i.e., $\psi(r_i/\sigma)$, to be zero, which will lead to convergence problems and may trigger the IFAIL = 7 error.

By suitable choice of the functions CHI and PSI this routine may be used for other applications of iterative weighted least-squares.

For the variance-covariance matrix of θ see G02HFF.

9 Example

Having input X, Y and the weights, a Schweppe type regression is performed using Huber's ψ function. The subroutine BETCAL calculates the appropriate value of β_2.

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.

```
*     G02HDF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX
      PARAMETER        (NMAX=9,MMAX=3)
*     .. Local Scalars ..
      real             BETA, EPS, PSIP0, SIGMA, TOL
      INTEGER          I, IFAIL, INDW, ISIGMA, IX, J, K, M, MAXIT, N,
     +                 NIT, NITMON
*     .. Local Arrays ..
      real             RS(NMAX), THETA(MMAX), WGT(NMAX),
     +                 WK(NMAX*(MMAX+4)), X(NMAX,MMAX), Y(NMAX)
*     .. External Functions ..
      real             CHI, PSI
      EXTERNAL         CHI, PSI
*     .. External Subroutines ..
      EXTERNAL         BETCAL, G02HDF, X04ABF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G02HDF 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.LE.NMAX) .AND. (M.LE.MMAX)) THEN
*        Read in the X matrix, the Y values and set X(i,1) to 1 for the
*        constant term
         DO 20 I = 1, N
            READ (NIN,*) (X(I,J),J=2,M), Y(I)
            X(I,1) = 1.0e0
```

```
   20    CONTINUE
*        Read in weights
         DO 40 I = 1, N
            READ (NIN,*) WGT(I)
   40    CONTINUE
         CALL BETCAL(N,WGT,BETA)
*        Set other parameter values
         IX = NMAX
         MAXIT = 50
         TOL = 0.5e-4
         EPS = 0.5e-5
         PSIP0 = 1.0e0
*        Set value of ISIGMA and initial value of SIGMA
         ISIGMA = 1
         SIGMA = 1.0e0
*        Set initial value of THETA
         DO 60 J = 1, M
            THETA(J) = 0.0e0
   60    CONTINUE
*        * Change NITMON to a positive value if monitoring information
*          is required *
         NITMON = 0
*        Schweppe type regression
         INDW = 1
         IFAIL = -1
*
         CALL G02HDF(CHI,PSI,PSIP0,BETA,INDW,ISIGMA,N,M,X,IX,Y,WGT,
     +               THETA,K,SIGMA,RS,TOL,EPS,MAXIT,NITMON,NIT,WK,IFAIL)
*
         WRITE (NOUT,*)
         IF (IFAIL.NE.0 .AND. IFAIL.NE.7) THEN
            WRITE (NOUT,99999) 'G02HDF fails, IFAIL = ', IFAIL
         ELSE
            IF (IFAIL.EQ.7) THEN
               WRITE (NOUT,99999) 'G02HDF returned IFAIL = ', IFAIL
               WRITE (NOUT,*)
     +           'Some of the following results may be unreliable'
            END IF
            WRITE (NOUT,99998) 'G02HDF required ', NIT,
     +        ' iterations to converge'
            WRITE (NOUT,99998) '                  K = ', K
            WRITE (NOUT,99997) '              Sigma = ', SIGMA
            WRITE (NOUT,*) '    THETA'
            DO 80 J = 1, M
               WRITE (NOUT,99996) THETA(J)
   80       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*) '  Weights  Residuals'
            DO 100 I = 1, N
               WRITE (NOUT,99995) WGT(I), RS(I)
  100       CONTINUE
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I2)
99998 FORMAT (1X,A,I4,A)
99997 FORMAT (1X,A,F9.4)
```

```
99996 FORMAT (1X,F9.4)
99995 FORMAT (1X,2F9.4)
      END
*
      real FUNCTION PSI(T)
*     .. Parameters ..
      real              C
      PARAMETER         (C=1.5e0)
*     .. Scalar Arguments ..
      real              T
*     .. Intrinsic Functions ..
      INTRINSIC         ABS
*     .. Executable Statements ..
      IF (T.LE.-C) THEN
         PSI = -C
      ELSE IF (ABS(T).LT.C) THEN
         PSI = T
      ELSE
         PSI = C
      END IF
      RETURN
      END
*
      real FUNCTION CHI(T)
*     .. Parameters ..
      real              DCHI
      PARAMETER         (DCHI=1.5e0)
*     .. Scalar Arguments ..
      real              T
*     .. Local Scalars ..
      real              PS
*     .. Intrinsic Functions ..
      INTRINSIC         ABS
*     .. Executable Statements ..
      PS = DCHI
      IF (ABS(T).LT.DCHI) PS = T
      CHI = PS*PS/2.0e0
      RETURN
      END
*
      SUBROUTINE BETCAL(N,WGT,BETA)
*     Calculate BETA for Schweppe type regression
*     .. Parameters ..
      real              DCHI
      PARAMETER         (DCHI=1.5e0)
*     .. Scalar Arguments ..
      real              BETA
      INTEGER           N
*     .. Array Arguments ..
      real              WGT(N)
*     .. Local Scalars ..
      real              AMAXEX, ANORMC, B, D2, DC, DW, DW2, PC, W2
      INTEGER           I, IFAIL
*     .. External Functions ..
      real              S15ABF, X01AAF, X02AKF
      EXTERNAL          S15ABF, X01AAF, X02AKF
*     .. Intrinsic Functions ..
      INTRINSIC         EXP, LOG, real, SQRT
```

```
*     .. Executable Statements ..
      AMAXEX = -LOG(X02AKF())
      ANORMC = SQRT(X01AAF(0.0e0)*2.0e0)
      D2 = DCHI*DCHI
      BETA = 0.0e0
      DO 20 I = 1, N
         W2 = WGT(I)*WGT(I)
         DW = WGT(I)*DCHI
         IFAIL = 0
         PC = S15ABF(DW,IFAIL)
         DW2 = DW*DW
         DC = 0.0e0
         IF (DW2.LT.AMAXEX) DC = EXP(-DW2/2.0e0)/ANORMC
         B = (-DW*DC+PC-0.5e0)/W2 + (1.0e0-PC)*D2
         BETA = B*W2/real(N) + BETA
   20 CONTINUE
      RETURN
      END
```

9.2 Program Data

```
G02HDF Example Program Data

    5    3           : N  M

  -1.0 -1.0 10.5     : X2  X3  Y
  -1.0  1.0 11.3
   1.0 -1.0 12.6
   1.0  1.0 13.4
   0.0  3.0 17.1     : End of X1 X2 and Y values

   0.4039            : WGT
   0.5012
   0.4039
   0.5012
   0.3862            : End of the weights
```

9.3 Program Results

```
 G02HDF Example Program Results

 G02HDF required    5 iterations to converge
                    K =    3
                Sigma =    2.7783
     THETA
   12.2321
    1.0500
    1.2464

   Weights  Residuals
    0.4039   0.5643
    0.5012  -1.1286
    0.4039   0.5643
    0.5012  -1.1286
    0.3862   1.1286
```