NAG Library Routine Document

G13BBF

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

G13BBF filters a time series by a transfer function model.

2   Specification

```
SUBROUTINE G13BBF (Y, NY, MR, NMR, PAR, NPAR, CY, WA, IWA, B, NB, IFAIL)

INTEGER          NY, MR(NMR), NMR, NPAR, IWA, NB, IFAIL
REAL (KIND=nag_wp) Y(NY), PAR(NPAR), CY, WA(IWA), B(NB)
```

3   Description

From a given series $y_1, y_2, \ldots, y_n$ a new series $b_1, b_2, \ldots, b_n$ is calculated using a supplied (filtering) transfer function model according to the equation

$$b_t = \delta_1 b_{t-1} + \delta_2 b_{t-2} + \cdots + \delta_p b_{t-p} + \omega_0 y_{t-b} - \omega_1 y_{t-b-1} - \cdots - \omega_q y_{t-b-q}. \qquad (1)$$

As in the use of G13BAF, large transient errors may arise in the early values of b_t due to ignorance of y_t for $t < 0$, and two possibilities are allowed.

(i)   The equation (1) is applied from $t = 1 + b + q, \ldots, n$ so all terms in y_t on the right-hand side of (1) are known, the unknown set of values b_t for $t = b + q, \ldots, b + q + 1 - p$ being taken as zero.

(ii)   The unknown values of y_t for $t \le 0$ are estimated by backforecasting exactly as for G13BAF.

4   References

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

5   Parameters

1:     Y(NY) – REAL (KIND=nag_wp) array   *Input*

On entry: the Q'_y backforecasts starting with backforecast at time $1 - Q'_y$ to backforecast at time 0 followed by the time series starting at time 1, where $Q'_y = \text{MR}(6) + \text{MR}(9) \times \text{MR}(10)$. If there are no backforecasts either because the ARIMA model for the time series is not known or because it is known but has no moving average terms, then the time series starts at the beginning of Y.

2:     NY – INTEGER   *Input*

On entry: the total number of backforecasts and time series data points in array Y.

Constraint: $\text{NY} \ge \max(1 + Q'_y, \text{NPAR})$.

3:     MR(NMR) – INTEGER array   *Input*

On entry: the orders vector for the filtering transfer function model followed by the orders vector for the ARIMA model for the time series if the latter is known. The transfer function model orders appear in the standard form (b, q, p) as given in the G13 Chapter Introduction. Note that if the ARIMA model for the time series is supplied then the routine will assume that the first Q'_y values of the array Y are backforecasts.

Constraints:

the filtering model is restricted in the following way:

$$\mathrm{MR}(1),\ \mathrm{MR}(2),\ \mathrm{MR}(3) \geq 0.$$

the ARIMA model for the time series is restricted in the following ways:

$$\mathrm{MR}(k) \geq 0, \text{ for } k = 4, 5, \ldots, 10;$$
$$\text{if } \mathrm{MR}(10) = 0,\ \mathrm{MR}(7) + \mathrm{MR}(8) + \mathrm{MR}(9) = 0;$$
$$\text{if } \mathrm{MR}(10) \neq 0,\ \mathrm{MR}(7) + \mathrm{MR}(8) + \mathrm{MR}(9) \neq 0;$$
$$\mathrm{MR}(10) \neq 1.$$

4: NMR – INTEGER   *Input*

On entry: the number of values supplied in the array MR. It takes the value 3 if no ARIMA model for the time series is supplied but otherwise it takes the value 10. Thus NMR acts as an indicator as to whether backforecasting can be carried out.

Constraint: NMR = 3 or 10.

5: PAR(NPAR) – REAL (KIND=nag_wp) array   *Input*

On entry: the parameters of the filtering transfer function model followed by the parameters of the ARIMA model for the time series. In the transfer function model the parameters are in the standard order of MA-like followed by AR-like operator parameters. In the ARIMA model the parameters are in the standard order of non-seasonal AR and MA followed by seasonal AR and MA.

6: NPAR – INTEGER   *Input*

On entry: the total number of parameters held in array PAR.

Constraints:

$$\text{if NMR} = 3,\ \mathrm{NPAR} = \mathrm{MR}(2) + \mathrm{MR}(3) + 1;$$
$$\text{if NMR} = 10,\ \mathrm{NPAR} = \mathrm{MR}(2) + \mathrm{MR}(3) + 1 + \mathrm{MR}(4) + \mathrm{MR}(6) + \mathrm{MR}(7) + \mathrm{MR}(9).$$

7: CY – REAL (KIND=nag_wp)   *Input*

On entry: if the ARIMA model is known (i.e., NMR = 10), CY must specify the constant term of the ARIMA model for the time series. If this model is not known (i.e., NMR = 3) then CY is not used.

8: WA(IWA) – REAL (KIND=nag_wp) array   *Workspace*
9: IWA – INTEGER   *Input*

On entry: the dimension of the array WA as declared in the (sub)program from which G13BBF is called.

Constraints:

let $K = \mathrm{MR}(3) + \mathrm{MR}(4) + \mathrm{MR}(5) + (\mathrm{MR}(7) + \mathrm{MR}(8)) \times \mathrm{MR}(10)$,
then

$$\text{if NMR} = 3,\ \mathrm{IWA} \geq \mathrm{MR}(1) + \mathrm{NPAR};$$
$$\text{if NMR} = 10,\ \mathrm{IWA} \geq \mathrm{MR}(1) + \mathrm{NPAR} + K \times (K + 2).$$

10: B(NB) – REAL (KIND=nag_wp) array   *Output*

On exit: the filtered output series. If the ARIMA model for the time series was known, and hence Q'_y backforecasts were supplied in Y, then B contains Q'_y 'filtered' backforecasts followed by the filtered series. Otherwise, the filtered series begins at the start of B just as the original series began at the start of Y. In either case, if the value of the series at time t is held in $Y(t)$, then the filtered value at time t is held in $B(t)$.

11: NB – INTEGER *Input*

On entry: the dimension of the array B as declared in the (sub)program from which G13BBF is called.

In addition to holding the returned filtered series, B is also used as an intermediate work array if the ARIMA model for the time series is known.

Constraints:

> if NMR = 3, NB ≥ NY;
> if NMR = 10, NB ≥ NY + max(MR(1) + MR(2), MR(3)).

12: 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, NMR ≠ 3 and NMR ≠ 10,
> or   MR(i) < 0, for $i = 1, 2, \ldots, $ NMR,
> or   NMR = 10 and MR(10) = 1,
> or   NMR = 10 and MR(10) = 0 and MR(7) + MR(8) + MR(9) ≠ 0,
> or   NMR = 10 and MR(10) ≠ 0, and MR(7) + MR(8) + MR(9) = 0,
> or   NPAR is inconsistent with the contents of MR,
> or   WA is too small,
> or   B is too small.

IFAIL = 2

> A supplied model has parameter values which have failed the validity test.

IFAIL = 3

> The supplied time series is too short to carry out the requested filtering successfully.

IFAIL = 4

> This only occurs when an ARIMA model for the time series has been supplied. The matrix which is used to solve for the starting values for MA filtering is singular.

IFAIL = −999

> Internal memory allocation failed.

7   Accuracy

Accuracy and stability are high except when the AR-like parameters are close to the invertibility boundary. All calculations are performed in *basic precision* except for one inner product type calculation which on machines of low precision is performed in *additional precision*.

8   Further Comments

If an ARIMA model is supplied, a local workspace array of fixed length is allocated internally by G13BBF. The total size of this array amounts to K integer elements, where K is the expression defined in the description of the parameter WA.

The time taken by G13BBF is roughly proportional to the product of the length of the series and number of parameters in the filtering model with appreciable increase if an ARIMA model is supplied for the time series.

9   Example

This example reads a time series of length 296. It reads one univariate ARIMA $(1, 1, 0, 0, 1, 1, 12)$ model for the series and the $(0, 13, 12)$ filtering transfer function model. 12 initial backforecasts are required and these are calculated by a call to G13AJF . The backforecasts are inserted at the start of the series and G13BBF is called to perform the filtering.

9.1   Program Text

```
      Program g13bbfe

!     G13BBF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g13ajf, g13bbf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: cx, cy, rms
      Integer                         :: i, idd, ifail, ifv, ii, ij, ipar,   &
                                         iqxd, ist, iw, iwa, nb, nmr, npar,   &
                                         nparx, nst, nx, ny, pp, qp, sy
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: b(:), fsd(:), fva(:), par(:),        &
                                         parx(:), st(:), w(:), wa(:), x(:),   &
                                         y(:)
      Integer                         :: isf(4), mrx(7)
      Integer, Allocatable            :: mr(:)
!     .. Intrinsic Procedures ..
      Intrinsic                       :: max, min, mod
!     .. Executable Statements ..
      Write (nout,*) 'G13BBF Example Program Results'
      Write (nout,*)

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

!     Read in problem size
      Read (nin,*) nx

!     Read univariate ARIMA for series
      Read (nin,*) mrx(1:7)
      Read (nin,*) cx

!     Calculate number of backforecasts required
      iqxd = mrx(3) + mrx(6)*mrx(7)
```

```
          If (iqxd/=0) Then
            nmr = 10
          Else
            nmr = 3
          End If

!       Back forecasts will be stored in first IQXD elements
!       of Y, the series will be stored in last NX elements of
!       Y, so calculate start point for the series
        sy = iqxd + 1

!       Calculate length of series with back forecasts
        ny = nx + iqxd

        Allocate (y(ny),mr(nmr))

!       Read in series
        Read (nin,*) y(sy:ny)

!       Get back forecasts if required
        If (iqxd/=0) Then

!         Calculate number of parameters in ARIMA model
          nparx = mrx(1) + mrx(3) + mrx(4) + mrx(6)

          ist = mrx(4) + mrx(7) + mrx(2) + mrx(5) + mrx(3) + &
            max(mrx(1),mrx(6)*mrx(7))
          ifv = max(1,iqxd)
          qp = mrx(6)*mrx(7) + mrx(3)
          pp = mrx(4)*mrx(7) + mrx(1)
          iw = 6*nx + 5*nparx + qp*(qp+11) + 3*pp + 7
          Allocate (parx(nparx),x(nx),st(ist),fva(ifv),fsd(ifv),w(iw))

!         Read in initial values
          Read (nin,*) parx(1:nparx)

!         Reverse series
          x(nx:1:-1) = y(sy:ny)

!         Possible sign reversal for ARIMA constant
          idd = mrx(2) + mrx(5)
          If (mod(idd,2)/=0) Then
            cx = -cx
          End If

!         Calculate back forecasts
          ifail = 0
          Call g13ajf(mrx,parx,nparx,cx,0,x,nx,rms,st,ist,nst,iqxd,fva,fsd,ifv, &
            isf,w,iw,ifail)

!         Move back forecasts into Y, in reverse order
          y(1:iqxd) = fva(iqxd:1:-1)

!         Reverse sign for ARIMA constant back again
          If (mod(idd,2)/=0) Then
            cx = -cx
          End If
        End If

!       Read model by which to filter series
        Read (nin,*) mr(1:3)

!       Calculate NPAR
        ipar = mr(2) + mr(3) + 1
        npar = ipar + nparx

        Allocate (par(npar))

!       Read in initial parameter values
        Read (nin,*) par(1:ipar)
```

```
       If (iqxd/=0) Then
!         Move ARIMA for series into MR
          mr(4:10) = mrx(1:7)

!         Move parameters of ARIMA for Y into PAR
          par((ipar+1):(ipar+nparx)) = parx(1:nparx)
       End If

!      Move constant
       cy = cx

!      Set parameters for call to filter routine G13BBF
       If (nmr==10) Then
         iwa = mr(3) + mr(4) + mr(5) + (mr(7)+mr(8))*mr(10)
         iwa = npar + iwa*(iwa+2)
         nb = ny + max(mr(1)+mr(2),mr(3))
       Else
         iwa = mr(1) + npar
         nb = ny
       End If

       Allocate (wa(iwa),b(nb))

!      Filter series by call to G13BBF
       ifail = 0
       Call g13bbf(y,ny,mr,nmr,par,npar,cy,wa,iwa,b,nb,ifail)

!      Display results
       If (iqxd/=0) Then
         Write (nout,*) '                       Original      Filtered'
         Write (nout,*) ' Backforecasts    y-series        series'
         ij = -iqxd
         Do i = 1, iqxd
           Write (nout,99999) ij, y(i), b(i)
           ij = ij + 1
         End Do
         Write (nout,*)
       End If
       Write (nout,*) &
         '         Filtered        Filtered        Filtered        Filtered'
       Write (nout,*) &
         '          series          series          series          series'
       Do i = iqxd + 1, ny, 4
         Write (nout,99998)(ii-iqxd,b(ii),ii=i,min(ny,i+3))
       End Do

99999 Format (1X,I8,F17.1,F16.1)
99998 Format (1X,I5,F10.1,I6,F10.1,I6,F10.1,I6,F10.1)
      End Program g13bbfe
```

9.2 Program Data

```
G13BBF Example Program Data
 158                                                        :: NX
 1    1    0    0    1    1    12                            :: MRX
 0.000                                                      :: CX
5312.0 5402.0 4960.0 4717.0 4383.0 3828.0 3665.0 3718.0
3744.0 3994.0 4150.0 4064.0 4324.0 4256.0 3986.0 3670.0
3292.0 2952.0 2765.0 2813.0 2850.0 3085.0 3256.0 3213.0
3514.0 3386.0 3205.0 3124.0 2804.0 2536.0 2445.0 2649.0
2761.0 3183.0 3456.0 3529.0 4067.0 4079.0 4082.0 4029.0
3887.0 3684.0 3707.0 3923.0 4068.0 4557.0 4975.0 5197.0
6054.0 6471.0 6277.0 5529.0 5059.0 4539.0 4236.0 4305.0
4299.0 4478.0 4561.0 4470.0 4712.0 4512.0 4129.0 3942.0
3572.0 3149.0 3026.0 3141.0 3145.0 3322.0 3384.0 3373.0
3630.0 3555.0 3413.0 3127.0 2966.0 2685.0 2642.0 2789.0
2867.0 3032.0 3125.0 3176.0 3359.0 3265.0 3053.0 2915.0
2690.0 2518.0 2523.0 2737.0 3074.0 3671.0 4355.0 4648.0
5232.0 5349.0 5228.0 5172.0 4932.0 4637.0 4642.0 4930.0
5033.0 5223.0 5482.0 5560.0 5960.0 5929.0 5697.0 5583.0
```

```
5316.0 5039.0 4972.0 5169.0 5138.0 5316.0 5409.0 5375.0
5803.0 5736.0 5643.0 5416.0 5059.0 4810.0 4937.0 5166.0
5187.0 5348.0 5483.0 5626.0 6077.0 6033.0 5996.0 5860.0
5499.0 5210.0 5421.0 5609.0 5586.0 3663.0 5829.0 6005.0
6693.0 6792.0 6966.0 7227.0 7089.0 6823.0 7286.0 7621.0
7758.0 8000.0 8393.0 8592.0 9186.0 9175.0              :: End of Y
 0.620  0.820                                           :: PARX
 0   13    12                                           :: MR
 1.0131  0.0806 -0.0150 -0.0150 -0.0150 -0.0150
-0.0150 -0.0150 -0.0150 -0.0150 -0.0150 -0.0150
 0.9981 -0.0956  0.0000  0.0000  0.0000  0.0000
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000  0.8200                                         :: End of PAR
```

9.3   Program Results

G13BBF Example Program Results

```
                Original        Filtered
Backforecasts   y-series        series
   -12          5159.0          4549.2
   -11          5165.9          4550.9
   -10          4947.5          4552.8
    -9          4729.8          4554.9
    -8          4424.5          4557.4
    -7          4072.5          4560.7
    -6          3995.5          4565.0
    -5          4142.7          4571.1
    -4          4219.7          4580.0
    -3          4452.1          4593.5
    -2          4758.0          4614.3
    -1          4834.6          4647.1

       Filtered         Filtered         Filtered         Filtered
       series           series           series           series
   1   4699.2     2     4782.2     3     4552.8     4     4550.4
   5   4525.7     6     4324.8     7     4256.9     8     4169.7
   9   4127.9    10     4154.6    11     4011.3    12     3878.7
  13   3705.1    14     3619.1    15     3603.1    16     3496.1
  17   3422.6    18     3463.5    19     3349.8    20     3262.1
  21   3225.9    22     3218.1    23     3103.6    24     3023.5
  25   2905.9    26     2758.5    27     2828.2    28     2958.4
  29   2926.2    30     3019.8    31     3010.7    32     3082.8
  33   3111.7    34     3286.3    35     3279.3    36     3324.4
  37   3461.7    38     3468.3    39     3709.0    40     3839.6
  41   4004.4    42     4146.3    43     4265.3    44     4344.6
  45   4419.8    46     4647.2    47     4802.6    48     4999.5
  49   5446.0    50     5861.0    51     5855.9    52     5310.7
  53   5202.5    54     5046.6    55     4857.1    56     4812.3
  57   4740.7    58     4631.1    59     4447.5    60     4317.7
  61   4079.8    62     3833.7    63     3667.7    64     3774.8
  65   3709.9    66     3648.5    67     3645.3    68     3619.8
  69   3549.4    70     3439.2    71     3250.3    72     3209.2
  73   3005.2    74     2912.4    75     2994.1    76     2947.9
  77   3103.7    78     3168.1    79     3226.0    80     3224.1
  81   3233.0    82     3119.2    83     2992.5    84     3014.8
  85   2763.7    86     2671.3    87     2664.9    88     2778.2
  89   2823.8    90     2989.0    91     3072.2    92     3132.1
  93   3394.6    94     3717.4    95     4180.5    96     4405.9
  97   4605.2    98     4733.0    99     4830.9   100     5030.8
 101   5079.0   102     5125.0   103     5236.7   104     5392.7
 105   5396.7   106     5300.7   107     5312.1   108     5336.6
 109   5347.9   110     5331.2   111     5322.0   112     5444.8
 113   5468.7   114     5532.9   115     5555.9   116     5603.4
 117   5483.2   118     5406.8   119     5250.5   120     5171.9
 121   5217.4   122     5162.3   123     5296.1   124     5268.2
 125   5204.9   126     5290.7   127     5500.0   128     5552.3
 129   5503.3   130     5419.2   131     5335.6   132     5447.6
 133   5495.1   134     5475.1   135     5643.8   136     5713.1
 137   5655.1   138     5691.9   139     5958.4   140     5959.0
```

```
141     5884.8      142     3714.7      143     5877.8      144     5814.1
145     6095.6      146     6210.7      147     6560.5      148     7013.9
149     7174.8      150     7230.8      151     7726.7      152     7880.0
153     7997.4      154     8428.5      155     8264.1      156     8443.1
157     8615.4      158     8644.6
```

_____