NAG Library Function Document

nag_tsa_spectrum_bivar_cov (g13ccc)

1 Purpose

nag_tsa_spectrum_bivar_cov (g13ccc) calculates the smoothed sample cross spectrum of a bivariate time series using one of four lag windows: rectangular, Bartlett, Tukey or Parzen.

2 Specification

```
#include <nag.h>
#include <nagg13.h>

void nag_tsa_spectrum_bivar_cov (Integer nxy,
                                NagMeanOrTrend mtxy_correction, double pxy, Integer iw, Integer mw,
                                Integer ish, Integer ic, Integer nc, double cxy[], double cyx[],
                                Integer kc, Integer l, double xg[], double yg[], Complex g[],
                                Integer *ng, NagError *fail)
```

3 Description

The smoothed sample cross spectrum is a complex valued function of frequency \( \omega \),

\[ f_{xy}(\omega) = cf(\omega) + iqf(\omega), \]

defined by its real part or co-spectrum

\[ cf(\omega) = \frac{1}{2\pi} \sum_{k=-M+1}^{M-1} w_k C_{xy}(k + S) \cos(\omega k) \]

and imaginary part or quadrature spectrum

\[ qf(\omega) = \frac{1}{2\pi} \sum_{k=-M+1}^{M-1} w_k C_{xy}(k + S) \sin(\omega k) \]

where \( w_k = w_{-k} \), for \( k = 0, 1, \ldots, M - 1 \), is the smoothing lag window as defined in the description of nag_tsa_spectrum_univar_cov (g13cac). The alignment shift \( S \) is recommended to be chosen as the lag \( k \) at which the cross-covariances \( c_{xy}(k) \) peak, so as to minimize bias.

The results are calculated for frequency values

\[ \omega_j = \frac{2\pi j}{L}, \quad j = 0, 1, \ldots, [L/2], \]

where \([\ ]\) denotes the integer part.

The cross-covariances \( c_{xy}(k) \) may be supplied by you, or constructed from supplied series \( x_1, x_2, \ldots, x_n; \)
\( y_1, y_2, \ldots, y_n \) as

\[ c_{xy}(k) = \sum_{i=1}^{\min(n-k, n)} \frac{x_i y_{i+k}}{n}, \quad k \geq 0 \]

\[ c_{xy}(k) = \sum_{i=1}^{\min(n-k, n)} \frac{x_i y_{i-k}}{n} = c_{yx}(-k), \quad k < 0 \]

this convolution being carried out using the finite Fourier transform.
The supplied series may be mean and trend corrected and tapered before calculation of the cross-covariances, in exactly the manner described in nag_tsa_spectrum_univar_cov (g13cac) for univariate spectrum estimation. The results are corrected for any bias due to tapering.

The bandwidth associated with the estimates is not returned. It will normally already have been calculated in previous calls of nag_tsa_spectrum_univar_cov (g13cac) for estimating the univariate spectra of \( y_t \) and \( x_t \).

4 References


5 Arguments

1:  
   **nxy** – Integer  
   
   *Input*
   
   *On entry:* \( n \), the length of the time series \( x \) and \( y \).
   
   *Constraint:* \( nxy \geq 1 \).

2:  
   **mtxy_correction** – NagMeanOrTrend  
   
   *Input*
   
   *On entry:* if cross-covariances are to be calculated by the function \( (ic = 0) \), \**mtxy_correction** must specify whether the data is to be initially mean or trend corrected.
   
   \**mtxy_correction** = Nag_NoCorrection
   
   For no correction.
   
   \**mtxy_correction** = Nag_Mean
   
   For mean correction.
   
   \**mtxy_correction** = Nag_Trend
   
   For trend correction.
   
   If cross-covariances are supplied \( (ic \neq 0) \), \**mtxy_correction** should be set to \**mtxy_correction** = Nag_NoCorrection
   
   *Constraint:* if \( ic = 0 \), \**mtxy_correction** = Nag_NoCorrection, Nag_Mean or Nag_Trend.

3:  
   **pxy** – double  
   
   *Input*
   
   *On entry:* if cross-covariances are to be calculated by the function \( (ic = 0) \), \**pxy** must specify 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.
   
   If cross-covariances are supplied \( (ic \neq 0) \), \**pxy** is not used.
   
   *Constraint:* if \( ic = 0 \), 0.0 \( \leq \) \**pxy** \( \leq \) 1.0.

4:  
   **iw** – Integer  
   
   *Input*
   
   *On entry:* the choice of lag window.
   
   \( iw = 1 \)
   
   Rectangular.
   
   \( iw = 2 \)
   
   Bartlett.
   
   \( iw = 3 \)
   
   Tukey.
\(iw = 4\)  
Parzen.

**Constraint:** \(1 \leq iw \leq 4\).

5: **mw** – Integer  
*Input*

*On entry:* \(M\), the ‘cut-off’ point of the lag window, relative to any alignment shift that has been applied. Windowed cross-covariances at lags \((-mw + ish)\) or less, and at lags \((mw + ish)\) or greater are zero.

**Constraints:**  
\[
\begin{align*}
mw & \geq 1; \\
mw + |ish| & \leq nxy.
\end{align*}
\]

6: **ish** – Integer  
*Input*

*On entry:* \(S\), the alignment shift between the \(x\) and \(y\) series. If \(x\) leads \(y\), the shift is positive.

**Constraint:** \(-mw < ish < mw\).

7: **ic** – Integer  
*Input*

*On entry:* indicates whether cross-covariances are to be calculated in the function or supplied in the call to the function.

\(ic = 0\)  
Cross-covariances are to be calculated.

\(ic \neq 0\)  
Cross-covariances are to be supplied.

8: **nc** – Integer  
*Input*

*On entry:* the number of cross-covariances to be calculated in the function or supplied in the call to the function.

**Constraint:** \(mw + |ish| \leq nc \leq nxy\).

9: **cxy[nc]** – double  
*Input/Output*

*On entry:* if \(ic \neq 0\), \(cxy\) must contain the \(nc\) cross-covariances between values in the \(y\) series and earlier values in time in the \(x\) series, for lags from 0 to \((nc - 1)\).

If \(ic = 0\), \(cxy\) need not be set.

*On exit:* if \(ic = 0\), \(cxy\) will contain the \(nc\) calculated cross-covariances.

If \(ic \neq 0\), the contents of \(cxy\) will be unchanged.

10: **cyx[nc]** – double  
*Input/Output*

*On entry:* if \(ic \neq 0\), \(cyx\) must contain the \(nc\) cross-covariances between values in the \(y\) series and later values in time in the \(x\) series, for lags from 0 to \((nc - 1)\).

If \(ic = 0\), \(cyx\) need not be set.

*On exit:* if \(ic = 0\), \(cyx\) will contain the \(nc\) calculated cross-covariances.

If \(ic \neq 0\), the contents of \(cyx\) will be unchanged.

11: **kc** – Integer  
*Input*

*On entry:* if \(ic = 0\), \(kc\) must specify the order of the fast Fourier transform (FFT) used to calculate the cross-covariances. \(kc\) should be a product of small primes such as \(2^m\) where \(m\) is the smallest integer such that \(2^m \geq n + nc\).
If \( ic \neq 0 \), that is if covariances are supplied, \( kc \) is not used.

**Constraint:** \( kc \geq nxy + nc \). 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.

12: \( l \) — Integer  

*Input*

*On entry:* \( L \), the frequency division of the spectral estimates as \( \frac{2\pi}{L} \). Therefore it is also the order of the FFT used to construct the sample spectrum from the cross-covariances. \( l \) should be a product of small primes such as \( 2^m \) where \( m \) is the smallest integer such that \( 2^m \geq 2M - 1 \).

**Constraint:** \( l \leq \frac{n}{2} \times mw - 1 \). The largest prime factor of \( l \) must not exceed 19, and the total number of prime factors of \( l \), counting repetitions, must not exceed 20. These two restrictions are imposed by the internal FFT algorithm used.

13: \( xg[dim] \) — double  

*Input/Output*

*Note:* the dimension, \( dim \), of the array \( xg \) must be at least \( \max(kc, l) \), when \( ic = 0 \);

*On entry:* if the cross-covariances are to be calculated (\( ic = 0 \)) \( xg \) must contain the \( nxy \) data points of the \( x \) series. If covariances are supplied (\( ic \neq 0 \)) \( xg \) may contain any values.

*On exit:* contains the real parts of the \( ng \) complex spectral estimates in elements \( xg[0] \) to \( xg[ng - 1] \), and \( xg[ng] \) to \( xg[dim - 1] \) contain 0.0. The \( y \) series leads the \( x \) series.

14: \( yg[dim] \) — double  

*Input/Output*

*Note:* the dimension, \( dim \), of the array \( yg \) must be at least \( \max(kc, l) \), when \( ic = 0 \);

*On entry:* if the cross-covariances are to be calculated (\( ic = 0 \)) \( yg \) must contain the \( nxy \) data points of the \( y \) series. If covariances are supplied (\( ic \neq 0 \)) \( yg \) may contain any values.

*On exit:* contains the imaginary parts of the \( ng \) complex spectral estimates in elements \( yg[0] \) to \( yg[ng - 1] \), and \( yg[ng] \) to \( yg[dim - 1] \) contain 0.0. The \( y \) series leads the \( x \) series.

15: \( g[l/2 + 1] \) — Complex  

*Output*

*On exit:* the complex vector that contains the \( ng \) cross spectral estimates in elements \( g[0] \) to \( g[ng - 1] \). The \( y \) series leads the \( x \) series.

16: \( ng \) — Integer  

*Output*

*On exit:* the number, \( \lfloor l/2 \rfloor + 1 \), of complex spectral estimates.

17: \( fail \) — NagError  

*Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 **Error Indicators and Warnings**

**NE_ALLOC_FAIL**  
Dynamic memory allocation failed.  
See Section 3.2.1.2 in the Essential Introduction for further information.
NE_BAD_PARAM
On entry, argument \( \langle \text{value} \rangle \) had an illegal value.

NE_INT
On entry, \( \text{ic} = 0 \) and \( \text{mtxy}\_\text{correction} \neq \text{Nag\_NoCorrection}, \text{Nag\_Mean} \) or \( \text{Nag\_Trend} \): \( \text{mtxy}\_\text{correction} = \langle \text{value} \rangle \).
On entry, \( \text{iw} = \langle \text{value} \rangle \).
Constraint: \( \text{iw} = 1, 2, 3 \) or \( 4 \).
On entry, \( \text{mw} = \langle \text{value} \rangle \).
Constraint: \( \text{mw} \geq 1 \).
On entry, \( \text{nxy} = \langle \text{value} \rangle \).
Constraint: \( \text{nxy} \geq 1 \).

NE_INT_2
On entry, \( \text{ish} = \langle \text{value} \rangle \) and \( \text{mw} = \langle \text{value} \rangle \).
Constraint: \( \text{ish} \leq \text{mw} \).
On entry, \( \text{l} = \langle \text{value} \rangle \) and \( \text{mw} = \langle \text{value} \rangle \).
Constraint: \( \text{l} \geq 2 \times \text{mw} - 1 \).
On entry, \( \text{nc} = \langle \text{value} \rangle \) and \( \text{nxy} = \langle \text{value} \rangle \).
Constraint: \( \text{nc} \leq \text{nxy} \).

NE_INT_3
On entry, \( \text{kc} = \langle \text{value} \rangle \), \( \text{nxy} = \langle \text{value} \rangle \) and \( \text{nc} = \langle \text{value} \rangle \).
Constraint: if \( \text{ic} = 0 \), \( \text{kc} \geq \text{nxy} + \text{nc} \).
On entry, \( \text{mw} = \langle \text{value} \rangle \), \( \text{ish} = \langle \text{value} \rangle \) and \( \text{nxy} = \langle \text{value} \rangle \).
Constraint: \( \text{mw} + |\text{ish}| \leq \text{nxy} \).
On entry, \( \text{nc} = \langle \text{value} \rangle \), \( \text{mw} = \langle \text{value} \rangle \) and \( \text{ish} = \langle \text{value} \rangle \).
Constraint: \( \text{nc} \geq \text{mw} + |\text{ish}| \).

NE_INT_REAL
On entry, \( \text{pxy} = \langle \text{value} \rangle \).
Constraint: if \( \text{ic} = 0 \), \( \text{pxy} \leq 1.0 \).
On entry, \( \text{pxy} = \langle \text{value} \rangle \).
Constraint: if \( \text{ic} = 0 \), \( \text{pxy} \geq 0.0 \).

NE_INTERNAL_ERROR
An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.
An unexpected error has been triggered by this function. Please contact NAG. See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE
Your licence key may have expired or may not have been installed correctly. See Section 3.6.5 in the Essential Introduction for further information.

NE_PRIME_FACTOR
\( \text{kc} \) has a prime factor exceeding 19, or more than 20 prime factors (counting repetitions):
\( \text{kc} = \langle \text{value} \rangle \).
\( \text{l} \) has a prime factor exceeding 19, or more than 20 prime factors (counting repetitions):
\( \text{l} = \langle \text{value} \rangle \).
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
Not applicable.

9 Further Comments
nag_tsa_spectrum_bivar_cov (g13ccc) carries out two FFTs of length \( kc \) to calculate the sample cross-covariances and one FFT of length \( L \) to calculate the sample spectrum. The timing of nag_tsa_spectrum_bivar_cov (g13ccc) is therefore dependent on the choice of these values. The time taken for an FFT of length \( n \) is approximately proportional to \( n \log(n) \) (but see Section 9 in nag_sum_fft_realherm_1d (c06pac) for further details).

10 Example
This example reads two time series of length 296. It then selects mean correction, a 10% tapering proportion, the Parzen smoothing window and a cut-off point of 35 for the lag window. The alignment shift is set to 3 and 50 cross-covariances are chosen to be calculated. The program then calls nag_tsa_spectrum_bivar_cov (g13ccc) to calculate the cross spectrum and then prints the cross-covariances and cross spectrum.

10.1 Program Text
/* nag_tsa_spectrum_bivar_cov (g13ccc) Example Program. *
 * Copyright 2014 Numerical Algorithms Group.
 * * Mark 7, 2002.
 */
#include <stdio.h>
#include <nag.h>
#include <nag.stdlib.h>
#include <nagg13.h>

int main(void)
{
    /* Scalars */
    double pxy;
    Integer exit_status, i, ic, ii, ish, iw, kc, lf,
              mw, nc, ng, nxy, nxyg;

    /* Arrays */
    double *cxy = 0, *cyx = 0, *xg = 0, *yg = 0;
    Complex *g = 0;

    NagMeanOrTrend mtxy;
    NagError fail;

    INIT_FAIL(fail);
    exit_status = 0;

    printf("nag_tsa_spectrum_bivar_cov (g13ccc) Example Program Results\n");

    /* Skip heading in data file */
    #ifdef _WIN32
    scanf_s("%*[\n] ");
    #endif

    /* Other code for the program */
else
    scanf("%*[\n] ");
#endif

#ifdef __WIN32
    scanf_s("%NAG_IFMT%NAG_IFMT%NAG_IFMT%*[\n] ", &nxy, &nc, &ic);
#else
    scanf("%NAG_IFMT%NAG_IFMT%NAG_IFMT%*[\n] ", &nxy, &nc, &ic);
#endif

if (nxy > 0 && nc > 0)
{
    /* Set parameters for call to nag_tsa_spectrum_bivar_cov (g13ccc) */
    /* Mean correction and 10 percent taper */
    mtxy = Nag_Mean;
    pxy = 0.1;

    /* Parzen window and zero covariance at lag 35 */
    iw = 4;
    mw = 35;

    /* Alignment shift of 3, 50 covariances to be calculated */
    ish = 3;
    kc = 350;
    lf = 80;

    if (ic == 0)
        nxyg = MAX(kc, lf);
    else
        nxyg = lf;

    /* Allocate arrays xg, yg, cxy and cyx */
    if (!(xg = NAG_ALLOC(nxyg, double)) ||
        !(yg = NAG_ALLOC(nxyg, double)) ||
        !(cxy = NAG_ALLOC(nc, double)) ||
        !(cyx = NAG_ALLOC(nc, double)) ||
        !(g = NAG_ALLOC((lf/2)+1, Complex)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    if (ic == 0)
    {
        for (i = 1; i <= nxy; ++i)
            #ifdef __WIN32
                scanf_s("%lf", &xg[i-1]);
            #else
                scanf("%lf", &xg[i-1]);
            #endif
            #ifdef __WIN32
                scanf_s("%*[\n] ");
            #else
                scanf("%*[\n] ");
            #endif
    }
    else
    {
        for (i = 1; i <= nc; ++i)
            #ifdef __WIN32
                scanf_s("%lf", &yg[i-1]);
            #else
                scanf("%lf", &yg[i-1]);
            #endif
            #ifdef __WIN32
                scanf_s("%*[\n] ");
            #else
                scanf("%*[\n] ");
            #endif
    }
}
#ifdef _WIN32
  scanf_s("%lf", &cxy[i-1]);
#else
  scanf("%lf", &cxy[i-1]);
#endif
#endif
/* nag_tsa_spectrum_bivar_cov (g13ccc).
 * Multivariate time series, smoothed sample cross spectrum
 * using rectangular, Bartlett, Tukey or Parzen lag window
 */
if (fail.code != NE_NOERROR)
{
  printf(
    "Error from nag_tsa_spectrum_bivar_cov (g13ccc).\n%s\n", fail.message);
  exit_status = 1;
  goto END;
}
print("\n");
printf(" Returned cross covariances\n");
printf(" Lag XY YX Lag XY YX Lag XY YX\n");
for (i = 1; i <= nc; i += 3)
{
  for (ii = i; ii <= MIN(i+2, nc); ++ii)
    printf("%4"NAG_IFMT"%9.4f%9.4f ", ii-1, cxy[ii-1],
      cyx[ii-1]);
  printf("\n");
}
print("\n");
printf(" Returned sample spectrum\n");
printf("\n");
printf(" Real Imaginary Real Imaginary "
      " Real Imaginary\n");
printf(" Lag part part Lag part part Lag"
      " part part\n");
for (i = 1; i <= ng; i += 3)
{
  for (ii = i; ii <= MIN(i+2, ng); ++ii)
    printf("%4"NAG_IFMT"%9.4f%9.4f ", ii-1, g[ii-1].re,
      g[ii-1].im);
  printf("\n");
}
END:
NAG_FREE(cxy);
NAG_FREE(cyx);
NAG_FREE(xg);
NAG_FREE(yg);
NAG_FREE(g);
return exit_status;
}

10.2 Program Data

nag_tsa_spectrum_bivar_cov (g13ccc) Example Program Data

```
296 50 0
-0.109 0.000 0.178 0.339 0.373 0.441 0.461 0.348
0.127 -0.180 -0.588 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
0.164 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 -1.746 -1.474 -1.201 -0.927
-0.475 -0.193 0.088 0.435 0.771 0.866 0.875 0.891
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 -0.423 -0.259 -0.423 -0.194 -0.049
1.000 1.337 1.460 1.353 0.772 0.218 -0.237 -0.714
-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 -0.039 -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
53.8 53.6 53.5 53.4 53.1 52.7 52.4 52.0 51.6 51.2 50.8 50.4 49.8 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 49.8 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 50.1 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 54.4 55.5 55.6 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 57.6 58.8 59.5 54.1 54.3 55.3 56.4 57.2
57.8 58.1 58.6 58.8 58.6 58.0 57.4 57.0 56.4 56.3 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 57.0
```
10.3 Program Results

nag_tsa_spectrum_bivar_cov (g13ccc) Example Program Results

Returned cross covariances

<table>
<thead>
<tr>
<th>Lag</th>
<th>XY</th>
<th>YX</th>
<th>Lag</th>
<th>XY</th>
<th>YX</th>
<th>Lag</th>
<th>XY</th>
<th>YX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1.6700</td>
<td>-1.6700</td>
<td>1</td>
<td>-2.0581</td>
<td>-1.3606</td>
<td>2</td>
<td>-2.4859</td>
<td>-1.1383</td>
</tr>
<tr>
<td>3</td>
<td>-2.8793</td>
<td>-0.9926</td>
<td>4</td>
<td>-3.1473</td>
<td>-0.9009</td>
<td>5</td>
<td>-3.2239</td>
<td>-0.8382</td>
</tr>
<tr>
<td>6</td>
<td>-3.0929</td>
<td>-0.7804</td>
<td>7</td>
<td>-2.7974</td>
<td>-0.7074</td>
<td>8</td>
<td>-2.4145</td>
<td>-0.6147</td>
</tr>
<tr>
<td>9</td>
<td>-2.0237</td>
<td>-0.5080</td>
<td>10</td>
<td>-1.6802</td>
<td>-0.4032</td>
<td>11</td>
<td>-1.4065</td>
<td>-0.3159</td>
</tr>
<tr>
<td>12</td>
<td>-1.2049</td>
<td>-0.2554</td>
<td>13</td>
<td>-1.0655</td>
<td>-0.2250</td>
<td>14</td>
<td>-0.9726</td>
<td>-0.2238</td>
</tr>
<tr>
<td>15</td>
<td>-0.9117</td>
<td>-0.2454</td>
<td>16</td>
<td>-0.8658</td>
<td>-0.2784</td>
<td>17</td>
<td>-0.8180</td>
<td>-0.3081</td>
</tr>
<tr>
<td>18</td>
<td>-0.7563</td>
<td>-0.3257</td>
<td>19</td>
<td>-0.6750</td>
<td>-0.3315</td>
<td>20</td>
<td>-0.5754</td>
<td>-0.3321</td>
</tr>
<tr>
<td>21</td>
<td>-0.4701</td>
<td>-0.3308</td>
<td>22</td>
<td>-0.3738</td>
<td>-0.3312</td>
<td>23</td>
<td>-0.3023</td>
<td>-0.3332</td>
</tr>
<tr>
<td>24</td>
<td>-0.2665</td>
<td>-0.3384</td>
<td>25</td>
<td>-0.2645</td>
<td>-0.3506</td>
<td>26</td>
<td>-0.2847</td>
<td>-0.3727</td>
</tr>
<tr>
<td>27</td>
<td>-0.3103</td>
<td>-0.3992</td>
<td>28</td>
<td>-0.3263</td>
<td>-0.4152</td>
<td>29</td>
<td>-0.3271</td>
<td>-0.4044</td>
</tr>
<tr>
<td>30</td>
<td>-0.3119</td>
<td>-0.3621</td>
<td>31</td>
<td>-0.2837</td>
<td>-0.2919</td>
<td>32</td>
<td>-0.2568</td>
<td>-0.2054</td>
</tr>
<tr>
<td>33</td>
<td>-0.2427</td>
<td>-0.1185</td>
<td>34</td>
<td>-0.2490</td>
<td>-0.0414</td>
<td>35</td>
<td>-0.2774</td>
<td>0.0227</td>
</tr>
<tr>
<td>36</td>
<td>-0.3218</td>
<td>0.0697</td>
<td>37</td>
<td>-0.3705</td>
<td>0.1039</td>
<td>38</td>
<td>-0.4083</td>
<td>0.1356</td>
</tr>
<tr>
<td>39</td>
<td>-0.4197</td>
<td>0.1805</td>
<td>40</td>
<td>-0.3920</td>
<td>0.2460</td>
<td>41</td>
<td>-0.3241</td>
<td>0.3319</td>
</tr>
<tr>
<td>42</td>
<td>-0.2273</td>
<td>0.4325</td>
<td>43</td>
<td>-0.1216</td>
<td>0.5331</td>
<td>44</td>
<td>-0.0245</td>
<td>0.6199</td>
</tr>
<tr>
<td>45</td>
<td>0.0528</td>
<td>0.6875</td>
<td>46</td>
<td>0.1074</td>
<td>0.7329</td>
<td>47</td>
<td>0.1448</td>
<td>0.7550</td>
</tr>
<tr>
<td>48</td>
<td>0.1713</td>
<td>0.7544</td>
<td>49</td>
<td>0.1943</td>
<td>0.7349</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Returned sample spectrum

<table>
<thead>
<tr>
<th>Lag</th>
<th>Real part</th>
<th>Imaginary part</th>
<th>Lag</th>
<th>Real part</th>
<th>Imaginary part</th>
<th>Lag</th>
<th>Real part</th>
<th>Imaginary part</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-6.5500</td>
<td>0.0000</td>
<td>1</td>
<td>-5.4267</td>
<td>-1.9842</td>
<td>2</td>
<td>-3.1323</td>
<td>-2.7307</td>
</tr>
<tr>
<td>3</td>
<td>-1.2649</td>
<td>-2.3998</td>
<td>4</td>
<td>-0.2102</td>
<td>-1.7520</td>
<td>5</td>
<td>0.3411</td>
<td>-1.1903</td>
</tr>
<tr>
<td>6</td>
<td>0.6063</td>
<td>-0.7420</td>
<td>7</td>
<td>0.6178</td>
<td>-0.3586</td>
<td>8</td>
<td>0.4391</td>
<td>-0.1008</td>
</tr>
<tr>
<td>9</td>
<td>0.2422</td>
<td>0.0061</td>
<td>10</td>
<td>0.1233</td>
<td>0.0409</td>
<td>11</td>
<td>0.0574</td>
<td>0.0529</td>
</tr>
<tr>
<td>12</td>
<td>0.0174</td>
<td>0.0452</td>
<td>13</td>
<td>-0.0008</td>
<td>0.0289</td>
<td>14</td>
<td>-0.0058</td>
<td>0.0161</td>
</tr>
<tr>
<td>15</td>
<td>-0.0051</td>
<td>0.0084</td>
<td>16</td>
<td>-0.0027</td>
<td>0.0040</td>
<td>17</td>
<td>-0.0010</td>
<td>0.0015</td>
</tr>
<tr>
<td>18</td>
<td>-0.0006</td>
<td>0.0006</td>
<td>19</td>
<td>-0.0005</td>
<td>0.0003</td>
<td>20</td>
<td>-0.0003</td>
<td>0.0003</td>
</tr>
<tr>
<td>21</td>
<td>-0.0003</td>
<td>0.0004</td>
<td>22</td>
<td>-0.0003</td>
<td>0.0003</td>
<td>23</td>
<td>-0.0003</td>
<td>0.0002</td>
</tr>
<tr>
<td>24</td>
<td>-0.0004</td>
<td>0.0001</td>
<td>25</td>
<td>-0.0004</td>
<td>-0.0000</td>
<td>26</td>
<td>-0.0003</td>
<td>-0.0001</td>
</tr>
<tr>
<td>27</td>
<td>-0.0002</td>
<td>-0.0001</td>
<td>28</td>
<td>-0.0001</td>
<td>0.0001</td>
<td>29</td>
<td>-0.0002</td>
<td>0.0003</td>
</tr>
<tr>
<td>30</td>
<td>-0.0003</td>
<td>0.0002</td>
<td>31</td>
<td>-0.0002</td>
<td>0.0001</td>
<td>32</td>
<td>-0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>33</td>
<td>-0.0000</td>
<td>0.0000</td>
<td>34</td>
<td>0.0001</td>
<td>-0.0001</td>
<td>35</td>
<td>0.0001</td>
<td>-0.0002</td>
</tr>
<tr>
<td>36</td>
<td>0.0001</td>
<td>-0.0001</td>
<td>37</td>
<td>0.0001</td>
<td>-0.0001</td>
<td>38</td>
<td>0.0001</td>
<td>-0.0001</td>
</tr>
<tr>
<td>39</td>
<td>0.0001</td>
<td>-0.0001</td>
<td>40</td>
<td>0.0001</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

NAG Library Manual