NAG Library Routine Document

G13MGF

1 Purpose

G13MGF provides a moving average, moving norm, moving variance and moving standard deviation operator for an inhomogeneous time series.

2 Specification

```
SUBROUTINE G13MGF (NB, MA, T, TAU, M1, M2, SINIT, INTER, FTYPE, P, PN, WMA,     &
                   RCOMM, LRCOMM, IFAIL)
INTEGER            NB, M1, M2, INTER(2), FTYPE, PN, LRCOMM, IFAIL
REAL (KIND=nag_wp) MA(NB), T(NB), TAU, SINIT(*), P, WMA(NB), RCOMM(LRCOMM)
```

3 Description

G13MGF provides a number of operators for an inhomogeneous time series. The time series is represented by two vectors of length n; a vector of times, t; and a vector of values, z. Each element of the time series is therefore composed of the pair of scalar values (t_i, z_i), for $i = 1, 2, \ldots, n$. The time t can be measured in any arbitrary units, as long as all elements of t use the same units.

The main operator available, the moving average (MA), with parameter τ is defined as

$$\mathrm{MA}[\tau, m_1, m_2; y](t_i) = \frac{1}{m_2 - m_1 + 1} \sum_{j=m_1}^{m_2} \mathrm{EMA}[\tilde{\tau}, j; y](t_i) \tag{1}$$

where $\tilde{\tau} = \frac{2\tau}{m_2 + m_1}$, m_1 and m_2 are user-supplied integers controlling the amount of lag and smoothing respectively, with $m_2 \geq m_1$ and $\mathrm{EMA}[\cdot]$ is the iterated exponential moving average operator.

The iterated exponential moving average, $\mathrm{EMA}[\tilde{\tau}, m; y](t_i)$, is defined using the recursive formula:

$$\mathrm{EMA}[\tilde{\tau}, m; y](t_i) = \mathrm{EMA}[\tilde{\tau}; \mathrm{EMA}[\tilde{\tau}, m-1; y](t_i)](t_i)$$

with

$$\mathrm{EMA}[\tilde{\tau}, 1; y](t_i) = \mathrm{EMA}[\tilde{\tau}; y](t_i).$$

and

$$\mathrm{EMA}[\tilde{\tau}; y](t_i) = \mu \mathrm{EMA}[\tilde{\tau}; y](t_{i-1}) + (\nu - \mu) y_{i-1} + (1 - \nu) y_i$$

where

$$\mu = e^{-\alpha} \qquad \text{and} \qquad \alpha = \frac{t_i - t_{i-1}}{\tilde{\tau}}.$$

The value of ν depends on the method of interpolation chosen and the relationship between y and the input series z depends on the transformation function chosen. G13MGF gives the option of three interpolation methods:

1. Previous point: $\nu = 1$.
2. Linear: $\nu = (1 - \mu)/\alpha$.
3. Next point: $\nu = \mu$.

and three transformation functions:

1. Identity: $y_i = z_i^{[p]}$.
2. Absolute value: $y_i = |z_i|^p$.
3. Absolute difference: $y_i = |z_i - \mathrm{MA}[\tau, m_1, m_2; z](t_i)|^p$.

where the notation $[p]$ is used to denote the integer nearest to p. In addition, if either the absolute value or absolute difference transformation are used then the resulting moving average can be scaled by p^{-1}.

The various parameter options allow a number of different operators to be applied by G13MGF, a few of which are:

(i) **Moving Average (MA)**, as defined in (1) (obtained by setting FTYPE $= 1$ and P $= 1$).

(ii) **Moving Norm (MNorm)**, defined as

$$\text{MNorm}(\tau, m, p; z) = \text{MA}[\tau, 1, m; |z|^p]^{1/p}$$

(obtained by setting FTYPE $= 4$, M1 $= 1$ and M2 $= m$).

(iii) **Moving Variance (MVar)**, defined as

$$\text{MVar}(\tau, m, p; z) = \text{MA}[\tau, 1, m; |z - \text{MA}[\tau, 1, m; z]|^p]$$

(obtained by setting FTYPE $= 3$, M1 $= 1$ and M2 $= m$).

(iv) **Moving Standard Deviation (MSD)**, defined as

$$\text{MSD}(\tau, m, p; z) = \text{MA}[\tau, 1, m; |z - \text{MA}[\tau, 1, m; z]|^p]^{1/p}$$

(obtained by setting FTYPE $= 5$, M1 $= 1$ and M2 $= m$).

For large datasets or where all the data is not available at the same time, z and t can be split into arbitrary sized blocks and G13MGF called multiple times.

4 References

Dacorogna M M, Gencay R, Müller U, Olsen R B and Pictet O V (2001) *An Introduction to High-frequency Finance* Academic Press

Zumbach G O and Müller U A (2001) Operators on inhomogeneous time series *International Journal of Theoretical and Applied Finance* **4(1)** 147–178

5 Parameters

1: NB – INTEGER *Input*

On entry: b, the number of observations in the current block of data. At each call the size of the block of data supplied in MA and T can vary; therefore NB can change between calls to G13MGF.

Constraint: NB ≥ 0.

2: MA(NB) – REAL (KIND=nag_wp) array *Input/Output*

On entry: z_i, the current block of observations, for $i = k + 1, \ldots, k + b$, where k is the number of observations processed so far, i.e., the value supplied in PN on entry.

On exit: the moving average:

if FTYPE $= 4$ or 5

$$\text{MA}(i) = \{\text{MA}[\tau, m_1, m_2; y](t_i)\}^{1/p},$$

otherwise

$$\text{MA}(i) = \text{MA}[\tau, m_1, m_2; y](t_i).$$

3: T(NB) – REAL (KIND=nag_wp) array *Input*

On entry: t_i, the times for the current block of observations, for $i = k + 1, \ldots, k + b$, where k is the number of observations processed so far, i.e., the value supplied in PN on entry.

If $t_i \leq t_{i-1}$, a warning will be issued, but G13MGF will continue as if t was strictly increasing by using the absolute value. The lagged difference, $t_i - t_{i-1}$ must be sufficiently small that $e^{-\alpha}$, $\alpha = (t_i - t_{i-1})/\tilde{\tau}$ can be calculated without overflowing, for all i.

4: TAU – REAL (KIND=nag_wp) *Input*

On entry: τ, the parameter controlling the rate of decay. τ must be sufficiently large that $e^{-\alpha}$, $\alpha = (t_i - t_{i-1})/\tilde{\tau}$ can be calculated without overflowing, for all i, where $\tilde{\tau} = \frac{2\tau}{m_2+m_1}$.

Constraint: TAU > 0.0.

5: M1 – INTEGER *Input*

On entry: m_1, the iteration of the EMA operator at which the sum is started.

Constraint: M1 ≥ 1.

6: M2 – INTEGER *Input*

On entry: m_2, the iteration of the EMA operator at which the sum is ended.

Constraint: M2 $\geq$ M1.

7: SINIT($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array SINIT must be at least $2 \times \text{M2} + 3$ if FTYPE $= 3$ or 5, and at least $\text{M2} + 2$ otherwise.

On entry: if PN $= 0$, the values used to start the iterative process, with

$\text{SINIT}(1) = t_0$,

$\text{SINIT}(2) = y_0$,

$\text{SINIT}(j+2) = \text{EMA}[\tau, j; y](t_0)$, for $i = 1, 2, \ldots, \text{M2}$.

In addition, if FTYPE $= 3$ or 5 then

$\text{SINIT}(\text{M2}+3) = z_0$,

$\text{SINIT}(\text{M2}+j+2) = \text{EMA}[\tau, j; z](t_0)$, for $j = 1, 2, \ldots, \text{M2}$.

i.e., initial values based on the original data z as opposed to the transformed data y.

If PN $\neq 0$, SINIT is not referenced.

Constraint: if FTYPE $\neq 1$, $\text{SINIT}(j) \geq 0$, for $j = 2, 3, \ldots, \text{M2} + 2$.

8: INTER(2) – INTEGER array *Input*

On entry: the type of interpolation used with INTER(1) indicating the interpolation method to use when calculating $\text{EMA}[\tau, 1; z]$ and INTER(2) the interpolation method to use when calculating $\text{EMA}[\tau, j; z]$, $j > 1$.

Three types of interpolation are possible:

INTER(i) $= 1$
Previous point, with $\nu = 1$.

INTER(i) $= 2$
Linear, with $\nu = (1 - \mu)/\alpha$.

INTER(i) $= 3$
Next point, $\nu = \mu$.

Zumbach and Müller (2001) recommend that linear interpolation is used in second and subsequent iterations, i.e., INTER(2) $= 2$, irrespective of the interpolation method used at the first iteration, i.e., the value of INTER(1).

Constraint: INTER(i) $= 1$, 2 or 3, for $i = 1, 2$.

9: FTYPE – INTEGER *Input*

On entry: the function type used to define the relationship between y and z when calculating $\mathrm{EMA}[\tau, 1; y]$. Three functions are provided:

FTYPE $= 1$
: The identity function, with $y_i = z_i^{[p]}$.

FTYPE $= 2$ or 4
: The absolute value, with $y_i = |z_i|^p$.

FTYPE $= 3$ or 5
: The absolute difference, with $y_i = |z_i - \mathrm{MA}[\tau, m; y](t_i)|^p$.

If FTYPE $= 4$ or 5 then the resulting vector of averages is scaled by p^{-1} as described in MA.

Constraint: FTYPE $= 1$, 2, 3, 4 or 5.

10: P – REAL (KIND=nag_wp) *Input/Output*

On entry: p, the power used in the transformation function.

On exit: if FTYPE $= 1$, then $[p]$, the actual power used in the transformation function is returned, otherwise P is unchanged.

Constraint: P $\neq 0$.

11: PN – INTEGER *Input/Output*

On entry: k, the number of observations processed so far. On the first call to G13MGF, or when starting to summarise a new dataset, PN should be set to 0. On subsequent calls it must be the same value as returned by the last call to G13MGF.

On exit: $k + b$, the updated number of observations processed so far.

Constraint: PN ≥ 0.

12: WMA(NB) – REAL (KIND=nag_wp) array *Output*

On exit: either the moving average or exponential moving average, depending on the value of FTYPE.

if FTYPE $= 3$ or 5
: $\mathrm{WMA}(i) = \mathrm{MA}[\tau; y](t_i)$

otherwise
: $\mathrm{WMA}(i) = \mathrm{EMA}[\tilde{\tau}; y](t_i)$.

13: RCOMM(LRCOMM) – REAL (KIND=nag_wp) array *Communication Array*

On entry: communication array, used to store information between calls to G13MGF. If LRCOMM $= 0$, RCOMM is not referenced, PN must be set to 0 and all the data must be supplied in one go.

14: LRCOMM – INTEGER *Input*

On entry: the dimension of the array RCOMM as declared in the (sub)program from which G13MGF is called.

Constraint: LRCOMM $= 0$ or LRCOMM $\geq 2 \times \mathrm{M2} + 20$.

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 $= 11$

On entry, NB $= \langle value \rangle$.
Constraint: NB ≥ 0.

IFAIL $= 31$

On entry, $i = \langle value \rangle$, $\mathrm{T}(i-1) = \langle value \rangle$ and $\mathrm{T}(i) = \langle value \rangle$.
Constraint: T should be strictly increasing.

IFAIL $= 32$

On entry, $i = \langle value \rangle$, $\mathrm{T}(i-1) = \langle value \rangle$ and $\mathrm{T}(i) = \langle value \rangle$.
Constraint: $\mathrm{T}(i) \neq \mathrm{T}(i-1)$ if linear interpolation is being used.

IFAIL $= 41$

On entry, TAU $= \langle value \rangle$.
Constraint: TAU > 0.0.

IFAIL $= 42$

On entry, TAU $= \langle value \rangle$.
On entry at previous call, TAU $= \langle value \rangle$.
Constraint: if PN > 0 then TAU must be unchanged since previous call.

IFAIL $= 51$

On entry, M1 $= \langle value \rangle$.
Constraint: M1 ≥ 1.

IFAIL $= 52$

On entry, M1 $= \langle value \rangle$.
On entry at previous call, M1 $= \langle value \rangle$.
Constraint: if PN > 0 then M1 must be unchanged since previous call.

IFAIL $= 61$

On entry, M1 $= \langle value \rangle$ and M2 $= \langle value \rangle$.
Constraint: M2 $\geq$ M1.

IFAIL $= 62$

On entry, M2 $= \langle value \rangle$.
On entry at previous call, M2 $= \langle value \rangle$.
Constraint: if PN > 0 then M2 must be unchanged since previous call.

IFAIL $= 71$

On entry, $j = \langle value \rangle$ and SINIT$(j) = \langle value \rangle$.
Constraint: SINIT$(j) \geq 0.0$, for $j = 2, 3, \ldots, \mathrm{M2} + 2$.

IFAIL $= 81$

On entry, INTER$(1) = \langle value \rangle$.
Constraint: INTER$(1) = 1$, 2 or 3.

IFAIL $= 82$

On entry, INTER$(2) = \langle value \rangle$.
Constraint: INTER$(2) = 1$, 2 or 3.

IFAIL $= 83$

On entry, INTER$(1) = \langle value \rangle$ and INTER$(2) = \langle value \rangle$.
On entry at previous call, INTER$(1) = \langle value \rangle$, INTER$(2) = \langle value \rangle$.
Constraint: if PN $\neq 0$, INTER must be unchanged since the last call.

IFAIL $= 91$

On entry, FTYPE $= \langle value \rangle$.
Constraint: FTYPE $= 1$, 2, 3, 4 or 5.

IFAIL $= 92$

On entry, FTYPE $= \langle value \rangle$, On entry at previous call, FTYPE $= \langle value \rangle$.
Constraint: if PN $\neq 0$, FTYPE must be unchanged since the previous call.

IFAIL $= 101$

On entry, P $= \langle value \rangle$.
Constraint: absolute value of P must be representable as an integer.

IFAIL $= 102$

On entry, P $= \langle value \rangle$.
Constraint: if FTYPE $\neq 1$, P $\neq 0.0$. If FTYPE $= 1$, the nearest integer to P $\neq 0$.

IFAIL $= 103$

On entry, $i = \langle value \rangle$, MA$(i) = \langle value \rangle$ and P $= \langle value \rangle$.
Constraint: if FTYPE $= 1$, 2 or 4 and MA$(i) = 0$ for all i then P ≥ 0.0.

IFAIL $= 104$

On entry, $i = \langle value \rangle$, MA$(i) = \langle value \rangle$, WMA$(i) = \langle value \rangle$ and P $= \langle value \rangle$.
Constraint: if P < 0.0, MA(i) – WMA$(i) \neq 0.0$, for all i.

IFAIL $= 105$

On entry, P $= \langle value \rangle$.
On exit from previous call, P $= \langle value \rangle$.
Constraint: if PN > 0 then P must be unchanged since previous call.

IFAIL $= 111$

On entry, PN $= \langle value \rangle$.
Constraint: PN ≥ 0.

IFAIL = 112

On entry, PN = ⟨*value*⟩.
On exit from previous call, PN = ⟨*value*⟩.
Constraint: if PN > 0 then PN must be unchanged since previous call.

IFAIL = 131

RCOMM has been corrupted between calls.

IFAIL = 141

On entry, PN = 0, LRCOMM = ⟨*value*⟩ and M2 = ⟨*value*⟩.
Constraint: if PN = 0, LRCOMM = 0 or LRCOMM ≥ 2M2 + 20.

IFAIL = 142

On entry, PN ≠ 0, LRCOMM = ⟨*value*⟩ and M2 = ⟨*value*⟩.
Constraint: if PN ≠ 0, LRCOMM ≥ 2M2 + 20.

IFAIL = 301

Truncation occurred to avoid overflow, check for extreme values in T, MA or for TAU. Results are returned using the truncated values.

IFAIL = −999

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

Approximately $4m_2$ real elements are internally allocated by G13MGF. If FTYPE = 3 or 5 then a further NB real elements are also allocated.

The more data you supply to G13MGF in one call, i.e., the larger NB is, the more efficient the routine will be, particularly if the routine is being run using more than one thread.

Checks are made during the calculation of α and y_i to avoid overflow. If a potential overflow is detected the offending value is replaced with a large positive or negative value, as appropriate, and the calculations performed based on the replacement values. In such cases IFAIL = 301 is returned. This should not occur in standard usage and will only occur if extreme values of MA, T or TAU are supplied.

9 Example

The example reads in a simulated time series, (t, z) and calculates the moving average. The data is supplied in three blocks of differing sizes.

9.1 Program Text

```
    Program g13mgfe
!     G13MGFE Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g13mgf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
```

```
!     .. Local Scalars ..
      Real (Kind=nag_wp)                :: p, tau
      Integer                           :: ftype, i, ierr, ifail, lrcomm,       &
                                           lsinit, m1, m2, nb, pn
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ma(:), rcomm(:), sinit(:), t(:),     &
                                          wma(:)
      Integer                           :: inter(2)
!     .. Intrinsic Procedures ..
      Intrinsic                         :: repeat
!     .. Executable Statements ..
      Write (nout,*) 'G13MGF Example Program Results'
      Write (nout,*)

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

!     Read in the number of iterations required
      Read (nin,*) m1, m2

!     Read in the transformation function, its parameter, the interpolation
!     method to use and the decay parameter tau
      Read (nin,*) ftype, p, inter(1:2), tau

!     Read in the initial values
      If (ftype==3 .Or. ftype==5) Then
        lsinit = 2*m2 + 3
      Else
        lsinit = m2 + 2
      End If
      Allocate (sinit(lsinit))
      Read (nin,*) sinit(1:lsinit)

!     Print some titles
      Write (nout,99997) 'Time', 'MA'
      Write (nout,99998) repeat('-',32)

      lrcomm = 20 + 2*m2
      Allocate (rcomm(lrcomm))

!     Loop over each block of data
      pn = 0
      Do
!       Read in the number of observations in this block
        Read (nin,*,Iostat=ierr) nb
        If (ierr/=0) Exit

!       Allocate MA, T and WMA to the required size
        Allocate (ma(nb),t(nb),wma(nb))

!       Read in the data for this block
        Do i = 1, nb
          Read (nin,*) t(i), ma(i)
        End Do

!       Update the moving average operator for this block of data
!       G13MGF overwrites the input data
        ifail = 0
        Call g13mgf(nb,ma,t,tau,m1,m2,sinit,inter,ftype,p,pn,wma,rcomm,lrcomm, &
          ifail)

!       Display the results for this block of data
        Write (nout,99999)(pn-nb+i,t(i),ma(i),i=1,nb)
        Write (nout,*)

        Deallocate (t,ma,wma)
      End Do
```

```
99999 Format (1X,I3,4X,F10.1,4X,F10.3)
99998 Format (1X,A)
99997 Format (14X,A,10X,A)
    End Program g13mgfe
```

9.2 Program Data

```
G13MGF Example Program Data
1 2                                       :: M1,M2
1  1.0   3 2   2.0                        :: FTYPE,P,INTER(1:2),TAU
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0           :: SINIT

5                                         :: NB
 7.5  0.6
 8.2  0.6
18.1  0.8
22.8  0.1
25.8  0.2                                 :: End of T and Z for first block

10                                        :: NB
26.8  0.2
31.1  0.5
38.4  0.7
45.9  0.1
48.2  0.4
48.9  0.7
57.9  0.8
58.5  0.3
63.9  0.2
65.2  0.5                                 :: End of T and Z for second block

15                                        :: NB
66.6  0.2
67.4  0.3
69.3  0.8
69.9  0.6
73.0  0.1
75.6  0.7
77.0  0.9
84.7  0.6
86.8  0.3
88.0  0.1
88.5  0.1
91.0  0.4
93.0  1.0
93.7  1.0
94.0  0.1                                 :: End of T and Z for third block
```

9.3 Program Results

```
 G13MGF Example Program Results

              Time            MA
 --------------------------------
   1           7.5         0.545
   2           8.2         0.567
   3          18.1         0.786
   4          22.8         0.214
   5          25.8         0.187

   6          26.8         0.192
   7          31.1         0.444
   8          38.4         0.680
   9          45.9         0.155
  10          48.2         0.298
  11          48.9         0.406
  12          57.9         0.777
  13          58.5         0.677
  14          63.9         0.258
```

```
  15          65.2          0.351

  16          66.6          0.291
  17          67.4          0.289
  18          69.3          0.572
  19          69.9          0.593
  20          73.0          0.244
  21          75.6          0.532
  22          77.0          0.715
  23          84.7          0.618
  24          86.8          0.426
  25          88.0          0.284
  26          88.5          0.240
  27          91.0          0.332
  28          93.0          0.723
  29          93.7          0.814
  30          94.0          0.744
```