NAG Toolbox

nag_tsa_inhom_iema (g13me)

1 Purpose

nag_tsa_inhom_iema (g13me) calculates the iterated exponential moving average for an inhomogeneous time series.

2 Syntax

```
[iema, pn, rcomm, ifail] = nag_tsa_inhom_iema(iema, t, tau, m, sinit, inter,
'nb', nb, 'pn', pn, 'rcomm', rcomm)
```

```
[iema, pn, rcomm, ifail] = g13me(iema, t, tau, m, sinit, inter, 'nb', nb, 'pn',
pn, 'rcomm', rcomm)
```

3 Description

nag_tsa_inhom_iema (g13me) 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)):

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

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. nag_tsa_inhom_iema (g13me) 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, $\mathrm{EMA}[\tau, m; z](t_i)$, $m > 1$, is defined using the recursive formula:

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

with

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

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 nag_tsa_inhom_iema (g13me) 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 Parameters

5.1 Compulsory Input Parameters

1: **iema**(**nb**) – REAL (KIND=nag_wp) array

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.

2: **t**(**nb**) – REAL (KIND=nag_wp) array

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 $t_i \leq t_{i-1}$, **ifail** $= 31$ will be returned, but nag_tsa_inhom_iema (g13me) will continue as if t was strictly increasing by using the absolute value.

3: **tau** – REAL (KIND=nag_wp)

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: **tau** > 0.0.

4: **m** – INTEGER

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

Constraint: **m** ≥ 1.

5: **sinit**(**m** + **2**) – REAL (KIND=nag_wp) array

If **pn** $= 0$, the values used to start the iterative process, with

sinit$(1) = t_0$,

sinit$(2) = z_0$,

sinit$(j+2) = \mathrm{EMA}[\tau, j; z](t_0)$, for $j = 1, 2, \ldots,$ **m**.

If **pn** $\neq 0$, **sinit** is not referenced.

6: **inter**(**2**) – INTEGER array

The type of interpolation used with **inter**(1) indicating the interpolation method to use when calculating $\mathrm{EMA}[\tau, 1; z]$ and **inter**(2) the interpolation method to use when calculating $\mathrm{EMA}[\tau, j; z]$, $j > 1$.

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 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 $i = 1, 2$.

5.2 Optional Input Parameters

1: **nb** – INTEGER

Default: the dimension of the arrays **iema**, **t**. (An error is raised if these dimensions are not equal.)

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 nag_tsa_inhom_iema (g13me).

Constraint: **nb** ≥ 0.

2: **pn** – INTEGER

Default: 0

k, the number of observations processed so far. On the first call to nag_tsa_inhom_iema (g13me), 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 nag_tsa_inhom_iema (g13me).

Constraint: **pn** ≥ 0.

3: **rcomm**$(lrcomm)$ – REAL (KIND=nag_wp) array

Communication array, used to store information between calls to nag_tsa_inhom_iema (g13me). On the first call to nag_tsa_inhom_iema (g13me), or if all the data is provided in one go, **rcomm** need not be provided.

5.3 Output Parameters

1: **iema**(**nb**) – REAL (KIND=nag_wp) array

The iterated EMA, with **iema**$(i) = \text{EMA}[\tau, m; z](t_i)$.

2: **pn** – INTEGER

Default: 0

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

3: **rcomm**$(lrcomm)$ – REAL (KIND=nag_wp) array

Communication array, used to store information between calls to nag_tsa_inhom_iema (g13me).

4: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 11$

Constraint: **nb** ≥ 0.

ifail $= 31$ (*warning*)

Constraint: **t** should be strictly increasing.

ifail $= 32$

Constraint: $\mathbf{t}(i) \neq \mathbf{t}(i-1)$ if linear interpolation is being used.

ifail $= 41$

Constraint: **tau** > 0.0.

ifail $= 42$

Constraint: if **pn** > 0 then **tau** must be unchanged since previous call.

ifail $= 51$

Constraint: **m** ≥ 1.

ifail $= 52$

Constraint: if **pn** > 0 then **m** must be unchanged since previous call.

ifail $= 71$

Constraint: **inter**$(1) = 1$, 2 or 3.

ifail $= 72$

Constraint: **inter**$(2) = 1$, 2 or 3.

ifail $= 73$

Constraint: if **pn** $\neq 0$, **inter** must be unchanged since the previous call.

ifail $= 81$

Constraint: **pn** ≥ 0.

ifail $= 82$

Constraint: if **pn** > 0 then **pn** must be unchanged since previous call.

ifail $= 91$

rcomm has been corrupted between calls.

ifail $= 101$

Constraint: if **pn** $= 0$, $lrcomm = 0$ or $lrcomm \geq$ **m** $+ 20$.

ifail $= 102$

Constraint: if **pn** $\neq 0$, $lrcomm \geq$ **m** $+ 20$.

ifail $= 301$ (*warning*)

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.

ifail $= -399$

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

ifail = −999

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

Approximately $4m$ real elements are internally allocated by nag_tsa_inhom_iema (g13me).

The more data you supply to nag_tsa_inhom_iema (g13me) in one call, i.e., the larger **nb** is, the more efficient the function will be, particularly if the function 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.

9 Example

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

9.1 Program Text

```
     function g13me_example

fprintf('g13me example results\n\n');

m     = nag_int(2);
inter = [nag_int(3); 2];
tau   = [0.5; 2; 8];
sinit = [5; 0.5; 0.5; 0.5];
nb    = [5, 10, 15];
t     = cell(3, 1);
iema  = cell(3, 1);
t{1}    = [ 7.5;  8.2; 18.1; 22.8; 25.8];
iema{1} = [ 0.6;  0.6;  0.8;  0.1;  0.2];
t{2}    = [26.8; 31.1; 38.4; 45.9; 48.2; 48.9; 57.9; 58.5; 63.9; 65.2];
iema{2} = [ 0.2;  0.5;  0.7;  0.1;  0.4;  0.7;  0.8;  0.3;  0.2;  0.5];
t{3}    = [66.6; 67.4; 69.3; 69.9; 73.0; 75.6; 77.0; 84.7; 86.8; 88.0; ...
           88.5; 91.0; 93.0; 93.7; 94.0];
iema{3} = [ 0.2;  0.3;  0.8;  0.6;  0.1;  0.7;  0.9;  0.6;  0.3;  0.1;  ...
            0.1;  0.4;  1.0;  1.0;  0.1];

fprintf('             Time       Iterated EMA\n');

fig1 = figure;
hold on
linecol = {'blue','green','red'};
xlabel('Time');
ylabel('Value');
title({'Simulated inhomogeneous time series and corresponding',
       'EMA(\tau,2;y) for 3 \tau values'});
tm = [t{1}; t{2}; t{3}];
jm = [iema{1}; iema{2}; iema{3}];
plot(tm,jm,'cs');

% Loop over different values of tau
for k = 1:numel(tau);
  % Loop over each block of data.
  for i = 1:numel(nb)
    if i == 1
      % process first block and create pn
```

```
      [ema, pn, rcomm, ifail] = ...
      g13me( ...
             iema{i}, t{i}, tau(k), m, sinit, inter, 'rcomm', zeros(22,1));
      jm = ema;
    else
      % Update the iterated EMA for this block of data, overwriting the input
      % data with the iterated EMA.
      [ema, pn, rcomm, ifail] = ...
      g13me( ...
             iema{i}, t{i}, tau(k), m, sinit, inter, 'pn', pn, 'rcomm', rcomm);
      jm = [jm; ema];
    end

    % Display the results for this block of data (tau = 2 only)
    if k==2
      for l=1:nb(i)
        fprintf('%3d    %10.1f    %10.3f\n', pn-nb(i)+l, t{i}(l), ema(l));
      end
      fprintf('\n');
    end
  end
  plot(tm,jm,linecol{k});
end
legend('Original data', '\tau=0.5', '\tau=2', '\tau=8', ...
       'Location', 'northwest');
legend('boxoff');
hold off
```

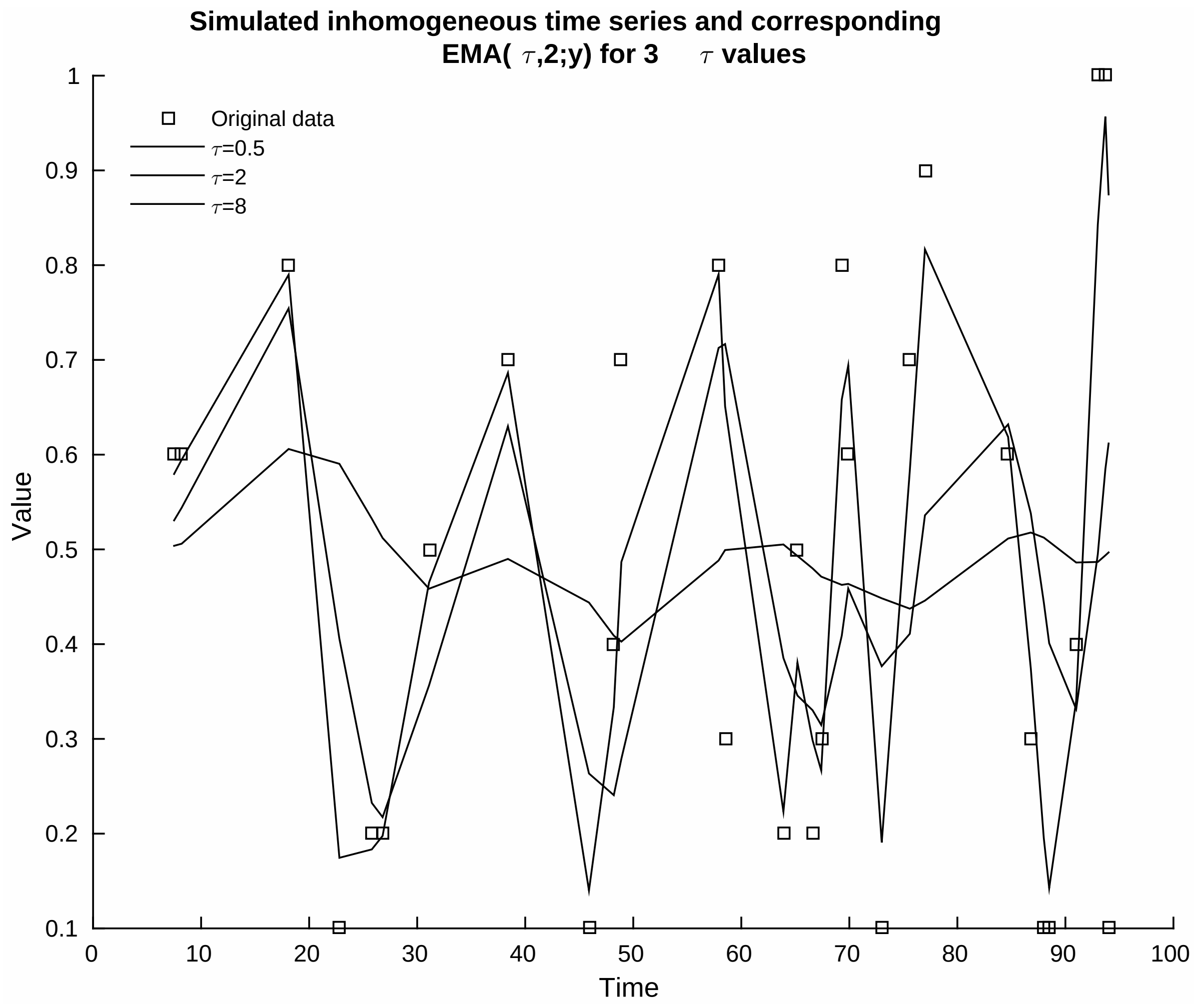
9.2 Program Results

```
     g13me example results

             Time       Iterated 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.
