NAG Library Routine Document

G13BCF

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

G13BCF calculates cross-correlations between two time series.

2 Specification

```
SUBROUTINE G13BCF (X, Y, NXY, NL, S, R0, R, STAT, IFAIL)
INTEGER          NXY, NL, IFAIL
REAL (KIND=nag_wp) X(NXY), Y(NXY), S, R0, R(NL), STAT
```

3 Description

Given two series $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$ the routine calculates the cross-correlations between x_t and lagged values of y_t:

$$r_{xy}(l) = \frac{\sum_{t=1}^{n-l}(x_t - \bar{x})(y_{t+l} - \bar{y})}{n s_x s_y}, \quad l = 0, 1, \ldots, L$$

where

$$\bar{x} = \frac{\sum_{t=1}^{n} x_t}{n}$$

$$s_x^2 = \frac{\sum_{t=1}^{n}(x_t - \bar{x})^2}{n}$$

and similarly for y.

The ratio of standard deviations s_y/s_x is also returned, and a portmanteau statistic is calculated:

$$\text{STAT} = n \sum_{l=1}^{L} r_{xy}(l)^2.$$

Provided n is large, L much less than n, and both x_t, y_t are samples of series whose true autocorrelation functions are zero, then, under the null hypothesis that the true cross-correlations between the series are zero, STAT has a χ^2-distribution with L degrees of freedom. Values of STAT in the upper tail of this distribution provide evidence against the null hypothesis.

4 References

Box G E P and Jenkins G M (1976) *Time Series Analysis: Forecasting and Control* (Revised Edition) Holden−Day

5   Parameters

1:    X(NXY) – REAL (KIND=nag_wp) array   *Input*

    On entry: the n values of the x series.

2:    Y(NXY) – REAL (KIND=nag_wp) array   *Input*

    On entry: the n values of the y series.

3:    NXY – INTEGER   *Input*

    On entry: n, the length of the time series.

    Constraint: NXY ≥ 2.

4:    NL – INTEGER   *Input*

    On entry: L, the maximum lag for calculating cross-correlations.

    Constraint: $1 \leq$ NL $<$ NXY.

5:    S – REAL (KIND=nag_wp)   *Output*

    On exit: the ratio of the standard deviation of the y series to the standard deviation of the x series, s_y/s_x.

6:    R0 – REAL (KIND=nag_wp)   *Output*

    On exit: the cross-correlation between the x and y series at lag zero.

7:    R(NL) – REAL (KIND=nag_wp) array   *Output*

    On exit: R(l) contains the cross-correlations between the x and y series at lags L, $r_{xy}(l)$, for $l = 1, 2, \ldots, L$.

8:    STAT – REAL (KIND=nag_wp)   *Output*

    On exit: the statistic for testing for absence of cross-correlation.

9:    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, NXY ≤ 1,
    or   NL < 1,

or   NL ≥ NXY.

IFAIL = 2

One or both of the x and y series have zero variance and hence cross-correlations cannot be calculated.

IFAIL = −99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL = −399

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL = −999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7   Accuracy

All computations are believed to be stable.

8   Parallelism and Performance

G13BCF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G13BCF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9   Further Comments

If $n < 100$, or $L < 10\log(n)$ then the autocorrelations are calculated directly and the time taken by G13BCF is approximately proportional to nL, otherwise the autocorrelations are calculated by utilizing fast Fourier transforms (FFTs) and the time taken is approximately proportional to $n\log(n)$. If FFTs are used then G13BCF internally allocates approximately $6n$ real elements.

10   Example

This example reads two time series of length 20. It calculates and prints the cross-correlations up to lag 15 for the first series leading the second series and then for the second series leading the first series.

10.1 Program Text

```
    Program g13bcfe

!     G13BCF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g13bcf, nag_wp
```

```
!      .. Implicit None Statement ..
       Implicit None
!      .. Parameters ..
       Integer, Parameter                :: nin = 5, nout = 6
!      .. Local Scalars ..
       Real (Kind=nag_wp)                :: r0xy, r0yx, statxy, statyx, sxy, syx
       Integer                           :: i, ifail, nl, nxy
!      .. Local Arrays ..
       Real (Kind=nag_wp), Allocatable :: rxy(:), ryx(:), x(:), y(:)
!      .. Executable Statements ..
       Write (nout,*) 'G13BCF Example Program Results'
       Write (nout,*)

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

!      Read series length and number of lags
       Read (nin,*) nxy, nl

       Allocate (x(nxy),y(nxy),rxy(nl),ryx(nl))

!      Read series
       Read (nin,*) x(1:nxy)
       Read (nin,*) y(1:nxy)

!      Call routine to calculate cross correlations between X and Y
       ifail = 0
       Call g13bcf(x,y,nxy,nl,sxy,r0xy,rxy,statxy,ifail)

!      Call routine to calculate cross correlations between Y and X
       ifail = 0
       Call g13bcf(y,x,nxy,nl,syx,r0yx,ryx,statyx,ifail)

!      Display results
       Write (nout,*) '                                Between         Between'
       Write (nout,*) '                                X and Y         Y and X'
       Write (nout,*)
       Write (nout,99999) 'Standard deviation ratio', sxy, syx
       Write (nout,*)
       Write (nout,*) 'Cross correlation at lag'
       Write (nout,99999) '                    0', r0xy, r0yx
       Write (nout,99998)(i,rxy(i),ryx(i),i=1,nl)
       Write (nout,*)
       Write (nout,99997) 'Test statistic          ', statxy, statyx

99999 Format (1X,A,F10.4,F15.4)
99998 Format (21X,I4,F10.4,F15.4)
99997 Format (1X,A,F10.4,F15.4)
      End Program g13bcfe
```

10.2 Program Data

```
G13BCF Example Program Data
    20    15
  0.02  0.05  0.08  0.03 -0.05  0.11 -0.01 -0.08 -0.08 -0.11
 -0.18 -0.19 -0.09  0.03  0.10  0.15 -0.14  0.07  0.09  0.16
  3.18  3.21  3.26  3.25  3.08  3.01  3.06  3.17  3.12  3.04
  3.26  3.45  3.33  3.70  3.31  3.81  3.33  2.96  3.28  3.10
```

10.3 Program Results

```
 G13BCF Example Program Results

                               Between         Between
                               X and Y         Y and X

 Standard deviation ratio   2.0053          0.4987

 Cross correlation at lag
                    0    0.0568          0.0568
```

```
                               1     0.0438      -0.0151
                               2    -0.3762       0.3955
                               3    -0.4864       0.3417
                               4    -0.6294       0.5486
                               5    -0.3871       0.2291
                               6    -0.1690       0.3190
                               7    -0.0678       0.1980
                               8     0.0962       0.0438
                               9     0.0788      -0.1428
                              10     0.2910      -0.1376
                              11     0.0950      -0.0387
                              12     0.0547      -0.0380
                              13     0.1855      -0.1551
                              14     0.0243      -0.1536
                              15     0.0034      -0.0696

Test statistic                      22.1269      17.2917
```