

NAG Library Routine Document

G13CBF

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

G13CBF calculates the smoothed sample spectrum of a univariate time series using spectral smoothing by the trapezium frequency (Daniell) window.

2 Specification

```
SUBROUTINE G13CBF (NX, MTX, PX, MW, PW, L, KC, LG, XG, NG, STATS, IFAIL)
INTEGER          NX, MTX, MW, L, KC, LG, NG, IFAIL
REAL (KIND=nag_wp) PX, PW, XG(KC), STATS(4)
```

3 Description

The supplied time series may be mean or trend corrected (by least squares), and tapered, the tapering factors being those of the split cosine bell:

$$\begin{aligned} \frac{1}{2}(1 - \cos(\pi(t - \frac{1}{2})/T)), & \quad 1 \leq t \leq T \\ \frac{1}{2}(1 - \cos(\pi(n - t + \frac{1}{2})/T)), & \quad n + 1 - T \leq t \leq n \\ 1, & \quad \text{otherwise,} \end{aligned}$$

where $T = \left\lceil \frac{np}{2} \right\rceil$ and p is the tapering proportion.

The unsmoothed sample spectrum

$$f^*(\omega) = \frac{1}{2\pi} \left| \sum_{t=1}^n x_t \exp(i\omega t) \right|^2$$

is then calculated for frequency values

$$\omega_k = \frac{2\pi k}{K}, \quad k = 0, 1, \dots, [K/2],$$

where $[]$ denotes the integer part.

The smoothed spectrum is returned as a subset of these frequencies for which k is a multiple of a chosen value r , i.e.,

$$\omega_{rl} = \nu_l = \frac{2\pi l}{L}, \quad l = 0, 1, \dots, [L/2],$$

where $K = r \times L$. You will normally fix L first, then choose r so that K is sufficiently large to provide an adequate representation for the unsmoothed spectrum, i.e., $K \geq 2 \times n$. It is possible to take $L = K$, i.e., $r = 1$.

The smoothing is defined by a trapezium window whose shape is supplied by the function

$$\begin{aligned} W(\alpha) &= 1, & |\alpha| \leq p \\ W(\alpha) &= \frac{1-|\alpha|}{1-p}, & p < |\alpha| \leq 1 \end{aligned}$$

the proportion p being supplied by you.

The width of the window is fixed as $2\pi/M$ by you supplying M . A set of averaging weights are constructed:

$$W_k = g \times W\left(\frac{\omega_k M}{\pi}\right), \quad 0 \leq \omega_k \leq \frac{\pi}{M},$$

where g is a normalizing constant, and the smoothed spectrum obtained is

$$\hat{f}(\nu_l) = \sum_{|\omega_k| < \frac{\pi}{M}} W_k f^*(\nu_l + \omega_k).$$

If no smoothing is required M should be set to n , in which case the values returned are $\hat{f}(\nu_l) = f^*(\nu_l)$. Otherwise, in order that the smoothing approximates well to an integration, it is essential that $K \gg M$, and preferable, but not essential, that K be a multiple of M . A choice of $L > M$ would normally be required to supply an adequate description of the smoothed spectrum. Typical choices of $L \simeq n$ and $K \simeq 4n$ should be adequate for usual smoothing situations when $M < n/5$.

The sampling distribution of $\hat{f}(\omega)$ is approximately that of a scaled χ_d^2 variate, whose degrees of freedom d is provided by the routine, together with multiplying limits mu, ml from which approximate 95% confidence intervals for the true spectrum $f(\omega)$ may be constructed as $[ml \times \hat{f}(\omega) mu \times \hat{f}(\omega)]$.

Alternatively, $\log \hat{f}(\omega)$ may be returned, with additive limits.

The bandwidth b of the corresponding smoothing window in the frequency domain is also provided. Spectrum estimates separated by (angular) frequencies much greater than b may be assumed to be independent.

4 References

Bloomfield P (1976) *Fourier Analysis of Time Series: An Introduction* Wiley

Jenkins G M and Watts D G (1968) *Spectral Analysis and its Applications* Holden-Day

5 Arguments

- 1: NX – INTEGER Input
On entry: n , the length of the time series.
Constraint: $NX \geq 1$.
- 2: MTX – INTEGER Input
On entry: whether the data are to be initially mean or trend corrected.
 $MTX = 0$
 For no correction.
 $MTX = 1$
 For mean correction.
 $MTX = 2$
 For trend correction.
Constraint: $0 \leq MTX \leq 2$.
- 3: PX – REAL (KIND=nag_wp) Input
On entry: the proportion of the data (totalled over both ends) to be initially tapered by the split cosine bell taper. (A value of 0.0 implies no tapering.)
Constraint: $0.0 \leq PX \leq 1.0$.

- 4: MW – INTEGER *Input*
On entry: the value of M which determines the frequency width of the smoothing window as $2\pi/M$. A value of n implies no smoothing is to be carried out.
Constraint: $1 \leq MW \leq NX$.
- 5: PW – REAL (KIND=nag_wp) *Input*
On entry: p , the shape parameter of the trapezium frequency window.
 A value of 0.0 gives a triangular window, and a value of 1.0 a rectangular window.
 If $MW = NX$ (i.e., no smoothing is carried out), PW is not used.
Constraint: $0.0 \leq PW \leq 1.0$.
- 6: L – INTEGER *Input*
On entry: L , the frequency division of smoothed spectral estimates as $2\pi/L$.
Constraints:
 $L \geq 1$;
 L must be a factor of KC.
- 7: KC – INTEGER *Input*
On entry: K , the order of the fast Fourier transform (FFT) used to calculate the spectral estimates.
Constraints:
 $KC \geq 2 \times NX$;
 KC must be a multiple of L .
- 8: LG – INTEGER *Input*
On entry: indicates whether unlogged or logged spectral estimates and confidence limits are required.
 $LG = 0$
 For unlogged.
 $LG \neq 0$
 For logged.
- 9: XG(KC) – REAL (KIND=nag_wp) array *Input/Output*
On entry: the n data points.
On exit: contains the NG spectral estimates $\hat{f}(\omega_i)$, for $i = 0, 1, \dots, [L/2]$, in XG(1) to XG(NG) (logged if $LG \neq 0$). The elements XG(i), for $i = NG + 1, \dots, KC$, contain 0.0.
- 10: NG – INTEGER *Output*
On exit: the number of spectral estimates, $[L/2] + 1$, in XG.
- 11: STATS(4) – REAL (KIND=nag_wp) array *Output*
On exit: four associated statistics. These are the degrees of freedom in STATS(1), the lower and upper 95% confidence limit factors in STATS(2) and STATS(3) respectively (logged if $LG \neq 0$), and the bandwidth in STATS(4).
- 12: 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, because for this routine the values of the output arguments may be useful even if $IFAIL \neq 0$ on exit, the recommended value is -1 . **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 $X04AAF$).

Note: G13CBF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

$IFAIL = 1$

On entry, $L = \langle value \rangle$.

Constraint: $L \geq 1$.

On entry, $MTX = \langle value \rangle$.

Constraint: $MTX \leq 2$.

On entry, $MTX = \langle value \rangle$.

Constraint: $MTX \geq 0$.

On entry, $MW = \langle value \rangle$.

Constraint: $MW \geq 1$.

On entry, $MW = \langle value \rangle$ and $NX = \langle value \rangle$.

Constraint: $MW \leq NX$.

On entry, $NX = \langle value \rangle$.

Constraint: $NX \geq 1$.

On entry, $PX = \langle value \rangle$, $MW = \langle value \rangle$ and $NX = \langle value \rangle$.

Constraint: if $PW < 0.0$, $MW = NX$.

On entry, $PX = \langle value \rangle$, $MW = \langle value \rangle$ and $NX = \langle value \rangle$.

Constraint: if $PW > 1.0$, $MW = NX$.

On entry, $PX = \langle value \rangle$.

Constraint: $PX \geq 0.0$.

On entry, $PX = \langle value \rangle$.

Constraint: $PX \leq 1.0$.

$IFAIL = 2$

On entry, $KC = \langle value \rangle$ and $L = \langle value \rangle$.

Constraint: KC must be a multiple of L .

On entry, $KC = \langle value \rangle$ and $NX = \langle value \rangle$.

Constraint: $KC \geq 2 \times NX$.

$IFAIL = 4$

One or more spectral estimates are negative.

Unlogged spectral estimates are returned in XG , and the degrees of freedom, unlogged confidence limit factors and bandwidth in $STATS$.

IFAIL = 5

The calculation of confidence limit factors has failed.

Spectral estimates (logged if requested) are returned in XG, and degrees of freedom and bandwidth in STATS.

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

The FFT is a numerically stable process, and any errors introduced during the computation will normally be insignificant compared with uncertainty in the data.

8 Parallelism and Performance

G13CBF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G13CBF 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

G13CBF carries out a FFT of length KC to calculate the sample spectrum. The time taken by the routine for this is approximately proportional to $KC \times \log(KC)$ (but see Section 9 in C06PAF for further details).

10 Example

This example reads a time series of length 131. It then calls G13CBF to calculate the univariate spectrum and prints the logged spectrum together with 95% confidence limits.

10.1 Program Text

```

Program g13cbfe

!      G13CBF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
!      Use nag_library, Only: g13cbf, nag_wp
!      .. Implicit None Statement ..
```

```

      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
      Real (Kind=nag_wp)          :: pw, px
      Integer                     :: i, ifail, kc, l, lg, lxx, mt, mw,    &
                                ng, nx
!      .. Local Arrays ..
      Real (Kind=nag_wp)          :: stats(4)
      Real (Kind=nag_wp), Allocatable :: xg(:)
!      .. Intrinsic Procedures ..
      Intrinsic                   :: max
!      .. Executable Statements ..
      Write (nout,*) 'G13CBF Example Program Results'
      Write (nout,*)

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

!      Read in the problem size
      Read (nin,*) nx

!      Read in smoothing parameters
      Read (nin,*) mt, px, mw, l, kc, lg
      If (mw/=nx) Then
         Read (nin,*) pw
      End If

      lxx = max(kc,nx)
      Allocate (xg(lxx))

!      Read in series
      Read (nin,*) xg(1:nx)

!      Calculate smooth spectrum
      ifail = -1
      Call g13cbf(nx,mt,px,mw,pw,l,kc,lg,xg,ng,stats,ifail)
      If (ifail/=0) Then
         If (ifail<4) Then
            Go To 100
         End If
      End If

!      Display results
      If (mw==nx) Then
         Write (nout,*) 'No smoothing'
      Else
         Write (nout,99999) 'Frequency width of smoothing window = 1/', mw
      End If
      Write (nout,99998) 'Degrees of freedom = ', stats(1),          &
         '      Bandwidth = ', stats(4)
      Write (nout,*)
      Write (nout,99997) '95 percent confidence limits - Lower = ',    &
         stats(2), ' Upper = ', stats(3)
      Write (nout,*)
      Write (nout,*)
      Write (nout,*)
      Write (nout,*)
      Write (nout,*)
      Write (nout,*)
      Write (nout,99996)(i,xg(i),i=1,ng)
      Write (nout,*)

100    Continue

99999 Format (1X,A,I0)
99998 Format (1X,A,F4.1,A,F7.4)
99997 Format (1X,A,F7.4,A,F7.4)
99996 Format (1X,I4,F10.4,I5,F10.4,I5,F10.4,I5,F10.4)
      End Program g13cbfe

```

10.2 Program Data

G13CBF Example Program Data

```

131
1 0.2 30 100 400 1
0.5
11.500  9.890  8.728  8.400  8.230  8.365  8.383  8.243
 8.080  8.244  8.490  8.867  9.469  9.786 10.100 10.714
11.320 11.900 12.390 12.095 11.800 12.400 11.833 12.200
12.242 11.687 10.883 10.138  8.952  8.443  8.231  8.067
 7.871  7.962  8.217  8.689  8.989  9.450  9.883 10.150
10.787 11.000 11.133 11.100 11.800 12.250 11.350 11.575
11.800 11.100 10.300  9.725  9.025  8.048  7.294  7.070
 6.933  7.208  7.617  7.867  8.309  8.640  9.179  9.570
10.063 10.803 11.547 11.550 11.800 12.200 12.400 12.367
12.350 12.400 12.270 12.300 11.800 10.794  9.675  8.900
 8.208  8.087  7.763  7.917  8.030  8.212  8.669  9.175
 9.683 10.290 10.400 10.850 11.700 11.900 12.500 12.500
12.800 12.950 13.050 12.800 12.800 12.800 12.600 11.917
10.805  9.240  8.777  8.683  8.649  8.547  8.625  8.750
 9.110  9.392  9.787 10.340 10.500 11.233 12.033 12.200
12.300 12.600 12.800 12.650 12.733 12.700 12.259 11.817
10.767  9.825  9.150
:: NX
:: MTX,PX,MW,L,KC,LG
:: PW
:: End of XG

```

10.3 Program Results

G13CBF Example Program Results

Frequency width of smoothing window = 1/30
Degrees of freedom = 7.0 Bandwidth = 0.1767

95 percent confidence limits - Lower = -0.8275 Upper = 1.4213

	Spectrum estimate		Spectrum estimate		Spectrum estimate		Spectrum estimate
1	-0.1776	2	-0.4561	3	-0.1784	4	1.9042
5	2.1094	6	1.7061	7	-0.7659	8	-1.4734
9	-1.5939	10	-2.1157	11	-2.9151	12	-2.7055
13	-2.8200	14	-3.4077	15	-3.8813	16	-3.6607
17	-4.0601	18	-4.4756	19	-4.2700	20	-4.3092
21	-4.5711	22	-4.8111	23	-4.5658	24	-4.7285
25	-5.4386	26	-5.5081	27	-5.2325	28	-5.0262
29	-4.4539	30	-4.4764	31	-4.9152	32	-5.8492
33	-5.5872	34	-4.9804	35	-4.8904	36	-5.2666
37	-5.7643	38	-5.8620	39	-5.5011	40	-5.7129
41	-6.3894	42	-6.4027	43	-6.1352	44	-6.5766
45	-7.3676	46	-7.1405	47	-6.1674	48	-5.8600
49	-6.1036	50	-6.2673	51	-6.4321		
