NAG Library Routine Document

G13MEF

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

G13MEF calculates the iterated exponential moving average for an inhomogeneous time series.

2 Specification

```
SUBROUTINE G13MEF (NB, IEMA, T, TAU, M, SINIT, INTER, PN, RCOMM, LRCOMM,    &
                   IFAIL)

INTEGER            NB, M, INTER(2), PN, LRCOMM, IFAIL
REAL (KIND=nag_wp) IEMA(NB), T(NB), TAU, SINIT(M+2), RCOMM(LRCOMM)
```

3 Description

G13MEF calculates the iterated exponential moving average 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 $i = 1, 2, \ldots, n$. Time can be measured in any arbitrary units, as long as all elements of t use the same units.

The exponential moving average (EMA), with parameter τ, is an average operator, with the exponentially decaying kernel given by

$$\frac{e^{-t_i/\tau}}{\tau}.$$

The exponential form of this kernel gives rise to the following iterative formula for the EMA operator (see Zumbach and MÏller (2001)):

$$\text{EMA}[\tau; z](t_i) = \mu \text{EMA}[\tau; z](t_{i-1}) + (\nu - \mu) z_{i-1} + (1 - \nu) z_i$$

where

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

The value of ν depends on the method of interpolation chosen. G13MEF gives the option of three interpolation methods:

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

The m-iterated exponential moving average, $\text{EMA}[\tau, m; z](t_i)$, $m > 1$, is defined using the recursive formula:

$$\text{EMA}[\tau, m; z] = \text{EMA}[\tau; \text{EMA}[\tau, m - 1; z]]$$

with

$$\text{EMA}[\tau, 1; z] = \text{EMA}[\tau; z].$$

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 G13MEF called multiple times.

4 References

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

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

5 Arguments

1: NB – INTEGER *Input*

On entry: b, the number of observations in the current block of data. The size of the block of data supplied in IEMA and T can vary; therefore NB can change between calls to G13MEF.

Constraint: $\text{NB} \geq 0$.

2: IEMA(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 iterated EMA, with $\text{IEMA}(i) = \text{EMA}[\tau, m; z](t_i)$.

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 $t_i \leq t_{i-1}$, $\text{IFAIL} = 31$ will be returned, but G13MEF will continue as if t was strictly increasing by using the absolute value.

4: TAU – REAL (KIND=nag_wp) *Input*

On entry: τ, the argument controlling the rate of decay, which must be sufficiently large that $e^{-\alpha}$, $\alpha = (t_i - t_{i-1})/\tau$ can be calculated without overflowing, for all i.

Constraint: $\text{TAU} > 0.0$.

5: M – INTEGER *Input*

On entry: m, the number of times the EMA operator is to be iterated.

Constraint: $\text{M} \geq 1$.

6: SINIT($\text{M} + 2$) – REAL (KIND=nag_wp) array *Input*

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

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

$\text{SINIT}(2) = z_0$,

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

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

7: 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:

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

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

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

Zumbach and MÏller (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 $i = 1, 2$.

8: PN – INTEGER *Input/Output*

On entry: k, the number of observations processed so far. On the first call to G13MEF, or when starting to summarise a new dataset, PN must be set to 0. On subsequent calls it must be the same value as returned by the last call to G13MEF.

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

Constraint: PN ≥ 0.

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

On entry: communication array, used to store information between calls to G13MEF. If LRCOMM = 0, RCOMM is not referenced, PN must be set to 0 and all the data must be supplied in one go.

10: LRCOMM – INTEGER *Input*

On entry: the dimension of the array RCOMM as declared in the (sub)program from which G13MEF is called.

Constraint: LRCOMM = 0 or LRCOMM $\geq$ M + 20.

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: 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 = ⟨*value*⟩.
Constraint: NB ≥ 0.

IFAIL = 31

On entry, i = ⟨*value*⟩, T($i - 1$) = ⟨*value*⟩ and T(i) = ⟨*value*⟩.
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, M $= \langle value \rangle$.
Constraint: M ≥ 1.

IFAIL $= 52$

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

IFAIL $= 71$

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

IFAIL $= 72$

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

IFAIL $= 73$

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 previous call.

IFAIL $= 81$

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

IFAIL $= 82$

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

IFAIL $= 91$

RCOMM has been corrupted between calls.

IFAIL $= 101$

On entry, PN $= 0$, LRCOMM $= \langle value \rangle$ and M $= \langle value \rangle$.
Constraint: if PN $= 0$, LRCOMM $= 0$ or LRCOMM $\geq$ M $+ 20$.

IFAIL $= 102$

On entry, PN $\neq 0$, LRCOMM $= \langle value \rangle$ and M $= \langle value \rangle$.
Constraint: if PN $\neq 0$, LRCOMM $\geq$ M $+ 20$.

IFAIL $= 301$

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

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

Not applicable.

8 Parallelism and Performance

G13MEF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G13MEF 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

Approximately $4m$ real elements are internally allocated by G13MEF.

The more data you supply to G13MEF 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 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 IEMA, T or TAU are supplied.

10 Example

The example reads in a simulated time series, (t, z) and calculates the iterated exponential moving average.

10.1 Program Text

```
    Program g13mefe
!     G13MEF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g13mef, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: tau
      Integer                          :: i, ierr, ifail, lrcomm, m, nb, pn
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: iema(:), rcomm(:), sinit(:), t(:)
      Integer                          :: inter(2)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: repeat
!     .. Executable Statements ..
      Write (nout,*) 'G13MEF Example Program Results'
      Write (nout,*)

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

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

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

!     Read in the initial values
      Allocate (sinit(m+2))
      Read (nin,*) sinit(1:m+2)

!     Print some titles
      Write (nout,99996) 'Iterated'
      Write (nout,99997) 'Time', 'EMA'
      Write (nout,99998) repeat('-',32)

      lrcomm = 20 + m
      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) Then
          Exit
        End If

!       Allocate IEMA and T to the required size
        Allocate (iema(nb),t(nb))

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

!       Update the iterated EMA for this block of data
!       G13MEF overwrites the input data with the iterated EMA
        ifail = 0
        Call g13mef(nb,iema,t,tau,m,sinit,inter,pn,rcomm,lrcomm,ifail)

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

```
        Deallocate (t,iema)
      End Do

99999 Format (1X,I3,4X,F10.1,4X,F10.3)
99998 Format (1X,A)
99997 Format (14X,A,10X,A)
99996 Format (25X,A)
    End Program g13mefe
```

10.2 Program Data

```
G13MEF Example Program Data
2                                         :: M
3 2   2.0                                 :: INTER(1:2),TAU
5.0 0.5 0.5 0.5                           :: 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 IEMA 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 IEMA 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 IEMA for third block
```

10.3 Program Results

```
 G13MEF Example Program Results

                        Iterated
              Time         EMA
 --------------------------------
   1           7.5        0.531
   2           8.2        0.544
   3          18.1        0.754
   4          22.8        0.406
   5          25.8        0.232

   6          26.8        0.217
   7          31.1        0.357
   8          38.4        0.630
   9          45.9        0.263
```

10	48.2	0.241
11	48.9	0.279
12	57.9	0.713
13	58.5	0.717
14	63.9	0.385
15	65.2	0.346
16	66.6	0.330
17	67.4	0.315
18	69.3	0.409
19	69.9	0.459
20	73.0	0.377
21	75.6	0.411
22	77.0	0.536
23	84.7	0.632
24	86.8	0.538
25	88.0	0.444
26	88.5	0.401
27	91.0	0.331
28	93.0	0.495
29	93.7	0.585
30	94.0	0.612

This example plot shows the exponential moving average for the same data using three different values of τ and illustrates the effect on the EMA of altering this argument.