NAG Library Routine Document

G13BDF

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

G13BDF calculates preliminary estimates of the parameters of a transfer function model.

2 Specification

```
SUBROUTINE G13BDF (R0, R, NL, NNA, S, NWDS, WA, IWA, WDS, ISF, IFAIL)
INTEGER             NL, NNA(3), NWDS, IWA, ISF(2), IFAIL
REAL (KIND=nag_wp) R0, R(NL), S, WA(IWA), WDS(NWDS)
```

3 Description

G13BDF calculates estimates of parameters $\delta_1, \delta_2, \ldots, \delta_p$, $\omega_0, \omega_1, \ldots, \omega_q$ in the transfer function model

$$y_t = \delta_1 y_{t-1} + \delta_2 y_{t-2} + \cdots + \delta_p y_{t-p} + \omega_0 x_{t-b} - \omega_1 x_{t-b-1} - \cdots - \omega_q x_{t-b-q}$$

given cross-correlations between the series x_t and lagged values of y_t:

$$r_{xy}(l), \quad l = 0, 1, \ldots, L$$

and the ratio of standard deviations s_y/s_x, as supplied by G13BCF.

It is assumed that the series x_t used to calculate the cross-correlations is a sample from a time series with true autocorrelations of zero. Otherwise the cross-correlations between the series b_t and a_t, as defined in the description of G13BAF, should be used in place of those between y_t and x_t.

The estimates are obtained by solving for $\delta_1, \delta_2, \ldots, \delta_p$ the equations

$$r_{xy}(b+q+j) = \delta_1 r_{xy}(b+q+j-1) + \cdots + \delta_p r_{xy}(b+q+j-p), \quad j = 1, 2, \ldots, p$$

then calculating

$$\omega_i = \pm\left(s_y/s_x\right)\left[r_{xy}(b+i) - \delta_1 r_{xy}(b+i-1) - \cdots - \delta_p r_{xy}(b+i-p)\right], \quad i = 0, 1, \ldots, q$$

where the '+' is used for ω_0 and '−' for ω_i, $i > 0$.

Any value of $r_{xy}(l)$ arising in these equations for $l < b$ is taken as zero. The parameters $\delta_1, \delta_2, \ldots, \delta_p$ are checked as to whether they satisfy the stability criterion.

4 References

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

5 Arguments

1: R0 – REAL (KIND=nag_wp) *Input*

On entry: the cross-correlation between the two series at lag 0, $r_{xy}(0)$.

Constraint: $-1.0 \leq \text{R0} \leq 1.0$.

2: R(NL) – REAL (KIND=nag_wp) array *Input*

On entry: the cross-correlations between the two series at lags 1 to L, $r_{xy}(l)$, for $l = 1, 2, \ldots, L$.

Constraint: $-1.0 \leq \mathrm{R}(i) \leq 1.0$, for $i = 1, 2, \ldots, \mathrm{NL}$.

3: NL – INTEGER *Input*

On entry: L, the number of lagged cross-correlations in the array R.

Constraint: $\mathrm{NL} \geq \max(\mathrm{NNA}(1) + \mathrm{NNA}(2) + \mathrm{NNA}(3), 1)$.

4: NNA(3) – INTEGER array *Input*

On entry: the transfer function model orders in the standard form b, q, p (i.e., delay time, number of moving-average MA-like followed by number of autoregressive AR-like parameters).

Constraint: $\mathrm{NNA}(i) \geq 0$, for $i = 1, 2, 3$.

5: S – REAL (KIND=nag_wp) *Input*

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

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

6: NWDS – INTEGER *Input*

On entry: the exact number of parameters in the transfer function model.

Constraint: $\mathrm{NWDS} = \mathrm{NNA}(2) + \mathrm{NNA}(3) + 1$.

7: WA(IWA) – REAL (KIND=nag_wp) array *Workspace*

8: IWA – INTEGER *Input*

On entry: the dimension of the array WA as declared in the (sub)program from which G13BDF is called.

Constraint: $\mathrm{IWA} \geq \mathrm{NNA}(3) \times (\mathrm{NNA}(3) + 1)$.

9: WDS(NWDS) – REAL (KIND=nag_wp) array *Output*

On exit: the preliminary estimates of the parameters of the transfer function model in the order of $q + 1$ MA-like parameters followed by the p AR-like parameters. If the estimation of either type of parameter fails then these arguments are set to 0.0.

10: ISF(2) – INTEGER array *Output*

On exit: indicators of the success of the estimation of MA-like and AR-like parameters respectively. A value 0 indicates that there are no parameters of that type to be estimated. A value of 1 or -1 indicates that there are parameters of that type in the model and the estimation of that type has been successful or unsuccessful respectively. Note that there is always at least one MA-like parameter in the model.

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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: $\mathrm{IFAIL} = 0$ 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, NNA$(i) < 0$, for $i = 1, 2, 3$,
or NL $< \max(\text{NNA}(1) + \text{NNA}(2) + \text{NNA}(3), 1)$,
or R0 < -1.0 or R0 > 1.0,
or R$(i) < -1.0$ or R$(i) > 1.0$, for some $i = 1, 2, \ldots, \text{NL}$,
or S ≤ 0.0,
or NWDS $\neq$ NNA(2) + NNA$(3) + 1$,
or IWA $<$ NNA$(3) \times ($NNA$(3) + 1)$.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Equations used in the computations may become unstable, in which case results are reset to zero with array ISF values set accordingly.

8 Parallelism and Performance

G13BDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G13BDF 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 NNA$(3) > 0$,a local workspace array of fixed length is allocated internally by G13BDF. The total size of this array amounts to NNA(3) integer elements and NNA$(3) \times ($NNA$(3) + 1)$ real elements.

The time taken by G13BDF is roughly proportional to NWDS^3.

10 Example

This example reads the cross-correlations between two series at lags 0 to 6. It then reads a $(3, 2, 1)$ transfer function model and calculates and prints the preliminary estimates of the parameters of the model.

10.1 Program Text

```
    Program g13bdfe

!     G13BDF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g13bdf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: r0, s
      Integer                          :: ifail, iwa, nl, nwds
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: r(:), wa(:), wds(:)
      Integer                          :: isf(2), nna(3)
!     .. Executable Statements ..
      Write (nout,*) 'G13BDF Example Program Results'
      Write (nout,*)

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

!     Read in problem size and cross-correlation at lag 0
      Read (nin,*) nl, r0

      Allocate (r(nl))

!     Read in rest of cross-correlations
      Read (nin,*) r(1:nl)

!     Read in transfer function model orders
      Read (nin,*) nna(1:3)

!     Read in standard deviation ratio
      Read (nin,*) s

      nwds = nna(2) + nna(3) + 1
      iwa = nna(3)*(nna(3)+1)
      Allocate (wa(iwa),wds(nwds))

!     Calculate parameter estimates
      ifail = 0
      Call g13bdf(r0,r,nl,nna,s,nwds,wa,iwa,wds,isf,ifail)

!     Display results
      Write (nout,99999) 'Success/failure indicator', isf(1), isf(2)
      Write (nout,*)
      Write (nout,99999) 'Transfer function model B, Q, P =', nna(1:3)
      Write (nout,*)
      Write (nout,*) 'Parameter initial estimates'
      Write (nout,99998) wds(1:nwds)

99999 Format (1X,A,3I4)
99998 Format (1X,4F10.4)
    End Program g13bdfe
```

10.2 Program Data

```
G13BDF Example Program Data
     6   -0.0155                                        :: NL,R0
  0.0339 -0.0374 -0.2895 -0.3430 -0.4518 -0.2787  :: R
     3     2     1                                      :: NNA
  1.9256                                                :: S
```

10.3 Program Results

```
 G13BDF Example Program Results

 Success/failure indicator   1   1

 Transfer function model B, Q, P =   3   2   1

 Parameter initial estimates
    -0.5575    0.3166    0.4626    0.6169
```