NAG Toolbox

nag_tsa_multi_spectrum_bivar (g13ce)

1 Purpose

For a bivariate time series, nag_tsa_multi_spectrum_bivar (g13ce) calculates the cross amplitude spectrum and squared coherency, together with lower and upper bounds from the univariate and bivariate (cross) spectra.

2 Syntax

```
[ca, calw, caup, t, sc, sclw, scup, ifail] = nag_tsa_multi_spectrum_bivar(xg,
yg, xyrg, xyig, stats, 'ng', ng)
```

```
[ca, calw, caup, t, sc, sclw, scup, ifail] = g13ce(xg, yg, xyrg, xyig, stats,
'ng', ng)
```

3 Description

Estimates of the cross amplitude spectrum $A(\omega)$ and squared coherency $W(\omega)$ are calculated for each frequency ω as

$$A(\omega) = \left|f_{xy}(\omega)\right| = \sqrt{cf(\omega)^2 + qf(\omega)^2} \quad \text{and}$$

$$W(\omega) = \frac{\left|f_{xy}(\omega)\right|^2}{f_{xx}(\omega)f_{yy}(\omega)},$$

where

> $cf(\omega)$ and $qf(\omega)$ are the co-spectrum and quadrature spectrum estimates between the series, i.e., the real and imaginary parts of the cross spectrum $f_{xy}(\omega)$ as obtained using nag_tsa_multi_spectrum_lag (g13cc) or nag_tsa_multi_spectrum_daniell (g13cd);
>
> $f_{xx}(\omega)$ and $f_{yy}(\omega)$ are the univariate spectrum estimates for the two series as obtained using nag_tsa_uni_spectrum_lag (g13ca) or nag_tsa_uni_spectrum_daniell (g13cb).

The same type and amount of smoothing should be used for these estimates, and this is specified by the degrees of freedom and bandwidth values which are passed from the calls of nag_tsa_uni_spectrum_lag (g13ca) or nag_tsa_uni_spectrum_daniell (g13cb).

Upper and lower 95% confidence limits for the cross amplitude are given approximately by

$$A(\omega)\left[1 \pm \left(1.96/\sqrt{d}\right)\sqrt{W(\omega)^{-1} + 1}\right],$$

except that a negative lower limit is reset to 0.0, in which case the approximation is rather poor. You are therefore particularly recommended to compare the coherency estimate $W(\omega)$ with the critical value T derived from the upper 5% point of the F-distribution on $(2, d-2)$ degrees of freedom:

$$T = \frac{2F}{d - 2 + 2F},$$

where d is the degrees of freedom associated with the univariate spectrum estimates. The value of T is returned by the function.

The hypothesis that the series are unrelated at frequency ω, i.e., that both the true cross amplitude and coherency are zero, may be rejected at the 5% level if $W(\omega) > T$. Tests at two frequencies separated by more than the bandwidth may be taken to be independent.

The confidence limits on $A(\omega)$ are strictly appropriate only at frequencies for which the coherency is significant. The same applies to the confidence limits on $W(\omega)$ which are however calculated at all frequencies using the approximation that $\operatorname{arctanh}\left(\sqrt{W(l)}\right)$ is Normal with variance $1/d$.

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: **xg**(**ng**) – REAL (KIND=nag_wp) array

The **ng** univariate spectral estimates, $f_{xx}(\omega)$, for the x series.

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

The **ng** univariate spectral estimates, $f_{yy}(\omega)$, for the y series.

3: **xyrg**(**ng**) – REAL (KIND=nag_wp) array

The real parts, $cf(\omega)$, of the **ng** bivariate spectral estimates for the x and y series. The x series leads the y series.

4: **xyig**(**ng**) – REAL (KIND=nag_wp) array

The imaginary parts, $qf(\omega)$, of the **ng** bivariate spectral estimates for the x and y series. The x series leads the y series.

Note: the two univariate and the bivariate spectra must each have been calculated using the same method of smoothing. For rectangular, Bartlett, Tukey or Parzen smoothing windows, the same cut-off point of lag window and the same frequency division of the spectral estimates must be used. For the trapezium frequency smoothing window, the frequency width and the shape of the window and the frequency division of the spectral estimates must be the same. The spectral estimates and statistics must also be unlogged.

5: **stats**(**4**) – REAL (KIND=nag_wp) array

The four associated statistics for the univariate spectral estimates for the x and y series. **stats**(1) contains the degrees of freedom, **stats**(2) and **stats**(3) contain the lower and upper bound multiplying factors respectively and **stats**(4) contains the bandwidth.

Constraints:

stats(1) ≥ 3.0;
$0.0 <$ **stats**(2) ≤ 1.0;
stats(3) ≥ 1.0.

5.2 Optional Input Parameters

1: **ng** – INTEGER

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

The number of spectral estimates in each of the arrays **xg**, **yg**, **xyrg** and **xyig**. It is also the number of cross amplitude spectral and squared coherency estimates.

Constraint: **ng** ≥ 1.

5.3 Output Parameters

1: **ca**(**ng**) – REAL (KIND=nag_wp) array

The **ng** cross amplitude spectral estimates $\hat{A}(\omega)$ at each frequency of ω.

2: **calw**(**ng**) – REAL (KIND=nag_wp) array

The **ng** lower bounds for the **ng** cross amplitude spectral estimates.

3: **caup**(**ng**) – REAL (KIND=nag_wp) array

The **ng** upper bounds for the **ng** cross amplitude spectral estimates.

4: **t** – REAL (KIND=nag_wp)

The critical value for the significance of the squared coherency, T.

5: **sc**(**ng**) – REAL (KIND=nag_wp) array

The **ng** squared coherency estimates, $\hat{W}(\omega)$ at each frequency ω.

6: **sclw**(**ng**) – REAL (KIND=nag_wp) array

The **ng** lower bounds for the **ng** squared coherency estimates.

7: **scup**(**ng**) – REAL (KIND=nag_wp) array

The **ng** upper bounds for the **ng** squared coherency estimates.

8: **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, **ng** < 1,
or **stats**$(1) < 3.0$,
or **stats**$(2) \leq 0.0$,
or **stats**$(2) > 1.0$,
or **stats**$(3) < 1.0$.

ifail $= 2$ (*warning*)

A bivariate spectral estimate is zero. For this frequency the cross amplitude spectrum and squared coherency and their bounds are set to zero.

ifail $= 3$ (*warning*)

A univariate spectral estimate is negative. For this frequency the cross amplitude spectrum and squared coherency and their bounds are set to zero.

ifail $= 4$ (*warning*)

A univariate spectral estimate is zero. For this frequency the cross amplitude spectrum and squared coherency and their bounds are set to zero.

ifail $= 5$ (*warning*)

A calculated value of the squared coherency exceeds 1.0. For this frequency the squared coherency is reset to 1.0 and this value for the squared coherency is used in the formulae for the calculation of bounds for both the cross amplitude spectrum and squared coherency. This has the consequence that both squared coherency bounds are 1.0.

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.

If more than one failure of the types 2, 3, 4 and 5 occurs then the failure type which occurred at lowest frequency is returned in **ifail**. However the actions indicated above are also carried out for failures at higher frequencies.

7 Accuracy

All computations are very stable and yield good accuracy.

8 Further Comments

The time taken by nag_tsa_multi_spectrum_bivar (g13ce) is approximately proportional to **ng**.

9 Example

This example reads the set of univariate spectrum statistics, the two univariate spectra and the cross spectrum at a frequency division of $\frac{2\pi}{20}$ for a pair of time series. It calls nag_tsa_multi_spectrum_bivar (g13ce) to calculate the cross amplitude spectrum and squared coherency and their bounds and prints the results.

9.1 Program Text

```
     function g13ce_example

fprintf('g13ce example results\n\n');

% Data
xg = [ 2.03490;     0.51554;     0.07640;
       0.01068;     0.00093;     0.00100;
       0.00076;     0.00037;     0.00021];
yg = [21.97712;     3.29761;     0.28782;
       0.02480;     0.00285;     0.00203;
       0.00125;     0.00107;     0.00191];
xyrg = ...
     [-6.54995;     0.34107;     0.12335;
      -0.00514;    -0.00033;    -0.00039;
      -0.00026;     0.00011;     0.00007];
xyig = ...
     [ 0.00000;    -1.19030;     0.04087;
       0.00842;     0.00032;    -0.00001;
       0.00018;    -0.00016;     0.00000];
ng = numel(xg);

% Statistics
stats = [30.00000;     0.63858;     1.78670;     0.33288];

% Calculate cross-amplitude spectrum
```

```
[ca, calw, caup, t, sc, sclw, scup, ifail] = ...
  g13ce( ...
  xg, yg, xyrg, xyig, stats);

% Display results
fprintf('      Cross amplitude spectrum\n\n');
fprintf('                          Lower     Upper\n');
fprintf('          Value     bound     bound\n');
for j=1:ng
  fprintf('%5d%10.4f%10.4f%10.4f\n', j-1, ca(j), calw(j), caup(j))
end
fprintf('\nSquared coherency test statistic = %12.4f\n\n', t);
fprintf('         Squared coherency\n\n');
fprintf('                          Lower     Upper\n');
fprintf('          Value     bound     bound\n');
for j=1:ng
  fprintf('%5d%10.4f%10.4f%10.4f\n', j-1, sc(j), sclw(j), scup(j))
end
```

9.2 Program Results

```
     g13ce example results

      Cross amplitude spectrum

                        Lower     Upper
              Value     bound     bound
    0    6.5499    3.9277   10.9228
    1    1.2382    0.7364    2.0820
    2    0.1299    0.0755    0.2236
    3    0.0099    0.0049    0.0197
    4    0.0005    0.0001    0.0017
    5    0.0004    0.0001    0.0015
    6    0.0003    0.0001    0.0010
    7    0.0002    0.0001    0.0007
    8    0.0001    0.0000    0.0018

Squared coherency test statistic =       0.1926

         Squared coherency

                        Lower     Upper
              Value     bound     bound
    0    0.9593    0.9185    0.9799
    1    0.9018    0.8093    0.9507
    2    0.7679    0.5811    0.8790
    3    0.3674    0.1102    0.6177
    4    0.0797    0.0000    0.3253
    5    0.0750    0.0000    0.3182
    6    0.1053    0.0000    0.3610
    7    0.0952    0.0000    0.3475
    8    0.0122    0.0000    0.1912
```