NAG Toolbox

nag_tsa_multi_spectrum_daniell (g13cd)

1 Purpose

nag_tsa_multi_spectrum_daniell (g13cd) calculates the smoothed sample cross spectrum of a bivariate time series using spectral smoothing by the trapezium frequency (Daniell) window.

2 Syntax

```
[xg, yg, ng, ifail] = nag_tsa_multi_spectrum_daniell(nxy, mtxy, pxy, mw, ish, pw, l, xg, yg, 'kc', kc)
[xg, yg, ng, ifail] = g13cd(nxy, mtxy, pxy, mw, ish, pw, l, xg, yg, 'kc', kc)
```

3 Description

The supplied time series may be mean and trend corrected and tapered as in the description of nag_tsa_uni_spectrum_daniell (g13cb) before calculation of the unsmoothed sample cross-spectrum

$$f^*_{xy}(\omega) = \frac{1}{2\pi n}\left\{\sum_{t=1}^{n} y_t \exp(i\omega t)\right\} \times \left\{\sum_{t=1}^{n} x_t \exp(-i\omega t)\right\}$$

for frequency values $\omega_j = \frac{2\pi j}{K}$, $0 \leq \omega_j \leq \pi$.

A correction is made for bias due to any tapering.

As in the description of nag_tsa_uni_spectrum_daniell (g13cb) for univariate frequency window smoothing, the smoothed spectrum is returned as a subset of these frequencies,

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

where [ ] denotes the integer part.

Its real part or co-spectrum $cf(\nu_l)$, and imaginary part or quadrature spectrum $qf(\nu_l)$ are defined by

$$f_{xy}(\nu_l) = cf(\nu_l) + iqf(\nu_l) = \sum_{|\omega_k| < \frac{\pi}{M}} \tilde{w}_k f^*_{xy}(\nu_l + \omega_k)$$

where the weights $\tilde{w}_k$ are similar to the weights w_k defined for nag_tsa_uni_spectrum_daniell (g13cb), but allow for an implicit alignment shift S between the series:

$$\tilde{w}_k = w_k \exp(-2\pi i S k/L).$$

It is recommended that S is chosen as the lag k at which the cross-covariances $c_{xy}(k)$ peak, so as to minimize bias.

If no smoothing is required, the integer M, which determines the frequency window width $\dfrac{2\pi}{M}$, should be set to n.

The bandwidth of the estimates will normally have been calculated in a previous call of nag_tsa_uni_spectrum_daniell (g13cb) for estimating the univariate spectra of y_t and x_t.

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: **nxy** – INTEGER

n, the length of the time series x and y.

Constraint: **nxy** ≥ 1.

2: **mtxy** – INTEGER

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

mtxy $= 0$
For no correction.

mtxy $= 1$
For mean correction.

mtxy $= 2$
For trend correction.

Constraint: $0 \leq$ **mtxy** ≤ 2.

3: **pxy** – REAL (KIND=nag_wp)

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: $0.0 \leq$ **pxy** ≤ 1.0.

4: **mw** – INTEGER

M, the frequency width of the smoothing window as $\frac{2\pi}{M}$.

A value of n implies that no smoothing is to be carried out.

Constraint: $1 \leq$ **mw** $\leq$ **nxy**.

5: **ish** – INTEGER

S, the alignment shift between the x and y series. If x leads y, the shift is positive.

Constraint: $-$**l** $<$ **ish** $<$ **l**.

6: **pw** – REAL (KIND=nag_wp)

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** $=$ **nxy** (i.e., no smoothing is carried out) then **pw** is not used.

Constraint: if **mw** $\neq$ **nxy**, $0.0 \leq$ **pw** ≤ 1.0.

7: **l** – INTEGER

L, the frequency division of smoothed cross spectral estimates as $\frac{2\pi}{L}$.

Constraints:

l ≥ 1;
l must be a factor of **kc**.

8: **xg**(**kc**) – REAL (KIND=nag_wp) array

The **nxy** data points of the x series.

9: **yg**(**kc**) – REAL (KIND=nag_wp) array

The **nxy** data points of the y series.

5.2 Optional Input Parameters

1: **kc** – INTEGER

Default: the dimension of the arrays **xg**, **yg**. (An error is raised if these dimensions are not equal.)

The dimension of the arrays **xg** and **yg**. the order of the fast Fourier transform (FFT) used to calculate the spectral estimates. **kc** should be a product 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 $\geq 2 \times$ **nxy**;
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 the internal FFT algorithm used.

5.3 Output Parameters

1: **xg**(**kc**) – REAL (KIND=nag_wp) array

The real parts of the **ng** cross spectral estimates in elements **xg**(1) to **xg**(**ng**), and **xg**(**ng** + 1) to **xg**(**kc**) contain 0.0. The y series leads the x series.

2: **yg**(**kc**) – REAL (KIND=nag_wp) array

The imaginary parts of the **ng** cross spectral estimates in elements **yg**(1) to **yg**(**ng**), and **yg**(**ng** + 1) to **yg**(**kc**) contain 0.0. The y series leads the x series.

3: **ng** – INTEGER

The number of spectral estimates, $[L/2] + 1$, whose separate parts are held in **xg** and **yg**.

4: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **nxy** < 1,
or **mtxy** < 0,
or **mtxy** > 2,
or **pxy** < 0.0,
or **pxy** > 1.0,
or **mw** < 1,
or **mw** $>$ **nxy**,
or **pw** < 0.0 and **mw** $\neq$ **nxy**,
or **pw** > 1.0 and **mw** $\neq$ **nxy**,
or **l** < 1,
or $|$**ish**$| \geq$ **l**.

ifail $= 2$

On entry, **kc** $< 2 \times$ **nxy**,
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 in calls to nag_tsa_multi_spectrum_daniell (g13cd). Seek expert help.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

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

nag_tsa_multi_spectrum_daniell (g13cd) carries out an FFT of length **kc** to calculate the sample cross spectrum. The time taken by the function for this is approximately proportional to **kc** $\times \log($**kc**$)$ (but see function document nag_sum_fft_realherm_1d (c06pa) for further details).

9 Example

This example reads two time series of length 296. It selects mean correction and a 10% tapering proportion. It selects a $2\pi/16$ frequency width of smoothing window, a window shape parameter of 0.5 and an alignment shift of 3. It then calls nag_tsa_multi_spectrum_daniell (g13cd) to calculate the smoothed sample cross spectrum and prints the results.

9.1 Program Text

```
function g13cd_example

fprintf('g13cd example results\n\n');

% Problem size
nxy  = nag_int(296);

% Control parameters
mtxy = nag_int(1);
pxy  = 0.1;
iw   = nag_int(4);
mw   = nag_int(16);
ish  = nag_int(3);
kc   = nag_int(640);
l    = nag_int(80);
pw = 0.5;

% Series
xg = zeros(kc, 1);
xg(1:nxy) = ...
```

```
    [-0.109; 0.000; 0.178; 0.339; 0.373; 0.441; 0.461; 0.348; 0.127;-0.180;
     -0.588;-1.055;-1.421;-1.520;-1.302;-0.814;-0.475;-0.193; 0.088; 0.435;
      0.771; 0.866; 0.875; 0.891; 0.987; 1.263; 1.775; 1.976; 1.934; 1.866;
      1.832; 1.767; 1.608; 1.265; 0.790; 0.360; 0.115; 0.088; 0.331; 0.645;
      0.960; 1.409; 2.670; 2.834; 2.812; 2.483; 1.929; 1.485; 1.214; 1.239;
      1.608; 1.905; 2.023; 1.815; 0.535; 0.122; 0.009; 0.164; 0.671; 1.019;
      1.146; 1.155; 1.112; 1.121; 1.223; 1.257; 1.157; 0.913; 0.620; 0.255;
     -0.280;-1.080;-1.551;-1.799;-1.825;-1.456;-0.944;-0.570;-0.431;-0.577;
     -0.960;-1.616;-1.875;-1.891;-1.746;-1.474;-1.201;-0.927;-0.524; 0.040;
      0.788; 0.943; 0.930; 1.006; 1.137; 1.198; 1.054; 0.595;-0.080;-0.314;
     -0.288;-0.153;-0.109;-0.187;-0.255;-0.299;-0.007; 0.254; 0.330; 0.102;
     -0.423;-1.139;-2.275;-2.594;-2.716;-2.510;-1.790;-1.346;-1.081;-0.910;
     -0.876;-0.885;-0.800;-0.544;-0.416;-0.271; 0.000; 0.403; 0.841; 1.285;
      1.607; 1.746; 1.683; 1.485; 0.993; 0.648; 0.577; 0.577; 0.632; 0.747;
      0.999; 0.993; 0.968; 0.790; 0.399;-0.161;-0.553;-0.603;-0.424;-0.194;
     -0.049; 0.060; 0.161; 0.301; 0.517; 0.566; 0.560; 0.573; 0.592; 0.671;
      0.933; 1.337; 1.460; 1.353; 0.772; 0.218;-0.237;-0.714;-1.099;-1.269;
     -1.175;-0.676; 0.033; 0.556; 0.643; 0.484; 0.109;-0.310;-0.697;-1.047;
     -1.218;-1.183;-0.873;-0.336; 0.063; 0.084; 0.000; 0.001; 0.209; 0.556;
      0.782; 0.858; 0.918; 0.862; 0.416;-0.336;-0.959;-1.813;-2.378;-2.499;
     -2.473;-2.330;-2.053;-1.739;-1.261;-0.569;-0.137;-0.024;-0.050;-0.135;
     -0.276;-0.534;-0.871;-1.243;-1.439;-1.422;-1.175;-0.813;-0.634;-0.582;
     -0.625;-0.713;-0.848;-1.039;-1.346;-1.628;-1.619;-1.149;-0.488;-0.160;
     -0.007;-0.092;-0.620;-1.086;-1.525;-1.858;-2.029;-2.024;-1.961;-1.952;
     -1.794;-1.302;-1.030;-0.918;-0.798;-0.867;-1.047;-1.123;-0.876;-0.395;
      0.185; 0.662; 0.709; 0.605; 0.501; 0.603; 0.943; 1.223; 1.249; 0.824;
      0.102; 0.025; 0.382; 0.922; 1.032; 0.866; 0.527; 0.093;-0.458;-0.748;
     -0.947;-1.029;-0.928;-0.645;-0.424;-0.276;-0.158;-0.033; 0.102; 0.251;
      0.280; 0.000;-0.493;-0.759;-0.824;-0.740;-0.528;-0.204; 0.034; 0.204;
      0.253; 0.195; 0.131; 0.017;-0.182;-0.262];
yg = zeros(kc, 1);
yg(1:nxy) = ...
    [53.8; 53.6; 53.5; 53.5; 53.4; 53.1; 52.7; 52.4; 52.2; 52.0; 52.0; 52.4;
     53.0; 54.0; 54.9; 56.0; 56.8; 56.8; 56.4; 55.7; 55.0; 54.3; 53.2; 52.3;
     51.6; 51.2; 50.8; 50.5; 50.0; 49.2; 48.4; 47.9; 47.6; 47.5; 47.5; 47.6;
     48.1; 49.0; 50.0; 51.1; 51.8; 51.9; 51.7; 51.2; 50.0; 48.3; 47.0; 45.8;
     45.6; 46.0; 46.9; 47.8; 48.2; 48.3; 47.9; 47.2; 47.2; 48.1; 49.4; 50.6;
     51.5; 51.6; 51.2; 50.5; 50.1; 49.8; 49.6; 49.4; 49.3; 49.2; 49.3; 49.7;
     50.3; 51.3; 52.8; 54.4; 56.0; 56.9; 57.5; 57.3; 56.6; 56.0; 55.4; 55.4;
     56.4; 57.2; 58.0; 58.4; 58.4; 58.1; 57.7; 57.0; 56.0; 54.7; 53.2; 52.1;
     51.6; 51.0; 50.5; 50.4; 51.0; 51.8; 52.4; 53.0; 53.4; 53.6; 53.7; 53.8;
     53.8; 53.8; 53.3; 53.0; 52.9; 53.4; 54.6; 56.4; 58.0; 59.4; 60.2; 60.0;
     59.4; 58.4; 57.6; 56.9; 56.4; 56.0; 55.7; 55.3; 55.0; 54.4; 53.7; 52.8;
     51.6; 50.6; 49.4; 48.8; 48.5; 48.7; 49.2; 49.8; 50.4; 50.7; 50.9; 50.7;
     50.5; 50.4; 50.2; 50.4; 51.2; 52.3; 53.2; 53.9; 54.1; 54.0; 53.6; 53.2;
     53.0; 52.8; 52.3; 51.9; 51.6; 51.6; 51.4; 51.2; 50.7; 50.0; 49.4; 49.3;
     49.7; 50.6; 51.8; 53.0; 54.0; 55.3; 55.9; 55.9; 54.6; 53.5; 52.4; 52.1;
     52.3; 53.0; 53.8; 54.6; 55.4; 55.9; 55.9; 55.2; 54.4; 53.7; 53.6; 53.6;
     53.2; 52.5; 52.0; 51.4; 51.0; 50.9; 52.4; 53.5; 55.6; 58.0; 59.5; 60.0;
     60.4; 60.5; 60.2; 59.7; 59.0; 57.6; 56.4; 55.2; 54.5; 54.1; 54.1; 54.4;
     55.5; 56.2; 57.0; 57.3; 57.4; 57.0; 56.4; 55.9; 55.5; 55.3; 55.2; 55.4;
     56.0; 56.5; 57.1; 57.3; 56.8; 55.6; 55.0; 54.1; 54.3; 55.3; 56.4; 57.2;
     57.8; 58.3; 58.6; 58.8; 58.8; 58.6; 58.0; 57.4; 57.0; 56.4; 56.3; 56.4;
     56.4; 56.0; 55.2; 54.0; 53.0; 52.0; 51.6; 51.6; 51.1; 50.4; 50.0; 50.0;
     52.0; 54.0; 55.1; 54.5; 52.8; 51.4; 50.8; 51.2; 52.0; 52.8; 53.8; 54.5;
     54.9; 54.9; 54.8; 54.4; 53.7; 53.3; 52.8; 52.6; 52.6; 53.0; 54.3; 56.0;
     57.0; 58.0; 58.6; 58.5; 58.3; 57.8; 57.3; 57];

[xg, yg, ng, ifail] = ...
  g13cd( ...
  nxy, mtxy, pxy, mw, ish, pw, l, xg, yg);

fprintf('                     Returned sample spectrum\n\n');
fprintf('%23s%22s%22s\n', 'Real  Imaginary', 'Real  Imaginary', ...
 'Real  Imaginary');
fprintf('%21s%22s%22s\n', 'Lag    part     part', '  Lag    part     part', ...
 '  Lag    part     part');
```

```
result = [double([0:ng-1]); xg(1:ng)'; yg(1:ng)'];
for j = 1:3:ng
  fprintf('%4d%9.4f%9.4f', result(:,j:min(j+2,ng)));
  fprintf('\n');
end
```

9.2 Program Results

```
     g13cd example results

                    Returned sample spectrum

        Real  Imaginary       Real  Imaginary       Real  Imaginary
 Lag    part      part  Lag    part      part  Lag    part      part
   0  -6.1563   0.0000    1  -5.5905  -2.0119    2  -3.2711  -2.7963
   3  -1.1803  -2.3264    4  -0.2061  -1.8132    5   0.3434  -1.1357
   6   0.6200  -0.7351    7   0.5967  -0.3449    8   0.4523  -0.0984
   9   0.2391   0.0177   10   0.1129   0.0402   11   0.0564   0.0523
  12   0.0134   0.0443   13  -0.0032   0.0299   14  -0.0057   0.0148
  15  -0.0057   0.0069   16  -0.0033   0.0038   17  -0.0011   0.0012
  18  -0.0004   0.0001   19  -0.0004   0.0002   20  -0.0003   0.0001
  21  -0.0001   0.0002   22  -0.0002   0.0003   23  -0.0002   0.0002
  24  -0.0002   0.0000   25  -0.0004   0.0000   26  -0.0002  -0.0002
  27  -0.0001  -0.0000   28  -0.0001   0.0002   29  -0.0001   0.0002
  30  -0.0002   0.0003   31  -0.0002   0.0001   32  -0.0001   0.0000
  33  -0.0000  -0.0000   34   0.0000  -0.0001   35   0.0001  -0.0001
  36   0.0001  -0.0001   37   0.0001  -0.0001   38   0.0000  -0.0001
  39   0.0000  -0.0001   40   0.0001   0.0000
```