NAG Library Routine Document

G02HFF

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

G02HFF calculates an estimate of the asymptotic variance-covariance matrix for the bounded influence regression estimates (M-estimates). It is intended for use with G02HDF.

2 Specification

```
SUBROUTINE G02HFF (PSI, PSP, INDW, INDC, SIGMA, N, M, X, LDX, RS, WGT, C,      &
                   LDC, WK, IFAIL)
INTEGER            INDW, INDC, N, M, LDX, LDC, IFAIL
REAL (KIND=nag_wp) PSI, PSP, SIGMA, X(LDX,M), RS(N), WGT(N), C(LDC,M),         &
                   WK(M*(N+M+1)+2*N)
EXTERNAL           PSI, PSP
```

3 Description

For a description of bounded influence regression see G02HDF. Let θ be the regression parameters and let C be the asymptotic variance-covariance matrix of $\hat{\theta}$. Then for Huber type regression

$$C = f_H\left(X^{\mathrm{T}}X\right)^{-1}\hat{\sigma}^2,$$

where

$$f_H = \frac{1}{n-m}\frac{\sum_{i=1}^{n}\psi^2(r_i/\hat{\sigma})}{\left(\frac{1}{n}\sum\psi'\left(\frac{r_i}{\hat{\sigma}}\right)\right)^2}\kappa^2$$

$$\kappa^2 = 1 + \frac{m}{n}\frac{\frac{1}{n}\sum_{i=1}^{n}\left(\psi'(r_i/\hat{\sigma}) - \frac{1}{n}\sum_{i=1}^{n}\psi'(r_i/\hat{\sigma})\right)^2}{\left(\frac{1}{n}\sum_{i=1}^{n}\psi'\left(\frac{r_i}{\hat{\sigma}}\right)\right)^2},$$

see Huber (1981) and Marazzi (1987).

For Mallows and Schweppe type regressions, C is of the form

$$\frac{\hat{\sigma}^2}{n}S_1^{-1}S_2S_1^{-1},$$

where $S_1 = \frac{1}{n}X^{\mathrm{T}}DX$ and $S_2 = \frac{1}{n}X^{\mathrm{T}}PX$.

D is a diagonal matrix such that the ith element approximates $E\left(\psi'(r_i/(\sigma w_i))\right)$ in the Schweppe case and $E\left(\psi'(r_i/\sigma)w_i\right)$ in the Mallows case.

P is a diagonal matrix such that the ith element approximates $E\left(\psi^2(r_i/(\sigma w_i))w_i^2\right)$ in the Schweppe case and $E\left(\psi^2(r_i/\sigma)w_i^2\right)$ in the Mallows case.

Two approximations are available in G02HFF:

1. Average over the r_i

Schweppe / Mallows

$$D_i = \left(\frac{1}{n}\sum_{j=1}^{n}\psi'\left(\frac{r_j}{\hat{\sigma}w_i}\right)\right)w_i \qquad D_i = \left(\frac{1}{n}\sum_{j=1}^{n}\psi'\left(\frac{r_j}{\hat{\sigma}}\right)\right)w_i$$

$$P_i = \left(\frac{1}{n}\sum_{j=1}^{n}\psi^2\left(\frac{r_j}{\hat{\sigma}w_i}\right)\right)w_i^2 \qquad P_i = \left(\frac{1}{n}\sum_{j=1}^{n}\psi^2\left(\frac{r_j}{\hat{\sigma}}\right)\right)w_i^2$$

2. Replace expected value by observed

Schweppe / Mallows

$$D_i = \psi'\left(\frac{r_i}{\hat{\sigma}w_i}\right)w_i \qquad D_i = \psi'\left(\frac{r_i}{\hat{\sigma}}\right)w_i$$

$$P_i = \psi^2\left(\frac{r_i}{\hat{\sigma}w_i}\right)w_i^2 \qquad P_i = \psi^2\left(\frac{r_i}{\hat{\sigma}}\right)w_i^2$$

See Hampel *et al.* (1986) and Marazzi (1987).

In all cases $\hat{\sigma}$ is a robust estimate of σ.

G02HFF is based on routines in ROBETH; see Marazzi (1987).

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 (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: PSI – REAL (KIND=nag_wp) FUNCTION, supplied by the user. *External Procedure*

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

The specification of PSI is:

```
FUNCTION PSI (T)

REAL (KIND=nag_wp) PSI
REAL (KIND=nag_wp) T
```

1: T – REAL (KIND=nag_wp) *Input*

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

PSI must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which G02HFF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: PSP – REAL (KIND=nag_wp) FUNCTION, supplied by the user. *External Procedure*

PSP must return the value of $\psi'(t) = \frac{d}{dt}\psi(t)$ for a given value of its argument.

The specification of PSP is:

```
FUNCTION PSP (T)

REAL (KIND=nag_wp) PSP
REAL (KIND=nag_wp) T
```

1: T – REAL (KIND=nag_wp) *Input*

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

PSP must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which G02HFF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

3: INDW – INTEGER *Input*

On entry: the type of regression for which the asymptotic variance-covariance matrix is to be calculated.

INDW < 0
Mallows type regression.

INDW $= 0$
Huber type regression.

INDW > 0
Schweppe type regression.

4: INDC – INTEGER *Input*

On entry: if INDW $\neq 0$, INDC must specify the approximation to be used.

If INDC $= 1$, averaging over residuals.

If INDC $\neq 1$, replacing expected by observed.

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

5: SIGMA – REAL (KIND=nag_wp) *Input*

On entry: the value of $\hat{\sigma}$, as given by G02HDF.

Constraint: SIGMA > 0.0.

6: N – INTEGER *Input*

On entry: n, the number of observations.

Constraint: N > 1.

7: M – INTEGER *Input*

On entry: m, the number of independent variables.

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

8: X(LDX,M) – REAL (KIND=nag_wp) array *Input*

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

9: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G02HFF is called.

Constraint: LDX $\geq$ N.

10: RS(N) – REAL (KIND=nag_wp) array *Input*

On entry: the residuals from the bounded influence regression. These are given by G02HDF.

11: WGT(N) – REAL (KIND=nag_wp) array *Input*

On entry: if INDW $\neq 0$, WGT must contain the vector of weights used by the bounded influence regression. These should be used with G02HDF.

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

12: C(LDC,M) – REAL (KIND=nag_wp) array *Output*

On exit: the estimate of the variance-covariance matrix.

13: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which G02HFF is called.

Constraint: LDC $\geq$ M.

14: WK$(\mathrm{M} \times (\mathrm{N} + \mathrm{M} + 1) + 2 \times \mathrm{N})$ – REAL (KIND=nag_wp) array *Output*

On exit: if INDW $\neq 0$, WK(i), for $i = 1, 2, \ldots, n$, will contain the diagonal elements of the matrix D and WK(i), for $i = n + 1, \ldots, 2n$, will contain the diagonal elements of matrix P.

The rest of the array is used as workspace.

15: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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, if you are 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.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

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 $\leq$ M,
or LDC $<$ M,
or LDX $<$ N.

IFAIL $= 2$

On entry, SIGMA ≤ 0.0.

IFAIL $= 3$

If INDW $= 0$ then the matrix $X^{\mathrm{T}}X$ is either not positive definite, possibly due to rounding errors, or is ill-conditioned.

If INDW $\neq 0$ then the matrix S_1 is singular or almost singular. This may be due to many elements of D being zero.

IFAIL $= 4$

Either the value of $\frac{1}{n}\sum_{i=1}^{n}\psi'\left(\frac{r_i}{\hat{\sigma}}\right) = 0$,

or $\kappa = 0$,

or $\sum_{i=1}^{n}\psi^2\left(\frac{r_i}{\hat{\sigma}}\right) = 0$.

In this situation G02HFF returns C as $\left(X^{\mathrm{T}}X\right)^{-1}$.

7 Accuracy

In general, the accuracy of the variance-covariance matrix will depend primarily on the accuracy of the results from G02HDF.

8 Further Comments

G02HFF is only for situations in which X has full column rank.

Care has to be taken in the choice of the ψ function since if $\psi'(t) = 0$ for too wide a range then either the value of f_H will not exist or too many values of D_i will be zero and it will not be possible to calculate C.

9 Example

The asymptotic variance-covariance matrix is calculated for a Schweppe type regression. The values of X, $\hat{\sigma}$ and the residuals and weights are read in. The averaging over residuals approximation is used.

9.1 Program Text

```
!   G02HFF Example Program Text
!   Mark 24 Release. NAG Copyright 2012.

    Module g02hffe_mod

!     G02HFF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter        :: tc = 1.5_nag_wp
      Integer, Parameter                   :: nin = 5, nout = 6
    Contains
      Function psi(t)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                   :: psi
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: t
!       .. Intrinsic Procedures ..
        Intrinsic                            :: abs
!       .. Executable Statements ..
        If (t<=-tc) Then
          psi = -tc
        Else If (abs(t)<tc) Then
          psi = t
        Else
```

```
          psi = tc
        End If
        Return
      End Function psi

      Function psp(t)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                    :: psp
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)       :: t
!       .. Intrinsic Procedures ..
        Intrinsic                             :: abs
!       .. Executable Statements ..
        psp = 0.0_nag_wp
        If (abs(t)<tc) Then
          psp = 1.0_nag_wp
        End If
        Return
      End Function psp
    End Module g02hffe_mod
    Program g02hffe

!     G02HFF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g02hff, nag_wp, x04caf
      Use g02hffe_mod, Only: nin, nout, psi, psp
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                    :: sigma
      Integer                               :: i, ifail, indc, indw, ldc, ldx,  &
                                               m, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable       :: c(:,:), rs(:), wgt(:), wk(:),    &
                                               x(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'G02HFF Example Program Results'
      Write (nout,*)
      Flush (nout)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size
      Read (nin,*) n, m

      ldx = n
      ldc = m
      Allocate (x(ldx,m),wgt(n),rs(n),wk(m*(n+m+1)+2*n),c(ldc,m))

!     Read in the data
      Read (nin,*)(x(i,1:m),i=1,n)

!     Read in SIGMA
      Read (nin,*) sigma

!     Read in weights and residuals
      Read (nin,*)(wgt(i),rs(i),i=1,n)

!     Read in control parameters
      Read (nin,*) indw, indc

!     Estimate variance-covariance matrix
      ifail = 0
      Call g02hff(psi,psp,indw,indc,sigma,n,m,x,ldx,rs,wgt,c,ldc,wk,ifail)
```

```
!     Display results
      ifail = 0
      Call x04caf('General',' ',m,m,c,ldc,'Covariance matrix',ifail)

    End Program g02hffe
```

9.2 Program Data

```
G02HFF Example Program Data
    5    3                : N  M
  1.0 -1.0 -1.0
  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
    20.7783               : SIGMA
    0.4039    0.5643
    0.5012   -1.1286
    0.4039    0.5643
    0.5012   -1.1286
    0.3862    1.1286      : End of weights and residuals, WGT and RS
 1  1                     : INDW,INDC
```

9.3 Program Results

```
 G02HFF Example Program Results

 Covariance matrix
          1       2       3
 1   0.2070  0.0000 -0.0478
 2   0.0000  0.2229  0.0000
 3  -0.0478  0.0000  0.0796
```