NAG Library Routine Document

G02HMF

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

G02HMF computes a robust estimate of the covariance matrix for user-supplied weight functions. The derivatives of the weight functions are not required.

2 Specification

```
SUBROUTINE G02HMF (UCV, RUSER, INDM, N, M, X, LDX, COV, A, WT, THETA, BL,       &
                   BD, MAXIT, NITMON, TOL, NIT, WK, IFAIL)

INTEGER            INDM, N, M, LDX, MAXIT, NITMON, NIT, IFAIL
REAL (KIND=nag_wp) RUSER(*), X(LDX,M), COV(M*(M+1)/2), A(M*(M+1)/2),            &
                   WT(N), THETA(M), BL, BD, TOL, WK(2*M)
EXTERNAL           UCV
```

3 Description

For a set of n observations on m variables in a matrix X, a robust estimate of the covariance matrix, C, and a robust estimate of location, θ, are given by

$$C = \tau^2 \left(A^{\mathrm{T}} A\right)^{-1},$$

where τ^2 is a correction factor and A is a lower triangular matrix found as the solution to the following equations.

$$z_i = A(x_i - \theta)$$

$$\frac{1}{n}\sum_{i=1}^{n} w\left(\|z_i\|_2\right) z_i = 0$$

and

$$\frac{1}{n}\sum_{i=1}^{n} u\left(\|z_i\|_2\right) z_i z_i^{\mathrm{T}} - v\left(\|z_i\|_2\right) I = 0,$$

where x_i is a vector of length m containing the elements of the ith row of X,

z_i is a vector of length m,

I is the identity matrix and 0 is the zero matrix.

and w and u are suitable functions.

G02HMF covers two situations:

(i) $v(t) = 1$ for all t,

(ii) $v(t) = u(t)$.

The robust covariance matrix may be calculated from a weighted sum of squares and cross-products matrix about θ using weights $wt_i = u(\|z_i\|)$. In case (i) a divisor of n is used and in case (ii) a divisor of $\sum_{i=1}^{n} wt_i$ is used. If $w(.) = \sqrt{u(.)}$, then the robust covariance matrix can be calculated by scaling each row of X by $\sqrt{wt_i}$ and calculating an unweighted covariance matrix about θ.

In order to make the estimate asymptotically unbiased under a Normal model a correction factor, τ^2, is needed. The value of the correction factor will depend on the functions employed (see Huber (1981) and Marazzi (1987)).

G02HMF finds A using the iterative procedure as given by Huber; see Huber (1981).

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

and

$$\theta_{j_k} = \frac{b_j}{D_1} + \theta_{j_{k-1}},$$

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

$$s_{jl} = \begin{cases} -\min\left[\max\left(h_{jl}/D_2, -BL\right), BL\right], & j > l \\ -\min\left[\max\left(\frac{1}{2}\left(h_{jj}/D_2 - 1\right), -BD\right), BD\right], & j = l \end{cases},$$

where

$$D_1 = \sum_{i=1}^{n} w\left(\|z_i\|_2\right)$$

$$D_2 = \sum_{i=1}^{n} u\left(\|z_i\|_2\right)$$

$$h_{jl} = \sum_{i=1}^{n} u\left(\|z_i\|_2\right) z_{ij} z_{il}, \text{ for } j \geq l$$

$$b_j = \sum_{i=1}^{n} w\left(\|z_i\|_2\right)\left(x_{ij} - b_j\right)$$

and *BD* and *BL* are suitable bounds.

The value of τ may be chosen so that C is unbiased if the observations are from a given distribution.

G02HMF is based on routines in ROBETH; see Marazzi (1987).

4 References

Huber P J (1981) *Robust Statistics* Wiley

Marazzi A (1987) 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 – SUBROUTINE, supplied by the user. *External Procedure*

UCV must return the values of the functions u and w for a given value of its argument.

The specification of UCV is:

```
SUBROUTINE UCV (T, RUSER, U, W)
```

```
REAL (KIND=nag_wp) T, RUSER(*), U, W
```

1: T – REAL (KIND=nag_wp) *Input*

On entry: the argument for which the functions u and w must be evaluated.

2: RUSER($*$) – REAL (KIND=nag_wp) array *User Workspace*

UCV is called with the parameter RUSER as supplied to G02HMF. You are free to use the array RUSER to supply information to UCV as an alternative to using COMMON global variables.

3: U – REAL (KIND=nag_wp) *Output*

On exit: the value of the u function at the point T.

Constraint: $\mathrm{U} \geq 0.0$.

4: W – REAL (KIND=nag_wp) *Output*

On exit: the value of the w function at the point T.

Constraint: $\mathrm{W} \geq 0.0$.

UCV must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which G02HMF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: RUSER($*$) – REAL (KIND=nag_wp) array *User Workspace*

RUSER is not used by G02HMF, but is passed directly to UCV and may be used to pass information to this routine as an alternative to using COMMON global variables.

3: INDM – INTEGER *Input*

On entry: indicates which form of the function v will be used.

$\mathrm{INDM} = 1$
$v = 1$.

$\mathrm{INDM} \neq 1$
$v = u$.

4: N – INTEGER *Input*

On entry: n, the number of observations.

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

5: M – INTEGER *Input*

On entry: m, the number of columns of the matrix X, i.e., number of independent variables.

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

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

On entry: $\mathrm{X}(i,j)$ must contain the ith observation on the jth variable, for $i = 1,2,\dots,n$ and $j = 1,2,\dots,m$.

7: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G02HMF is called.

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

8: COV($\mathrm{M} \times (\mathrm{M}+1)/2$) – REAL (KIND=nag_wp) array *Output*

On exit: a robust estimate of the covariance matrix, C. The upper triangular part of the matrix C is stored packed by columns (lower triangular stored by rows), that is C_{ij} is returned in $\mathrm{COV}(j \times (j-1)/2 + i)$, $i \leq j$.

9: A$(\mathrm{M} \times (\mathrm{M}+1)/2)$ – REAL (KIND=nag_wp) 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 $\neq 0$, and 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.

Constraint: $\mathrm{A}(j \times (j-1)/2 + j) \neq 0.0$, for $j = 1, 2, \ldots, m$.

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

10: WT(N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{WT}(i)$ contains the weights, $wt_i = u\left(\|z_i\|_2\right)$, for $i = 1, 2, \ldots, n$.

11: THETA(M) – REAL (KIND=nag_wp) array *Input/Output*

On entry: an initial estimate of the location parameter, θ_j, for $j = 1, 2, \ldots, m$.

In many cases an initial estimate of $\theta_j = 0$, for $j = 1, 2, \ldots, m$, will be adequate. Alternatively medians may be used as given by G07DAF.

On exit: contains the robust estimate of the location parameter, θ_j, for $j = 1, 2, \ldots, m$.

12: BL – REAL (KIND=nag_wp) *Input*

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

Suggested value: $\mathrm{BL} = 0.9$.

Constraint: $\mathrm{BL} > 0.0$.

13: BD – REAL (KIND=nag_wp) *Input*

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

Suggested value: $\mathrm{BD} = 0.9$.

Constraint: $\mathrm{BD} > 0.0$.

14: MAXIT – INTEGER *Input*

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

Suggested value: $\mathrm{MAXIT} = 150$.

Constraint: $\mathrm{MAXIT} > 0$.

15: NITMON – INTEGER *Input*

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

$\mathrm{NITMON} > 0$
The value of A, θ and δ (see Section 7) will be printed at the first and every NITMON iterations.

$\mathrm{NITMON} \leq 0$
No iteration monitoring is printed.

When printing occurs the output is directed to the current advisory message channel (See X04ABF.)

16: TOL – REAL (KIND=nag_wp) *Input*

On entry: the relative precision for the final estimate of the covariance matrix. Iteration will stop when maximum δ (see Section 7) is less than TOL.

Constraint: $\mathrm{TOL} > 0.0$.

17: NIT – INTEGER *Output*

On exit: the number of iterations performed.

18: WK($2 \times$ M) – REAL (KIND=nag_wp) array *Workspace*

19: 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 $<$ M,
or LDX $<$ N.

IFAIL $= 2$

On entry, TOL ≤ 0.0,
or MAXIT ≤ 0,
or diagonal element of A $= 0.0$,
or BL ≤ 0.0,
or BD ≤ 0.0.

IFAIL $= 3$

A column of X has a constant value.

IFAIL $= 4$

Value of U or W returned by UCV < 0.

IFAIL $= 5$

The routine has failed to converge in MAXIT iterations.

IFAIL $= 6$

Either the sum D_1 or the sum D_2 is zero. This may be caused by the functions u or w being too strict for the current estimate of A (or C). You should either try a larger initial estimate of A or make the u and w functions less strict.

7 Accuracy

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

(i) $d1 =$ the maximum value of $\left|s_{jl}\right|$

(ii) $d2 =$ the maximum absolute change in $wt(i)$

(iii) $d3 =$ the maximum absolute relative change in θ_j

and let $\delta = \max(d1, d2, d3)$. Then the iterative procedure is assumed to have converged when $\delta <$ TOL.

8 Further Comments

The existence of A will depend upon the function u (see Marazzi (1987)); 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.

If derivatives of the u and w functions are available then the method used in G02HLF will usually give much faster convergence.

9 Example

A sample of 10 observations on three variables is read in along with initial values for A and θ and parameter values for the u and w functions, c_u and c_w. The covariance matrix computed by G02HMF is printed along with the robust estimate of θ.

UCV computes the Huber's weight functions:

$$u(t) = 1, \quad \text{if} \quad t \leq c_u^2$$

$$u(t) = \frac{c_u}{t^2}, \quad \text{if} \quad t > c_u^2$$

and

$$w(t) = 1, \quad \text{if} \quad t \leq c_w$$

$$w(t) = \frac{c_w}{t}, \quad \text{if} \quad t > c_w.$$

9.1 Program Text

```
!   G02HMF Example Program Text
!   Mark 24 Release. NAG Copyright 2012.

    Module g02hmfe_mod

!     G02HMF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                   :: iset = 1, nin = 5, nout = 6
    Contains
      Subroutine ucv(t,ruser,u,w)

!       u function

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: t
        Real (Kind=nag_wp), Intent (Out)     :: u, w
!       .. Array Arguments ..
```

```
        Real (Kind=nag_wp), Intent (Inout)   :: ruser(*)
!       .. Local Scalars ..
        Real (Kind=nag_wp)                   :: cu, cw, t2
!       .. Executable Statements ..
        cu = ruser(1)
        u = 1.0_nag_wp
        If (t/=0.0_nag_wp) Then
          t2 = t*t
          If (t2>cu) Then
            u = cu/t2
          End If
        End If
!       w function
        cw = ruser(2)
        If (t>cw) Then
          w = cw/t
        Else
          w = 1.0_nag_wp
        End If
      End Subroutine ucv
    End Module g02hmfe_mod
    Program g02hmfe

!     G02HMF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g02hmf, nag_wp, x04abf, x04ccf
      Use g02hmfe_mod, Only: iset, nin, nout, ucv
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                   :: bd, bl, tol
      Integer                              :: i, ifail, indm, la, lcov, ldx,   &
                                              lruser, m, maxit, n, nadv, nit,  &
                                              nitmon
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable      :: a(:), cov(:), ruser(:),          &
                                              theta(:), wk(:), wt(:), x(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'G02HMF Example Program Results'
      Write (nout,*)

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

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

      ldx = n
      lruser = 2
      la = ((m+1)*m)/2
      lcov = la
      Allocate (x(ldx,m),ruser(lruser),cov(lcov),a(la),wt(n),theta(m),wk(2*m))

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

!     Read in the initial value of A
      Read (nin,*) a(1:la)

!     Read in the initial value of THETA
      Read (nin,*) theta(1:m)

!     Read in the values of the parameters of the ucv functions
      Read (nin,*) ruser(1:lruser)

!     Read in the control parameters
      Read (nin,*) indm, nitmon, bl, bd, maxit, tol

!     Set the advisory channel to NOUT for monitoring information
      If (nitmon/=0) Then
```

```
        nadv = nout
        Call x04abf(iset,nadv)
      End If

!     Compute robust estimate of variance / covariance matrix
      ifail = 0
      Call g02hmf(ucv,ruser,indm,n,m,x,ldx,cov,a,wt,theta,bl,bd,maxit,nitmon, &
        tol,nit,wk,ifail)

!     Display results
      Write (nout,99999) 'G02HMF required ', nit, ' iterations to converge'
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04ccf('Upper','Non-Unit',m,cov,'Robust covariance matrix',ifail)
      Write (nout,*)
      Write (nout,*) 'Robust estimates of THETA'
      Write (nout,99998) theta(1:m)

99999 Format (1X,A,I0,A)
99998 Format (1X,F10.3)
    End Program g02hmfe
```

9.2 Program Data

```
G02HMF Example Program Data
    10    3                    : N,M
  3.4  6.9  12.2
  6.4  2.5  15.1
  4.9  5.5  14.2
  7.3  1.9  18.2
  8.8  3.6  11.7
  8.4  1.3  17.9
  5.3  3.1  15.0
  2.7  8.1   7.7
  6.1  3.0  21.9
  5.3  2.2  13.9               : End of X
  1.0 0.0 1.0 0.0 0.0 1.0      : A
  0.0 0.0 0.0                  : THETA
  4.0 2.0                      : RUSER
1 0 0.9 0.9 50 5.0E-5          : INDM,NITMON,BL,BD,MAXIT,TOL
```

9.3 Program Results

```
 G02HMF Example Program Results

 G02HMF required 34 iterations to converge

 Robust covariance matrix
             1          2          3
 1      3.2779    -3.6918     4.7391
 2                 5.2841    -6.4087
 3                           11.8373

 Robust estimates of THETA
      5.700
      3.864
     14.704
```