NAG Toolbox for MATLAB

g13cb

1 Purpose

g13cb calculates the smoothed sample spectrum of a univariate time series using spectral smoothing by the trapezium frequency (Daniell) window.

2 Syntax

```
[xg, ng, stats, ifail] = g13cb(nx, mtx, px, mw, pw, l, lg, xg, 'kc', kc)
```

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:

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

where $T = \left[\frac{np}{2}\right]$ 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

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

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.,

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

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

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

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:

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

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^2_d variate, whose degrees of freedom d is provided by the function, together with multiplying limits mu, ml from which approximate 95% confidence intervals for the true spectrum $f(\omega)$ may be constructed as $\left[ml \times \hat{f}(\omega) mu \times \hat{f}(\omega)\right]$. 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 Parameters

5.1 Compulsory Input Parameters

1: **nx – int32 scalar**

n, the length of the time series.

Constraint: $\mathbf{nx} \geq 1$.

2: **mtx – int32 scalar**

Whether the data are to be initially mean or trend corrected.

$\mathbf{mtx} = 0$

For no correction.

$\mathbf{mtx} = 1$

For mean correction.

$\mathbf{mtx} = 2$

For trend correction.

Constraint: $0 \leq \mathbf{mtx} \leq 2$.

3: **px – double scalar**

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: $0.0 \leq \mathbf{px} \leq 1.0$.

4: **mw – int32 scalar**

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 – double scalar**

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** ≤ 1.0.

6: **l – int32 scalar**

L, the frequency division of smoothed spectral estimates as $2\pi/L$.

Constraints:

l ≥ 1;
l must be a factor of **kc**.

7: **lg – int32 scalar**

Indicates whether unlogged or logged spectral estimates and confidence limits are required.

lg $= 0$

For unlogged.

lg $\neq 0$

For logged.

8: **xg(kc) – double array**

The n data points.

5.2 Optional Input Parameters

1: **kc – int32 scalar**

Default: The dimension of the array **xg**.

K, the order of the fast Fourier transform (FFT) used to calculate the spectral estimates. **kc** should be a multiple of small primes such as 2^m where m is the smallest integer such that $2^m \geq 2n$, provided $m \leq 20$.

Constraints:

kc $\geq 2 \times$ **nx**;
kc must be a multiple of **l**. The largest prime factor of **kc** must not exceed 19, and the total number of prime factors of **kc**, counting repetitions, must not exceed 20. These two restrictions are imposed by c06ea which performs the FFT.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **xg(kc) – double array**

Contains the **ng** spectral estimates $\hat{f}(\omega_i)$, for $i = 0, 1, \ldots, [L/2]$, in **xg**(1) to **xg**(**ng**) (logged if **lg** $\neq 0$). The elements **xg**(i), for $i =$ **ng** $+ 1, \ldots,$ **kc** contain 0.0.

2: **ng – int32 scalar**

The number of spectral estimates, $[L/2] + 1$, in **xg**.

3: **stats(4) – double array**

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).

4: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Note: g13cb may return useful information for one or more of the following detected errors or warnings.

ifail $= 1$

On entry, **nx** < 1,
or **mtx** < 0,
or **mtx** > 2,
or **px** < 0.0,
or **px** > 1.0,
or **mw** < 1,
or **mw** $>$ **nx**,
or **pw** < 0.0 and **mw** $\neq$ **nx**,
or **pw** > 1.0 and **mw** $\neq$ **nx**,
or **l** < 1.

ifail $= 2$

On entry, **kc** $< 2 \times$ **nx**,
or **kc** is not a multiple of **l**,
or **kc** has a prime factor exceeding 19,
or **kc** has more than 20 prime factors, counting repetitions.

ifail $= 3$

This indicates that a serious error has occurred. Check all array subscripts and (sub)program parameter lists in calls to g13cb. Seek expert help.

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. This error will not normally occur. Spectral estimates (logged if requested) are returned in **xg**, and degrees of freedom and bandwidth in **stats**.

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 Further Comments

g13cb carries out a FFT of length **kc** to calculate the sample spectrum. The time taken by the function for this is approximately proportional to **kc** $\times$ log(**kc**) (but see Section 8 of the document for c06ea for further details).

9 Example

```
nx = int32(131);
mtx = int32(1);
px = 0.2;
mw = int32(131);
pw = 0.5;
l = int32(100);
lg = int32(1);
xg = zeros(400, 1);
xg(1:131) = [11.5;
     9.890000000000001;
     8.728;
     8.4;
     8.23;
     8.365;
     8.382999999999999;
     8.243;
     8.08;
     8.244;
     8.49;
     8.867000000000001;
     9.468999999999999;
     9.786;
     10.1;
     10.714;
     11.32;
     11.9;
     12.39;
     12.095;
     11.8;
     12.4;
     11.833;
     12.2;
     12.242;
     11.687;
     10.883;
     10.138;
     8.952;
     8.443;
     8.231;
     8.067;
     7.871;
     7.962;
     8.217000000000001;
     8.689;
     8.989000000000001;
     9.449999999999999;
     9.882999999999999;
     10.15;
     10.787;
     11;
     11.133;
     11.1;
```

```
    11.8;
    12.25;
    11.35;
    11.575;
    11.8;
    11.1;
    10.3;
    9.725;
    9.025;
    8.048;
    7.294;
    7.07;
    6.933;
    7.208;
    7.617;
    7.867;
    8.308999999999999;
    8.640000000000001;
    9.179;
    9.57;
    10.063;
    10.803;
    11.547;
    11.55;
    11.8;
    12.2;
    12.4;
    12.367;
    12.35;
    12.4;
    12.27;
    12.3;
    11.8;
    10.794;
    9.675000000000001;
    8.9;
    8.208;
    8.087;
    7.763;
    7.917;
    8.029999999999999;
    8.212;
    8.669;
    9.175000000000001;
    9.683;
    10.29;
    10.4;
    10.85;
    11.7;
    11.9;
    12.5;
    12.5;
    12.8;
    12.95;
    13.05;
    12.8;
    12.8;
    12.8;
    12.6;
    11.917;
    10.805;
    9.24;
    8.776999999999999;
    8.683;
    8.648999999999999;
    8.547000000000001;
    8.625;
    8.75;
    9.109999999999999;
    9.391999999999999;
```

```
     9.787000000000001;
     10.34;
     10.5;
     11.233;
     12.033;
     12.2;
     12.3;
     12.6;
     12.8;
     12.65;
     12.733;
     12.7;
     12.259;
     11.817;
     10.767;
     9.824999999999999;
     9.15];
[xgOut, ng, stats, ifail] = g13cb(nx, mtx, px, mw, pw, l, lg, xg)
```

```
xgOut =
     array elided
ng =
          51
stats =
    2.0000
   -1.3053
    3.6762
    0.0480
ifail =
           0
```