

NAG Fortran Library Routine Document

G10CAF

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

G10CAF computes a smoothed data sequence using running median smoothers.

2 Specification

```
SUBROUTINE G10CAF(ITYPE, N, Y, SMOOTH, ROUGH, IFAIL)
INTEGER          ITYPE, N, IFAIL
real            Y(N), SMOOTH(N), ROUGH(N)
```

3 Description

Given a sequence of n observations recorded at equally spaced intervals, G10CAF fits a smooth curve through the data using one of two smoothers. The two smoothers are based on the use of running medians and averages to summarize overlapping segments. The fit and the residuals are called the smooth and the rough respectively. They obey the following:

$$\text{Data} = \text{Smooth} + \text{Rough}.$$

The two smoothers are:

1. 4253H,twice consisting of a running median of 4, then 2, then 5, then 3 followed by hanning. Hanning is a running weighted average, the weights being 1/4, 1/2 and 1/4. The result of this smoothing is then reroughed by computing residuals, applying the same smoother to them and adding the result to the smooth of the first pass.
2. 3RSSH,twice consisting of a running median of 3, two splitting operations named S to improve the smooth sequence, each of which is followed by a running median of 3, and finally hanning. The end points are dealt with using the method described by Velleman and Hoaglin (1981). The full smoother 3RSSH,twice is produced by reroughing as described above.

The compound smoother 4253H,twice is recommended. The smoother 3RSSH,twice is popular when calculating by hand as it requires simpler computations and is included for comparison purposes.

4 References

Tukey J W (1977) *Exploratory Data Analysis* Addison-Wesley

Velleman P F and Hoaglin D C (1981) *Applications, Basics, and Computing of Exploratory Data Analysis* Duxbury Press, Boston, MA

5 Parameters

1: ITYPE – INTEGER *Input*

On entry: specifies the method to be used.

If ITYPE = 0, 4253H,twice is used.

If ITYPE = 1, 3RSSH,twice is used.

Constraint: ITYPE = 0 or 1.

- 2: N – INTEGER *Input*
On entry: the number of observations, n .
Constraint: $N > 6$.
- 3: Y(N) – *real* array *Input*
On entry: the sample observations.
- 4: SMOOTH(N) – *real* array *Output*
On exit: contains the smooth.
- 5: ROUGH(N) – *real* array *Output*
On exit: contains the rough.
- 6: IFAIL – INTEGER *Input/Output*
On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.
On exit: IFAIL = 0 unless the routine detects an error (see Section 6).
 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, for users 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.**

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, ITYPE < 0,
 or ITYPE > 1.

IFAIL = 2

On entry, $N \leq 6$.

7 Accuracy

Not applicable.

8 Further Comments

Alternative methods of smoothing include the use of splines; see G10ABF and G10ACF.

9 Example

The example program reads in a sequence of 49 observations on bituminous coal production (in millions of net tons per year) in the USA., 1920–1968 and is taken from Tukey (1977). For comparison purposes, both smoothers are applied to the data and the results are printed.

9.1 Program Text

Note: the listing of the example program presented below uses *bold italicised* terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      G10CAF Example Program Text
*      Mark 16 Release. NAG Copyright 1992.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=100)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, ITYPE, N
*      .. Local Arrays ..
      real             ROUGH(NMAX), ROUGH1(NMAX), SMOOT1(NMAX),
+                     SMOOTH(NMAX), Y(NMAX)
*      .. External Subroutines ..
      EXTERNAL         G10CAF
*      .. Executable Statements ..
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.GT.0 .AND. N.LE.NMAX) THEN
        READ (NIN,*) (Y(I),I=1,N)
*
        ITYPE = 1
        IFAIL = 0
        CALL G10CAF(ITYPE,N,Y,SMOOTH,ROUGH,IFAIL)
        ITYPE = 0
        IFAIL = 0
        CALL G10CAF(ITYPE,N,Y,SMOOT1,ROUGH1,IFAIL)
*
        WRITE (NOUT,*) ' G10CAF Example Program Results'
        WRITE (NOUT,*)
        WRITE (NOUT,99999)
        WRITE (NOUT,99998)
        DO 20 I = 1, N
          WRITE (NOUT,99997) I, Y(I), SMOOTH(I), ROUGH(I), SMOOT1(I),
+            ROUGH1(I)
        20  CONTINUE
        ELSE
          WRITE (NOUT,*) ' N is out of range'
        END IF
        STOP
*
99999 FORMAT ('          Using 3RSSH,twice          Using 4',
+          '253H,twice')
99998 FORMAT (' Index      Data      Smooth      Rough      Smooth ',
+          '      Rough')
99997 FORMAT (1X,I4,F11.1,4F13.4)
      END
```

9.2 Program Data

G10CAF Example Program Data

```
49
569.0 416.0 422.0 565.0 484.0 520.0 573.0 518.0 501.0 505.0
468.0 382.0 310.0 334.0 359.0 372.0 439.0 446.0 349.0 395.0
461.0 511.0 583.0 590.0 620.0 578.0 534.0 631.0 600.0 438.0
516.0 534.0 467.0 457.0 392.0 467.0 500.0 493.0 410.0 412.0
416.0 403.0 422.0 459.0 467.0 512.0 534.0 552.0 545.0
```

9.3 Program Results

G10CAF Example Program Results

Index	Data	Using 3RSSH,twice		Using 4253H,twice	
		Smooth	Rough	Smooth	Rough
1	569.0	416.0000	153.0000	491.3750	77.6250
2	416.0	416.0000	0.0000	491.3750	-75.3750
3	422.0	431.5000	-9.5000	491.3750	-69.3750
4	565.0	473.0000	92.0000	498.8828	66.1172
5	484.0	509.5000	-25.5000	514.9375	-30.9375
6	520.0	520.6875	-0.6875	524.6602	-4.6602
7	573.0	521.5625	51.4375	525.0352	47.9648
8	518.0	518.0000	0.0000	521.1602	-3.1602
9	501.0	510.0000	-9.0000	512.5742	-11.5742
10	505.0	496.5000	8.5000	493.1680	11.8320
11	468.0	455.2500	12.7500	449.7422	18.2578
12	382.0	387.5000	-5.5000	391.6133	-9.6133
13	310.0	339.7500	-29.7500	353.4297	-43.4297
14	334.0	334.9375	-0.9375	343.8438	-9.8438
15	359.0	353.9375	5.0625	355.1602	3.8398
16	372.0	376.1250	-4.1250	382.7930	-10.7930
17	439.0	392.2500	46.7500	405.5469	33.4531
18	446.0	396.2500	49.7500	411.8633	34.1367
19	349.0	403.0000	-54.0000	411.5586	-62.5586
20	395.0	427.2500	-32.2500	420.9375	-25.9375
21	461.0	461.3750	-0.3750	456.1250	4.8750
22	511.0	513.3125	-2.3125	513.8516	-2.8516
23	583.0	567.5625	15.4375	565.2422	17.7578
24	590.0	590.0000	0.0000	589.4688	0.5312
25	620.0	593.5000	26.5000	594.7188	25.2812
26	578.0	595.2500	-17.2500	594.5625	-16.5625
27	534.0	590.9375	-56.9375	591.8125	-57.8125
28	631.0	566.8125	64.1875	583.8438	47.1562
29	600.0	531.5000	68.5000	569.0312	30.9688
30	438.0	516.0000	-78.0000	546.3438	-108.3438
31	516.0	516.0000	0.0000	517.2578	-1.2578
32	534.0	501.8750	32.1250	489.6445	44.3555
33	467.0	473.6250	-6.6250	471.2383	-4.2383
34	457.0	457.0000	0.0000	463.4844	-6.4844
35	392.0	452.0000	-60.0000	464.1875	-72.1875
36	467.0	440.1250	26.8750	468.4688	-1.4688
37	500.0	421.3750	78.6250	470.6094	29.3906
38	493.0	412.0000	81.0000	462.2617	30.7383
39	410.0	412.0000	-2.0000	438.5703	-28.5703
40	412.0	412.0000	0.0000	416.1094	-4.1094
41	416.0	411.0625	4.9375	408.8711	7.1289
42	403.0	410.6875	-7.6875	412.1836	-9.1836
43	422.0	422.0000	0.0000	424.8750	-2.8750
44	459.0	446.6250	12.3750	448.1445	10.8555
45	467.0	476.3750	-9.3750	478.7578	-11.7578
46	512.0	509.0000	3.0000	510.0234	1.9766
47	534.0	534.0000	0.0000	534.1250	-0.1250
48	552.0	545.0000	7.0000	547.0000	5.0000
49	545.0	547.7500	-2.7500	550.9375	-5.9375
