NAG 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
double precision  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: n, the number of observations.

Constraint: $\mathrm{N} > 6$.

3: Y(N) – ***double precision*** array *Input*

On entry: the sample observations.

4: SMOOTH(N) – ***double precision*** array *Output*

On exit: contains the smooth.

5: ROUGH(N) – ***double precision*** array *Output*

On exit: contains the rough.

6: 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.

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, 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.**

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, $\mathrm{N} \leq 6$.

7 Accuracy

Not applicable.

8 Further Comments

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

9 Example

This example 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

```
*     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 ..
      DOUBLE PRECISION ROUGH(NMAX), ROUGH1(NMAX), SMOOT1(NMAX),
     +                 SMOOTH(NMAX), Y(NMAX)
*     .. External Subroutines ..
*     Skip heading in data file
      EXTERNAL         G10CAF
*     .. Executable Statements ..
      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
*
99999 FORMAT ('                      Using 3RSSH,twice         Using 4',
     +       '253H,twice')
99998 FORMAT (' Index     Data      Smooth        Rough       Smooth  ',
     +       '      Rough')
99997 FORMAT (1X,I4,F11.1,4F13.3)
      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

                     Using 3RSSH,twice         Using 4253H,twice
 Index     Data      Smooth        Rough       Smooth        Rough
    1      569.0      416.000      153.000      491.375       77.625
    2      416.0      416.000        0.000      491.375      -75.375
    3      422.0      431.500       -9.500      491.375      -69.375
    4      565.0      473.000       92.000      498.883       66.117
    5      484.0      509.500      -25.500      514.938      -30.938
```

6	520.0	520.688	-0.688	524.660	-4.660
7	573.0	521.562	51.438	525.035	47.965
8	518.0	518.000	0.000	521.160	-3.160
9	501.0	510.000	-9.000	512.574	-11.574
10	505.0	496.500	8.500	493.168	11.832
11	468.0	455.250	12.750	449.742	18.258
12	382.0	387.500	-5.500	391.613	-9.613
13	310.0	339.750	-29.750	353.430	-43.430
14	334.0	334.938	-0.938	343.844	-9.844
15	359.0	353.938	5.062	355.160	3.840
16	372.0	376.125	-4.125	382.793	-10.793
17	439.0	392.250	46.750	405.547	33.453
18	446.0	396.250	49.750	411.863	34.137
19	349.0	403.000	-54.000	411.559	-62.559
20	395.0	427.250	-32.250	420.938	-25.938
21	461.0	461.375	-0.375	456.125	4.875
22	511.0	513.312	-2.312	513.852	-2.852
23	583.0	567.562	15.438	565.242	17.758
24	590.0	590.000	0.000	589.469	0.531
25	620.0	593.500	26.500	594.719	25.281
26	578.0	595.250	-17.250	594.562	-16.562
27	534.0	590.938	-56.938	591.812	-57.812
28	631.0	566.812	64.188	583.844	47.156
29	600.0	531.500	68.500	569.031	30.969
30	438.0	516.000	-78.000	546.344	-108.344
31	516.0	516.000	0.000	517.258	-1.258
32	534.0	501.875	32.125	489.645	44.355
33	467.0	473.625	-6.625	471.238	-4.238
34	457.0	457.000	0.000	463.484	-6.484
35	392.0	452.000	-60.000	464.188	-72.188
36	467.0	440.125	26.875	468.469	-1.469
37	500.0	421.375	78.625	470.609	29.391
38	493.0	412.000	81.000	462.262	30.738
39	410.0	412.000	-2.000	438.570	-28.570
40	412.0	412.000	0.000	416.109	-4.109
41	416.0	411.062	4.938	408.871	7.129
42	403.0	410.688	-7.688	412.184	-9.184
43	422.0	422.000	0.000	424.875	-2.875
44	459.0	446.625	12.375	448.145	10.855
45	467.0	476.375	-9.375	478.758	-11.758
46	512.0	509.000	3.000	510.023	1.977
47	534.0	534.000	0.000	534.125	-0.125
48	552.0	545.000	7.000	547.000	5.000
49	545.0	547.750	-2.750	550.938	-5.938