NAG Library Routine Document

G13MFF

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

G13MFF calculates the iterated exponential moving average for an inhomogeneous time series, returning the intermediate results.

2 Specification

```
SUBROUTINE G13MFF (SORDER, NB, Z, IEMA, LDIEMA, T, TAU, M1, M2, SINIT,     &
                   INTER, FTYPE, P, X, PN, RCOMM, LRCOMM, IFAIL)
INTEGER            SORDER, NB, LDIEMA, M1, M2, INTER(2), FTYPE, PN,         &
                   LRCOMM, IFAIL
REAL (KIND=nag_wp) Z(NB), IEMA(LDIEMA,*), T(NB), TAU, SINIT(M2+2), P,       &
                   X(*), RCOMM(LRCOMM)
```

3 Description

G13MFF 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 (Zumbach and MÏller (2001)) for the EMA operator:

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

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 and the relationship between y and the input series z depends on the transformation function chosen. G13MFF 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: $y_i = |z_i - x_i|^p$;

where the notation $[p]$ is used to denote the integer nearest to p. In the case of the absolute difference x is a user-supplied vector of length n and therefore each element of the time series is composed of the triplet of scalar values, (t_i, z_i, x_i).

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

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

with

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

For large datasets or where all the data is not available at the same time, z, t and, where required, x can be split into arbitrary sized blocks and G13MFF 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: SORDER – INTEGER *Input*

On entry: determines the storage order of output returned in IEMA.

Constraint: SORDER $= 1$ or 2.

2: 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 Z, T and X can vary; therefore NB can change between calls to G13MFF.

Constraint: NB ≥ 0.

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

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.

Constraint: if FTYPE $= 1$ or 2 and P < 0.0, Z$(i) \neq 0$, for $i = 1, 2, \ldots,$ NB.

4: IEMA(LDIEMA, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array IEMA must be at least M2 $-$ M1 $+ 1$ if SORDER $= 1$, otherwise at least NB.

On exit: the iterated exponential moving average.

If SORDER $= 1$, IEMA$(i, j) = \mathrm{EMA}[\tau, j + \mathrm{M1} - 1; y](t_{i+k})$.

If SORDER $= 2$, IEMA$(j, i) = \mathrm{EMA}[\tau, j + \mathrm{M1} - 1; y](t_{i+k})$.

For $i = 1, 2, \ldots,$ NB, $j = 1, 2, \ldots,$ M2 $-$ M1 $+ 1$ and k is the number of observations processed so far, i.e., the value supplied in PN on entry.

5: LDIEMA – INTEGER *Input*

On entry: the first dimension of the array IEMA as declared in the (sub)program from which G13MFF is called.

Constraints:

if SORDER $= 1$, LDIEMA $\geq$ NB;
otherwise LDIEMA $\geq$ M2 $-$ M1 $+ 1$.

6: 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 $t_i \leq t_{i-1}$, IFAIL $= 61$ will be returned, but G13MFF will continue as if t was strictly increasing by using the absolute value.

7: TAU – REAL (KIND=nag_wp) *Input*

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

Constraint: TAU > 0.0.

8: M1 – INTEGER *Input*

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

Constraint: M1 ≥ 1.

9: M2 – INTEGER *Input*

On entry: the maximum number of times the EMA operator is to be iterated. Therefore G13MFF returns EMA$[\tau, m; y]$, for $m =$ M1, M1 $+ 1, \ldots,$ M2.

Constraint: M2 $\geq$ M1.

10: SINIT(M2 $+ 2$) – REAL (KIND=nag_wp) array *Input*

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

SINIT$(1) = t_0$,

SINIT$(2) = y_0$,

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

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

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

11: INTER(2) – INTEGER array *Input*

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

12: 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 $= 2$
: The absolute value, with $y_i = |z_i|^p$.

FTYPE $= 3$
: The absolute difference, with $y_i = |z_i - x_i|^p$, where the vector x is supplied in X.

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

13: 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 $\neq 0$.

14: X($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array X must be at least NB if FTYPE $= 3$.

On entry: if FTYPE $= 3$, x_i, the vector used to shift 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 FTYPE $\neq 3$ then X is not referenced.

Constraint: if FTYPE $= 3$ and P < 0, X(i) $\neq$ Z(i), for $i = 1, 2, \ldots,$ NB.

15: PN – INTEGER *Input/Output*

On entry: k, the number of observations processed so far. On the first call to G13MFF, 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 G13MFF.

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

Constraint: PN ≥ 0.

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

On entry: communication array, used to store information between calls to G13MFF. If LRCOMM $= 0$, RCOMM is not referenced, PN must be set to 0 and all the data must be supplied in one go.

17: LRCOMM – INTEGER *Input*

On entry: the dimension of the array RCOMM as declared in the (sub)program from which G13MFF is called.

Constraint: LRCOMM $= 0$ or LRCOMM $\geq$ M2 $+ 20$.

18: 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, SORDER $= \langle value \rangle$.
Constraint: SORDER $= 1$ or 2.

IFAIL $= 21$

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

IFAIL $= 51$

On entry, SORDER $= 1$, LDIEMA $= \langle value \rangle$ and NB $= \langle value \rangle$.
Constraint: LDIEMA $\geq$ NB.

On entry, SORDER $= 2$, LDIEMA $= \langle value \rangle$ and M2 $-$ M1 $+ 1 = \langle value \rangle$.
Constraint: LDIEMA $\geq$ M2 $-$ M1 $+ 1$.

IFAIL $= 61$

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

IFAIL $= 62$

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

IFAIL $= 71$

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

IFAIL $= 72$

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 $= 81$

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

IFAIL $= 82$

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 $= 91$

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

IFAIL $= 92$

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

IFAIL $= 101$

On entry, $\mathrm{FTYPE} \neq 1$, $j = \langle value \rangle$ and $\mathrm{SINIT}(j) = \langle value \rangle$.
Constraint: if $\mathrm{FTYPE} \neq 1$, $\mathrm{SINIT}(j) \geq 0.0$, for $j = 2, 3, \ldots, \mathrm{M2} + 2$.

IFAIL $= 111$

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

IFAIL $= 112$

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

IFAIL $= 113$

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

IFAIL $= 121$

On entry, $\mathrm{FTYPE} = \langle value \rangle$.
Constraint: $\mathrm{FTYPE} = 1$, 2 or 3.

IFAIL $= 122$

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

IFAIL $= 131$

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

IFAIL $= 132$

On entry, $\mathrm{P} = \langle value \rangle$.
Constraint: if $\mathrm{FTYPE} \neq 1$, $\mathrm{P} \neq 0.0$. If $\mathrm{FTYPE} = 1$, the nearest integer to P must not be 0.

IFAIL $= 133$

On entry, $i = \langle value \rangle$, $\mathrm{Z}(i) = \langle value \rangle$ and $\mathrm{P} = \langle value \rangle$.
Constraint: if $\mathrm{FTYPE} = 1$ or 2 and $\mathrm{Z}(i) = 0$ for any i then $\mathrm{P} > 0.0$.

IFAIL $= 134$

On entry, $i = \langle value \rangle$, $\mathrm{Z}(i) = \langle value \rangle$, $\mathrm{X}(i) = \langle value \rangle$ and $\mathrm{P} = \langle value \rangle$.
Constraint: if $\mathrm{FTYPE} = 3$ and $\mathrm{Z}(i) = \mathrm{X}(i)$ for any i then $\mathrm{P} > 0.0$.

IFAIL = 135

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

IFAIL = 151

On entry, PN = ⟨*value*⟩.
Constraint: PN ≥ 0.

IFAIL = 152

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

IFAIL = 161

RCOMM has been corrupted between calls.

IFAIL = 171

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

IFAIL = 172

On entry, PN $\neq 0$, LRCOMM = ⟨*value*⟩ and M2 = ⟨*value*⟩.
Constraint: if PN $\neq 0$ then LRCOMM $\geq$ M2 $+ 20$.

IFAIL = 301

Truncation occurred to avoid overflow, check for extreme values in T, Z, X 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

G13MFF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G13MFF 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 $4 \times \mathrm{M2}$ real elements are internally allocated by G13MFF.

The more data you supply to G13MFF 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 $\mathrm{IFAIL} = 301$ is returned. This should not occur in standard usage and will only occur if extreme values of Z, T, X or TAU are supplied.

10 Example

This example reads in three blocks of simulated data from an inhomogeneous time series, then calculates and prints the iterated EMA for m between 2 and 6.

10.1 Program Text

```
    Program g13mffe
!     G13MFF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g13mff, 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, ldiema,       &
                                          lrcomm, m1, m2, miema, nb, pn,       &
                                          sdiema, sorder
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: iema(:,:), rcomm(:), sinit(:), t(:), &
                                          x(:), z(:)
      Integer                          :: inter(2)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: repeat
!     .. Executable Statements ..
      Write (nout,*) 'G13MFF Example Program Results'
      Write (nout,*)

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

!     Read in the required order for the output matrix
      Read (nin,*) sorder

!     Read in the problem size
      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
      Allocate (sinit(m2+2))
      Read (nin,*) sinit(1:m2+2)

      miema = m2 - m1 + 1

!     Print some titles
      Write (nout,99997) repeat(' ',5*miema), 'Iteration'
      Write (nout,99996) 'Time', (i,i=m1,m2)
      Write (nout,99998) repeat('-',22+10*miema)

      lrcomm = m2 + 20
      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 Z and T to the required size
        Allocate (z(nb),t(nb))

!       Read in the data for this block
        If (ftype/=3) Then
          Allocate (x(0))
          Do i = 1, nb
            Read (nin,*) t(i), z(i)
          End Do
        Else
          Allocate (x(nb))
          Do i = 1, nb
            Read (nin,*) t(i), z(i), x(i)
          End Do
        End If

        If (sorder==1) Then
          ldiema = nb
          sdiema = miema
        Else
          ldiema = miema
          sdiema = nb
        End If
        Allocate (iema(ldiema,sdiema))

!       Update the iterated EMA for this block of data
        ifail = 0
        Call g13mff(sorder,nb,z,iema,ldiema,t,tau,m1,m2,sinit,inter,ftype,p,x, &
          pn,rcomm,lrcomm,ifail)

!       Display the results for this block of data
        If (sorder==1) Then
!         IEMA(NB,M2-M1+1)
          Do i = 1, nb
            Write (nout,99999) pn - nb + i, t(i), iema(i,1:miema)
          End Do
        Else
!         IEMA(NB,M2-M1+1)
          Do i = 1, nb
            Write (nout,99999) pn - nb + i, t(i), iema(1:miema,i)
          End Do
        End If
        Write (nout,*)

        Deallocate (z,t,x,iema)
      End Do
```

```
99999 Format (1X,I3,4X,F10.1,4X,20(2X,F8.3))
99998 Format (1X,A)
99997 Format (20X,A,A)
99996 Format (14X,A,10X,20(I2,8X))
    End Program g13mffe
```

10.2 Program Data

```
G13MFF Example Program Data
2                                         :: SORDER
2 6                                       :: 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
```

10.3 Program Results

```
 G13MFF Example Program Results

                                            Iteration
             Time          2          3          4          5          6
 ------------------------------------------------------------------------
   1          7.5        0.433      0.320      0.237      0.175      0.130
   2          8.2        0.479      0.361      0.268      0.198      0.147
   3         18.1        0.756      0.700      0.631      0.558      0.485
   4         22.8        0.406      0.535      0.592      0.600      0.577
   5         25.8        0.232      0.351      0.459      0.530      0.561

   6         26.8        0.217      0.301      0.406      0.491      0.540
   7         31.1        0.357      0.309      0.318      0.364      0.422
   8         38.4        0.630      0.556      0.490      0.445      0.425
   9         45.9        0.263      0.357      0.407      0.428      0.432
  10         48.2        0.241      0.284      0.343      0.388      0.413
  11         48.9        0.279      0.277      0.325      0.372      0.403
  12         57.9        0.713      0.617      0.543      0.496      0.469
```

13	58.5	0.717	0.643	0.566	0.511	0.478
14	63.9	0.385	0.495	0.541	0.546	0.531
15	65.2	0.346	0.432	0.502	0.533	0.535
16	66.6	0.330	0.384	0.453	0.504	0.526
17	67.4	0.315	0.364	0.427	0.483	0.515
18	69.3	0.409	0.367	0.389	0.435	0.478
19	69.9	0.459	0.385	0.386	0.423	0.465
20	73.0	0.377	0.403	0.394	0.398	0.419
21	75.6	0.411	0.399	0.399	0.397	0.403
22	77.0	0.536	0.440	0.410	0.401	0.401
23	84.7	0.632	0.606	0.563	0.524	0.493
24	86.8	0.538	0.587	0.583	0.557	0.526
25	88.0	0.444	0.542	0.574	0.567	0.542
26	88.5	0.401	0.515	0.564	0.567	0.548
27	91.0	0.331	0.404	0.481	0.529	0.545
28	93.0	0.495	0.418	0.438	0.483	0.518
29	93.7	0.585	0.455	0.438	0.469	0.506
30	94.0	0.612	0.475	0.441	0.465	0.500