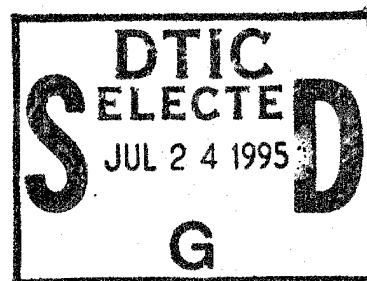
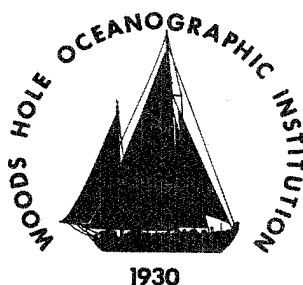


Woods Hole Oceanographic Institution



AUTO AND CROSS-BISPECTRAL ANALYSIS
OF SCALAR AND VECTOR TIME SERIES:
PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS
WITH ARTIFICIAL DATA

by

Gerard H. Martineau
and
Melbourne G. Briscoe

April 1978

TECHNICAL REPORT

*Prepared for the National Science Foundation
under Grant OCE76-14739.*

19950720 027

WOODS HOLE, MASSACHUSETTS 02543

DTIC QUALITY INSPECTED 8

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

JK

WHOI-77-75

AUTO AND CROSS-BISPECTRAL ANALYSIS
OF SCALAR AND VECTOR TIME SERIES:
PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS
WITH ARTIFICIAL DATA

by

Gerard H. Martineau
and
Melbourne G. Briscoe

WOODS HOLE OCEANOGRAPHIC INSTITUTION
Woods Hole, Massachusetts 02543

April 1978

TECHNICAL REPORT

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

*Prepared for the National Science Foundation
under Grant OCE76-14739.*

*Reproduction in whole or in part is permitted
for any purpose of the United States Government.
In citing this manuscript in a bibliography,
the reference should be followed by the phrase:
UNPUBLISHED MANUSCRIPT.*

Approved for Distribution:

Valentine Worthington
Valentine Worthington, Chairman
Department of Physical Oceanography

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Table of Contents

- I. Introduction, Summary, and Examples
 - A. Introduction
 - B. Summary
 - C. Examples based on Artificial Signals
 - D. Empirical Confidence Levels for Bicoherence
 - E. References for Section I
- II. Bispectral Programs
 - A. Introduction to the Bispectral Program Reports
 - 1. Summary
 - 2. Flow Chart for the Bispectral Programs
 - 3. Compatibility with Other Systems
 - 4. References for Section IIA
 - B. Building RWDISC Data File Preparatory to Fourier Transform
 - 1. FRAGTAP; ORDAT
 - 2. GENRAN
 - C. Modifying RWDISC File from B by Replacing Data Series with Fourier Coefficients in a Special Order
 - 1. FOURIER
 - D. Bispectral Calculations and Plotting
 - 1. BISCAL
 - 2. BPLOT
 - 3. BISUM
 - 4. BIVEC
 - 5. RBPLOT
 - E. FORTRAN IV Listings, Alphabetically

I. Introduction, Summary, and Examples

A. Introduction

The usual spectral analysis is an examination of the second-order properties of a time series. The spectrum is, in fact, a description of how the various frequency bands contribute to the variance (the second moment) of the series, or the covariance in the case of cross-spectra. If the time series is a record from a Gaussian process, then the second-order spectrum provides a "complete" description because all higher moments of the series (for example the third moment, skewness, or the fourth moment, Kurtosis) are either zero or representable in terms of the second moment. One particular feature of stationary Gaussian processes is that the individual frequency components are independent; second-order spectral analysis is consistent with this feature, in that nothing in the analysis is sensitive to any non-independence of frequency components.

But phase-locking of frequency components is a significant feature of numerous important processes. For instance, spectral lines may show up at some frequency and at twice that frequency; second-order spectral analysis can only detect the existence of the lines, but an analysis that can detect whether the two lines are phase-locked can tell if the higher-frequency line is simply the first harmonic of the lower-frequency line. A common example would

be lines at 0.08 and 0.16 cph; the first is probably the M_2 tide, but is the second the M_4 tide or the first harmonic of the M_2 tide? A less common example would be quadratic nonlinear interactions, in which the product of frequency components at ω_1 and ω_2 would give rise to a phase-locked frequency component at ω_3 , where $\omega_1 + \omega_2 - \omega_3 = 0$. This is analogous to the usual spectral analysis in which only two frequency components contribute to the energy, such that $\omega_1 + \omega_2 = 0$; this is usually written as $\omega_2 = -\omega_1$, and interpreted as the energy being due to the frequency components at $\pm\omega_1$.

Thus the bispectrum is a third-order spectrum (c.f., Hasselman, Munk, and McDonald, 1963;¹ Nagata, 1970;² Yao, Neshyba, and Crew, 1975³ and 1977;⁴ and Haubrich, 1965⁵) that is sensitive to phase-locked frequency triplets, such that $\omega_1 + \omega_2 - \omega_3 = 0$. Because only ω_1 and ω_2 need be mentioned ($\omega_3 = \omega_1 + \omega_2$ always), the pair (ω_1, ω_2) is referred to as a bi-frequency, and the display of the bispectrum is usually in terms of contours in the bi-frequency plane.

If all three frequencies from the triplet $(\omega_1, \omega_2, \omega_3)$ are from a single scalar (real) series, it is becoming customary to refer to the "auto-bispectrum," whereas if the three frequencies are not from the same scalar series, one refers to the "cross-bispectrum." If the series are not scalar, but rather vector random processes, such as

wind or ocean current velocity, one can choose a method of treating the vectors developed by Yao et al., 1975,³ and use a rotary decomposition of the vectors that is independent of the rectangular coordinate system that might be used, and which is particularly useful for physical processes involving rotation. In this case one refers to "auto-rotary bispectra," and so on.

B. Summary

This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBLOT) as contour plots on the bi-frequency plane.* The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \pm\omega_1 \pm\omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs,

*See pages 75-77 for precise definitions of the bispectrum and bicoherence.

there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.

C. Examples Based on Artificial Signals

Definition of "Hanning" and "Overlapping"

In the following discussion, the terms "Hanning" and "overlapping" are used. Hanning* refers to a particular data window applied to the time domain in order to reduce side lobes and consequent leakage in the frequency domain. The window is a "raised cosine" over 100 percent of the time record; many windows are possible and some are more efficacious, but the Hanning window is extremely efficient to implement. See Reference 6 for a general discussion of windows.

One effect of windowing is to lessen the effective length of the time domain segment, which equivalently reduces the useful degrees of freedom in the segment. This effect is compensable by segmenting the entire time domain record into overlapping pieces so that the losses due to windowing in one segment are recovered in the overlapped segment. Nuttall⁷ has shown that 62.5 percent overlapping allows a complete recovery of degrees-of-freedom losses, but that 50 percent overlapping is much simpler and yields a 92 percent recovery of losses.

The computational recipes applied here are as follows:

*"Hanning" is a traditional misnomer: the term is based on the name of Julius Von Hann, who first described the window, by analogy to Hamming, which is another window that is due to Richard Hamming.

Let a series consist of $(2T_m)$ points spaced at a "time" $\Delta\tau = 1$ apart so that the Nyquist frequency is $(1/2)$ and there are (T_m+1) spectral estimates. Let $u'_{T_m}(\tau)$ be the Hanning window to be applied at time τ . Then:

$$u'_{T_m}(\tau) = \begin{cases} 0 & \tau < 0 \\ \frac{1}{2} [1 - \cos(\frac{\pi\tau}{T_m})] & 0 \leq \tau \leq 2 T_m \\ 0 & \tau > 2 T_m \end{cases} \quad (1)$$

The Fourier transform of $u'_{T_m}(\tau)$ is $U'_{T_m}(f)$ as follows:

$$U'_{T_m}(f) = \frac{1}{2}U_{T_m}(f) - \frac{1}{4}U_{T_m}(f - \frac{1}{2T_m}) - \frac{1}{4}U_{T_m}(f + \frac{1}{2T_m}) \quad (2)$$

where $0 \leq f \leq 1/2$ and $\Delta f = 1/(2T_m)$ and $U_{T_m}(f)$ is the Fourier transform of the rectangular window of unit height over the time $2T_m$ for which the expression (1) is nonzero. By the convolution theorem the application of expression (2) to the Fourier transformed series is equivalent to applying (1) to the original series. It is expression (2) that is used to accomplish Hanning below. Specifically, if we consider a real series of n points having the following Fourier representation:

$$U_i = a_0 + \sum_{k=1}^{n/2} [a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)],$$

then the i^{th} Hanned coefficient is:

$$U'_i = 0.5 U_i - 0.25(U_{i-1} + U_{i+1}) .$$

The Fourier coefficients a_1 through $(a_{n/2} - 1)$, and b_2 through $b_{n/2} - 2$ are changed by Hanning, leaving the following unchanged: a_0 , $a_{n/2}$, b_1 , and $b_{n/2-1}$.

If the variance of the Hanned series is now computed, it will be found to be low because of the cosine taper. To correct for this, the transformed and Hanned series can be multiplied by the ratio of the variance of the transformed, non-Hanned, series to the variance of the transformed, Hanned series. An approximation to this value, valid for a sufficiently large number of data pieces, is a theoretical value² for Gaussian, white noise: $\sqrt{8/3}$. All transformed and Hanned series in this report have been multiplied by $\sqrt{8/3}$.

In calculating the power spectrum of a data series, it is desirable to divide the series into segments, and to overlap these segments when using a data or spectral window such as the Hanning window (Welch, 1967⁹). The segmenting is desirable for many reasons, such as the reduction in the series length to be transformed, a possible reduction in the number of overall computations, greater ease of observing nonstationary trends in the data, and greater control over frequency resolution.

Specifically, for a 50% overlap, let the number of points per segment or "piece" be S . If the pieces are numbered consecutively $i = 1, 2, \dots$, and the data points are numbered consecutively $1, 2, \dots$, then the i^{th} piece starts

at point $[1+(i-1)S/2]$. This is done until no more complete pieces can be constructed from the data, the remaining data being discarded if necessary. Note that because of the Fast Fourier Transform algorithm used, the piece length is never odd.

All real (i.e., scalar) series or real components of vector series to be used below consist of 10 pieces of length 64, resulting in 32 complex Fourier coefficients for each piece. These 10 pieces are contiguous in the case of no Hanning and no overlapping, meaning that 640 points of the series are used. For Hanning and 50% overlapping, 352 points of the series are used.

All contour plots except Figures 16 and 17 are contoured at the 80, 85, 90, 95, and 99% confidence levels determined from a Gaussian noise background.* Figures 16 and 17 emphasize the bicoherence peaks by displaying contours at the 95, 96, 97, 98, and 99% confidence levels. All series used for auto- and cross-bicoherence plots are Hanned and overlapped. A comparison of Hanning with no Hanning is given in the demonstration of auto-rotary bicoherence plots.

* See page 7 in program BISCAL report below, and pages 6,7 in program BIVEC below.

1. Auto-bicoherence

Figures 3, 4, and 5 are auto-bicoherence contour plots of the following three Hanned artificial signals:*

$$\text{Fig. 3: } s = g + a(\cos 8t + \cos 16t + \cos 24t)$$

$$\text{Fig. 4: } s = g + a(\sin 8t + \sin 16t + \sin 24t)$$

$$\text{Fig. 5: } s = g + a(-\sin 8t - \sin 16t - \sin 24t)$$

In all three cases, g is Gaussian noise with zero mean and unity variance, and $a = 10$. Figure 3 uses the Gaussian noise background whose auto-bicoherence is displayed in Figure 1, while Figures 4 and 5 both use the Gaussian noise background having the auto-bicoherence shown in Figure 2. It is observed that all three figures are essentially alike, with two peaks at the only possible coupled frequency triplets $(\omega_1, \omega_2, \omega_3)$:

$$(\omega_1, \omega_2, \omega_3) = (8, 8, 16) \text{ and } (8, 16, 24) .$$

Note that ω_3 is the horizontal intercept of the straight line $\omega_2 = -\omega_1 + \omega_3$ [i.e., $\omega_1 + \omega_2 - \omega_3 = 0$] having slope (-1) and passing through the bi-frequency (ω_1, ω_2) . The contour plots are alike because the signals differ only in the phasing of the terms and the bicoherence does not give any phase information.

*In generating all of the artificial series used in this report, an error in indexing in program GENRAN introduced into each sinusoidal term an additional phase angle of $(-2\pi\omega/L)$, where ω = frequency and L = piece length. Since this made no difference in the appearance or interpretation of the plots, the examples presented were not repeated.

Now notice that if Figures 3 and 4 were presented as the "east" and "north" components of velocity, for example, they would be representing three coupled counterclockwise rotating vectors of frequencies 8, 16, and 24. If Figures 3 and 5 were so presented, they would be representing three coupled clockwise rotating vectors. Since the pairs of figures are identical, could one detect the presence of these coupled rotating vectors? In order to do this, it is necessary to examine the biphase at the coincident peaks in the east and north components. The bicoherence and rotary bicoherence programs can furnish this information. For Figure 3 the biphase value for all peaks is essentially 0. For all peaks of Figure 4, it is $\pi/2$, and for all peaks of Figure 5, it is $-\pi/2$. This means that for the coincident peaks in the pair consisting of Figures 3 and 4, the north biphase leads east by 90° and for the pair 3 and 5, north lags east by 90° . The fact that there are coincident peaks implies the existence of coupled sinusoids of the same frequency. The biphase difference of 90° implies that the sinusoids can represent the east and north components of a rotating vector, and the sign of the difference gives the sense of rotation. Thus the conditions above can exist together only if Figures 3 and 4, given as east and north, represent a coupled counterclockwise-rotating vector, and if Figures 3 and 5 represent a coupled clockwise-rotating vector.

It will be demonstrated below that the same conclusions about coupled rotating vectors and their sense of rotation can be reached by examining an auto-rotary bicoherence plot of these pairs of signals.

Figure 6 is an auto-bicoherence plot of the Hanned, overlapped, artificial signal

$$s = g^2 + 10 g \cdot \sin(16t) - 50 \cos(32t)$$

where g is Gaussian noise of zero mean and unity variance. The ridge pattern in Figure 6 comes from the joint behavior of the second and third terms in the signal. The second term is a product in the time domain of g (white noise, i.e., contains all frequency components) and a sinusoid (a delta function at 16), which could be expressed as a convolution in the frequency domain. The convolution reproduces the delta function at all the frequencies, not just 16. The $\sin(16t)$ and the $\cos(32t)$ are phase-locked, because no randomness is introduced into the phases, hence the $g \cdot \sin(16t)$ is also phase-locked to the third term.

Noting that $\omega_1 + \omega_2 - \omega_3 = 0$ is a requirement for a significant bispectrum, and noting that the second term describes a signal that may be thought of as a modulation of a signal at 16 to the new frequencies $16 + \omega_i$ and $16 - \omega_i$ and letting $\omega_1 = 16 + \omega_i$, $\omega_2 = 16 - \omega_i$, and $\omega_3 = 32$, we see that $\omega_1 + \omega_2 - \omega_3 = 0$ for all ω_i ! Also,

$\omega_1 + \omega_2 - \omega_3 = 0$ if $\omega_i = 16$. Therefore, we obtain significant bicoherence whenever $\omega_1 + \omega_2 = 32$ or when either $\omega_1 = 32$ or $\omega_2 = 32$.

2. Cross-bicoherence

Figures 7 and 7a are cross-bicoherence plots of the following two Hanned signals, s_x and s_y :

$$s_x = g_x + a(\cos 8t + \cos 16t + \cos 24t)$$

$$s_y = g_y + a(\sin(-8t) + \sin(-16t) + \sin(-24t)) .$$

Here g_x is the Gaussian noise of Figure 1 and g_y is the Gaussian noise of Figure 2. The cross-bicoherence between g_x and g_y is shown in Figure 7b. The form is $g_x g_x g_y$ for $\omega_1 \omega_2 \omega_3$ (see below). The amplitude a is again 10. Figure 7a displays bicoherences normalized by amplitude as described in the report on program BISCAL. Notice first that the domain has increased because the portion of the bi-frequency plane in which the bicoherence is unique has increased for cross-bicoherence.* The cross-bicoherence is computed here by using s_x as the source of ω_1 , s_x again as the source for ω_2 , and s_y for ω_3 . The source signals could have been taken in the only other unique sequence $s_x s_y s_y$. It can be seen that there are peaks at the only possible coupled frequency triplets:

* See page 5 of the program BISCAL report below.

$$(\omega_1, \omega_2, \omega_3) = (8, 8, 16), (8, 16, 24), (-8, 16, 8),$$

$$(-8, 24, 16), \text{ and } (-16, 24, 8) .$$

Notice that in effect s_x and s_y contain both positive and negative frequencies; i.e., 8, 16, 24, -8, -16, -24. There are "edges" of peaks at the frequency triplets $(-8, 32, 24)$, $(-16, 32, 16)$, and $(-24, 32, 8)$. These are visible because of the frequency-spreading due to Hanning, and are in the portion of the bi-frequency plane which is redundant with the domain plotted.

Figure 7c is a plot of average bicoherence where the average is over N , the number of points used in calculating a line integral over the diagonal straight line

$$\omega_2 = -\omega_1 + \omega_3 \quad \text{corresponding to the locus of all frequency triplets having a given "sum frequency"}$$

ω_3 . The abscissa is the sum frequency and the plot is directly based on Figure 7. The integrals are calculated by program BISUM. It is seen that there are peaks at 16 and 24 for positive ω_2 , corresponding to the two frequency triplets $(8, 8, 16)$ and $(8, 16, 24)$. The apparent peak at $\omega_3 = 2$ is due to the fact that only one point contributes to this sum, hence the "average" is poorly specified. For negative ω_2 , there is a peak at 8 corresponding to the frequency triplets $(-8, 16, 8)$ and $(-16, 24, 8)$ and another at 16 corresponding to $(-8, 24, 16)$. The peak at 24 is due solely to the redundant bicoherence peak at the frequency triplet $(-8, 32, 24)$ as mentioned above.

Figure 7d is a plot of average squared-bicoherence based on Figure 7, which emphasizes the peaks.

3. Auto-rotary Bicoherence

Figures 8 and 9 are auto-rotary bicoherence plots of Gaussian noise signals used as background noise in the plots having contours at the confidence levels 80, 85, 90, 95, 99% (all except Figure 17), and for determining these confidence levels. These Gaussian noise signals are processed in the same way as the signals in which they serve as noise. Figure 8 differs from Figure 9 in that the noise of Figure 8 has been Hanned and overlapped, while that of Figure 9 has undergone neither. It is observed that both plots appear to have a random distribution of peaks.

Figures 10, 11, 12, and 13 are auto-rotary bicoherence plots of the following complex signal: $\vec{s} = (g_1 + au) + i(g_2 + av)$

$$\begin{aligned} u &= \cos 8t + \cos 16t + \cos 24t + \cos(-8t) + \cos(-16t) + \cos(-24t) \\ &= 2(\cos 8t + \cos 16t + \cos 24t) \end{aligned}$$

$$\begin{aligned} v &= \sin 8t + \sin 16t + \sin 24t + \sin(-8t) + \sin(-16t) + \sin(-24t) \\ &= 0 \end{aligned}$$

$$g_1, g_2 = \text{Gaussian noise, mean} = 0, \text{variance} = 1$$

$$\begin{aligned} a &= 10 \text{ for Fig. 10, } 1.0 \text{ for Fig. 11, } 0.1 \text{ for Fig. 12,} \\ &10 \text{ for Fig. 13.} \end{aligned}$$

This complex signal has three counter-rotating vectors of frequencies 8, 16, 24. In Figures 10, 11, and 12 the signal

has been Hanned and the signal-to-noise ratio (S/N) is 10, 1.0 and 0.1 respectively. There are marked peaks at the same frequencies in Figures 10 and 11, but Figure 12, where S/N is down to 0.1, does not seem to differ significantly from Gaussian noise. It is apparent from the relative sizes of the contours that they are higher on the peaks of Figure 11 (S/N = 1.0) than on those of Figure 10 (S/N = 10) as expected. There are peaks at the following coupled frequency triplets:

(8,8,16), (8,16,24)	quadrant I
(16,-8,8), (24,-8,16), (24,-16,8)	quadrant II
(-16,-8,-24), (-8,-8,-16)	quadrant III
(-16,8,-8), (-24,8,-16), (-24,16,-8)	quadrant IV

In a rotary bicoherence plot, a peak indicates a coupling of two rotating vectors of frequencies ω_1 and ω_2 with a third of frequency ω_3 . The sign of the frequency gives the sense of rotation in all cases, positive indicating counterclockwise. Figures 10 and 11 demonstrate that there are indeed three counter-rotating vectors of frequencies 8, 16, 24, -8, -16, -24 which are pairwise coupled so that the sum and difference frequencies exist in the same set.

Figure 13 is an auto-rotary bicoherence plot of the same signal with a S/N = 10, but this time not Hanned. Comparing it with the Hanned case of Figure 10, it is apparent that the peaks are still present at the same frequencies, but they are much narrower because they are each due to a

single point. The peaks $(8,8,16)$ and $(-8,-8,-16)$ are ill-formed because the one contributing point lies on the boundary of the contour plot. In general, these peaks are difficult to distinguish from the Gaussian noise background, and they would require something like a stereoscopic or perspective display. Alternatively one can raise the contour levels to emphasize the peaks and suppress the noise background. This is done in Figure 17 in which the contour levels are at the 95, 96, 97, 98, and 99% confidence levels, and the highlighting is apparent. The noise background itself is separately displayed in Figure 16.

Figures 14 and 15 are for comparison with Figures 3, 4, and 5 in connection with a hypothetical presentation of Figures 3 and 4, and 3 and 5 as two "east" and "north" component pairs of auto-bicoherence plots. Figure 14 is an auto-rotary bicoherence plot using a vector series whose u and v components are identically the signals of Figures 3 and 4 respectively. Figure 14 displays two peaks, at $(8,8,16)$ and $(16,8,24)$. These indicate directly that counterclockwise pairs of rotating vectors of rotation frequencies $(8,8)$ and $(16,8)$ are phase-locked to counterclockwise rotating vectors of frequencies 16 and 24 respectively. Recall that Figures 3 and 4 had to be used in conjunction with biphasic information to arrive at the same conclusion. Figure 15 displays two peaks, at $(-8,-8,-16)$ and $(-16,-8,-24)$,

indicating a similar set of clockwise rotating vectors, for which conclusion biphase information had to be added to the auto-bicoherence plots of Figures 3 and 5.

Figure 18 is a cross-rotary bicoherence plot between the following two complex signals, called X and Y:

X and Y are of form $\vec{s} = (g_1 + au) + i(g_2 + av)$, where:

g_1, g_2 = Gaussian noise, mean = 0, variance = 1

$a = 10$

For X:

$u = \cos(-8) + \cos(-16) + \cos(16)$

$v = \sin(-8) + \sin(-16) + \sin(16)$

For Y:

$u = \cos 8 + \cos 16 + \cos(-24)$

$v = \sin 8 + \sin 16 + \sin(-24)$.

The cross-rotary bispectrum and bicoherence are computed with X supplying ω_1 , X supplying ω_2 , and Y supplying ω_3 ("Form XXY"). Figure 18 displays only two peaks, at the coupled frequency triplets (16,-8,8) and (-16,-8,-24). The first indicates phase-locking of a counterclockwise rotating vector of frequency 16 with a clockwise rotating

one of frequency (-8) to produce a counterclockwise-rotating vector of frequency 8. The second indicates coupling of three clockwise-rotating vectors. Examination of the input frequencies shows that these are indeed the only two possibilities.

D. Empirical Confidence Levels for Bicoherence

Figure 19 displays empirically determined confidence levels at 90, 95, 99, and 99.9 percent confidence that the bicoherence lies below the value given by the graph as a function of the number of pieces over which the averaging was done and the equivalent number of independent pieces. The data were supplied by program BISCAL, which produced a record of 4096 auto-bicoherence values from Gaussian noise having mean 0 and unity variance. The record was processed by splitting it into 256-point piece lengths, Hanning, and 50% overlapping. Program BISCAL placed the bicoherences into bins of width 0.01, determined the fraction of the total number (4096 here) lying below the upper bin limit, and calculated the above confidence levels by linearly interpolating between bins.

The dashed line in Figure 19 is the modified theoretical 95% confidence level determined from the following expression (Haubrich, 1965):⁵

$$95\% \text{ confidence level} = (6/2P')^{1/2}$$

where:

$2P'$ = equivalent degrees of freedom

P' = equivalent number of independent pieces
 $= 18P^2/(19P-1)$

P = number of Hanned, 50% overlapped pieces

Examination of Figure 19 shows that 5.9 would be slightly better than 6.0 as the constant in Haubrich's expression.

The horizontal axis of Figure 19 is given both in P' and P . If one had neither Hanned nor overlapped,* then the P' scale should be used to determine confidence limits; if the pieces were Hanned and overlapped, the P scale is applicable.

The curves in Figure 19 are based on the usual version of bicoherence (see BISCAL write-up) which normalizes the modulus of the averaged bispectrum by the autospectral energy in each of the three frequency bands being examined. An alternate bicoherence calculation (see also BISCAL write-up) normalizes each bispectrum before the average is

* One should only overlap if windowing is also being done, otherwise the equivalent number of degrees of freedom may actually be reduced (Nuttall, personal communication, 1973).

calculated and the modulus taken; this is done to emphasize phase relations and minimize amplitude relations amongst the three frequency bands being examined. The confidence limits (for $P = 9$ and 39) are shown as solid symbols in Figure 19; for 95% confidence levels on "phase bicoherence" the usual 95% confidence level is satisfactory. One might also use Haubrich's simple expression with 5.8 as the constant.

E. References for Section I

1. "Bispectra of Ocean Waves," by K. Hasselman, W. H. Munk, and G. MacDonald, Time Series Analysis, edited by M. Rosenblatt, pp. 125-139, John Wiley & Sons, New York, 1963.
2. "Lag Joint Probability, Higher Order Covariance Function, and Higher Order Spectrum," by Yutaka Nagata, La mer (Bulletin de la Société franco-japonaise d'océanographie), 8(2), pp. 78-94, May, 1970.
3. "Rotary Cross-Bispectra and Energy Transfer Functions between Non-Gaussian Vector Processes I. Development and Example," by Neng-chun Yao, Steve Neshyba, and Henry Crew, Journal of Physical Oceanography, 5, 164-172, January, 1975.
4. "Rotary Cross-Bispectra and Energy Transfer Functions between Non-Gaussian Vector Processes II. Winds and Currents off the Oregon Coast," by Neng-chun Yao, S. Neshyba, and H. Crew, Journal of Physical Oceanography, 7, 892-903, November, 1977.
5. "Earth Noise, 5 to 500 Millicycles per Second," by Richard H. Haubrich, Journal of Geophysical Research, 70, 1415-1427, 1965.
6. "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," by Fredric J. Harris, Proceedings of the IEEE, 66(1), 51-83, January, 1978.

7. "Spectral Estimation by Means of Overlapped Fast Fourier Transform Processing of Windowed Data," by A. H. Nuttall, Naval Underwater Systems Center Report No. 4169, October 13, 1971.
8. "Energy Loss in Surface Wave Spectra Due to Data Windowing," by Melbourne G. Briscoe, Deep-Sea Research, 19, 665-667, 1972.
9. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms," P. D. Welch, IEEE Transactions on Audio and Electroacoustics, AU-15, 70-73, 1967.

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENRANOUTPUT 0
 FREQUENCY 2: GENRANOUTPUT 0
 FREQUENCY 3: GENRANOUTPUT 0

PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
 *NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT: 0.430 80.0
 19:41 DEC 21, '77 0.458 85.0
 0.479 90.0
 0.552 95.0
 0.667 99.0

GAUSS NOISE HANNED
 6 SEED=3333

FILE BVPL

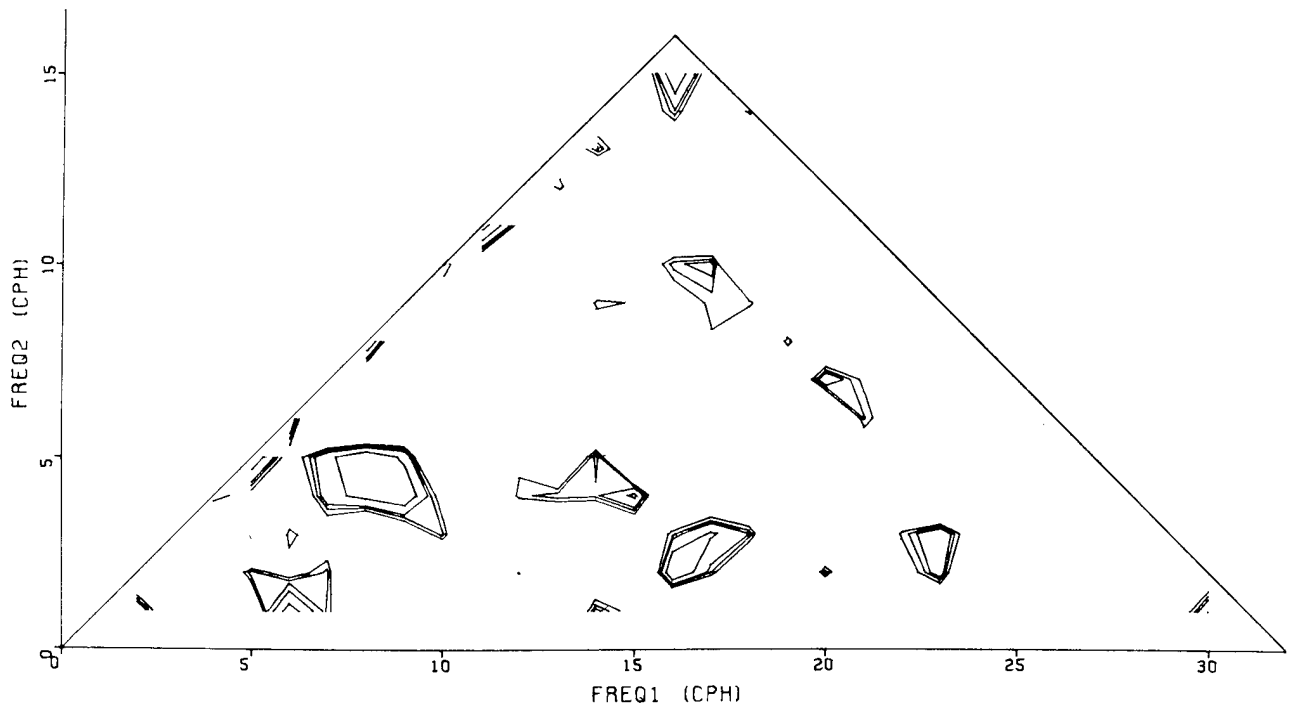


Figure 1

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENRANOUTPUT 0
 FREQUENCY 2: GENRANOUTPUT 0
 FREQUENCY 3: GENRANOUTPUT 0

PROCESSING HISTORY

NO LF'S 4*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
 *NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 17:18 DEC 15, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

	0.420	80.0
TIME OF PLOT:	0.447	85.0
09:48 DEC 27, '77	0.498	90.0
	0.591	95.0
	0.834	99.0

GAUSSIAN NOISE HANNED FILE BVPL5
 SEED=7777

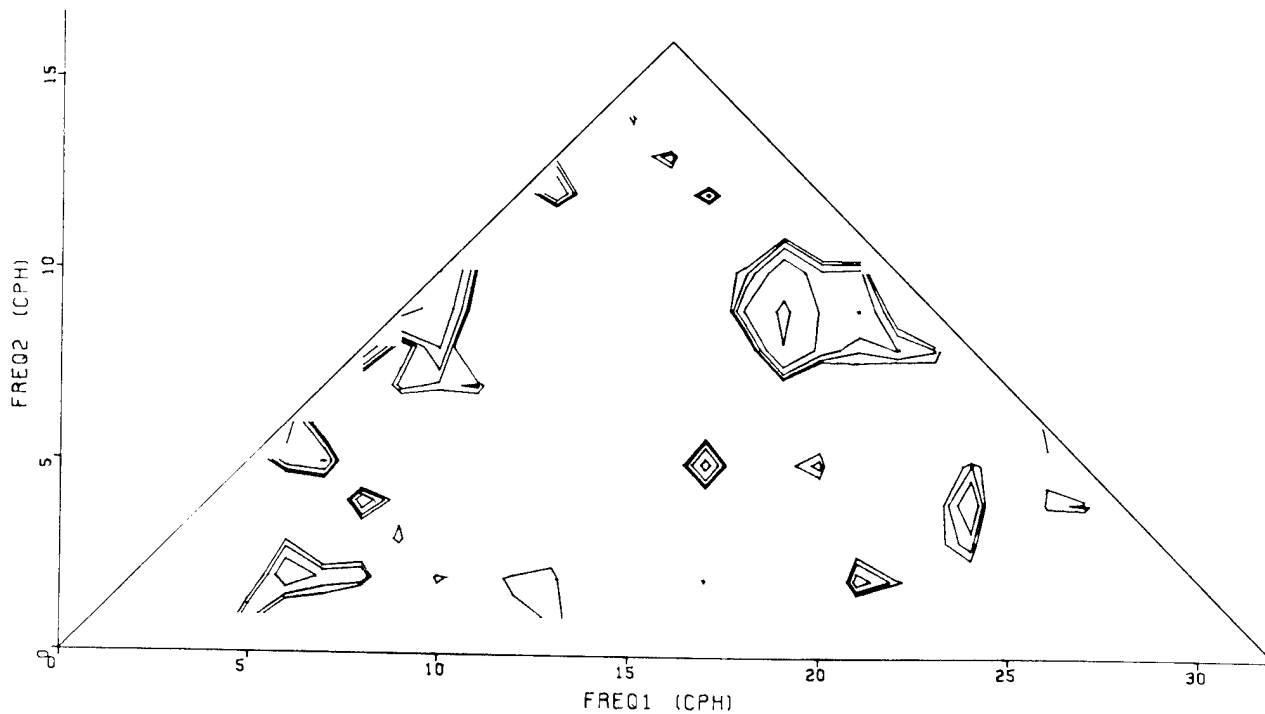


Figure 2

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:

FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*SUBSAMP 0*PC SIZE 64
 *NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

	0.430	80.0
TIME OF PLOT:	0.458	85.0
09:01 DEC 22, '77	0.479	90.0
	0.552	95.0
	0.667	99.0

C8+C16+C24 NYQUIST=32 HANNED
 FILE BVPL10 S/N=10

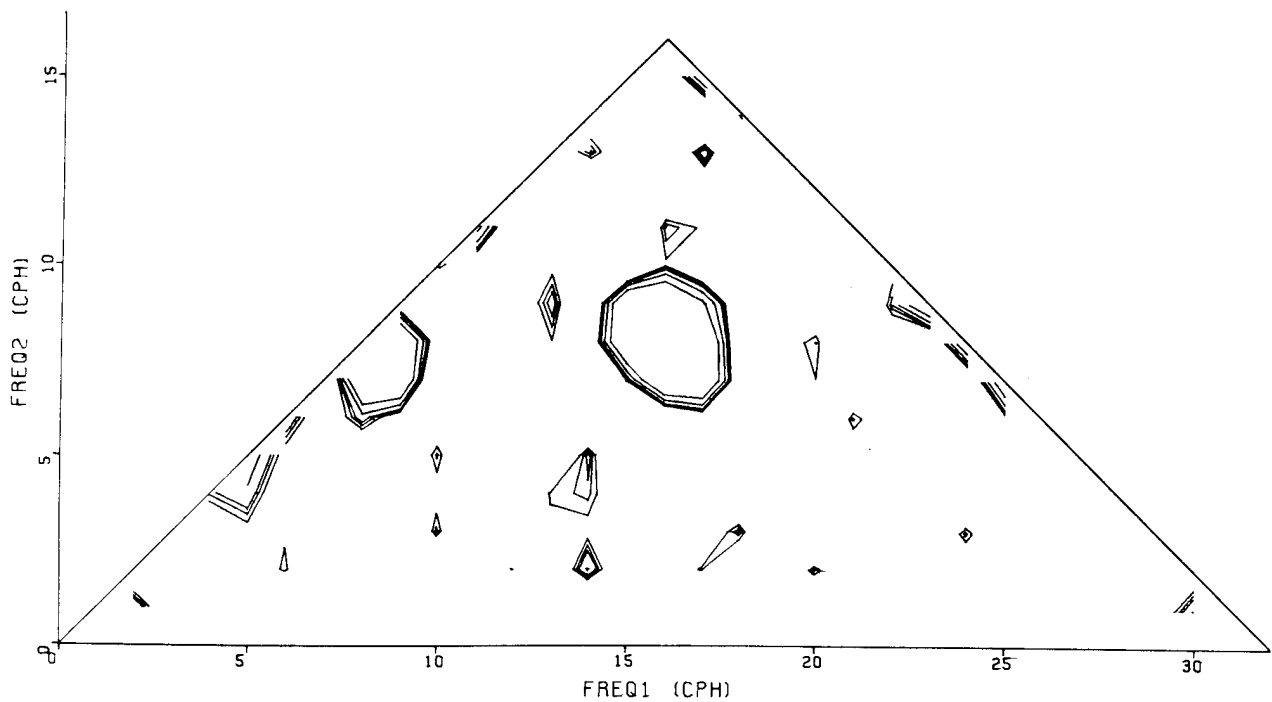


Figure 3

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENRANOUTPUT 0
 FREQUENCY 2: GENRANOUTPUT 0
 FREQUENCY 3: GENRANOUTPUT 0

PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
 *NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

	0.420	80.0
TIME OF PLOT:	0.447	85.0
09:06 DEC 22, '77	0.498	90.0
	0.591	95.0
	0.834	99.0

S8+S16+S24 NYQUIST=32 HANNED
 FILE BVPL11 S/N=10

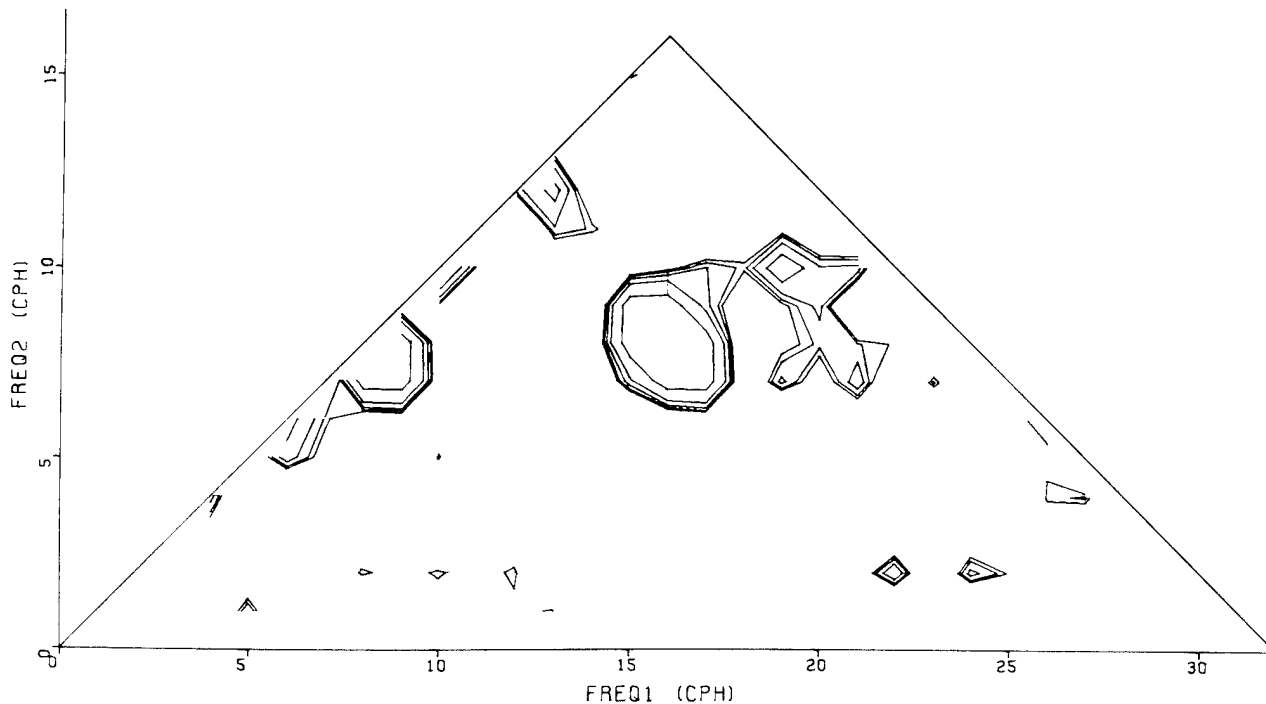


Figure 4

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENRANOUTPUT 0
 FREQUENCY 2: GENRANOUTPUT 0
 FREQUENCY 3: GENRANOUTPUT 0

PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
 *NO PCS 10*CLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT: 09:11 DEC 22, '77

0.420	80.0
0.447	85.0
0.498	90.0
0.591	95.0
0.834	99.0

-S8-S16-S24 NYQUIST=32 HANNED
 FILE BVPL12 S/N=10

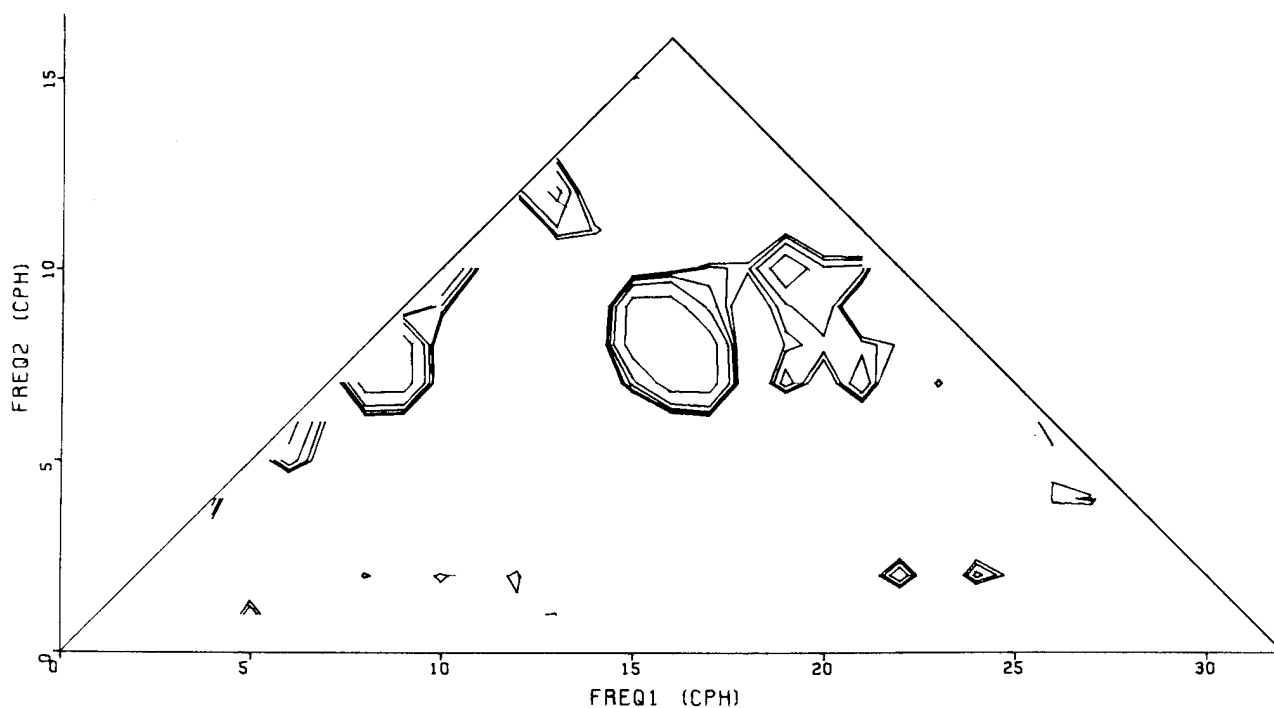


Figure 5

AUTO BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

	0.398	95.0
TIME OF PLOT:	0.489	99.0
15:58 MAR 07, '78	0.605	99.9
	0.720	100.0

G**2+10GSIN(16T)-50COS(32T)
MEAN REM'D OLAP HANNED 19PCS

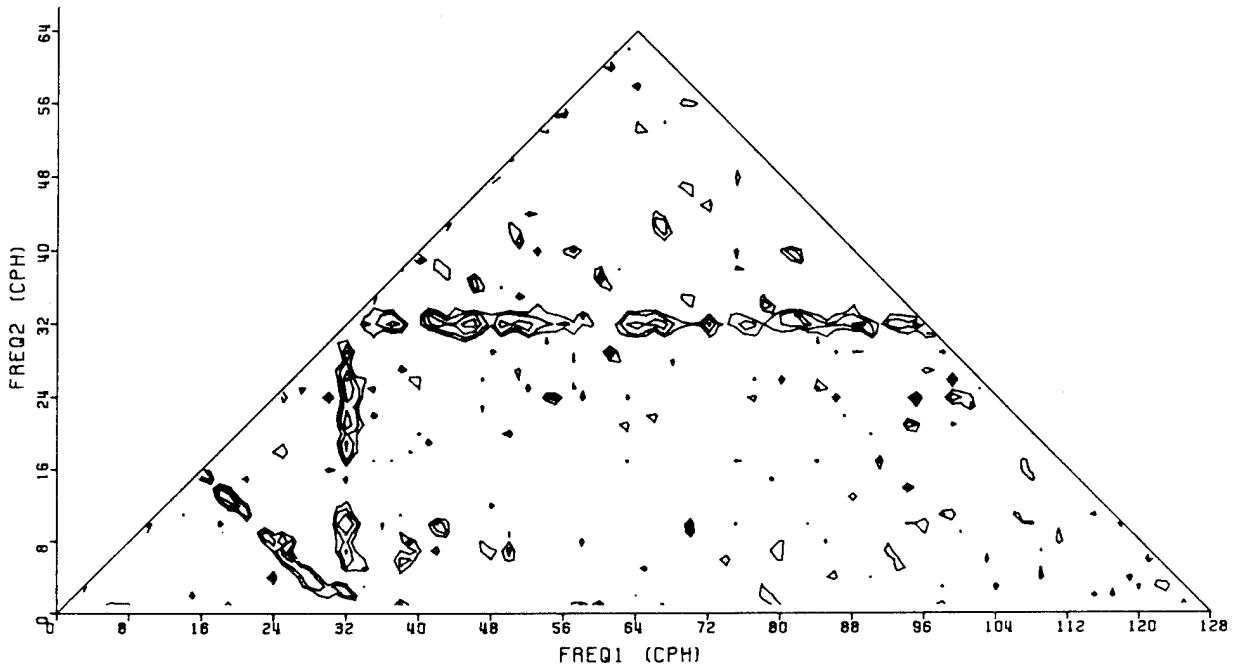


Figure 6

CROSS BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENRANOUTPUT 0
 FREQUENCY 2: GENRANOUTPUT 0
 FREQUENCY 3: GENRANOUTPUT 0

PROCESSING HISTORY

NO LF'S 4=NO NOS DATA 352=PAEW 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 84
 =NO PCS 10=GLAP 1=SUBT 0=HANN 1=CREATED 17:18 DEC 15, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.400 80.0
 0.431 85.0
 0.468 90.0
 0.551 95.0
 0.675 99.0

X=C8+C16+C24;Y=-S8-S16-S24;FORM XXY
 NYQ=32 HANNED S/N=10 FILE BVPL3

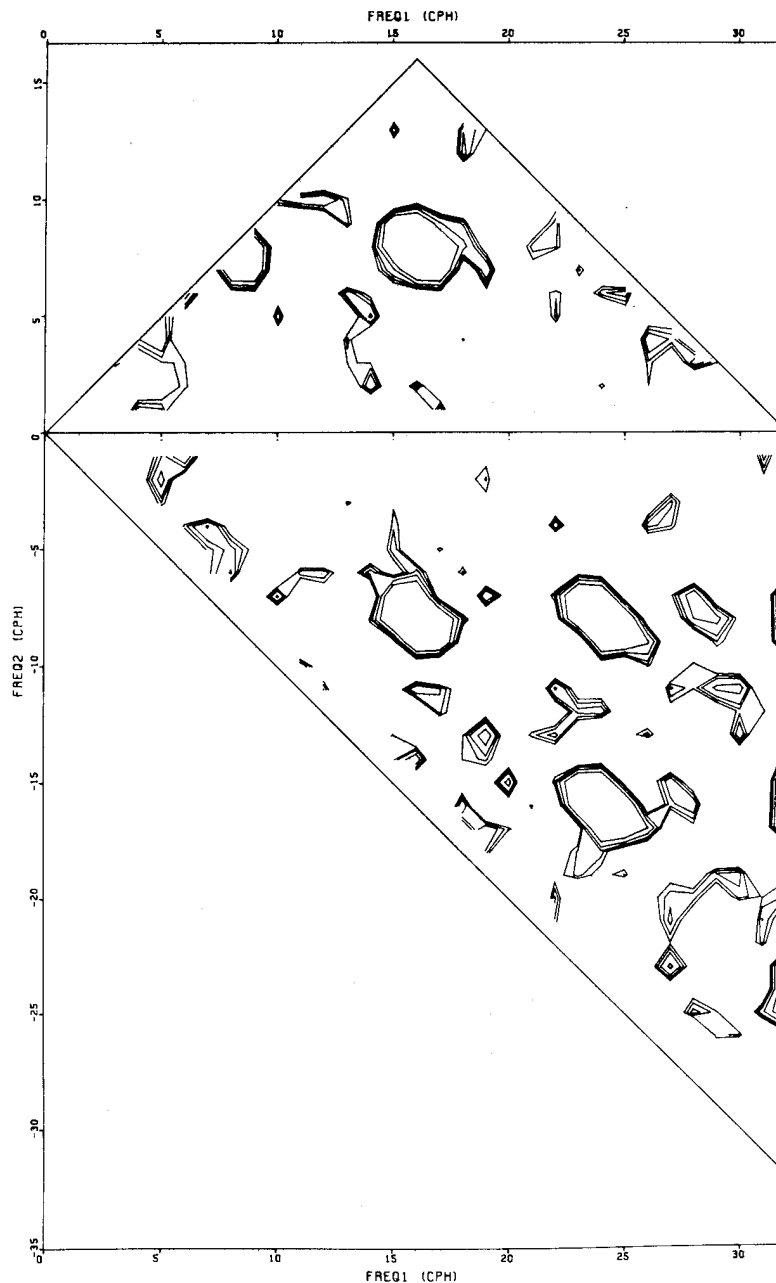


Figure 7

CROSS BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENRANOUTPUT 0
 FREQUENCY 2: GENRANOUTPUT 0
 FREQUENCY 3: GENRANOUTPUT 0

PROCESSING HISTORY

NO LF'S 0=NO WDS DATA 552=PREW 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 84
 *NO PCS 10=OLAP 1=ISUBT 0=HANN 1 *CREATED 18:20 DEC 21, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT:
 09:28 DEC 22, '77

0.400	80.0
0.431	85.0
0.468	90.0
0.551	95.0
0.675	99.0

X=C8+C16+C24; Y=-S8-S16-S24; FORM XXY
 NYQ=32 HANNED UNITIZED S/N=10

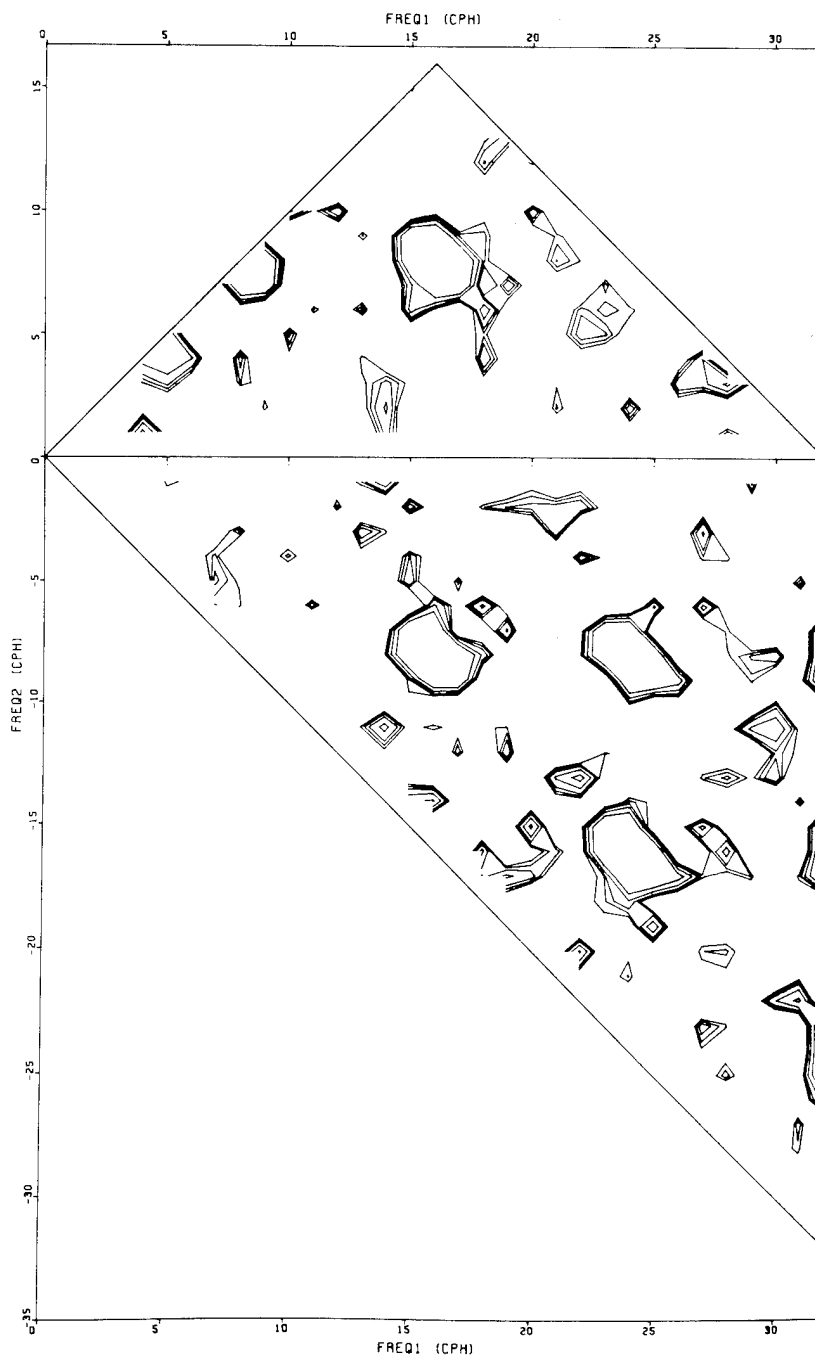


Figure 7a

CROSS BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENARMOUTPUT 0
 FREQUENCY 2: GENARMOUTPUT 0
 FREQUENCY 3: GENARMOUTPUT 0

PROCESSING HISTORY

NO L.F.'S 4=NO NOS DATA 352=PREV 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 64
 =NO PCS 10=BLAP 1=ISUBT 0=HANN 1 =CREATED 17:18 DEC 15, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.400 80.0
 0.431 85.0
 0.468 90.0
 0.551 95.0
 0.675 99.0

GAUSS NOISE HANNED MEAN=0 STD=1
 FORM XXY FILE BVPL1

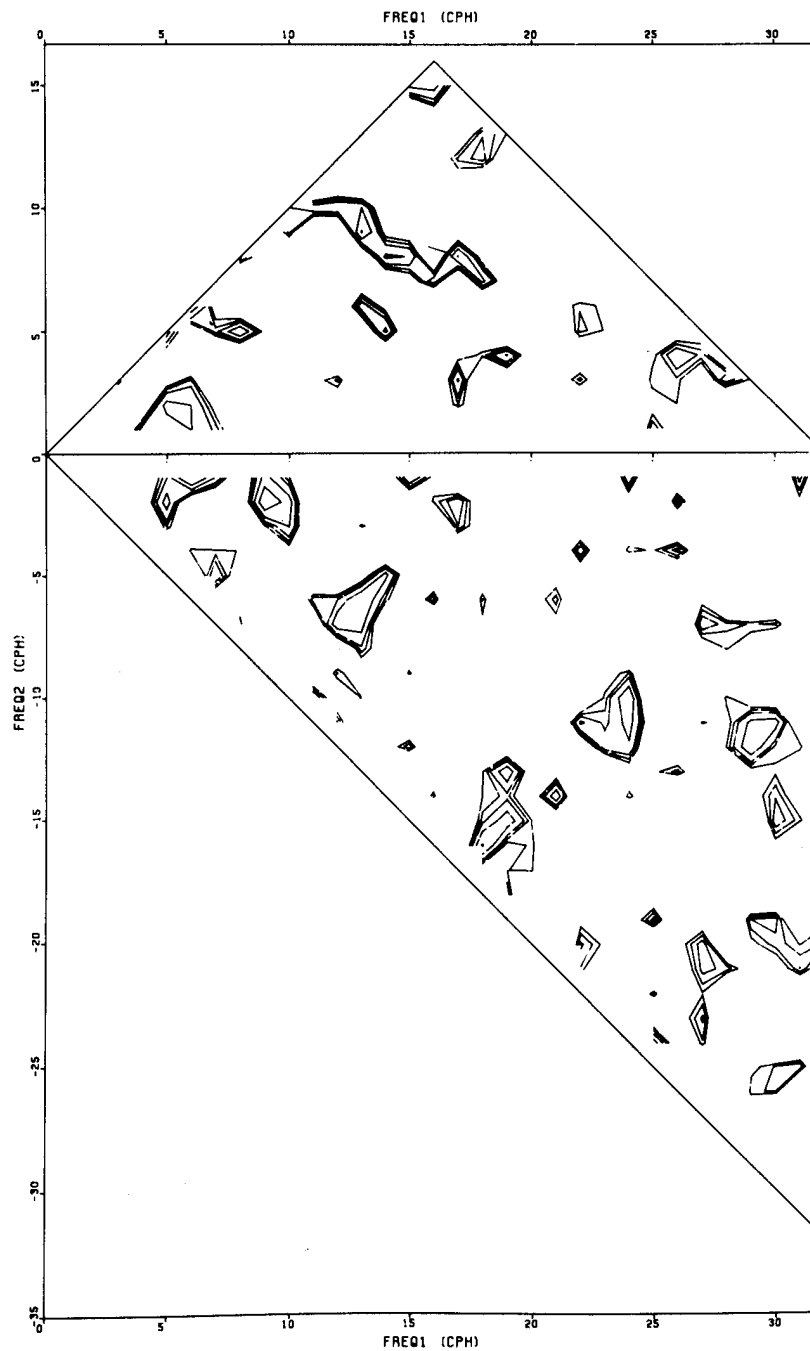


Figure 7b

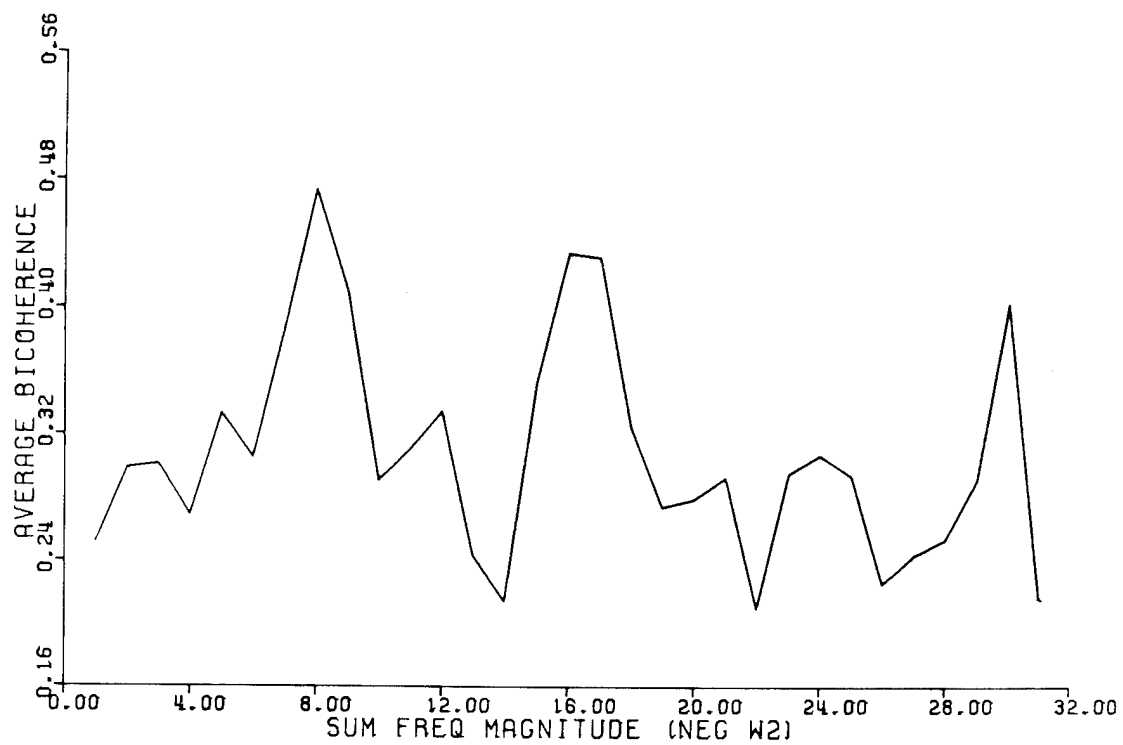
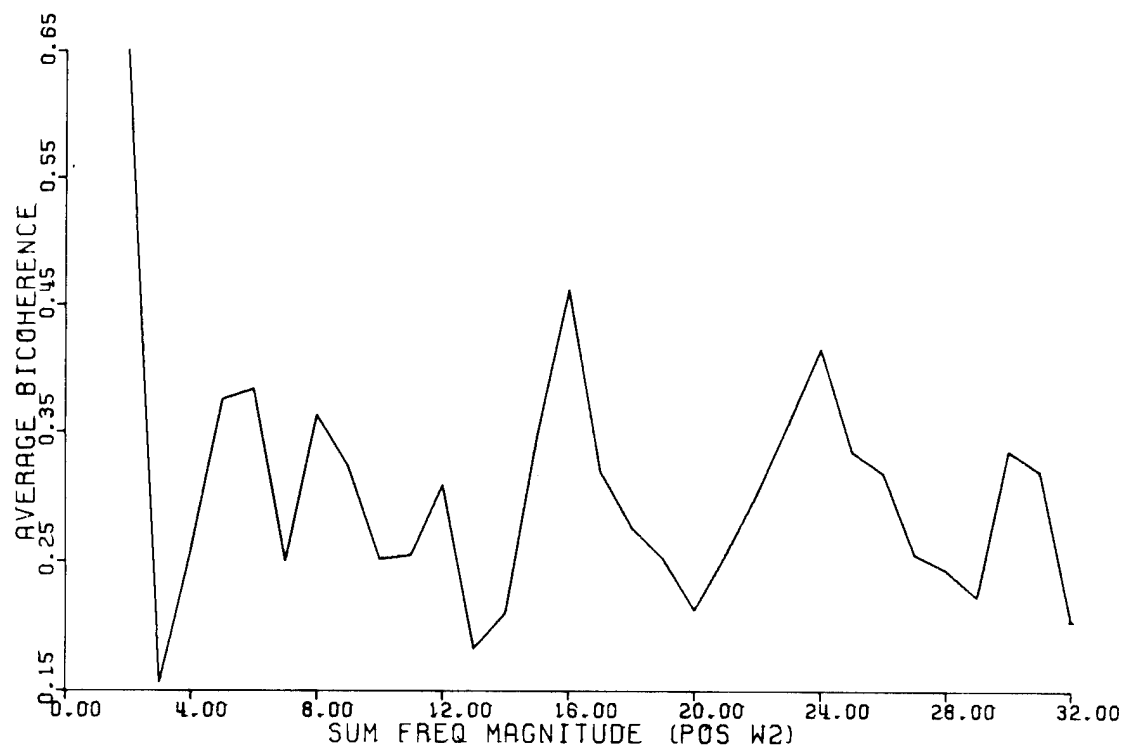


Figure 7c

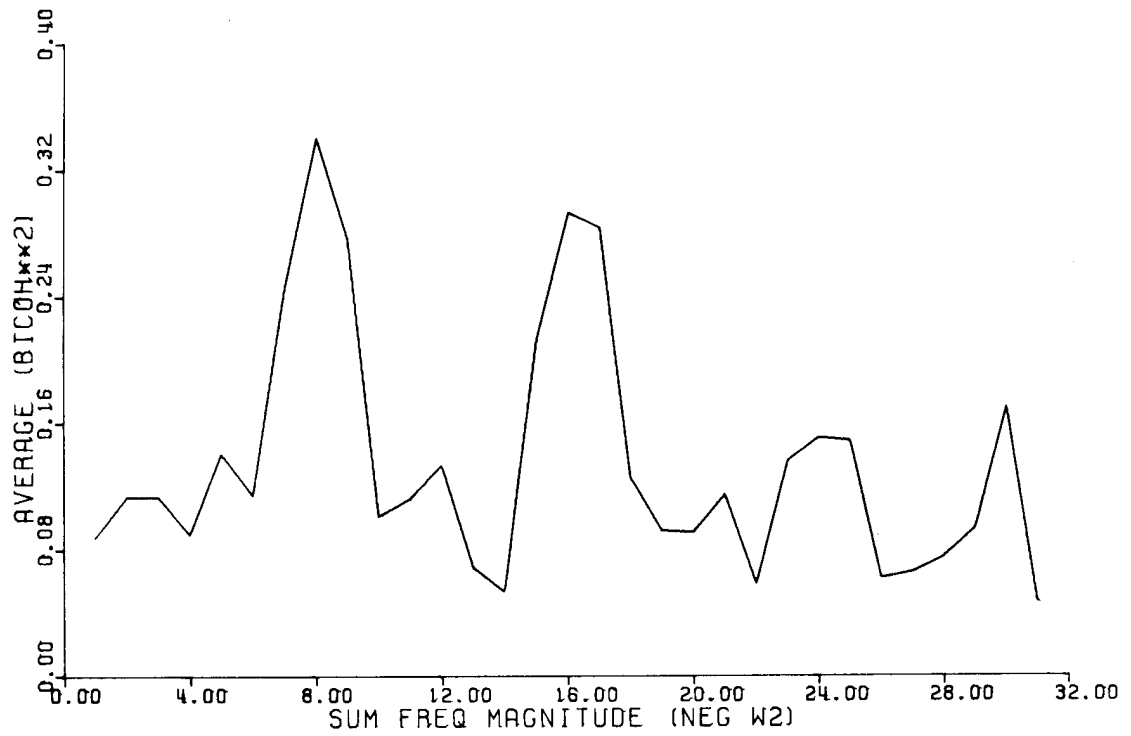
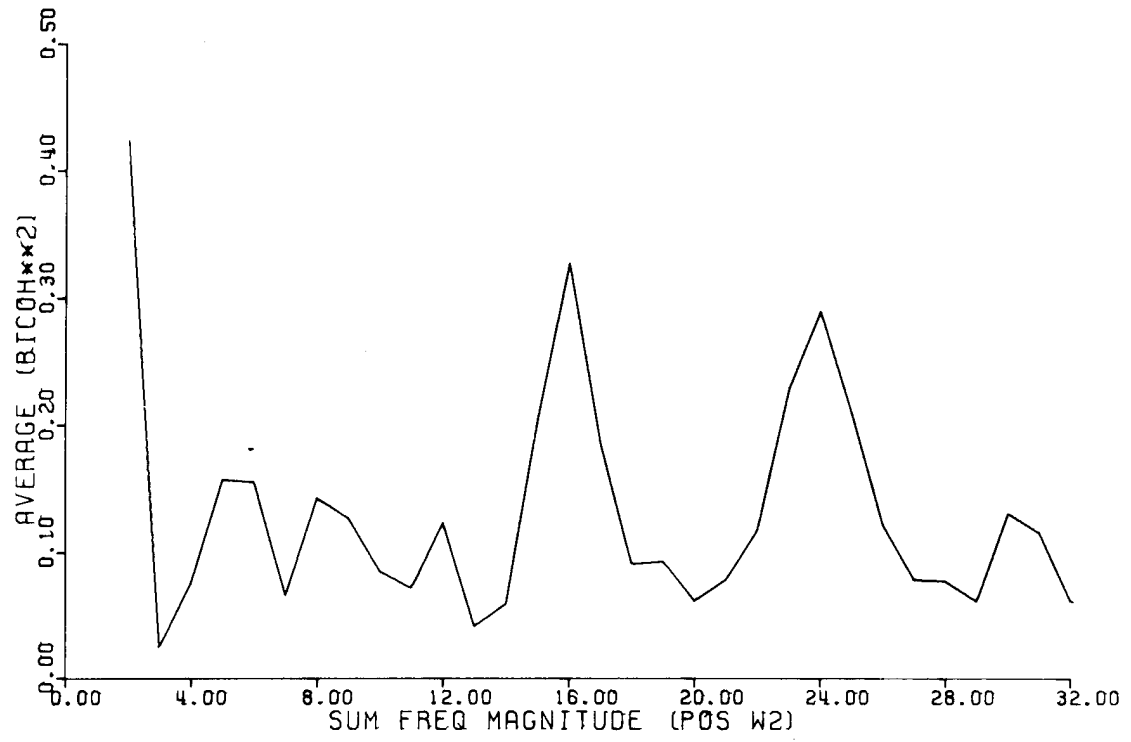


Figure 7d

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

GAUSSIAN NOISE MEAN=0. S.D.=1
YVEC,YV1,YV2,YVPL1

PROCESSING HISTORY

NO LF'S 11=NO WDS DATA 352=PREW 0=

SUBSAMP 0=NSUBSAMP 0=PC SIZE 64=NO PCS 10=

OLAP 1=ISUBT 0=HANN 1 =CREATED 14:28 DEC 07, '77

TIME OF PLOT:
14:52 DEC 07, '77

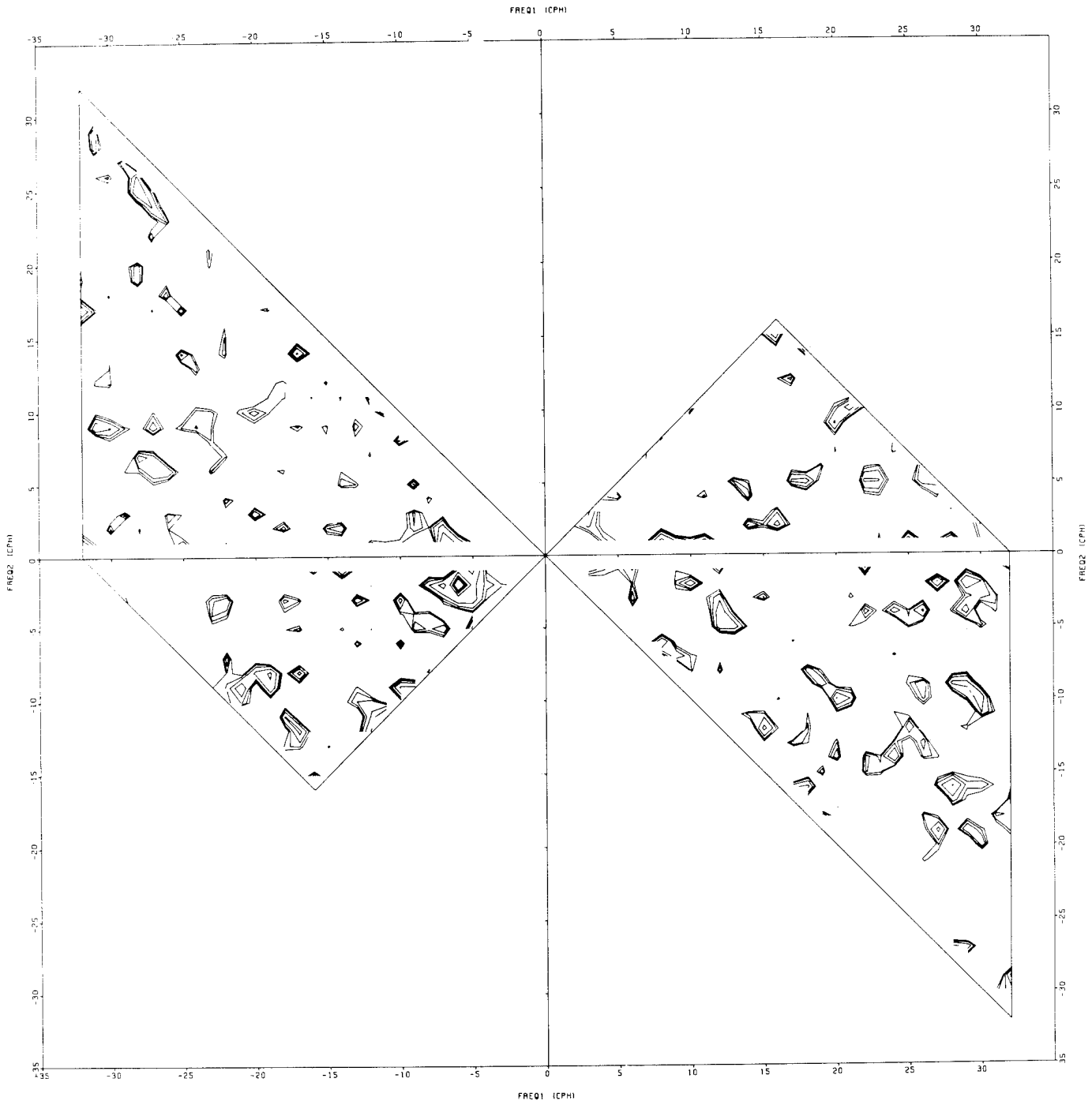


Figure 8

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.404	80.000
0.431	85.000
0.476	90.000
0.541	95.000
0.662	99.000

PROCESSING HISTORY

NO LF'S 11=NO WDS DATA 640=PREW 0=
 SUBSAMP 0=NSUBSAMP 0=PC SIZE 64=NO PCS 10=
 OLAP 0=ISUBT 0=HANN 0 =CREATED 14:29 DEC 07, '77

TIME OF PLOT:
 16:22 DEC 07, '77

GAUSSIAN NOISE MEAN=0 S.D.=1
 NOT HANNED

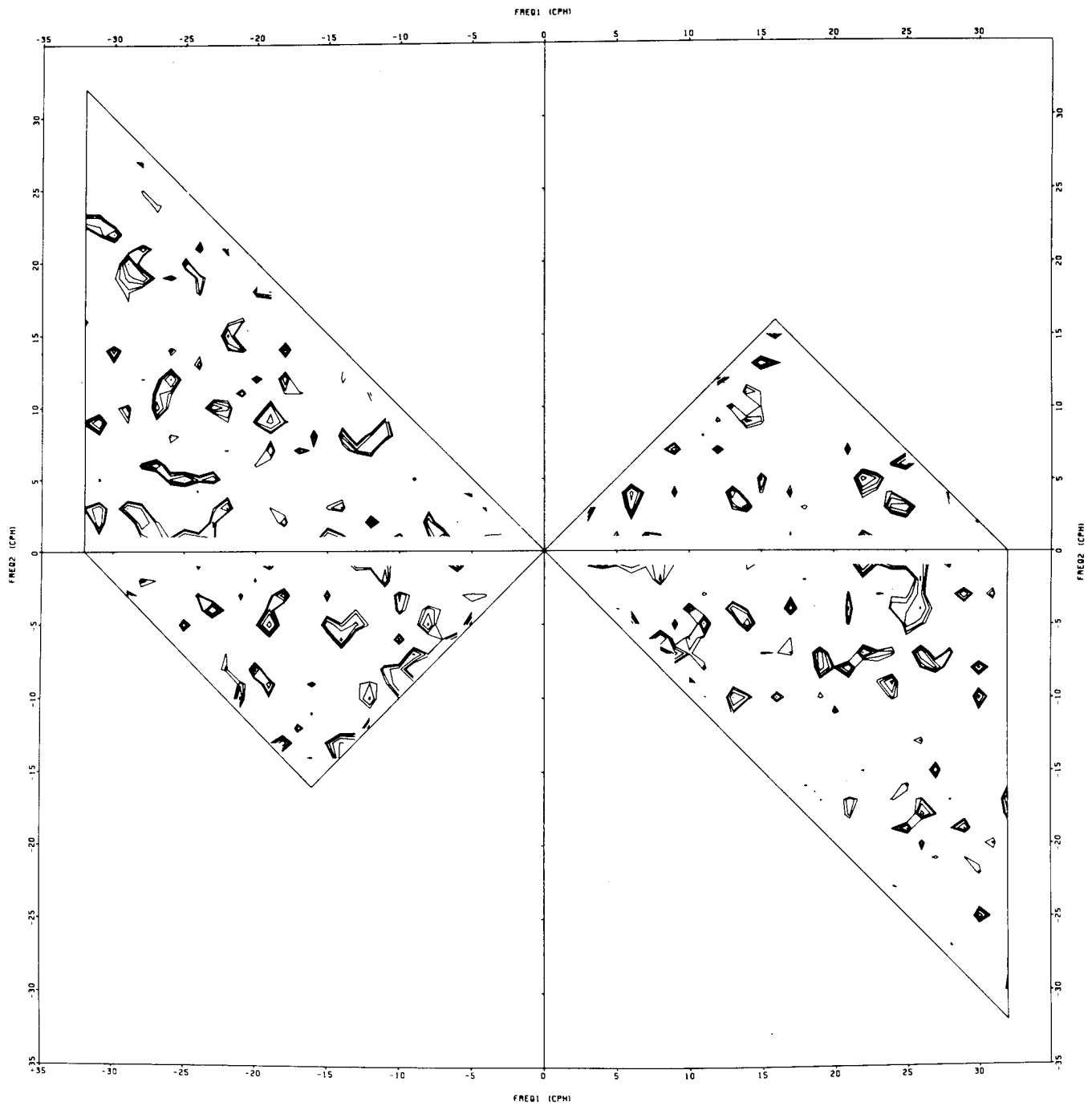


Figure 9

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 350*PREH 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 1*SUBT 0*HANN 1*CREATED 14:26 DEC 07, '77

TIME OF PLOT:
 15:43 DEC 07, '77

3 COUNTER ACT VEC'S S/N 10 HANNED
 FREQ 8.16,24 NYQUIST=32

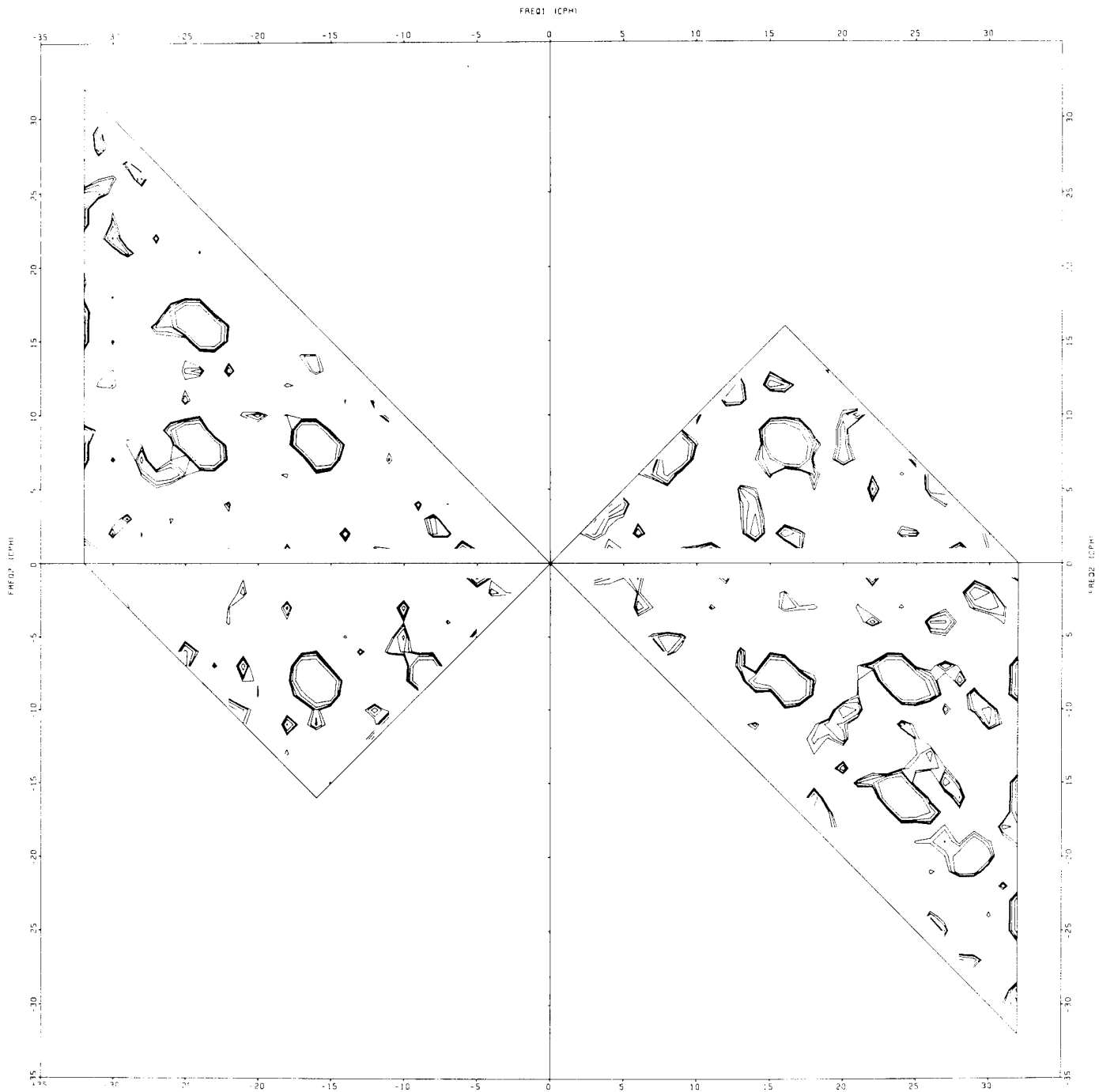


Figure 10

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 352*PREW 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*10 PCS 10*
 OLAP 1*ISUBT 0*HANN 1*CREATED 14:28 DEC 07, '77

3 COUNTER ROT VEC'S S/N=1 HANNED
 FREQ 8.16.24 NYQUIST=32

TIME OF PLOT:
 15:48 DEC 07, '77

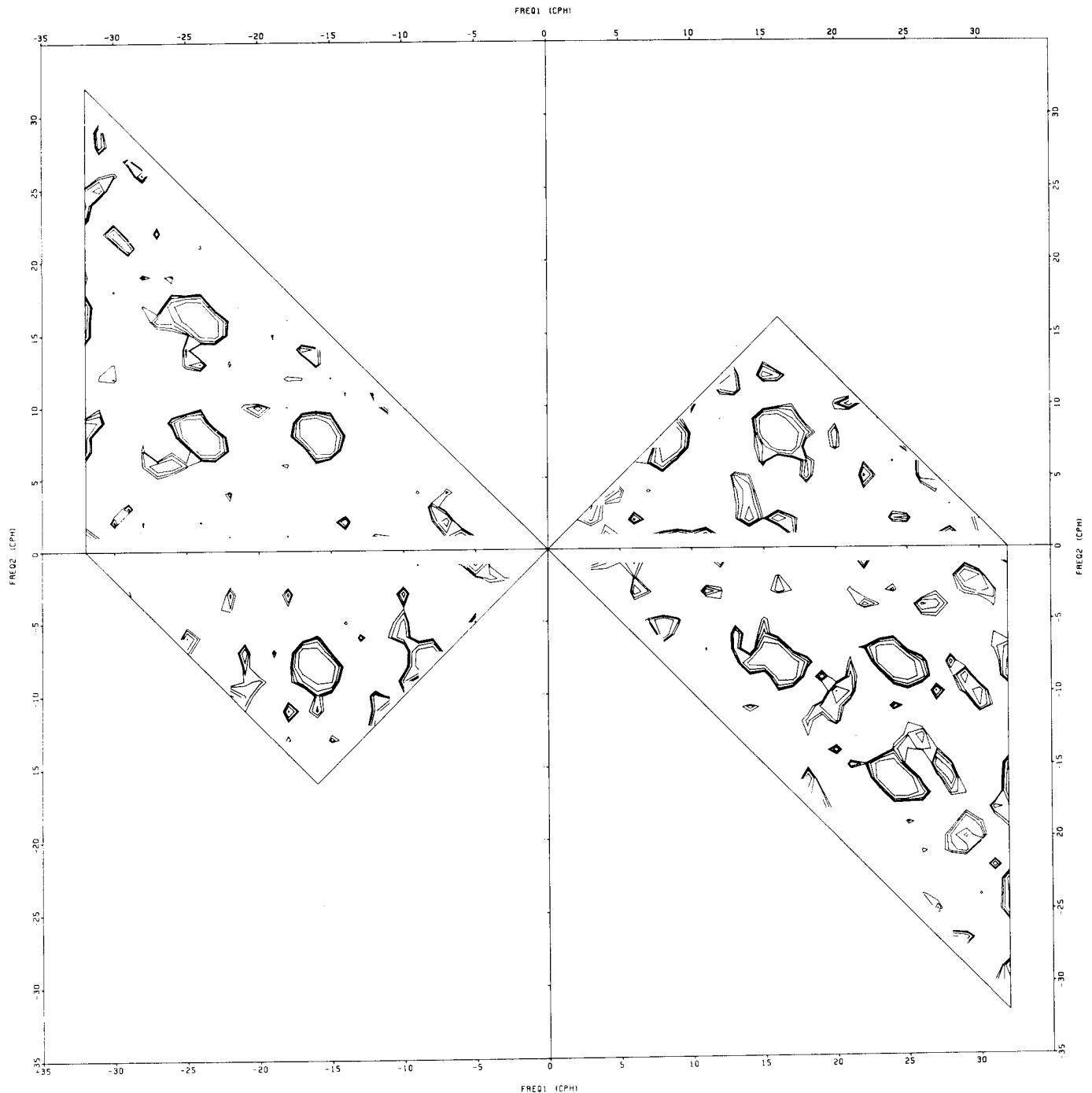


Figure 11

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

PROCESSING HISTORY

NO LF'S !! NO WDS DATA 352*PREW 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07, '77

3 COUNTER ROT VEC'S S-N=0.1 HANNED
 FREQ 8.16.24 NYQU:ST=32

TIME OF PLOT:
 15:55 DEC 07, '77

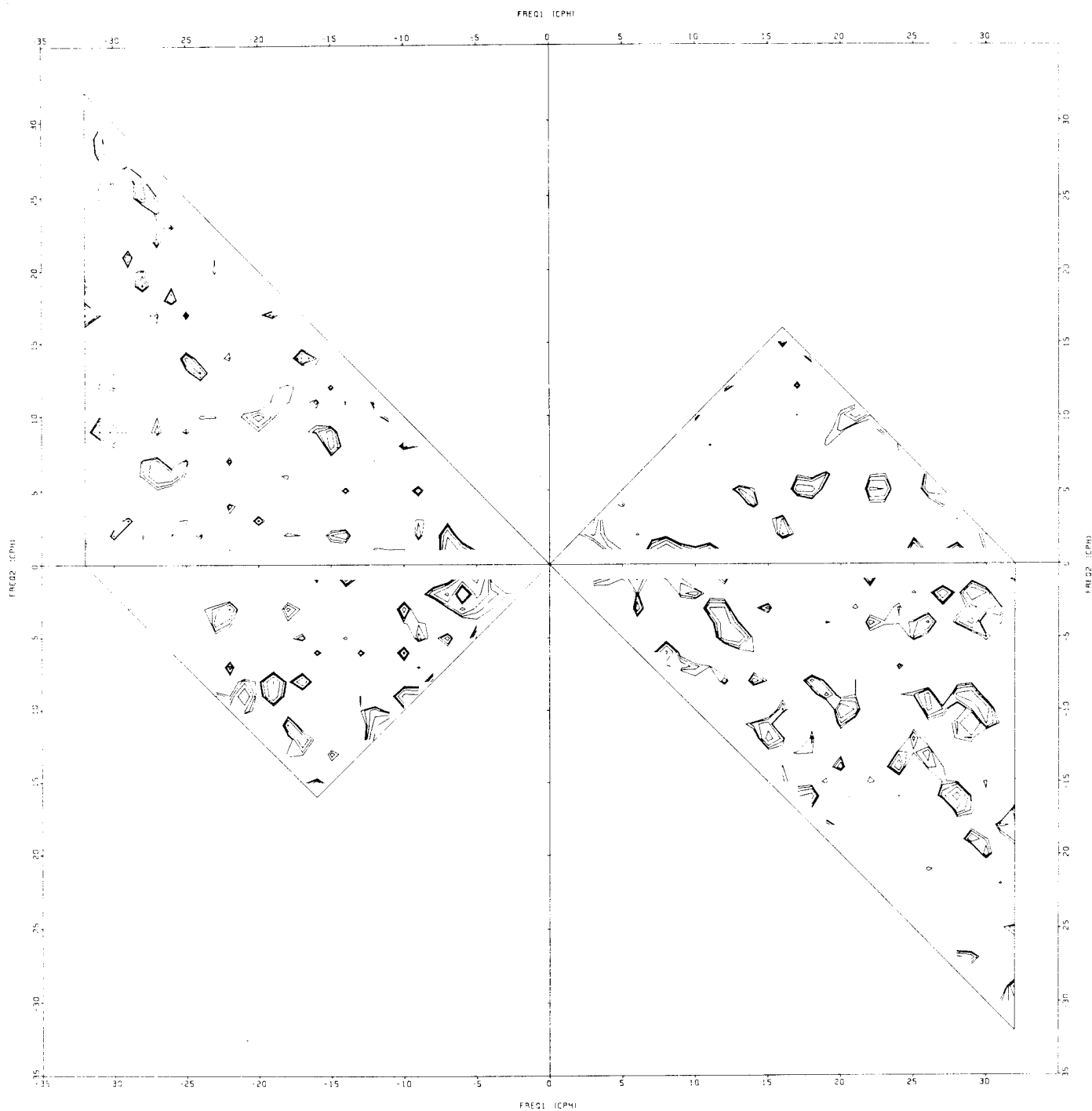


Figure 12

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.404	80.000
0.431	85.000
0.476	90.000
0.541	95.000
0.662	99.000

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 640*PREH 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 0*ISUBT 0*HANN 0 *CREATED 14:29 DEC 07, '77

TIME OF PLOT:
 14:47 DEC 08, '77

3 COUNTER ROT VEC'S S/N=10 NOT HND
 FREQ 8.16.24 NYQUIST=32

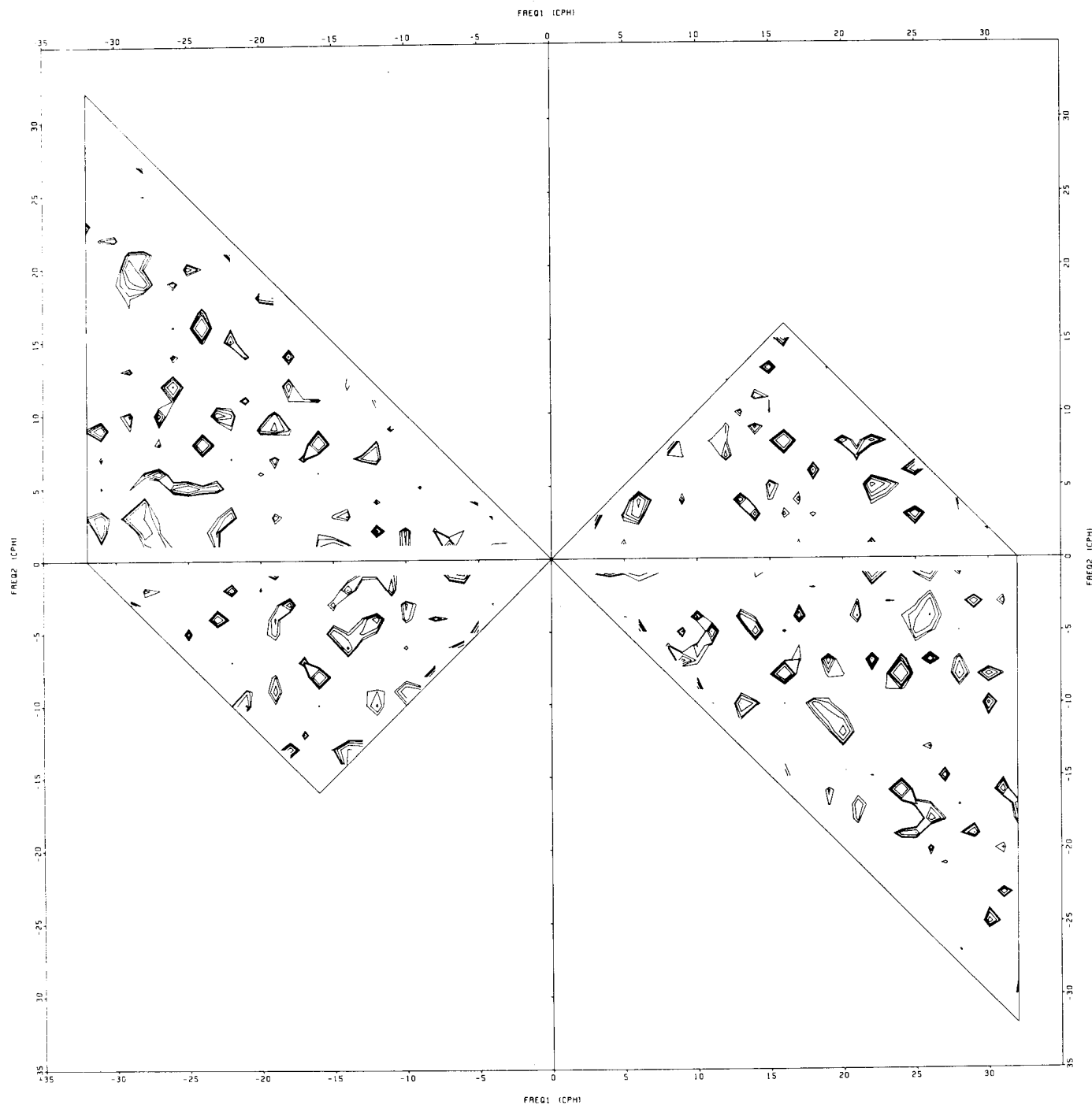


Figure 13

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

PROCESSING HISTORY

NO LF'S 4*NO WDS DATA 352*PREW C*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 1*ISUBT 0*HANN 1 *CREATED 17:18 DEC 15, '77

TIME OF PLOT:
 10:57 DEC 16, '77

3 CW VEC'S S/N=10 HANNED
 FREQ 8.16,24 NYQUIST=32

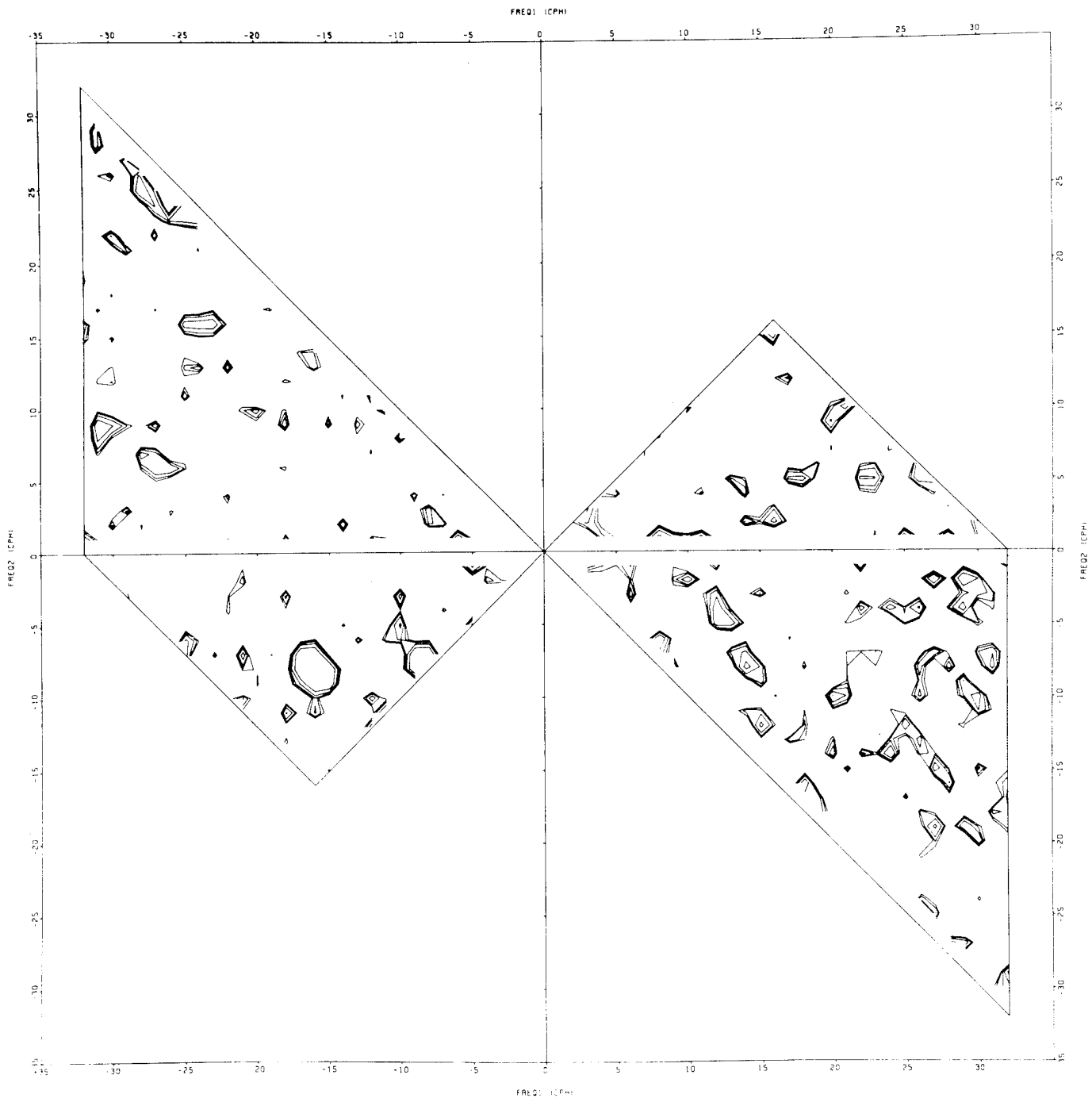


Figure 14

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 352*PREW 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07, '77

TIME OF PLOT:
 15:00 DEC 08, '77

3 CCW VEC'S S/N=10 HANNED
 FREQ 8,16,24 NYQUIST=32

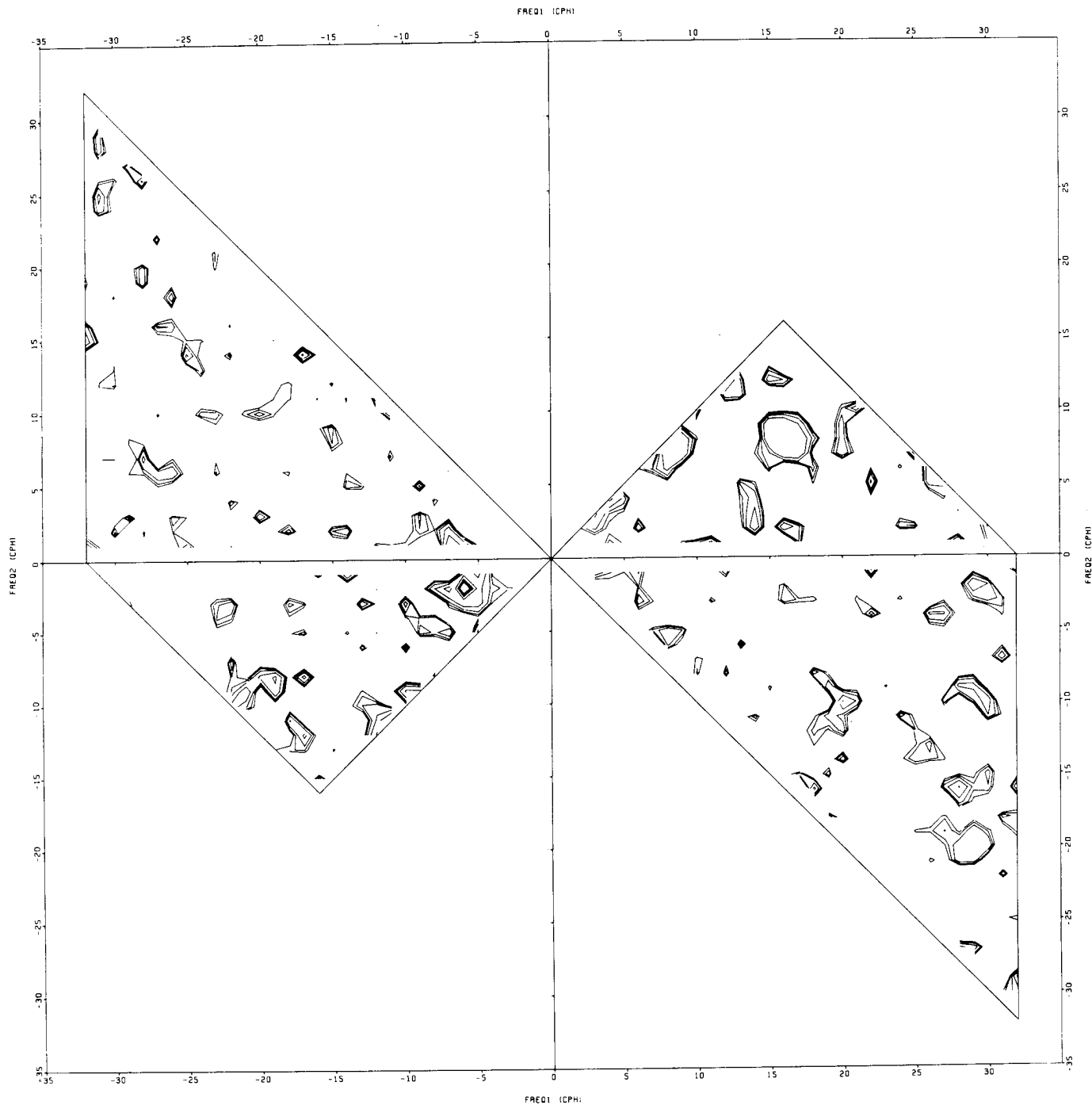


Figure 15

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.541	95.000
0.556	96.000
0.579	97.000
0.616	98.000
0.662	99.000

GAUSSIAN NOISE MEAN=0, S.D.=1
NOT HANNED

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 640*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 0*ISUBT 0*HANN 0 *CREATED 14:29 DEC 07, '77

TIME OF PLOT:
13:55 DEC 08, '77

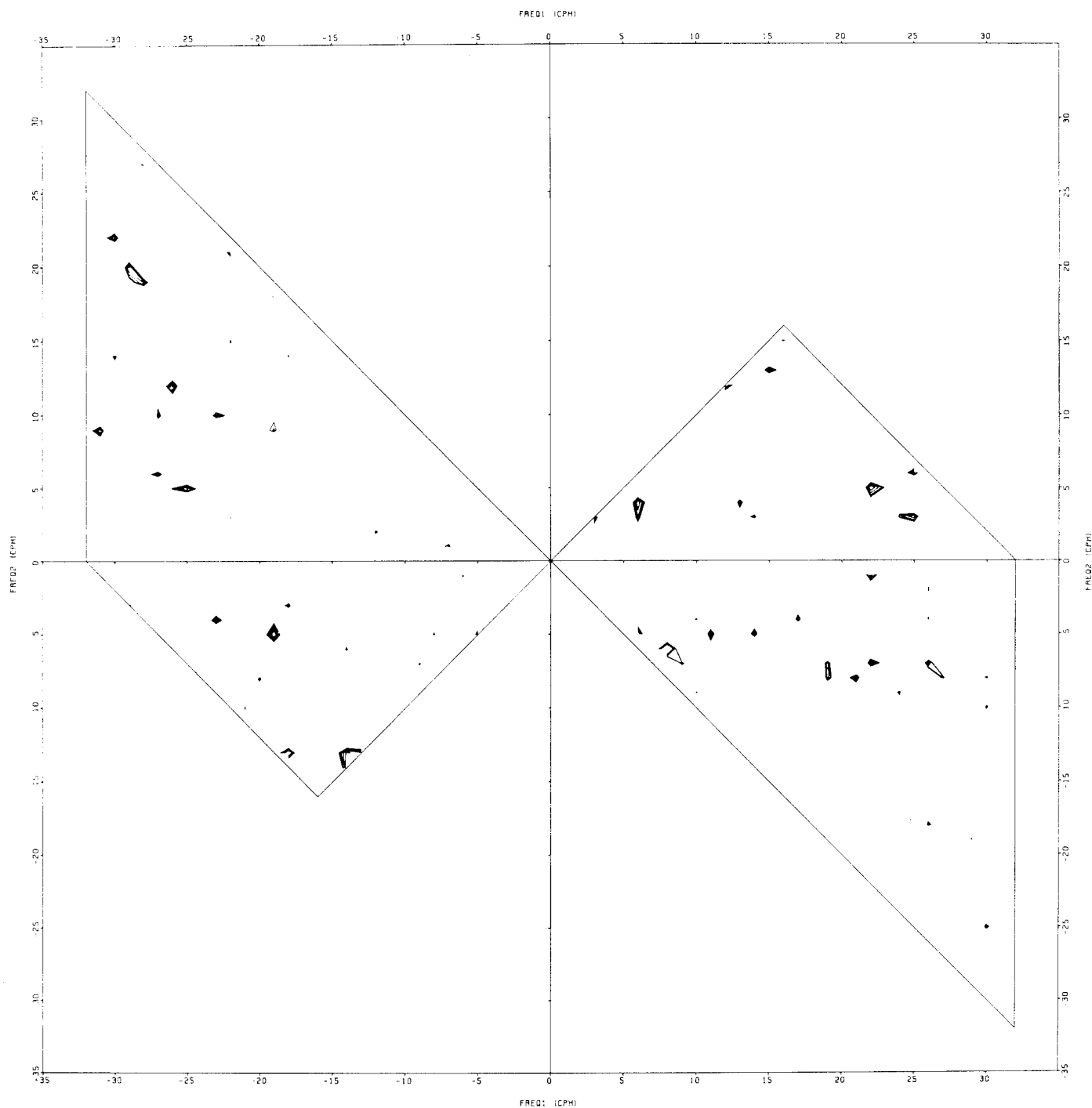


Figure 16

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.541	95.000
0.556	96.000
0.579	97.000
0.616	98.000
0.662	99.000

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 640*PREW 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 0*ISUBT 0*HANN 0 *CREATED 14:29 DEC 07, '77

3 COUNTER ROT VEC'S S/N=10 NOT HND
 FREQ 8.16.24 NYQUIST=32

TIME OF PLOT:
 13:50 DEC 08, '77

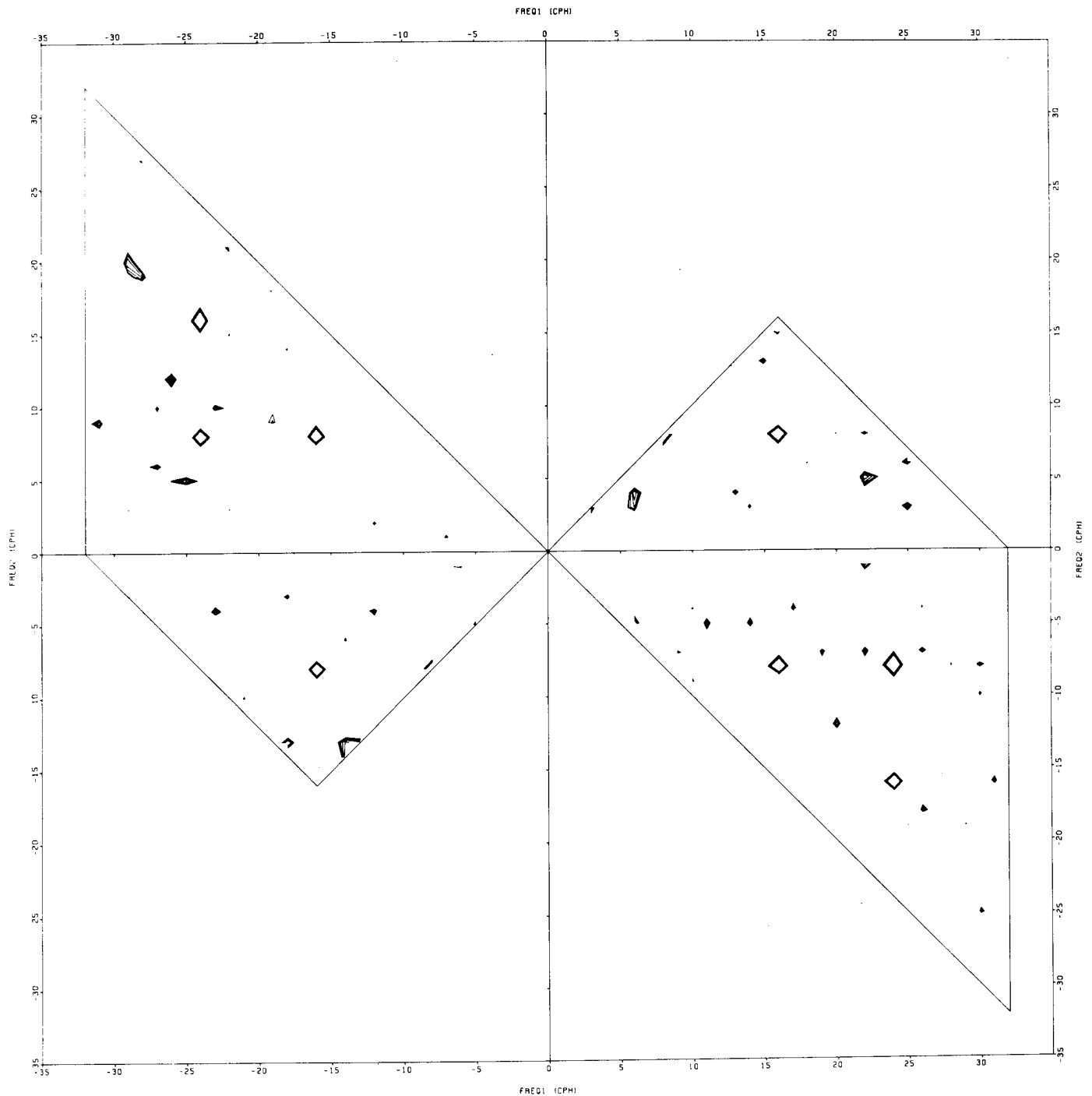


Figure 17

CROSS ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 352*PREW 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07, '77

TIME OF PLOT:
 16:14 DEC 07, '77

S/N=10 HANNED SER 1: -8, -16, 16
 SER 2: 8, 16, -24 NYQUIST=32

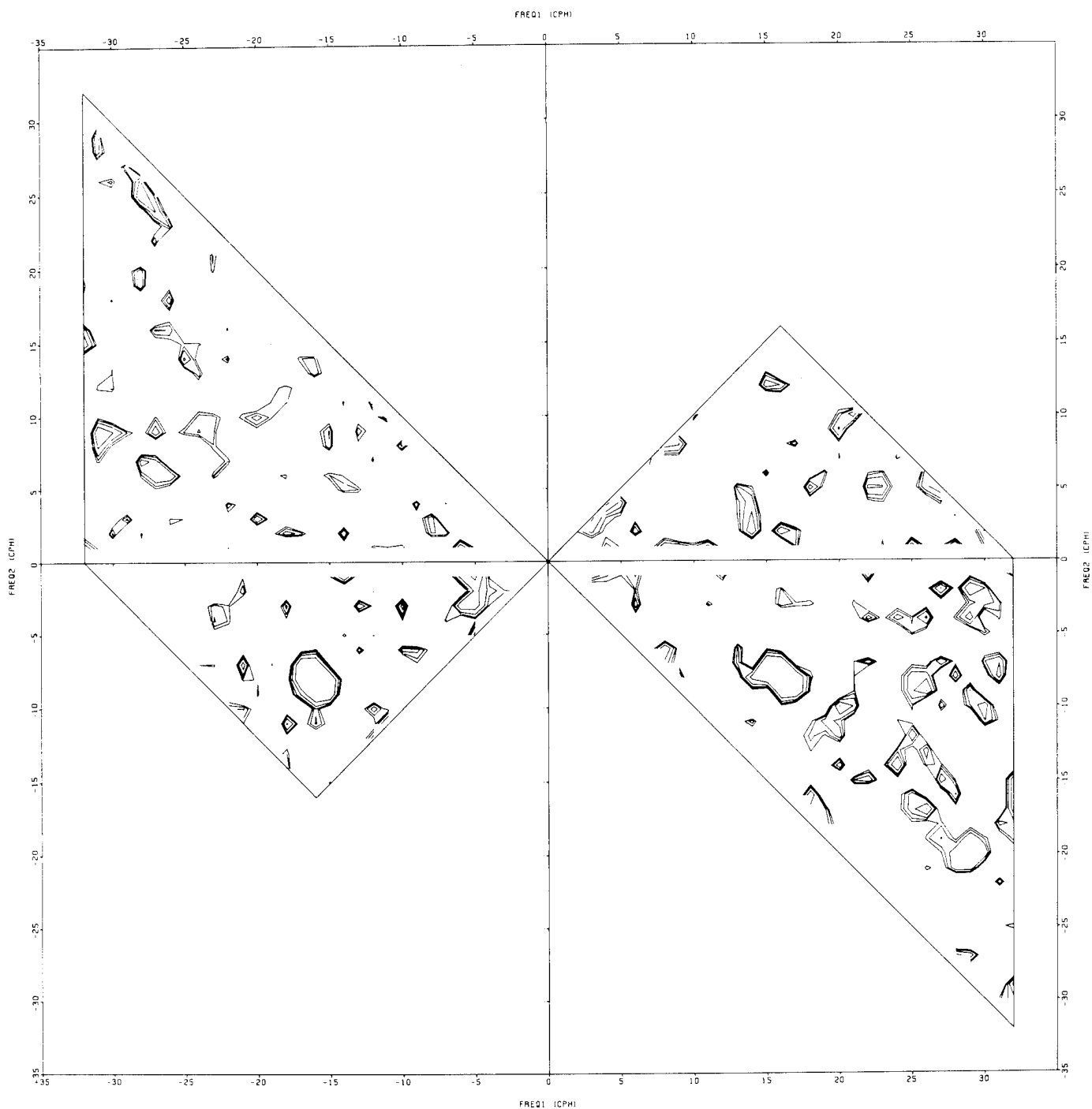


Figure 18

Confidence Levels Determined by Program BISCAL

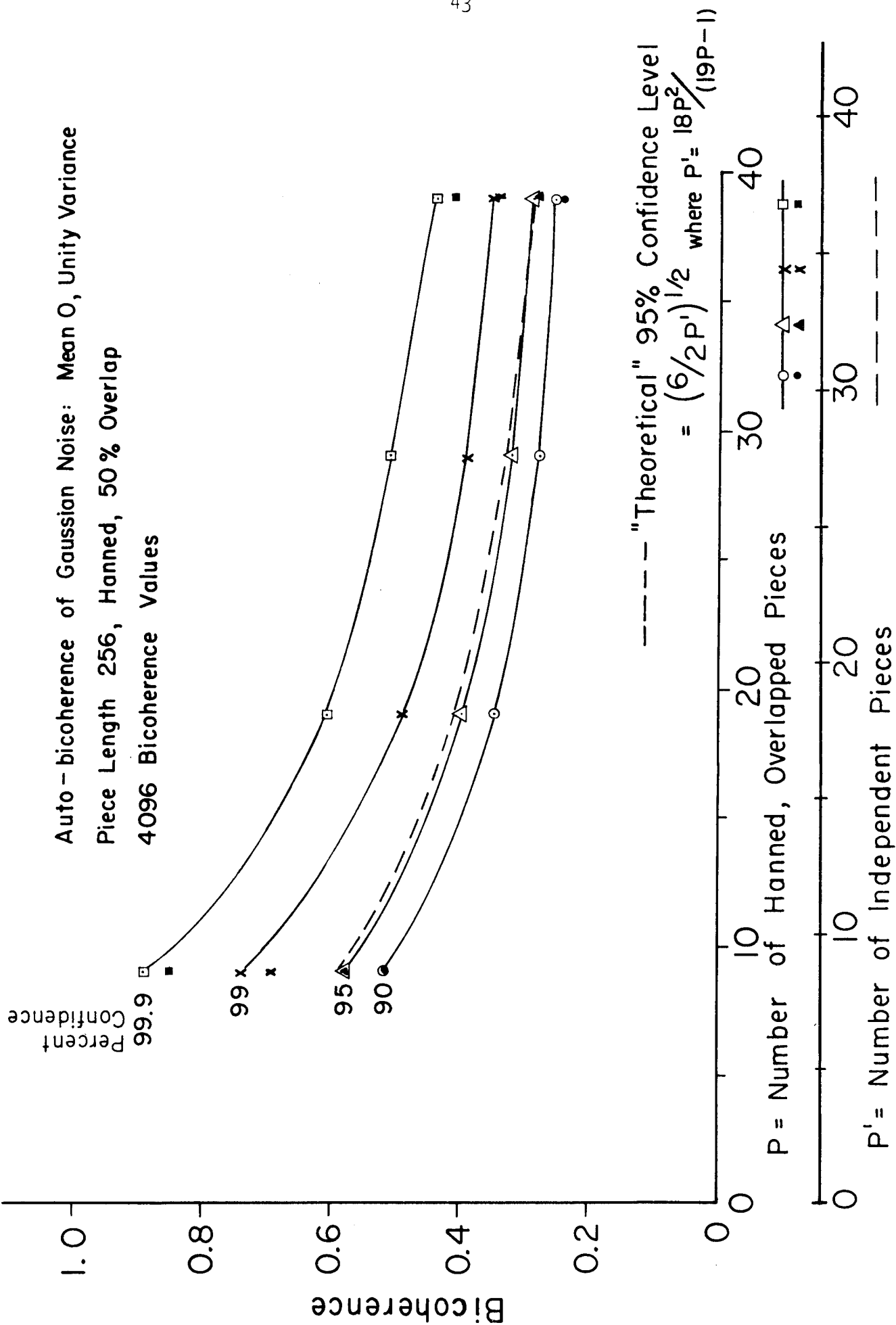


Figure 19

II. Bispectral Programs

A. Introduction To the Bispectral Program Reports

1. Summary

Following is a brief summary of the tasks performed by the programs described in Sections B, C, and D below. In reviewing this summary it will be helpful to refer to the flowchart presented immediately following the summary in Section II.A.2.

Building RWDISC Data File Preparatory to Fourier Transform (Section B)

- a. FRAGTAP; ORDAT: Two main programs, run consecutively from TAPDIS (RWDISC) file having multiple pseudo-files to a different RWDISC file having 21 logical files, the 21st of which is a facsimile of TAPDIS pseudo-file 1 (labels)
- b. GENRAN:* Main program which generates RWDISC file of series containing any combination of:
 - 1) Gaussian noise
 - 2) non-Gaussian noise of form:

$$ax^2 + bx^3$$

where x is Gaussian noise

- 3) signals of form:

$$\sum_{i=1}^4 A_i \sin(f_{i1}t + \phi_{i1}) \sin(f_{i2}t + \phi_{i2})$$

- 4) harmonics of form:

$$\sum_{i=1}^6 A_i \sin(ift + \phi_i)$$

Modifying RWDISC File from I by Replacing Data Series with Fourier Coefficients in a Special Order (Section C)

- a. FOURIER: Main program which does Fourier transform, including options of prewhitening, triangular weighting of consecutive subsamples, splitting into contiguous or overlapping pieces, subtraction of mean or trend, and Hanning.

Bispectral Calculations and Plotting (Section D)

- a. BISCAL: Works from specially organized RWDISC file produced by program FOURIER, to calculate auto and cross-bispectra of two real series, to write a disc file of bicoherences for transmittal to plotting program, BPLOT, and to compute confidence levels for the bicoherences.
- b. BPLOT: Works from disc file produced by programs BISCAL or GENRAN to plot auto and cross-bicoherences.
- c. BISUM: Works from RWDISC file produced by program FOURIER to integrate certain bispectral quantities along paths of constant sum frequency in the ω_1, ω_2 plane. Produces disc file containing integrals, as well as another disc file for input into plotting program BPLOT.
- d. BIVVEC: Works from RWDISC FILE produced by program FOURIER, to calculate auto and cross rotary bispectra of two vector series, to write disc files of rotary bicoherences for transmittal to plotting program RBPLOT, and to compute confidence levels for the rotary bicoherences.
- e. RBPLOT: Works from disc files produced by programs BIVVEC or GENRAN to make contour plots of auto and cross rotary bicoherences.

2. Flowchart for the Bispectral Programs

Sections IIB, C, D, and E contain reports and listings for nine programs connected with bispectral analysis. The following flowchart is presented as an aid to understanding the functions of the various programs, and should be consulted in reviewing the reports on them:

Section B:

Building RWDISC
Data File pre-
paratory to Fourier
Transform

FRAGTAP

for actual
Data Series

GENRAN

for arti-
ficial Data
Series

ORDAT

Section C:

Modifying RWDISC
File from B by
replacing Data
Series with
Fourier
coefficients

FOURIER

Section D:

Bispectral
Calculations
and
Plotting

BISCAL

BISUM

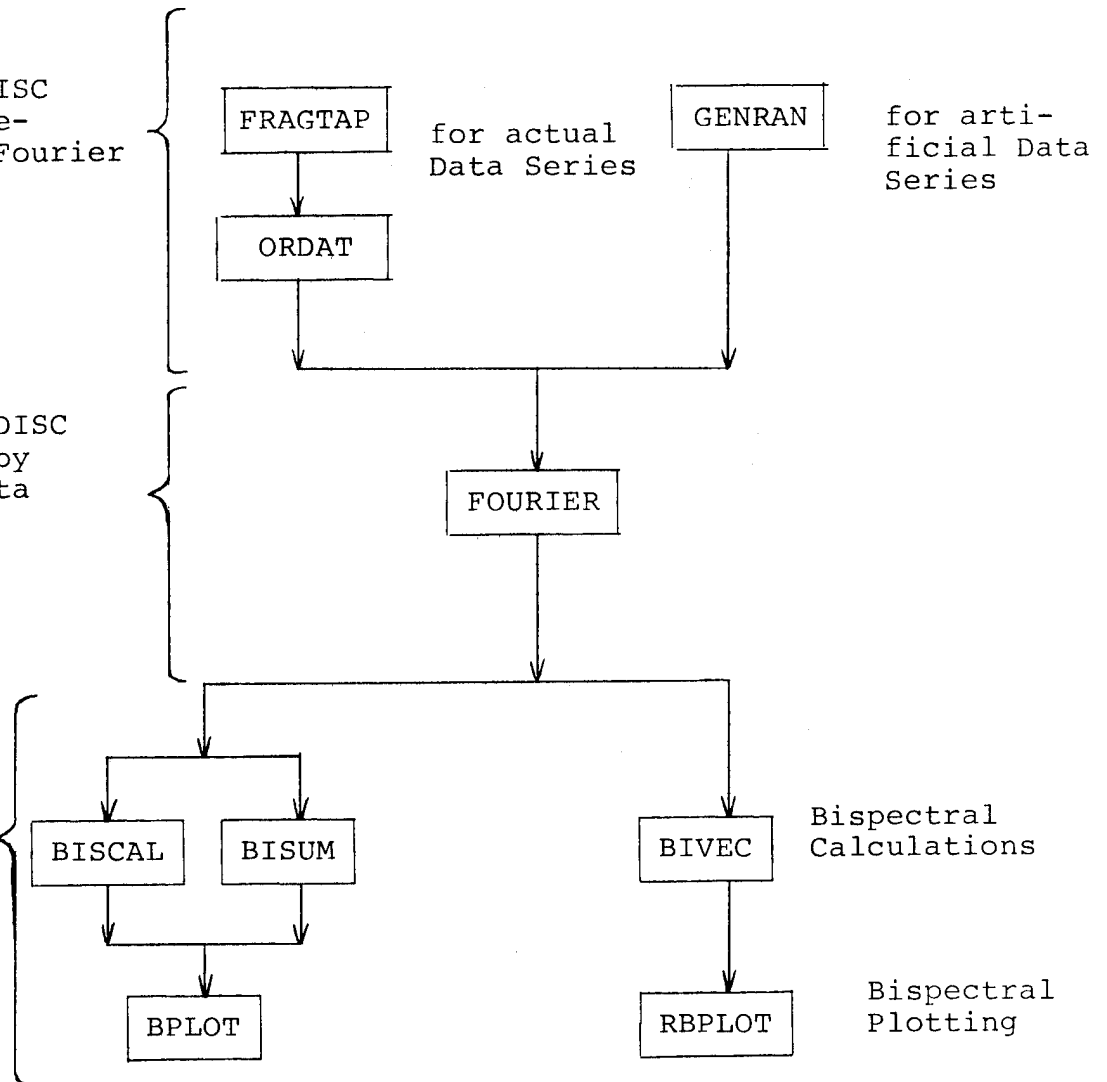
BPLOT

BIVEC

Bispectral
Calculations

RBPLOT

Bispectral
Plotting



3. Compatibility with Other Systems

The programs described in the reports and listings in Section II are all system-dependent, the system being the Xerox CP-V system on the XEROX SIGMA-7 computer at Woods Hole Oceanographic Institution (WHOI). This is particularly true of those programs which retrieve and organize the data: FRAGTAP, ORDAT, and FOURIER. If one were to attempt to adapt any of the programs described below to another system, the most tedious but straightforward task would be to convert the Xerox Extended Fortran-IV statements used by these programs to new Fortran-IV statements wherever necessary. A helpful discussion of differences between Xerox Extended Fortran-IV and ANS Standard Fortran is given in the reference manual for the former.¹ One would next have to turn attention to the subroutines used by these programs, and which reside in the system library at WHOI. These subroutines are not generally listed in the reports. It would be necessary to either rewrite the subroutines to be compatible with the new system, or replace them by equivalent subroutines. The following points would be helpful:

(a) Mention is often made of "RWDISC" data files throughout these reports. RWDISC is a machine-language program written at WHOI to provide users with a convenient method of using the disc as a true random-access file(s). It uses FORTRAN-callable entry points for reading, writing, and initializing, and it sets up a conceptually simple disc file organization upon which all random-access input and output in the following programs is based. One would probably not attempt to adapt RWDISC to another system, but rather exploit the conceptual simplicity of the disc file structure and FORTRAN calls by providing similar alternatives. The first step in doing this would be to consult the report on subroutine RWDISC.²

(b) FRAGTAP; ORDAT: Data are assumed to originate from an RWDISC-type disc file of a particular organization used by the Moored Array Project (the "Buoy Group") at WHOI. This is referred to in the report as a TAPDIS³ disc file. Programs FRAGTAP and ORDAT simply reorganize the data into another different type of RWDISC file to be used in all subsequent processing. These programs would probably be irrelevant to the problem of running the bispectral programs on another system.

(c) FOURIER: Besides RWDISC, the major subroutine used by this program is "HARML", which does the Fast Fourier Transform (FFT) on the data. This is a FORTRAN-IV subroutine

which resides in the library on the system disc at WHOI.³ Again, one could either adapt this program to another system, or provide an alternate subroutine for HARML. It is also possible to substitute an entire program for FOURIER, in order to supply Fourier coefficients from appropriately processed data, in the order stated in the report, and in the RWDISC-type disc organization required by the bispectral programs BISCAL, BISUM, and BIVEC. There are other minor subroutines used by program FOURIER, and these will be found in Reference 3.

(d) BISCAL, BISUM, BIVEC: These are the main bispectral programs. The input is an RWDISC file of Fourier coefficients as described in the respective reports, and output is one or more consecutive disc files. The RWDISC file has been discussed above. The binary WRITE statements used to create the consecutive files should not have to be modified in another system using FORTRAN IV. The subroutines other than RWDISC used by these programs are minor and can be found in Reference 3.

(e) BPLOT, RBPLOT: These programs make contour plots from consecutive files produced by programs BISCAL, BISUM, and BIVEC. These two plotting programs use a set of standard CALCOMP-plotter subroutines which should be available on other systems.⁴ These include the following:

PLOT (with entry point PLOTS)

SYMBOL

NUMBER

Additionally, programs BPLOT and RBPLOT use subroutines which reside in the system library at WHOI.³ These include:

WHCNTR, for generating contour plots.
Has entry points: DIMWH, GRID,
NOLABL, FLAGZ

AXDRAW }
AXWJS } used to draw axes via CALCOMP sub-
routine calls

- plus some minor programs from the WHOI library.³

The binary READ statements used for input should not require any change.

(f) GENRAN: This utility program to generate artificial series is intended to be run on-line in the context of the XEROX CP-V time-sharing system. It can possibly be adapted to other time-sharing systems if the required subroutines or substitutes are supplied. Program GENRAN again uses subroutine RWDISC to create a disc file organized in the RWDISC format. The only other subroutine it uses, from the WHOI system library, is subroutine NORAN, which generates pseudo-random numbers which can have a Gaussian distribution of specified mean and variance.³

4. References for Section IIA:

1. "Xerox Extended FORTRAN IV" Language Reference Manual, Xerox Corporation Publication No. 90/09/56E.
2. "Handbook for Computer Users," Woods Hole Oceanographic Institution, pp. V-E-1 ff.
3. "Reprints of WHOI Programs," Information Processing Center, Woods Hole Oceanographic Institution.
4. "Handbook for Computer Users," Woods Hole Oceanographic Institution, pp. V-E-36 ff.

B. BUILDING RWDISC DATA FILE PREPARATORY TO FOURIER TRANSFORM

NAMES: FRAGTAP; ORDAT

PURPOSE: To transfer data series from one or more TAPDIS¹ disc files to another RWDISC² random access file of different organization, for use in bispectral programs.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Programs FRAGTAP and ORDAT are two main programs which are intended to be run consecutively in the same job. Their purpose is to create an RWDISC² random access file which has a different organization from the input TAPDIS file(s).¹ The major reason for doing this is that the bispectral programs which use the data tend to repeatedly access long series sequentially from the disc. The segment length into which the logical files of the new RWDISC file are divided is therefore set at 512 (the size of the monitor buffer used by RWDISC), rather than 73 in the TAPDIS file. Hence there is one disc access per 512 words rather than one per 73 as there would otherwise be. Other reasons for reorganizing the TAPDIS files are to give the user the opportunity to select from several TAPDIS files only those series of interest, and hence produce an output file which is no longer than necessary, and to simplify the structure of the RWDISC files to be input into the bispectral programs.

The reason for two main programs is to allow the opening of a newly-structured RWDISC file. Program FRAGTAP creates up to 10 consecutive files from the TAPDIS file which it reads. There is one consecutive file per TAPDIS pseudo-file¹ and an additional consecutive file for the TAPDIS label logical file. Any logical files from any TAPDIS pseudo-file may be selected for transmittal. Program FRAGTAP may be run repeatedly on several TAPDIS files to generate more consecutive files. Program ORDAT reads any or all of these consecutive files, and places the contents into logical files of the user's choice in the new RWDISC file.

INPUT: (FRAGTAP-ORDAT sequence considered as a whole) Input is one or more TAPDIS files.

OUTPUT:I. Disc

Upon completion of the FRAGTAP-ORDAT sequence, disc output consists of a single RWDISC file having 21 logical files. The 21st logical file is reserved as a facsimile of one of the TAPDIS label files read by FRAGTAP. There is one data series per logical file in logical files 1 through 20 if all are used. It is not necessary to have 20 data series in an RWDISC file having 21 logical files. Data begins at location 5, the first three words being the Buoy-format tape* data file name from which this series originated, and the fourth word being the Buoy-format variable number.

Upon completion of the FRAGTAP program alone, there are up to ten consecutive files. One of these (normally with DCB assignment 10) is a facsimile of the TAPDIS label logical file (no. 1). The others each contain all series in sequence, called for by the user, from one TAPDIS pseudo-file, associated with one Buoy-format tape file. The data start in location 10. Locations 1 through 9 contain:

- #1,2,3: Buoy tape file name associated with this pseudo-file.
- #4: Number of variables read from the TAPDIS pseudo-file.
- #5: Number of cycles for this TAPDIS pseudo-file (from TAPDIS label).
- #6,7,8,9: Buoy format variable numbers for up to four data series.

II. Line Printer

Line printer output from a typical FRAGTAP-ORDAT sequence is shown in Appendix 1. In this job, 20 data files were transferred from a single TAPDIS file. Input and output originated and terminated on labeled tape, and the programs worked from the scratch pack.

FRAGTAP output

- (1) facsimile of input data cards;
- (2) starting locations of each successive pseudo-file in TAPDIS file;
- (3) summary of contents of each consecutive file produced.

*This is a name used for a standard format originated and used by the Moored Array Project at Woods Hole Oceanographic Institution. See Reference 3.

ORDAT output

ORDAT furnishes a complete summary of the contents and disposition of each input consecutive file. This includes the original Buoy tape file name associated with the TAPDIS pseudo-file, the number of series transmitted from that pseudo-file, the data series length (excluding label) for that pseudo-file, the logical files (destination) of each series, and the Buoy variable numbers associated with each data series transmitted.

USAGE:

The user must first determine whether only the public (RAD) disc pack will be used, or whether he will ask for a private, scratch pack. This will be determined by the amount of data to be handled. When the scratch pack is used, labeled tape will probably also be used because of the size of the files involved. Two versions of the control card deck will be presented. The first assumes that the TAPDIS file exists on the RAD, that the final RWDISC file will be on the RAD; and that only this disc pack is used throughout. The second assumes that the RWDISC file is originally on labeled tape, that the programs will work from the scratch pack, and that the final RWDISC file will be on labeled tape.

CONTROL CARDS:

Let	uuu	= user number
	aaa	= account number
	INNN	= labeled tape with input TAPDIS file
	OUTT	= labeled tape with output TAPDIS file
	TAPDISFILE	= input TAPDIS file on RAD or labeled tape
	OUTFILE	= output file (21 logical files) on RAD or labeled tape

Suppose 20 series from 9 TAPDIS pseudo-files from a single TAPDIS file are to be placed in logical files 1 through 20 in order, in the output file.

Version 1 (Everything on RAD)

```

!JOB aaa,uuu
!LIMIT (TIME,1), (CORE,26)
!ASSIGN F:RAD, (FILE,TAPDISFILE), (DIRECT), (KEYED)
!ASSIGN F:1, (FILE,T1), (OUT), (SAVE)
!ASSIGN F:2, (FILE,T2), (OUT), (SAVE)
      : (consecutive files 3 through 9)
(for label file)!ASSIGN F:10, (FILE,T10), (OUT), (SAVE)
!LOAD (EF, (FRAGTAPR)), (UNSAT, (3))
!RUN
!DATA

```

```

-----> FRAGTAP data cards
!ASSIGN F:RAD, (FILE,OUTFILE), (DIRECT), (KEYED), (OUTIN), (SAVE)
!ASSIGN F:1, (FILE,T1)
      ⋮ (Consecutive files 2 through 9)
!ASSIGN F:10, (FILE,T10)
!LOAD (EF, (ORDATR)), (UNSAT, (3))
!RUN
!DATA

```

```

-----> ORDAT data cards

```

Version 2 (Initial input, final output on labeled tape;
all work done from scratch pack)

```

!JOB aaa,uuu
!LIMIT (TIME,1), (CORE,26), (9T,1), (SP,1), (MOUNT, (X,PACK))
!MESSAGE USES 9T INNN
!MESSAGE USES 9T OUTT**WRITE
!SPAK (NRAN,1), (1500)
!ASSIGN M:EO, (INOUT)
!PCL
COPY LT#INNN/TAPDISFILE TO DP#PACK/R1.:PAK(WR(ALL),FA)
REM LT#INNN
END
!ASSIGN F:RAD, (SN,PACK), (FILE,R1,:PAK), (INOUT)
!ASSIGN F:1, (SN,PACK), (FILE,C1,:PAK), (INOUT)
      ⋮ (Consecutive files 2 through 9)
!ASSIGN F:10, (SN,PACK), (FILE,C10,:PAK), (INOUT)
!LOAD (EF, (FRAGTAPR)), (UNSAT, (3))
!RUN
!DATA

```

```

-----> FRAGTAP data cards
!ASSIGN M:EO, (OUT)
!LOAD (EF, (ORDATR((, (UNSAT, (3))
!RUN
!DATA

```

```

-----> ORDAT data cards

```

```

!PCL
SPE LT#OUTT
COPY DP#PACK/R1.:PAK TO LT#OUTT/OUTFILE(WR(ALL),FA)
END

```

Note in the above that program FRAGTAP could have been run more than once to access several TAPDIS files, and that program ORDAT need not use all consecutive files produced by FRAGTAP. There cannot be more than 10 DCB assignments at any time, however.

DATA CARDS:

All data card entries are in FORTRAN IV free-field generalized format.

FRAGTAP

Card(1): NFILES - number of pseudo-files (one per Buoy-format tape file) to access from TAPDIS file

JDCB - NFILES DCB assignments (one consecutive file per TAPDIS pseudo-file)

LDCB - one DCB assignment for TAPDIS label logical file

Cards(2 through NFILES+1)

NSEQ - sequential position of TAPDIS pseudo-file

NV - number of variables to access from each TAPDIS pseudo-file

ILF - actual RWDISC logical files of the NV variables in TAPDIS pseudo-file

ORDAT

Card(1): NDCB - number of consecutive files, not counting label file

LDCB - DCB for label file

Cards(2 through NDCB+1)

IDCB - DCB of a consecutive file to read (applies to this entire input card)

NSERIES - number of data series in consecutive file having IDCB DCB assignment

ILF - NSERIES logical files in output RWDISC file, in which to place data series from consecutive file associated with IDCB

RESTRICTIONS:

- (1) No more than ten DCB assignments for consecutive files can be in effect at any time.
- (2) No logical file past the 20th can be used for data in the output RWDISC file. The 21st file is reserved for labels.
- (3) The limiting data series length is 16000 words.

STORAGE REQUIREMENTS:

Peak core cited in a typical run of FRAGTAP-ORDAT was 50 512-word pages.

SUBPROGRAMS REQUIRED: None

TIMING: The processing of 20 12000-word data series on the scratch pack, starting and ending with RWDISC files on labeled tape, took 0.735 min of CPU time.

ERRORS AND DIAGNOSTICS:

I. FRAGTAP

- (1) NERR = ...followed by "STOP 2"
RDISC² error (type NERR) in reading word 100
(number of pseudo-files) in TAPDIS label logical file.
- (2) NERR = ...followed by "STOP 3"
RDISC error (type NERR) in reading total TAPDIS label logical file.
- (3) NERR = ...followed by "STOP 4"
RDISC error (type NERR) in reading a TAPDIS data file.
- (4) ISTAT = ...followed by "STOP 3"
BUFFER OUT error (type ISTAT) in writing labels to a consecutive data file.
- (5) ISTAT = ...followed by "STOP 1"
BUFFER OUT error (type ISTAT) in writing fascimile of TAPDIS label file to consecutive file.
- (6) ISTAT = ...followed by "STOP 4"
BUFFER OUT error (type ISTAT) in writing a consecutive file of data.

II. ORDAT

- (1) ISTAT = ...followed by "STOP 1"
BUFFER IN error (type ISTAT) in reading label portion of a consecutive file.
- (2) INWDS = ...followed by IDCB = ...followed by "STOP 10"
BUFFER IN signalled something other than 9 words for number read in label portion of consecutive file with IDCB
- (3) IDCB = ...; NUMVAR = ...; NSERIES = ...;
"STOP DISCREPANCY IN NO. OF VARIABLES"
Number of series specified by user (NSERIES) disagrees with that in TAPDIS label file for pseudo-file corresponding to IDCB.

- (4) N = ...; INWDS=...; JCYC = ...; "STOP 50"
No. of words of data(INWDS) read by BUFFER IN
disagrees with that in TAPDIS label file.
- (5) ISTAT = ...; "STOP 2"
BUFFER IN error (type ISTAT) in reading
consecutive file containing TAPDIS labels.

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: June, 1977

REFERENCES:

1. Report on Program TAPDIS by John Maltais, Woods Hole Oceanographic Institution.
2. "Handbook for Computer Users," Information Processing Center, Woods Hole Oceanographic Institution, pp. V-E-1 ff.
3. "A Nine Channel Digital Magnetic Tape Format for Storing Oceanographic Data," by John A. Maltais. Woods Hole Oceanographic Institution Reference No. 69-55.

APPENDIX 1
(FRAGTAP; ORDAT)

```

13:29 MAR 25, 1977 ID=0377
JBR 462,1719
LIMIT (TIME,4),(CORE,30),(9T,1),(Sp,1),(MOUNT,(X,PACK))
MESSAGE USES 9T IWJM
MESSAGE USES 9T IWJN***WRITE
SPAK (NRAN,1),1500
ASSIGN M:EB,(INPUT)
PCL
COPY LT#IWJM/IWEX TO DP#PACK/R1.:PAK(WR(ALL),FA)
REM LT#IWJM
END
PCL PROCESSING TERMINATED
ASSIGN F:RAD,(SN,PACK),(FILE,R1.:PAK),(INPUT)
ASSIGN F:1,(SN,PACK),(FILE,C1.:PAK),(INPUT)
ASSIGN F:2,(SN,PACK),(FILE,C2.:PAK),(INPUT)
ASSIGN F:3,(SN,PACK),(FILE,C3.:PAK),(INPUT)
ASSIGN F:4,(SN,PACK),(FILE,C4.:PAK),(INPUT)
ASSIGN F:5,(SN,PACK),(FILE,C5.:PAK),(INPUT)
ASSIGN F:6,(SN,PACK),(FILE,C6.:PAK),(INPUT)
ASSIGN F:7,(SN,PACK),(FILE,C7.:PAK),(INPUT)
ASSIGN F:8,(SN,PACK),(FILE,C8.:PAK),(INPUT)
ASSIGN F:9,(SN,PACK),(FILE,C9.:PAK),(INPUT)
ASSIGN F:10,(SN,PACK),(FILE,C10.:PAK),(INPUT)
LOAD (EF,(FRAGTAPR)),(UNSAT,(3))
:P1 ASSOCIATED.

```

* * ALLOCATION SUMMARY * *

PROTECTION	LOCATION	PAGES			
DATA (00)	A000	27			
PROCEDURE (01)	F400	2			
DCB (10)	EE00	3			
*****			SGN= C7L	SIZE=020.9K	*****
***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC					
		SEGH1-0 E091	SEGH1-1 F735	SEGH1-2 F3FF	
		SEGL0-0 A000	SEGL0-1 F400	SEGL0-2 EE00	
		00 SIZE= 4092	01 SIZE= 336	10 SIZE= 600	

```

RUN
INPUT RECORDS FOLLOW (IN GENERALIZED FORMAT)
9 1 2 3 4 5 6 7 8 9 10

```

```

1 2 2 3
2 3 2 3 4
3 1 2
4 3 2 3 4
5 1 2
6 2 2 3
7 2 2 3
8 3 2 3 4
9 3 2 3 4

```

SEQUENTIAL INPUT FILE READ BY TAPDIS	REL LOCATION IN TAPDIS FILE (KREL)
1	1

2	12001
3	24001
4	36001
5	48001
6	60001
7	72001
8	84001
9	96001

OUTPUT FILE SUMMARY IN ORDER OF ACCESS

OUTPUT DCB	TAPE FILE	RUBY TAPE VARI ACCESSED THIS FILE	TOTAL WORDS IN OUTPUT FILE
1	IWEX A1 3	2 3	12009
2	IWEX A2	1 2 3	12009
3	IWEX B2 3	3	12009
4	IWEX A4	1 2 3	12009
5	IWEX A5	1	12009
6	IWEX A6	1 2	12009
7	IWEX B6	1 2	12009
8	IWEX A8	1 2 3	12009
9	IWEX C10 3	1 2 3	12009

STOP OUTPUT COMPLETED

ASSIGN M:EB,(OUT)
 LOAD (EF,(ORDATR)),(UNSAT,(3))
 !P1 ASSOCIATED.

* * ALLOCATION SUMMARY * *

PROTECTION	LOCATION	PAGES
DATA (00)	A000	27
PROCEDURE (01)	F400	2
DCB (10)	EE00	3

***** SGN= C7L

SIZE=020.5K *****

***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC

SEGHI=0	ECB3	SEGHI=1	F6A1	SEGHI=2	F3FF
SEGLB=0	A000	SEGLB=1	F400	SEGLB=2	EE00

00 SIZE= 4CB4 01 SIZE= 2A2 10 SIZE= 600

RUN

OUTPUT SUMMARY FOR DCB 1
 RUBY FILE: IWEX A1 3
 NO. OF SERIES: 2
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 1 2
 RUBY VAR. NOS.: 2 3

OUTPUT SUMMARY FOR DCB 2
 RUBY FILE: IWEX A2
 NO. OF SERIES: 3
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 3 4 5

BUOY VAR. NOS.: 1 2 3

OUTPUT SUMMARY FOR DCB 3

BUOY FILE: IWEX B2 3

NO. OF SERIES: 1

DATA WDS. EACH SERIES: 12000

WRITTEN TO LOG. FILES: 6

BUOY VAR. NOS.: 3

OUTPUT SUMMARY FOR DCB 4

BUOY FILE: IWEX A4

NO. OF SERIES: 3

DATA WDS. EACH SERIES: 12000

WRITTEN TO LOG. FILES: 7 8 9

BUOY VAR. NOS.: 1 2 3

OUTPUT SUMMARY FOR DCB 5

BUOY FILE: IWEX A5

NO. OF SERIES: 1

DATA WDS. EACH SERIES: 12000

WRITTEN TO LOG. FILES: 10

BUOY VAR. NOS.: 1

OUTPUT SUMMARY FOR DCB 6

BUOY FILE: IWEX A6

NO. OF SERIES: 2

DATA WDS. EACH SERIES: 12000

WRITTEN TO LOG. FILES: 11 12

BUOY VAR. NOS.: 1 2

OUTPUT SUMMARY FOR DCB 7

BUOY FILE: IWEX B6

NO. OF SERIES: 2

DATA WDS. EACH SERIES: 12000

WRITTEN TO LOG. FILES: 13 14

BUOY VAR. NOS.: 1 3

OUTPUT SUMMARY FOR DCB 8

BUOY FILE: IWEX A8

NO. OF SERIES: 3

DATA WDS. EACH SERIES: 12000

WRITTEN TO LOG. FILES: 15 16 17

BUOY VAR. NOS.: 1 2 3

OUTPUT SUMMARY FOR DCB 9

BUOY FILE: IWEX C10 3

BUBY VAR. NOS.: 1 2 3

OUTPUT SUMMARY FOR DCB 3

BUBY FILE: IWEX B2 3
 NO. OF SERIES: 1
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 6
 BUBY VAR. NOS.: 3

OUTPUT SUMMARY FOR DCB 4

BUBY FILE: IWEX A4
 NO. OF SERIES: 3
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 7 8 9
 BUBY VAR. NOS.: 1 2 3

OUTPUT SUMMARY FOR DCB 5

BUBY FILE: IWEX A5
 NO. OF SERIES: 1
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 10
 BUBY VAR. NOS.: 1

OUTPUT SUMMARY FOR DCB 6

BUBY FILE: IWEX A4
 NO. OF SERIES: 2
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 11 12
 BUBY VAR. NOS.: 1 2

OUTPUT SUMMARY FOR DCB 7

BUBY FILE: IWEX B4
 NO. OF SERIES: 2
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 13 14
 BUBY VAR. NOS.: 1 3

OUTPUT SUMMARY FOR DCB 8

BUBY FILE: IWEX A2
 NO. OF SERIES: 3
 DATA WDS. EACH SERIES: 12000
 WRITTEN TO LOG. FILES: 15 16 17
 BUBY VAR. NOS.: 1 2 3

OUTPUT SUMMARY FOR DCB 9

BUBY FILE: IWEX C10 3

NO. OF SERIES: 3
DATA WDS. EACH SERIES: 12000
WRITTEN TO LOG. FILES: 18 19 20
BUGY VAR. NOS.: 1 2 3

900 WORDS WRITTEN IN LABEL L. F. NO. 21
STOP NORMAL END

PCL

SPE LT#IWJN

COPY DP#PACK/R1.:PAK IN LT#IWJN/JWEX22(WR(ALL),FA)

END

PCL PROCESSING TERMINATED

C. MODIFYING RWDISC FILE FROM B BY REPLACING DATA
SERIES WITH FOURIER COEFFICIENTS IN
A SPECIAL ORDER

NAME: FOURIER

TYPE: Main Program

PURPOSE: To create RWDISC⁽¹⁾ file of FOURIER coefficients from RWDISC file of data series, for input into bispectral programs

MACHINE: SIGMA-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program FOURIER takes an RWDISC file containing a number of data series and replaces each one by a series of Fourier coefficients which is specially formatted for efficient access by the bispectral programs written by this author. The series may be processed in a number of ways including: a) prewhitening (done on the original series before any other processing); b) compression of the series by triangular weighting on consecutive subsamples (done after any prewhitening but before any further operations); c) splitting into pieces, which may be contiguous or: d) overlapping of pieces by 50%; e) subtraction of mean of each piece, or trend based on endpoints, or trend based on linear regression; f) "Hanning" of pieces, i.e., 100 percent cosine tapering.

Calculation of the Fourier coefficients is done by IPC program HARM1.⁽²⁾

INPUT:

Input is assumed to be an RWDISC file consisting of one data series per logical file, starting at word 5 of each logical file, the first four words being reserved for label information. Every series to be accessed by program FOURIER is assumed to be of equal length. There may be up to 21 logical files in the input RWDISC file. If there are 1 through 20 inclusive, every logical file may contain a data series. The case of 21 logical files is considered to be standard, and in that case the 21st logical file is reserved for label information.

OUTPUT: (1) Disc

Disc output is a modified version of the input RWDISC file, and therefore has the same number of logical files. Each data series is replaced by a series of Fourier coefficients starting at word 5 and organized as follows:

Let the n -point data series be represented by the sequence:

$$\{Y_j\}, j = 0, n-1$$

and the Y_j -th term by the expression:

$$Y_j = a_0 + \sum_{k=1}^{n/2} [a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)] .$$

Now let a_{km} , b_{km} represent the k -th order coefficients, for the m -th piece. Further, let $\{a_{km}\}$ represent the sequence of k -th order cosine coefficients over all pieces taken in order, the overlapped pieces contributing the even sequence $m = (2, 4, 6, \dots)$ in the case of overlapping. Then a logical file of Fourier coefficients produced by program FOURIER has the appearance

$$\{a_{0m}\}, [\{a_{km}\}, \{b_{km}\}], k = 1, n/2-1, \{a_{n/2 m}\}$$

All coefficients $b_{n/2 m}$ are zero in the above representation of the data series.

It will be noted that the coefficients necessary to calculate a spectral estimate at a given frequency are always contiguous.

(2) Line Printer (see Appendix 2)

Normally, line printer output consists of three lines giving the processing conditions of this run, and the date of the run. Line 3 is a facsimile of an EBCDIC label written on logical file 21 of the output RWDISC file (starting at word 1) if 21 logical files are chosen. The entries are self-explanatory if it is noted that when referring to the execution of a given process, 0 means "no" and 1 means "yes". The entry "NO WDS DATA" refers to the length of each data series entering the program.

USAGE:

(1) Control cards

Let aaa = account number

uuu = user number

fff = input name

and let all element files exist in account 462.
Then the control card portion of the input card
deck appears as follows:

```
!JOB aaa,uuu
!LIMIT (TIME,4),(CORE,50)
!ASSIGN F:RAD,(FILE,fff),(DIRECT),(KEYED),(INOUT)
* !LOAD(EF,(FOURIERR,462),(CSUBR,462),(CTRLR,462),(INPUTR,462),;
! (KOEFHARMR,462),(OUTOMEGAR,462)),(UNSAT,(3))
!RUN
!DATA
```

* If all element files are contained in a single
file FOURIERR, then these two cards become
simply: !LOAD (EF,(FOURIERR,462)),(UNSAT,(103),(3))

(2) Data cards

There is a single data card followed by NAMELIST input,
which must be terminated by an * card, whether or
not there is any actual namelist input.

The data card has 8 or 9 entries, all in FORTRAN gen-
eralized format. The order of these entries is:

IFIRST	- the first logical file to be processed by program FOURIER
ILAST	- the last logical file to be processed by program FOURIER. See "RESTRICTIONS"
NDO2	- the data length in words
IPREW	- prewhitening switch (0 or 1)
ISUBSAMP	- subsampling switch (0 or 1). If ISUBSAMP = 1, triangular weighting of the subsample will be done automatically.
NSSMP	- an <u>odd number</u> , the length of each sub- sample; to be entered <u>only</u> if ISUBSAMP = 1
LPiece	- piece length. <u>SEE RESTRICTIONS BELOW</u>
IOVERLAP	- 50% overlap switch (0 or 1). This will be set to 0 unless Hanning is specified

```

      / 0 Don't subtract mean or trend of each
        piece
      1 Subtract trend based on endpoints of
ISUBT  - { each piece
      2 Subtract trend based on linear
        regression
      3 Subtract mean of each piece

IHANN  - Hanning switch (0 or 1)

```

The NAME LIST at present has only one relevant variable: MF = no. of logical files in the RWDISC file. The default for MF is 21, and it is absolutely essential to enter it as a name list variable if MF is not in fact 21. Failure to do so will not produce an error message, and can result in subtly incorrect answers.

If MF is four, for example, follow the data card by

MF = 4 .

If MF is twenty-one, follow the data card simply by

* .

RESTRICTIONS:

- (1) $NDO2 \leq 16000$. If NPIECES is the total number of pieces, then in the limit as $NPIECES \rightarrow \infty$, the total length of the output series of Fourier coefficients will approach $2 \times NDO2$. If the user wishes to process many series of length considerably shorter than 16000, it may be advisable to recompile the program. The main working array WK is in COMMON and is normally dimensioned WK(33000). When editing the source program, this array dimension should be set at $(2 \times NDO2)$. This is most readily done in the COMMON of every program and subroutine by combining all into a single file.
- (2) LPIECE. There are several important restrictions on the length of a piece; all of which must be met before the program will proceed:
 - i) LPIECE must be an even number with no prime factor greater than 5.² Additionally, it is assumed that the output of program FOURIER will be used for bispectral calculations. The scalar bispectral program BISCAL requires that LPIECE be evenly divisible by 4, and the vector (rotary) bispectral program

BIVEC requires that LPIECE be evenly divisible by 8. Hence these added restrictions are placed on LPIECE. Appendix 1 is a list of piece lengths ≤ 4000 which can be used for programs BISCAL and BIVEC. Numbers in parentheses are not divisible by 8, and so cannot be used in connection with program BIVEC. Piece lengths must further meet the following criteria:

- ii) LPIECE must satisfy one of the following inequalities: Let IND be the data series length after any subsampling. Then:

$$450 \geq \text{LPIECE} \geq (\text{IND}/75) \text{ for 50\% overlap}$$

- or -

$$450 \geq \text{LPIECE} \geq (\text{IND}/150) \text{ for no overlap}$$

These limits can be changed by minor source program modifications. Briefly, to increase the upper limit, change the dimension of all arrays of size 450 to the new upper limit. To decrease the lower limit, all arrays of size 150 must be changed to size J, where the denominator of the lower limit above is (J/2) for 50% overlap and J for no overlap.

- (3) i) $\text{MF} \leq 21$
 ii) If $\text{MF} = 21$, then $\text{ILAST} < 21$
 iii) If $\text{MF} < 21$, then $\text{ILAST} \leq \text{MF}$

ERRORS AND DIAGNOSTICS:

- (1) "PIECE LENGTH OF...INVALID. MUST BE FROM THE FOLLOWING SET:"

Meaning: The user has specified a piece length which fails to meet criterion i), item (2) of "RESTRICTIONS" section for program BISCAL. The piece lengths which are output are those of Appendix 1.

Action: Abort

- (2) "PIECE LENGTH DOES NOT SATISFY RESTRICTION FOR (NO) OVERLAP"

Meaning: The user has specified a piece length which does not satisfy one of the criteria ii), item (2) of "RESTRICTIONS"

Action: Abort

(3) "SUBSAMPLE LENGTH MUST BE ODD"

Meaning: The user has specified subsampling with an even subsample length.

Action: Abort

(4) "OVERLAP POSSIBLE ONLY IF HANN. IOVERLAP SET TO 0"

Meaning: Self-explanatory

Action: Proceed with no overlapping.

(5) "MF CANNOT EXCEED 21"

Meaning: User must not specify more than 21 logical files

Action: Abort

(6) "LAST LOG FILE CANNOT EXCEED MF OR 20"

Meaning: One of the following conditions was not met:

(a) If $MF = 21$, the last logical file to be processed must be less than 21;

(b) If $MF < 21$, the last logical file must not exceed MF.

Action: Abort

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: June, 1977

REFERENCES: (1) "Handbook for Computer Users," Woods Hole Oceanographic Institution, pp V-E-1 ff.
(2) "Reprints of WHOI Programs," Woods Hole Oceanographic Institution, HARMI Report.

APPENDIX 1
(FOURIER)

APPENDIX 1

Numbers divisible by four and ≤ 4000 containing no prime factor greater than 5:

(Numbers in parentheses may not be used for piece lengths to be handled by program BIVEC)

(12)	(108)	(324)	768	(1500)	2560
16	120	360	800	1536	2592
(20)	128	384	864	1600	(2700)
24	144	400	(900)	(1620)	2880
32	160	432	960	1728	2916
(36)	(180)	480	(972)	1800	3000
40	(192)	(500)	(1000)	1920	3072
48	200	512	1024	1944	3200
(60)	216	(540)	1080	2000	3240
64	240	576	1152	2048	3456
72	256	600	1200	2160	3600
80	288	640	1280	2304	3848
96	(300)	648	1296	2400	3888
(100)	320	720	1440	(2500)	4000

APPENDIX 2
(FOURIER)

```

***** SIGMA 5/6/7/9 LEAD MODULE MAP *****
***** ACN= 462 <RRGT> START=1540C *****
***** LMN= D7L BIAS= A000 *****
***** SGN= D7L SIZE=012.9K *****
***** + =DRLE DEF - =LIB DEF * =UNUSED DEF @ =UNUSED LIB DEF

```

```

***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC

```

```

SEGHI=0 15070 SEGHI=1 1584F SEGHI=2 153FF
SEGL=0 A000 SEGL=1 15400 SEGL=2 15200
00 SIZE= 807E 01 SIZE= 750 10 SIZE= 200

```

```

***** SECT - PROGRAM SECTIONS MAP *****
A2F8 DSECT 0 A2F8 0 F:ICOM
12642 CSECT 0 12642 0 *V#MAIN
12A16 CSECT 0 12A16 0 *V#CTRL
12DE6 CSECT 0 12DE6 0 *V#INPUT
131C4 CSECT 0 131C4 0 *V#K8EFF
131DA CSECT 0 131DA 0 *V#BUT#MEGA
1521E LDCB 2 1521E 0 M:
15250 LDCB 2 15250 0 M:DA
1528C LDCB 2 1528C 0 M:PC
152C8 LDCB 2 152C8 0 M:LL
15304 LDCB 2 15304 0 F:108
15340 LDCB 2 15340 0 F:105
1537C LDCB 2 1537C 0 F:RAD
1540C CSECT 1 1540C 0 *MAIN
1549E CSECT 1 1549E 0 CTRL
15502 CSECT 1 15502 0 INPUT
155FA CSECT 1 155FA 0 K8EFF
156F8 CSECT 1 156F8 0 BUT#MEGA
15A24 CSECT 1

```

```

***** RELLOCATABLE DEFINITIONS SORTED BY NAME *****
1549E 0 CTRL 1537C 0 F:RAD 15340 0 F:105 15304 0 F:108
A2F8 0 F:ICOM 15502 0 INPUT 155FA 0 K8EFF 15433 0 *LABEL
1521E 0 M: 15250 0 M:DA 152C8 0 M:LL 1528C 0 M:PC
15304 0 BUT#MEGA 15340 0 *V#CTRL 12DE6 0 *V#INPUT
156F8 0 *V#K8EFF 131DA 0 *V#BUT#MEGA 12642 0 *V#MAIN
131C4 0

```

RUN

```

LABEL AS WRITTEN ON L.F. 21 IF ME#21:
NO CTS 4*NO WDS DATA 978*PREW 0*SUBSAMP 0*PC SIZE 256*NR PCS 38*BLAP 1*ISUBT 3*HANN 0*CREATED 20143 MAY 19177

```

STOP PROCESSING COMPLETED

D. BISPECTRAL CALCULATIONS AND PLOTTING

NAME: BISCAL

TYPE: Main Program

PURPOSE: To calculate auto and cross-bispectra of two real series, to write a consecutive file of bicoherences for transmittal to plotting program BPLOT,¹ and to compute confidence levels for the bicoherences.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BISCAL uses as input an RWDISC² file of FOURIER coefficients having a particular order to be described below. These coefficients are assumed to result from real data series and are normally the output of program FOURIER,³ which has done any necessary prewhitening, subsampling, overlapping, mean and trend removal, and Hanning. If L is the piece length and NP is the number of pieces resulting from program FOURIER, there are (L)(NP) words input into BISCAL for each data series.

Program BISCAL then computes approximately $(3/4)(L/2)^2$ values of bicoherence, along with the corresponding biphasic and biphasic error if desired. Two alternate forms of bicoherence may be calculated. The default form is that defined by N. C. G. Yao.⁴ The other form is a normalization by amplitude, so that only phase information remains (see below). The user can elect to have the program write a consecutive file of bicoherences to be used as input to program BPLOT for plotting. The user can also instruct the program to determine those bicoherences which correspond to specified confidence levels. A listing of partitioning of bicoherences into bins of specified size can be obtained, as well as a listing of bicoherences and related quantities when the bicoherences fall between limits set by the user.

Computations PerformedA. BispectraForm 1

The default definitions of bispectrum, bicoherence and biphas used by program BISCAL are those given by N. C. G. Yao⁴ except for the definition of the "sum frequency" ω_3 . Consider two real time series from which are derived three complex Fourier coefficients, each of which may come from either series. Let these coefficients be $X(\omega_1)$, $X(\omega_2)$, and $X(\omega_3)$. Since we are considering real time series, the real part of $X(\omega_k)$ is an even function and the imaginary part is an odd function about the frequency origin. Using this property,

$$X(\omega_k) = \begin{cases} a_k + ib_k & \text{when } \omega_k > 0 \\ a_k - ib_k & \text{when } \omega_k < 0 \end{cases} \quad (1)$$

The average bispectrum over the pieces under consideration is then defined as $\langle B(\omega_1, \omega_2) \rangle$, where:

$$\begin{aligned} \langle B(\omega_1, \omega_2) \rangle &= \langle X(\omega_1) X(\omega_2) X^*(\omega_3) \rangle & (2) \\ &\quad \text{when } \omega_1 + \omega_2 = \omega_3 \\ &= 0 & \text{when } \omega_1 + \omega_2 \neq \omega_3 \end{aligned}$$

In the frequency domain of interest (see below), $\omega_1 > 0$ always, and $|\omega_1| > |\omega_2|$. Since $\omega_3 = \omega_1 + \omega_2$, we have that $\omega_3 > 0$ in this domain always, and therefore

$$X^*(\omega_3) = a_3 - ib_3$$

The second Fourier coefficient $X(\omega_2)$ ranges over both positive and negative frequencies in the domain of interest and is evaluated according to equation (1).

Expanding equation (2) above, we have that:

$$\text{Re}\langle B(\omega_1, \omega_2) \rangle = \langle a_1 a_2 a_3 \pm a_1 b_2 b_3 + b_1 a_2 b_3 \mp b_1 b_2 a_3 \rangle \quad (2a)$$

$$\text{Im}\langle B(\omega_1, \omega_2) \rangle = \langle \pm a_1 b_2 a_3 - a_1 a_2 b_3 + b_1 a_2 a_3 \pm b_1 b_2 b_3 \rangle \quad (2b)$$

where the upper signs correspond to $\omega_2 > 0$ and the lower to $\omega_2 < 0$.

The bicoherence $\text{bic}(\omega_1, \omega_2)$ is then defined as:

$$\text{bic}(\omega_1, \omega_2) = \frac{|\langle B(\omega_1, \omega_2) \rangle|}{[\langle |X(\omega_1)|^2 \rangle \langle |X(\omega_2)|^2 \rangle \langle |X(\omega_1 + \omega_2)|^2 \rangle]^{1/2}} \quad (3)$$

and the biphas $[\text{biphase}(\omega_1, \omega_2)]$ as:

$$\text{biphase}(\omega_1, \omega_2) = \tan^{-1} \frac{\text{Im}\langle B(\omega_1, \omega_2) \rangle}{\text{Re}\langle B(\omega_1, \omega_2) \rangle} \quad (4)$$

Form 2

Program BISCAL can compute an alternate form of bicoherence based on an amplitude-normalized bispectrum. Let:

$$B_R = \text{Real}[X(\omega_1)X(\omega_2)X^*(\omega_1 + \omega_2)]$$

$$B_I = \text{Imag}[X(\omega_1)X(\omega_2)X^*(\omega_1 + \omega_2)]$$

$$B = \sqrt{B_R^2 + B_I^2}$$

The average bispectrum over the pieces under consideration is then defined as $\langle B'(\omega_1, \omega_2) \rangle$, where:

$$\langle B'(\omega_1, \omega_2) \rangle = \langle \frac{B_R}{B} + i \frac{B_I}{B} \rangle = \langle \frac{B_R}{B} \rangle + i \langle \frac{B_I}{B} \rangle .$$

The amplitude-normalized bicoherence is then defined as:

$$\text{bic}'(\omega_1, \omega_2) = |\langle B'(\omega_1, \omega_2) \rangle|$$

The biphase is now:

$$\text{biphase}(\omega_1, \omega_2) = \tan^{-1} \frac{\text{Im} \langle B'(\omega_1, \omega_2) \rangle}{\text{Re} \langle B'(\omega_1, \omega_2) \rangle}$$

Frequency Domains

Because of symmetries in the bispectrum, it is not necessary to let the constituent frequencies range over all possible values. If all three coefficients in (1) above are based on the same series, then the "auto-bispectrum" and "auto-bicoherence" are being calculated and the minimum necessary domain is:

$$\{A\} = \{ \{ \omega_2 \leq \omega_1 \} \cap \{ \omega_2 \leq \omega_1 + L/2 \} \cap \{ \omega_2 > 0 \} \} \quad (5)$$

where $\{ \omega_2 \leq \omega_1 \}$ actually means $\{ (\omega_1, \omega_2) \ni \omega_2 \leq \omega_1 \}$ etc., and where $\omega_1 + \omega_2 - \omega_3 = 0$ and L is the piece length, or pictorially:

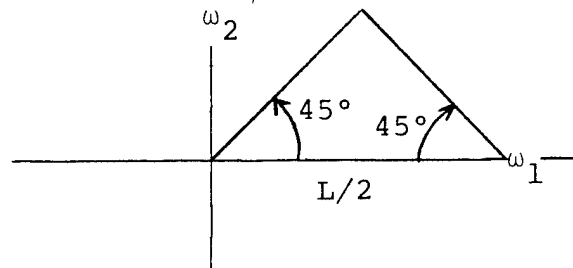


Figure 1

Whenever more than one data series enter the computation of the three coefficients in (1), then the minimum necessary domain becomes:

$$\{C\} = \{\{A\} \cup \{\{\omega_2 > -\omega_1\} \cap \{\omega_1 \leq L/2\} \cap \{\omega_2 < 0\}\}\} \quad (6)$$

where $\omega_1 + \omega_2 - \omega_3 = 0$, L is the piece length, and $\{A\}$ is defined in (5), or pictorially:

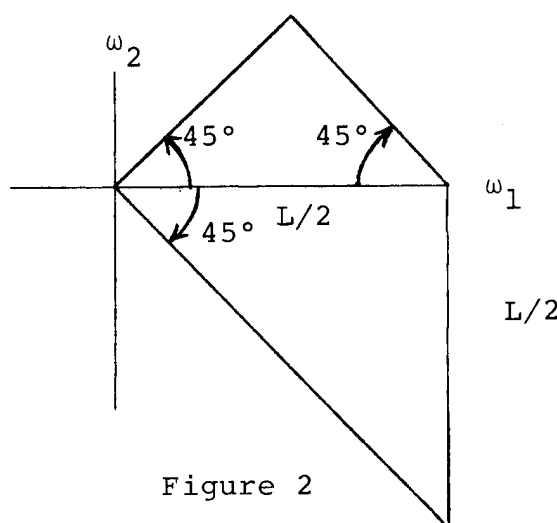


Figure 2

In this case, we say that "cross-bispectrum" and "cross-bicoherence" are being computed.

Order of the Computations

Program BISCAL can handle two data series, each of length 16K words. This means that for 50% overlap and for the limiting case of an infinite number of pieces, each data series entering the program consists of 32K FOURIER coefficients. In order to permit computation of the bispectra, and to minimize the number of disc accesses, the following steps are taken:

1) Organization of input file

Let $\{a_i\}$ be the set of all real parts of the i^{th} -frequency FOURIER coefficient taken over all pieces, and let $\{b_i\}$ be the corresponding imaginary parts. Then, for a piece length of L , the input file elements have the following order:

$\{a_0\}, \{a_1\}, \{b_1\}, \{a_2\}, \{b_2\}, \dots, \{a_{\frac{L}{2}-1}\}, \{b_{\frac{L}{2}-1}\}, \{a_{\frac{L}{2}}\}$

The elements of $b_{\frac{L}{2}}$ are all zero and do not appear in the input file. Program FOURIER³ automatically outputs coefficients in this order. This permits all elements associated with a given frequency to be input by entering the series only once.

2) Organization of frequency domain and computation procedure

No more than $(\frac{L}{2} + 2)(NP)$ elements from each series are in core at any time, where L = piece length, and NP = no. of pieces. These elements correspond essentially to the "top" and "bottom" halves of each series. This causes the domain of Fig. 2 to divide naturally into five sub-domains or "rasters" as indicated by the dashed lines in the following figure:

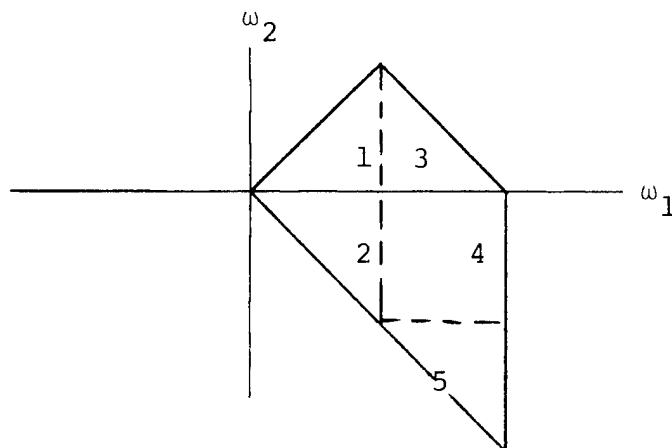


Figure 3

Computations of bicoherences proceed for each raster in order. The reading of series "halves" from the disc is a function of each particular combination of source series for the three frequencies, an attempt being made to maximize efficiency. Within each raster, computation of bicoherences is done from left to right along diagonal scans of slope (-1), each scan thus corresponding to a single sum frequency ω_3 . The disc is accessed for the coefficients of frequency ω_3 only when these coefficients are not already available in core.

Averaging over user-specified pieces is done separately on the real and imaginary parts of the bispectrum, equations (2a) and (2b), and on each autospectrum in the denominator of equation (3). The averages are then used in calculating bicoherence and biphase.

B. Confidence levels

As each bicoherence is computed, the user may elect to assign it to a bin having a size of his choice. User may also determine the number of bins by specifying the upper bicoherence limit to be considered. After this sorting has been done, program BISCAL determines the bicoherences below which specified fractions ("confidence levels") of the total number of bicoherences calculated by the program lie. This is done by calculating the fraction of the total which lies below the upper limit of each bin, finding the two closest such fractions which "bracket" the specified confidence level, and linearly interpolating the two corresponding bicoherences to find the one which corresponds to the specified confidence level. A dump of the distribution is available.

USAGE:

(1) Tailoring load module size

For long series of FOURIER coefficients, it is desirable to have large data buffers to keep disc accesses to a minimum. On the other hand these large data buffers can be wasteful for shorter series both in terms of (page · minutes) charges and the penalty in turnaround time. The user is therefore advised to recompile the program to match his needs if he is going to be handling a number of data series of comparable length. To edit the program prior to recompilation, the following steps should be observed:

- a. Copy the source file to a duplicate file, and then work with the duplicate file;
- b. Change the DIMENSION's of OM1 and OM2 arrays to at least $[(L/2 + 2)(NP)]$, where L = piece length and NP is the number of pieces.
- c. Check that array W3 is DIMENSIONed at least $2(NP)$.
- d. Check that SQ is DIMENSIONed at least $SQ(L/4 + 1, L/4 + 1)$ (both dimensions identical).
- e. Check that the initialization of variable ISQD is identical to both dimensions of SQ.
- f. Recompile.

(2) Control cards

Let aaa = account number

uuu = user number

fff = input file name (RWDISC, keyed)

ttt = file name for transmittal to plotting
program (consecutive)

Assume the element file is called BISCALO and exists
in account aaa. The Control Card portion of the in-
put deck is then as follows:

```
!JOB aaa,uuu
* !LIMIT (TIME,4), (CORE,48)
!ASSIGN F:RAD, (FILE,fff), (DIRECT), (KEYED)
† !ASSIGN F:l, (FILE,ttt), (OUT), (SAVE)
!LOAD (EF, (BISCALO)), (UNSAT, (3))
!RUN
!DATA
(DATA CARDS)
```

(3) Data cards

Program BISCAL has NAMELIST input followed by one or
two data cards depending on the application. The
NAMELIST input must be terminated by an * card whether
or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are
as follows. In general the switches are to be set
to 0 for "off" and 1 for "on".

* Time and core limits will vary widely depending on
application. See below.

† Necessary only if storing matrix of bicoherences on
disc, for transmittal to plotting program BPLOT.
DCB assignment may be changed by NAMELIST parameter
ISTORE.

<u>NAMelist VARIABLE</u>	<u>MEANING</u>	<u>DEFAULT VALUE</u>
MF	Number of logical files in input RAD file	21
IDATSTART	Starting location of data in each input logical file	5
INORM	Switch to calculate amplitude-normalized bicoherences	0
LISTBI	Switch to list bicoherences and associated quantities	0
IDUMP	Switch to store bicoherences in a consecutive file for transmittal to plotting program BPLOt	1
ISTORE	DCB assignment for consecutive file of bicoherences created by setting switch IDUMP	1
KONF	Switch for determining bicoherence confidence levels	0
BINSIZE	Bin width for sorting bicoherences. A given bicoherence is placed in bin I, where $I = \text{INT}(\text{bicoherence}/\text{BINSIZE}) + 1$	0.01
BICOHLIM	The bicoherence "limit" for sorting which determines the number of bins of size BINSIZE, by determining the integer NBINS obtained by rounding $(\text{BICOHLIM}/\text{BINSIZE})$, and then redefining BICOHLIM as $(\text{NBINS})(\text{BINSIZE})$	1.5
KB1	Switch to display partitioning of bicoherences	1
KB2	Switch to display fraction of total bicoherences under each bin limit	1
IHANN	Determines method of calculating phase error when LISTBI $\neq$ 0 (See PHSErr, Page 12) IHANN=0 not Hanned and not overlapped IHANN=1 Hanned and overlapped	1

<u>NAMelist VARIABLE</u>	<u>MEANING</u>	<u>DEFAULT VALUE</u>
SIGLVL	95% confidence level for bicoherences, above which phase error PHSERR is calculated and printed when LISTBI=1	*

The data cards follow the * card which terminates NAMelist input. They are all generalized, free-field format and are as follows:

Card 1 (Mandatory)

<u>Variable No.</u>	<u>Meaning</u>
1	Logical file for first series [†]
2	Logical file for second series [†]
3	Logical file for sum frequency [†]
4	Total number of pieces in each series
5	Piece length
6 }	Lower, upper piece sequence numbers inclusive, between which user wants to average
7 }	
8 }	Lower, upper limits determining when to print bicoherences when LISTBI≠0. Bicoherences must satisfy: (var. #8) < bicoherence ≤ (var. #9)
9 }	

Card 2 (Input only when KONF≠0)

<u>Variable No.</u>	<u>Meaning</u>
1 (NCONF)	Number of confidence levels to follow (≤10)
2 (CONF(I))	NCONF confidence levels expressed as a decimal, 0 to 1

* Default is an approximate value. If P = number of pieces, let the number of degrees of freedom EDof be defined as follows:

$$EDof = \begin{cases} 36P^2/(19P-1) & \text{for Hanning, overlapping} \\ 2P & \text{for no Hanning, no overlapping} \end{cases}$$

Then the default value of SIGLVL is

$$SIGLVL = (6/EDof)^{1/2}$$

[†] If the first three variables are equal, the program automatically reverts to calculation of auto-bicoherences based on a single file, choosing the minimum necessary domain.

OUTPUT:(1) Line Printer

Typical line printer output is displayed in Appendix 1. First a dump of NAMELIST parameter values is printed, followed by a facsimile of input data cards. Next, if confidence levels are called for, the "adjusted bicoherence limit" (as determined by the rule given in describing variable BICOHLIM in the section on Data Cards above) is printed.

This is followed by a listing of individual bicoherences if called for. There is one bicoherence value per line, plus other output described below. The listing occurs only within the limits given by variables #8 and #9 on Card 1. The user is cautioned that for cross-bicoherences the total number of bicoherences computed is about $0.75 (LP)^2$ where LP = piece length. Without sufficiently narrow limits on the bicoherences actually printed, this can result in a very large number of output pages. The items output in each line depend on which form of bicoherence is called for in the program. They are as follows, in order of appearance from left to right on the line:

(A) Default bicoherence (after Yao):

<u>Heading(s)</u>	<u>Meaning</u>
F1 F2 F3	frequency triplet (sequence numbers) associated with this bicoherence
BIC	bicoherence
BIPH	biphase
BIMOD	$ \langle B(\omega_1, \omega_2) \rangle $
BISPECR	$\text{Re } \langle B(\omega_1, \omega_2) \rangle$
BISPECI	$\text{Im } \langle B(\omega_1, \omega_2) \rangle$
AUTOF1	$\langle X(\omega_1) ^2 \rangle$
AUTOF2	$\langle X(\omega_2) ^2 \rangle$
AUTOF1F2	$\langle X(\omega_3) ^2 \rangle$

<u>Heading(s)</u>	<u>Meaning</u>
SD	$[< X(\omega_1) ^2> < X(\omega_2) ^2> < X(\omega_3) ^2>]^{1/2}$
PHSERR*	phase error; depends on IHANN (see page 9) $= \sqrt{2P}$ for IHANN=0, P = no. of pieces $= 57.296 \sin^{-1} [1.96/(BIC \cdot W)]$ for IHANN=1 (default), where $W = [36P^2/(19P-1)]^{1/2}$

(B) Alternate bicoherence (normalized by amplitude)

<u>Heading(s)</u>	<u>Meaning</u>
F1 F2 F3	frequency triplet (sequence nubmers) associated with this bicoherence
BIC	bicoherence
BIPH	biphase
BISPPR	$\text{Re } \langle B'(\omega_1, \omega_2) \rangle$
BISPPI	$\text{Im } \langle B'(\omega_1, \omega_2) \rangle$
PHSERR	Defined as above

After listing individual bicoherences, there is an optional listing of how the bicoherences are partitioned among bins, followed by the total number of bicoherences submitted. The format is displayed in Appendix 1.

* PHSERR is calculated and printed only above 95% confidence level given by SIGLVL (See NAMELIST variable table). In the line printer output shown in Appendix 1, SIGLVL = 0.5612.

Following this there is an optional display of cumulative fractional numbers of total bicoherence lying below each upper bin limit. Again, the format is displayed in Appendix 1.

Finally, when a calculation of confidence levels is called for, a summary of the result for each confidence level is displayed as in Appendix 1. This is followed by a statement of the number of bicoherences which happened to fall above the limit used in the program (also stated).

(2) Disc

Normally a consecutive disc file is produced by program BISCAL, which is used to transmit the computed bicoherences to program BPLOT for plotting. The first task of program BPLOT is to construct from this file a two-dimensional array whose elements occupy the same relative position in the matrix as do the grid points in the final plot. By using the coding which does this (2 statements for auto, or 5 statements for cross-bicoherences) the user can access this file directly. This will not be delved into here, but the author can be consulted for details.

RESTRICTIONS:

1. The length of the data series which program BISCAL can handle depends on the dimensions of OM1, OM2, and array SQ, as well as on how the data series was processed in the FOURIER transform. The following relations can be used:

Let $\mathcal{L}$ = data series length adjusted so that $2\mathcal{L}$ is evenly divisible by L, the piece length

L = piece length

D = dimensions of OM1 and OM2

NP = number of pieces

50% overlap

$$\mathcal{L} = \frac{L}{2} \left(\frac{2D + L + 4}{L + 4} \right) \text{ and } NP = \frac{2\mathcal{L}}{L} - 1 \quad (7)$$

$$\mathcal{L} = \frac{2DL}{L + 4} \text{ and } NP = \frac{\mathcal{L}}{L} \quad (8)$$

In both cases the number of points in the series entering BISCAL is $(L)(NP)$, and OM1, OM2 must be dimensioned at least $(NP)(L/2 + 2)$.

The latter requirement has been incorporated in the derivation of expressions (7) and (8).

Example. Program BISCAL has been successfully run with $D = 16400$ and SQ dimensioned at $SQ(70,70)$. Assuming 50% overlap, eqn. (7) above yields $\mathcal{L} = 16256$ as the maximum data series length which could have been handled. This is subject to restriction (2) below, which requires that SQ should be dimensioned at least $SQ(M,M)$ where $M = L/4 + 1$, a condition which is met here.

2. Array W3 must be DIMENSIONED at least $2(NP)$.
3. The dimension of array SQ must be at least $SQ(M,M)$ where $M = L/4 + 1$, $L = \text{piece length}$.
4. Variable ISQD must be initialized at either dimension of SQ in (3).
5. L must be an even number with no prime factor greater than 5 and evenly divisible by 4.
6. (No. of confidence levels) ≤ 10 .
7. (No. of bins used in sorting bicoherences) ≤ 200 .

STORAGE AND CPU TIME REQUIREMENTS:

As already mentioned economy and convenience may be achieved by tailoring the size of the program to the user's needs if he foresees using it for a number of data series of somewhat comparable length.

The following operating history is provided to assist in this and for general information:

Auto or Cross- Bicoherence	Dimensions of OM1/and OM2	Dimensions of SQ	Piece Length	No. of pieces	Task	Peak core (pg size 512)	cpu time (min)	Charge (CU)
auto	5400	70	256	39	disc file sorting	44	1.895	3.383
auto	5400	70	256	39	disc file	44	1.774	3.034
cross	5400	70	256	39	disc file	44	~ 6	~ 5.3

SUBPROGRAMS REQUIRED: None

ERRORS AND DIAGNOSTICS:

<u>Message</u>	<u>Meaning</u>	<u>Action Taken</u>
"(PIECE LENGTH)/2 MUST BE EVEN"	Self-explanatory	abort
"CANNOT HAVE MORE THAN TWO INPUT SERIES"	User has specified 3 distinct logical files for first three entries on Data Card 1	abort

PROGRAMMER: Gerard H. Martineau

ORIGINATOR: Melbourne G. Briscoe

DATE: July 1977

REFERENCES:

- (1) Report on Program BPLLOT, by G. H. Martineau.
- (2) "Handbook for Computer Users," Information Processing Center of W.H.O.I., pp. V-E-1 ff.
- (3) Report on Program FOURIER, by G. H. Martineau.
- (4) "Bispectral and Cross-Bispectral Analysis of Wind and Currents off Oregon Coast," Ph.D. Thesis at Oregon State University, by N. C. G. Yao, June, 1974.

APPENDIX 1
(BISCAL)

17:43 JAN 26, '78 ID=04F2

JOB 462,1719

LIMIT (TIME,2),(CARE,28),(ORDER),(ACCRUNT)
 ASSIGN F:RAD,(FILE,BVECI1),(DIRECT),(KEYED)
 LOAD (EF,(BISCALOR)),(UNSAT,(3))
 :P1 ASSOCIATED.

* * ALLOCATION SUMMARY * *

PROTECTION LOCATION PAGES

DATA (00) A000 7
 PROCEDURE (01) R000 3
 DCB (10) AE00 1
 ***** SGN= D2L SIZE=0064K *****

***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC

SEGL1=0 AC78 SEGL1=1 RE61 SEGL1=2 AFFF
 SEGL9=0 A000 SEGL9=1 B000 SEGL9=2 AE00
 00 SIZE= C7C 01 SIZE= E62 10 SIZE= 200

RUN

IDATSTART = 5

MF = 21

ISTORE = 1

BINSIZE = 1.000000E-02

BICOLIM = 1.50000

IH1 = 1

KB1 = 1

KB2 = 1

K0NF = 1

LISTBI = 1

IDUMP = 0

INORM = 0

IWANN = 1

1 1 1 10 64 1 10 .545 .575

4 .90 .95 .99 .999

ADJUSTED BICOHERENCE LIMIT, HALDING BINSIZE AS INPUT 1.50

F1	F2	F3	BIC	RIPH	RIMD	BISPECR	BISPECI	AUT9F1	AUT9F2	AUT9F1F2	SD	PHSERR
6	2	8	.568	73	.610E-02	.183E-02	.582E-02	.558E-01	.353E-01	.586E-01	.107E-01	52.2
6	5	11	.546	114	.745E-02	.301E-02	.681E-02	.558E-01	.561E-01	.594E-01	.136E-01	
8	4	12	.553	35	.623E-02	.508E-02	.361E-02	.586E-01	.432E-01	.502E-01	.113E-01	
20	9	29	.567	-150	.135E-01	.117E-01	.667E-02	.113E 00	.874E-01	.574E-01	.238E-01	52.3

BIN NO BIN LIMITS NO OF BIC'S

BIN NO	BIN LIMITS	NO OF BIC'S
1	.00	1
2	.01	0
3	.02	1
4	.03	4
5	.04	2
6	.05	4
7	.06	3
8	.07	0
9	.08	4

NO OF RICS

BIN LIMITS

BIN NO

10	.09	2
11	.10	3
12	.11	2
13	.12	9
14	.13	4
15	.14	4
16	.15	4
17	.16	6
18	.17	6
19	.18	11
20	.19	8
21	.20	4
22	.21	5
23	.22	8
24	.23	1
25	.24	5
26	.25	7
27	.26	4
28	.27	12
29	.28	8
30	.29	6
31	.30	7
32	.31	2
33	.32	9
34	.33	5
35	.34	5
36	.35	8
37	.36	8
38	.37	2
39	.38	3
40	.39	9
41	.40	4
42	.41	5
43	.42	2
44	.43	7
45	.44	5
46	.45	4
47	.46	3
48	.47	3
49	.48	1
50	.49	1
51	.50	3
52	.51	2
53	.52	1
54	.53	1
55	.54	1
56	.55	2
57	.56	1
58	.57	2
59	.58	1
60	.59	1
61	.60	3
62	.61	1
63	.62	0
64	.63	0
65	.64	1
66	.65	1
67	.66	0
68	.67	0
69	.68	0
	.69	0

70	.69	.70	0
71	.70	.71	1
72	.71	.72	1
73	.72	.73	1
74	.73	.74	0
75	.74	.75	0
76	.75	.76	0
77	.76	.77	0
78	.77	.78	0
79	.78	.79	0
80	.79	.80	0
81	.80	.81	0
82	.81	.82	0
83	.82	.83	1
84	.83	.84	1
85	.84	.85	0
86	.85	.86	0
87	.86	.87	0
88	.87	.88	0
89	.88	.89	1
90	.89	.90	0
91	.90	.91	0
92	.91	.92	0
93	.92	.93	0
94	.93	.94	0
95	.94	.95	0
96	.95	.96	1
97	.96	.97	0
98	.97	.98	0
99	.98	.99	0
100	.99	1.00	0
101	1.00	1.01	0
102	1.01	1.02	0
103	1.02	1.03	0
104	1.03	1.04	0
105	1.04	1.05	0
106	1.05	1.06	0
107	1.06	1.07	0
108	1.07	1.08	0
109	1.08	1.09	0
110	1.09	1.10	0
111	1.10	1.11	0
112	1.11	1.12	0
113	1.12	1.13	0
114	1.13	1.14	0
115	1.14	1.15	0
116	1.15	1.16	0
117	1.16	1.17	0
118	1.17	1.18	0
119	1.18	1.19	0
120	1.19	1.20	0
121	1.20	1.21	0
122	1.21	1.22	0
123	1.22	1.23	0
124	1.23	1.24	0
125	1.24	1.25	0
126	1.25	1.26	0
127	1.26	1.27	0
128	1.27	1.28	0
129	1.28	1.29	0

NO OF BIC'S

BIN LIMITS

BIN NO

130	1.29	1.30	0
131	1.30	1.31	0
132	1.31	1.32	0
133	1.32	1.33	0
134	1.33	1.34	0
135	1.34	1.35	0
136	1.35	1.36	0
137	1.36	1.37	0
138	1.37	1.38	0
139	1.38	1.39	0
140	1.39	1.40	0
141	1.40	1.41	0
142	1.41	1.42	0
143	1.42	1.43	0
144	1.43	1.44	0
145	1.44	1.45	0
146	1.45	1.46	0
147	1.46	1.47	0
148	1.47	1.48	0
149	1.48	1.49	0
150	1.49	1.50	0

TOTAL NO OF BIC'S SUBMITTED: 256

FRACT BELOW
UPPER BIN LIM

BIN LIMITS

BIN NO

1	.00	.01	.004
2	.01	.02	.004
3	.02	.03	.008
4	.03	.04	.023
5	.04	.05	.031
6	.05	.06	.047
7	.06	.07	.059
8	.07	.08	.059
9	.08	.09	.074
10	.09	.10	.082
11	.10	.11	.094
12	.11	.12	.102
13	.12	.13	.137
14	.13	.14	.152
15	.14	.15	.168
16	.15	.16	.184
17	.16	.17	.207
18	.17	.18	.230
19	.18	.19	.273
20	.19	.20	.305
21	.20	.21	.320
22	.21	.22	.340
23	.22	.23	.371
24	.23	.24	.375
25	.24	.25	.395
26	.25	.26	.410
27	.26	.27	.438
28	.27	.28	.484
29	.28	.29	.516
30	.29	.30	.539
31	.30	.31	.566
32	.31	.32	.574

33	.32	.33	.609
34	.33	.34	.629
35	.34	.35	.648
36	.35	.36	.680
37	.36	.37	.711
38	.37	.38	.719
39	.38	.39	.730
40	.39	.40	.766
41	.40	.41	.781
42	.41	.42	.801
43	.42	.43	.809
44	.43	.44	.836
45	.44	.45	.855
46	.45	.46	.871
47	.46	.47	.883
48	.47	.48	.887
49	.48	.49	.891
50	.49	.50	.902
51	.50	.51	.910
52	.51	.52	.914
53	.52	.53	.918
54	.53	.54	.922
55	.54	.55	.930
56	.55	.56	.934
57	.56	.57	.941
58	.57	.58	.945
59	.58	.59	.949
60	.59	.60	.961
61	.60	.61	.965
62	.61	.62	.965
63	.62	.63	.965
64	.63	.64	.969
65	.64	.65	.973
66	.65	.66	.973
67	.66	.67	.973
68	.67	.68	.973
69	.68	.69	.973
70	.69	.70	.973
71	.70	.71	.977
72	.71	.72	.980
73	.72	.73	.984
74	.73	.74	.984
75	.74	.75	.984
76	.75	.76	.984
77	.76	.77	.984
78	.77	.78	.984
79	.78	.79	.984
80	.79	.80	.984
81	.80	.81	.984
82	.81	.82	.984
83	.82	.83	.988
84	.83	.84	.992
85	.84	.85	.992
86	.85	.86	.992
87	.86	.87	.992
88	.87	.88	.992
89	.88	.89	.996
90	.89	.90	.996
91	.90	.91	.996
92	.91	.92	.996

FRACT BELOW
UPPER BIN LIM

BIN LIMITS

BIN NO

93	.92	.93	.996
94	.93	.94	.996
95	.94	.95	.996
96	.95	.96	1.000
97	.96	.97	1.000
98	.97	.98	1.000
99	.98	.99	1.000
100	.99	1.00	1.000
101	1.00	1.01	1.000
102	1.01	1.02	1.000
103	1.02	1.03	1.000
104	1.03	1.04	1.000
105	1.04	1.05	1.000
106	1.05	1.06	1.000
107	1.06	1.07	1.000
108	1.07	1.08	1.000
109	1.08	1.09	1.000
110	1.09	1.10	1.000
111	1.10	1.11	1.000
112	1.11	1.12	1.000
113	1.12	1.13	1.000
114	1.13	1.14	1.000
115	1.14	1.15	1.000
116	1.15	1.16	1.000
117	1.16	1.17	1.000
118	1.17	1.18	1.000
119	1.18	1.19	1.000
120	1.19	1.20	1.000
121	1.20	1.21	1.000
122	1.21	1.22	1.000
123	1.22	1.23	1.000
124	1.23	1.24	1.000
125	1.24	1.25	1.000
126	1.25	1.26	1.000
127	1.26	1.27	1.000
128	1.27	1.28	1.000
129	1.28	1.29	1.000
130	1.29	1.30	1.000
131	1.30	1.31	1.000
132	1.31	1.32	1.000
133	1.32	1.33	1.000
134	1.33	1.34	1.000
135	1.34	1.35	1.000
136	1.35	1.36	1.000
137	1.36	1.37	1.000
138	1.37	1.38	1.000
139	1.38	1.39	1.000
140	1.39	1.40	1.000
141	1.40	1.41	1.000
142	1.41	1.42	1.000
143	1.42	1.43	1.000
144	1.43	1.44	1.000
145	1.44	1.45	1.000
146	1.45	1.46	1.000
147	1.46	1.47	1.000
148	1.47	1.48	1.000
149	1.48	1.49	1.000
150	1.49	1.50	1.000

CONFIDENCE

LEVEL (PERCENT)	BICOHERENCE
90.0	.498
95.0	.591
99.0	.834
99.9	.957

NO. OF BICOHERENCES GREATER THAN SPECIFIED MAXIMUM OF 1.50 IS 0

STOP NORMAL PROGRAM COMPLETION

NAME: BPLOT

PURPOSE: To construct auto or cross-bicoherence
plot from disc file produced by program
BISCAL⁽¹⁾

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BPLOT reads a consecutive file of auto or cross-bicoherences produced by program BISCAL and then makes a plot of these bicoherences over the appropriate domain.

USAGE:

(1) Control cards

Let aaa = account number
 uuu = user number
 fff = input file name
 ppp = plot file name

Assume element file BPLOTR (object of BPLOT) is in account aaa. Assume any plotting is on Versatec or graphics terminal. Then the Control Card portion of the input deck is:

```
!JOB aaa,uuu
!LIMIT (TIME,2),(CORE,38)
!ASSIGN F:1,(FILE,fff)
*!ASSIGN F:95,(FILE,ppp),(OUT),(SAVE)
!LOAD (EF,(BPLOTR),(PLOTDFER,3)),(UNSAT,(3))
!RUN
!DATA
      → (DATA CARDS)
†!PLOTV
```

* - Necessary only if plot file is to be saved after this job.

† - Necessary only for a Versatec plot.

(2) Data Cards

Program BPLOT has NAMELIST input followed by three data cards. The NAMELIST input must be terminated by an * card whether or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are as follows:

<u>NAMELIST variable</u>	<u>Meaning</u>	<u>Default Value</u>
ISTORE	DCB assignment for input consecutive file	1
NDECPL	Format control for axis annotation: >0: Number of digits to the right of the decimal point which are plotted, after proper rounding =0: Only integer portion of number and decimal point are plotted, after rounding =-1: Only integer portion of number is plotted, after rounding <-1: NDECPL -1 digits are truncated from the integer portion after rounding	1
WIDTH	Width, in inches, of plot of L/2 points, where L=pc. length	8.75
CMPPT	Spacing in centimeters between adjacent data (grid) points on plot. If entered by user, this will override any specification of variable WIDTH	$WIDTH * 2.54 / (L/2)$ (L=pc. length)
IHISTORY	Switch to include information about data origin on plot; i.e., original buoy format file names and variable numbers, and to print processing history on plot (from program FOURIER) IHISTORY=1 for "yes", 0 for "no"	1

The data cards follow the * card which terminates NAMELIST input. They are all in generalized, free-field format and are as follows:

Card 1

<u>Variable No.</u>	<u>Meaning</u>
1	{ 1 for auto-bicoherence 2 for cross-bicoherence
2	
	piece length

Card 2

<u>Variable No.</u>	<u>Meaning</u>
1 (SAMPSEC)	Sample interval in seconds
2 (FREQTIC)	*Distance between tic marks on plot in $(\text{hr})^{-1}$
3 (NCONLV)	Number of contour levels to follow (≤ 10)
4 thru (3+NCONLV)	NCONLV contour levels for plot
(4+NCONLV) thru (3+2•NCONLV) }	NCONLV confidence levels (percent) for contour levels

Card 3

Up to 72 characters of identification, which will appear immediately above the plot. The first 36 characters will appear on one line, and the next 36 characters on the next line. The user is responsible for a proper transition of text from one line to the next, occurring between columns 36 and 37.

OUTPUT:(1) Line Printer

Sample line printer output appears in Appendix 1. A facsimile of the three input cards is printed, followed by a listing of the contour levels from Card 2 under a separate heading, "CONTOUR LEVELS."

(2) Plots

Either a disc plot file or a Calcomp plot tape can be output by the program. See Ref. 2 for the latter. By using the PLOTV or PLOTTC processor, Versatec or Calcomp plots can be produced either in the same job as BPLOT or separately. Additionally, the Versatec plot file can be displayed on a graphics terminal via the PLOTT processor.

Two sample reduced Calcomp plots constitute Appendices 2 and 3, which display auto and cross-bicoherences respectively. The notation is self-explanatory.

*Full-scale frequency (Nyquist frequency) is $1800/\text{SAMPSEC}$. Variable FREQTIC does not have to divide evenly into the full-scale frequency.

RESTRICTIONS:

- (1) Program BPLOT uses a large two-dimensional buffer, "DATA", which can be thought of as a matrix whose elements correspond one-for-one with the grid points of the plot, and whose rectangular "boundary" completely contains the plot, the non-data points being padded with flags to that effect. The dimensions of DATA therefore govern the maximum piece length which can be plotted. The rule is as follows: If L is the piece length, then array DATA must be dimensioned at least DATA(R,C) where:

$$C \geq L/2$$

$$R \geq 1.5C + 1$$

The "standard" dimensions of DATA are DATA(200,130). If recompilation is necessary, then the only other change which must be made is to the initialization of variables MD1 and MD2 in the statement following the DIMENSION statement. MD1, MD2 must be the first and second dimensions of DATA respectively.

- (2) The number of contour levels must be ≤ 15 .

SUBPROGRAMS REQUIRED: PLOTDFER, for a Versatec or graphics terminal plot file.

STORAGE REQUIREMENTS: In a typical BPLOT run in which BPLOT plus PLOTDFER were loaded, and array DATA in BPLOT was dimensioned DATA (200,130), a peak core of 75 512-word pages were used.

TIMING: In the run just cited, 0.397 min of CPU time was used.

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: July 1977

REFERENCES: (1) Report on Program BISCAL, by Gerard H. Martineau, Woods Hole Oceanographic Inst.
(2) Handbook for Computer Users, Information Processing Center of Woods Hole Oceanographic Inst.

APPENDIX 1
(B PLOT)

13:46 DEC 30, '77 ID=0929
 JOB 460,1719
 LIMIT (TIME,2),(CORE,38)
 ASSIGN F:1,(FILE,TIGOUT)
 ASSIGN F:95,(FILE,BISC1),(OUT),(SAVE)
 LOAD (EF,(BPL0TR),(PL0TDFR,3)),(UNSAT,(3))
 :P1 ASSOCIATED.

* * ALLOCATION SUMMARY * *

PROTECTION	LOCATION	PAGES
DATA (00)	A000	3C
PROCEDURE (01)	11A00	6
DCB (10)	11800	1

***** SGN= IZL SIZE=032.3K *****

***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC

SEGHI=0 116F9	SEGHI=1 1254B	SEGHI=2 119FF
SEGL0=0 A000	SEGL0=1 11A00	SEGL0=2 11800
00 SIZE= 76FA	01 SIZE= B4C	10 SIZE= 200

RUN
 1 256
 225 0.8 3 .42 .52 .637 95. 99. 99.9
 IWEX B10EAST

CONTOUR LEVELS:
 .420 .520 .637
 STOP NORMAL PROGRAM COMPLETION

APPENDIX 2
(BPLOT)

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENRANOUTPUT 0
 FREQUENCY 2: GENRANOUTPUT 0
 FREQUENCY 3: GENRANOUTPUT 0

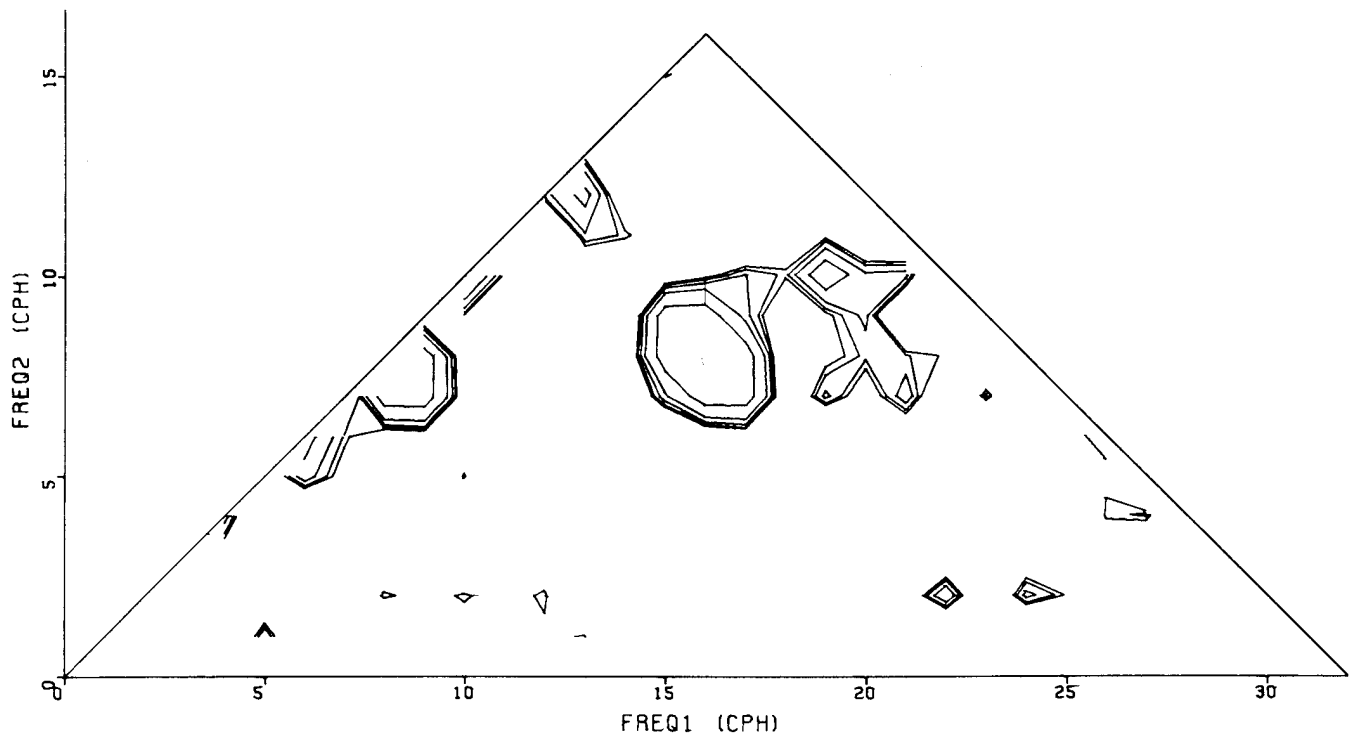
PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
 *NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT: 0.420 80.0
 09:06 DEC 22, '77 0.447 85.0
 0.498 90.0
 0.591 95.0
 0.834 99.0

S8+S16+S24 NYQUIST=32 HANNED
 FILE BVPL11 S/N=10



APPENDIX 3
(BPLOT)

CROSS BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:
 FREQUENCY 1: GENARNOOUTPUT 0
 FREQUENCY 2: GENARNOOUTPUT 0
 FREQUENCY 3: GENARNOOUTPUT 0

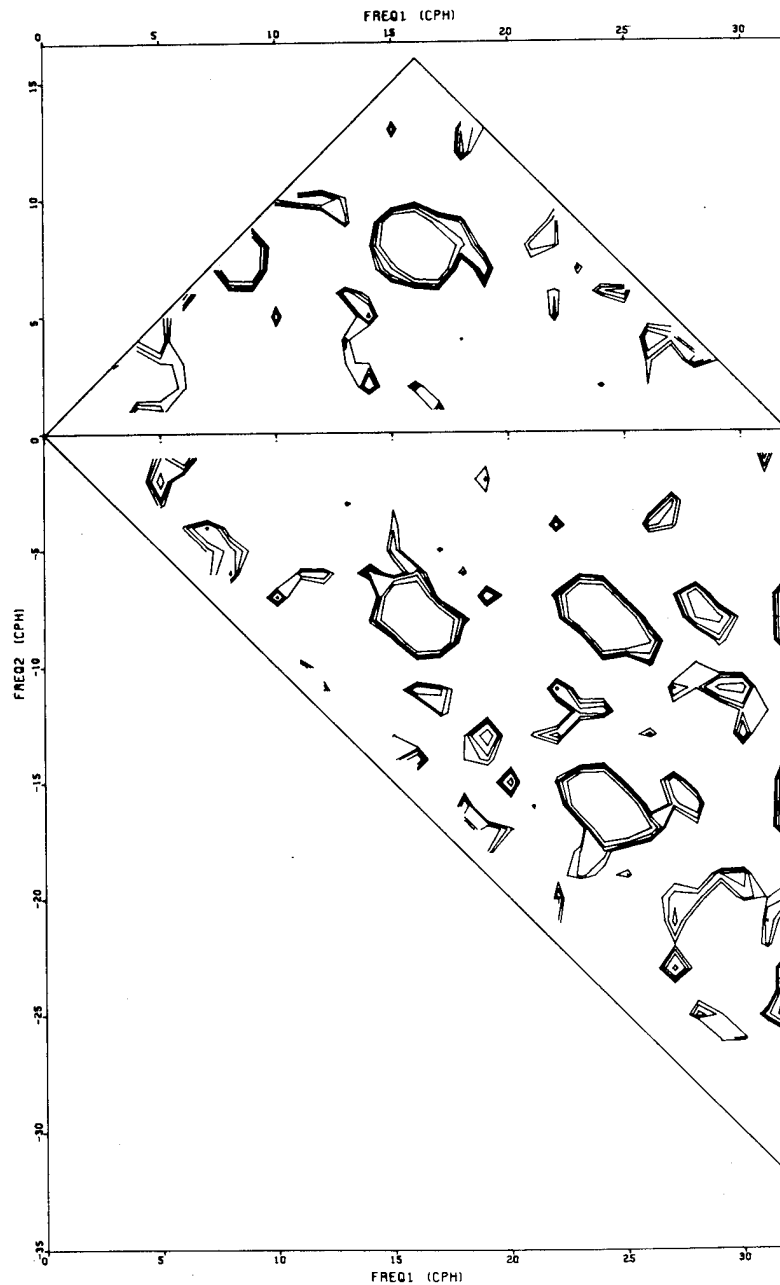
PROCESSING HISTORY

NO LF'S N=NO MOS DATA SS2=PREM D=SUBSRAP D=PC SIZE 84
 WND PCS 10=OLAP 1=ISUBT D=HANN 1 WCREATED 17:18 DEC 15, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.400 80.0
 TIME OF PLOT: 0.431 85.0
 09:38 DEC 22, '77 0.468 90.0
 0.551 95.0
 0.675 99.0

X=C8+C16+C24;Y=-S8-S16-S24;FORM XXY
 NYQ=32 HANNED S/N=10 FILE BVPL3



NAME: BISUM

PURPOSE: To integrate certain bispectral quantities along paths of constant sum frequency in the $\omega_1 - \omega_2$ plane.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BISUM is similar to the bispectral program BISCAL,¹ and the user is referred to the report on BISCAL for computational procedures. The difference in the two programs is that the sorting capability of BISCAL has been replaced in BISUM by summations of bicoherences and other quantities related to bispectra, over paths of constant sum frequencies (ω_3) in the $\omega_1 - \omega_2$ plane. The paths are diagonal straight lines of slope (-1). The summing is done separately above and below the ω_1 axis; i.e., for positive and negative ω_2 . A formatted listing of these integrals is presented. The integrals may additionally be written to a disc file. It is also possible to write the same file of bicoherences that is written by BISCAL for transmittal to the plotting program BPLOTT.²

INPUT:

Input consists of the same type of RWDISC file of Fourier coefficients as is used by program BISCAL.¹

OUTPUT: (I) Line Printer

A sample of the printed output appears in Appendix 1. First the values of all NAMELIST variables are listed, along with a facsimile of the input data card. This is followed by a formatted list of integrals.

The integrals presented in the case of calculation of bicoherences by Form 1 as described in the program BISCAL report, are as follows:

Let N be the number of points over the path of integration.

- (1) BR : real part of bispectrum at frequency ω_3
- (2) BR/N
- (3) BI : imaginary part of bispectrum at frequency ω_3
- (4) BI/N
- (5) BIC : bicoherence¹
- (6) BIC/N
- (7) (BIC)²
- (8) (BIC)²/N
- (9) $S(\omega_3) * (BIC)^2$ where $S(\omega_3)$ is square of auto-spectrum at frequency ω_3
- (10) $(S(\omega_3) * (BIC)^2)/N$

In the case of calculation of bicoherences of Form 2 (from amplitude-normalized bispectrum) in the program BISCAL report, all of the above quantities corresponding to the amplitude-normalized bispectrum are calculated, except (9) and (10) which are listed as zero.

It will be noted that both the sum frequency sequence number ($|\omega_3|$) and the number of independent variable values that enter each integral are printed for each line of output, each line corresponding to a single value of $|\omega_3|$.

If L is the piece length, there are (L/2-1) lines of output for an auto-bispectrum and twice that many for a cross-bispectrum.

The user may also get a listing of bicoherences and related quantities between limits which he can set. This listing is described in the report on program BISCAL.¹ The limits are set in a data card to be described in the "USAGE" section.

(II) Disc

Two types of disc files may be written by program BISUM. The first is the same type of file of bicoherences written by BISCAL¹ for transmittal to plotting program BPLOT. The second is a consecutive file containing all integrals computed by BISUM. This file is organized as follows (L = piece length):

cross- bispectrum	{	auto- bispectrum	{	1 EBCID header record followed by (L/2-1) binary records followed by (L/2-1) binary records
----------------------	---	---------------------	---	---

This file should be read by a FORTRAN formatted READ statement (for the header record, to fill an integer array dimensioned 33), followed by one FORTRAN binary READ statement for each line of printed output. Each READ statement should read 13 words. If the user wishes to access words 1, 2, or 13, he is notified that these are integer type, all others being real.

USAGE: (I) Input Card Deck

1) Control Cards

```
Let aaa      = account number
    uuu      = user number
    INFILE   = input file of FOURIER coefficients
    OUTBIC   = output file of bicoherences
    OUTSUMS  = output file of integrals
    BISUMR   = object file of program BISUM in
                account aaa
```

The control card deck can appear as follows:

```
!JOB aaa,uuu
* !LIMIT (TIME,4), (CORE,50)
!ASSIGN F:RAD, (FILE,INFILE), (DIRECT), (KEYED)
† !ASSIGN F:1, (FILE,OUTBIC), (OUT), (SAVE)
†† !ASSIGN F:2, (FILE,OUTSUMS), (OUT), (SAVE)
!LOAD (EF, (BISUMR)), (UNSAT, (3))
!RUN
!DATA
      → Data cards
```

```
* Core limit can vary greatly with dimension-
  ing of working arrays. See below.
† Necessary only when outputting file of
  bicoherences.
†† Necessary only when outputting file of
  integrals.
```

2) Data Cards

Data card input format is similar to that for program BISCAL.¹ There is NAMELIST input, followed by an * card whether or not NAMELIST parameters are entered, followed by a single data card in free-field, generalized format.

The NAMELIST parameters follow, with their default values. "Switches" are set to 0 for "off", 1 for "on":

<u>Variable</u>	<u>Meaning</u>	<u>Default Value</u>
MF	Number of logical files in input RWDISC file of FOURIER coefficients	21
IDATSTART	Location of first word of data in each logical file of input file	5
INORM	Switch to calculate amplitude- normalized bicoherences	0
LISTBI	Switch to list bicoherences and related quantities ¹ within user- specified limits	0
IDUMP	Switch to output file of bicoher- ences for transmittal to plotting program BPLOT	0
ISTORE	DCB assignment for output file of bicoherences created by setting switch IDUMP	1
KDISC	Switch to output file of integrals	0
KDCB	DCB assignment for output file of integrals	2
IHANN	Determines method of calculating phase error when LISTBI≠0 (See PHSERR on page 12 of BISCAL report). IHANN = 0 not Hanned and not over- lapped IHANN=1 Hanned and overlapped	1
SIGLVL	95% confidence level for bicoherences, above which phase error PHSERR is calculated and printed when LISTBI=1	†

The single data card, which follows the * NAMELIST terminator card has the following entries in free-field, generalized format:

† Default is an approximate value. If P = number of pieces, let the number of degrees of freedom EDOF be defined as follows:

$$EDOF = \begin{cases} 36P^2/(19P-1) & \text{for Hanning, overlapping} \\ 2P & \text{for no Hanning, no overlapping} \end{cases}$$

Then the default value of SIGLVL is:

$$SIGLVL = (6/EDOF)^{1/2}$$

<u>Variable No.</u>	<u>Meaning</u>
*1	Logical file in input RWDISC file associated with frequency ω_1
*2	Logical file associated with ω_2
*3	Logical file associated with $\omega_3 = -\omega_1 - \omega_2$
4	Number of pieces in input series of Fourier coefficients
5	Piece length
6 } 7 }	Two piece sequence numbers between which it is desired to average, inclusively
8 } 9 }	Lower, upper limits determining when to print bicoherence when LISTBI $\neq$ 0. Bi-coherences must satisfy: (var. #9) < bicoherence $\leq$ (var. #10)

(II) Altering Load Module Size

As in BISCAL, program BISUM attempts to minimize disc accesses in the same way.¹ This involves the use of two large buffers which largely determine the size of the load module. There may be worthwhile benefits to the user, in terms of page-minute charges and turn-around time, in recompiling the program using buffers which are no longer than needed.

The user is therefore advised to recompile the program to match his needs if he is going to be handling a number of data series of comparable length. To edit the program prior to recompilation, the following steps should be observed:

- a. Copy the source file to a duplicate file, and then work with the duplicate file;
- b. Change the DIMENSION's of OM1 and OM2 arrays to at least $[(L/2+2)(NP)]$, where L = piece length and NP is the number of pieces.

* If variables 1, 2, 3 are all equal, then auto-bispectra will be computed and the minimum necessary domain will be used.

- c. Check that array W3 is DIMENSIONed at least 2(NP).
- d. Check that SQ is DIMENSIONed at least SQ(L/4 + 1, L/4 + 1) (both dimensions identical).
- e. Check that the initialization of variable ISQD is identical to both dimensions of SQ.
- f. There are ten arrays used to accumulate sums of bicoherences and related quantities. These must be all DIMENSIONed at least (L/2). The arrays are BRU, BIU, BICU, BIC2U, BCSU, BRL, BIL, BICL, BIC2L, and BCSL.
- g. Variable JDIM must be initialized at the dimension of the arrays in (f).
- h. Recompile.

A further reduction in size of the program can be achieved if the user does not wish to produce a file of bicoherences. Array SQ has two equal dimensions, which must be at least (L/2+1) if this file is produced. If there is to be no file of bicoherences, then SQ can be dimensioned SQ(2,2). Variable ISQD must be initialized at one of the dimensions of SQ.

RESTRICTIONS:

1. The length of the data series which program BISUM can handle depends on the dimensions of OM1, OM2, and array SQ, as well as on how the data series was processed in the FOURIER transform. The following relations can be used:

Let $\mathcal{L}$ = data series length adjusted so that $2\mathcal{L}$ is evenly divisible by L, the piece length

L = piece length

D = dimensions of OM1 and OM2

NP = number of pieces

50% overlap

$$\mathcal{L} = \frac{L}{2} \left(\frac{2D + L + 4}{L + 4} \right) \text{ and } NP = \frac{2\mathcal{L}}{L} - 1 \quad (1.)$$

$$\mathcal{L} = \frac{2DL}{L + 4} \text{ and } NP = \frac{\mathcal{L}}{L} \quad (2.)$$

In both cases the number of points in the series entering BISUM is $(L)(NP)$, and OM1, OM2 must be dimensioned at least $(NP)(L/2 + 2)$.

The latter requirement has been incorporated in the derivation of expressions 1. and 2.

Example. Program BISUM has been successfully run with $D = 16400$ and SQ dimensioned at SQ(70,70). Assuming 50% overlap, Eq. (7) above yields $L = 16256$ as the maximum data series length which could have been handled. This is subject to restriction (2) below, which requires that SQ should be dimensioned at least SQ(M,M) where $M = L/4 + 1$, a condition which is met here.

2. Array W3 must be DIMENSIONED at least $2(NP)$.
3. The dimension of array SQ must be at least SQ(M,M) where $M = L/4 + 1$, $L =$ piece length.
4. Variable ISQD must be initialized at either dimension of SQ in (3).
5. The arrays used to accumulate sums of bicoherences and related quantities must all be DIMENSIONED at least $(L/2)$. These arrays are BRU, BIU, BICU, BIC2U, BCSU, BRL, BIL, BICL, BIC2L, and BCSL.
6. Variable JDIM must be initialized to the dimension of the arrays in 5.
7. L must be an even number with no prime factor greater than 5, and evenly divisible by 4.

SUBPROGRAMS REQUIRES: None

ERRORS AND DIAGNOSTICS:

<u>Message</u>	<u>Meaning</u>	<u>Action Taken</u>
"(PIECE LENGTH)/2 MUST BE EVEN"	Piece length must be evenly divisible by four	Abort
"CANNOT HAVE MORE THAN TWO INPUT SERIES"	Three distinct logical files have been referenced by first three entries on data card	Abort

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: July, 1977

REFERENCES:

1. Report on Progrqm BISCAL, by G. H. Martineau, Woods Hole Oceanographic Institution.
2. Report on Program BPLOT, by G. H. Martineau, Woods Hole Oceanographic Insitution.

APPENDIX 1
(BISUM)

15:18 JAN 26, '78 ID=0462
 J98 462,1719
 LIMIT (TIME,2),(CORE,35),(ORDER),(ACCOUNT)
 ASSIGN FIRAD,(FILE,BVEC11),(DIRECT),(KEYED)
 LOAD (EF,(BISUM02R)),(UNSAT,(3))
 :P1 ASSOCIATED.

*** ALLLOCATION SUMMARY ***

PROTECTION LOCATION PAGES

DATA (00) A000 6
 PROCEDURE (01) B000 8
 DCB (10) AC00 2

***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC SIZE=007.0K *****

***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC
 SEGHI=0 A8BD SEGHI=1 BF65 SEGHI=2 AFFF
 SEGL=0 A000 SEGL=1 B000 SEGL=2 AC00
 00 SIZE= BBE 01 SIZE= F66 10 SIZE= 400

RUN
 IDATSTART = 5
 MF = 21
 ISTORE = 0
 BINSIZE = 1.000000E+02
 BIC0HLIM = 1.50000
 IHI = 1
 KB1 = 1
 KB2 = 1
 LISTBI = 0
 IDUMP = 0
 INARM = 0
 IHANN = 1
 KDISC = 0
 KDCB = 2
 4 4 7 10 64 1 10 .0001 1000.

INTEGRALS OVER POSITIVE FREQUENCIES ALONE

SUM FREQ SEQ NO	NO IN SCAN (N)	BR	BR/N	BI	BI/N	BIC	PIC/N	BIC**2	(BIC**2)/N	S(W3)* BIC**2	(S(W3))* BIC**2	SUM FREQ SEQ NO
2	1	..720E-02	..720E-02	..191E-01	..191E-01	.651	.651	.124	.424	.149E+01	.149E+01	2
3	1	..181E-02	..181E-02	..247E-02	..247E-02	.156	.156	.243E-01	.243E-01	.926E+03	.926E+03	3
4	2	..452E-03	..226E-03	..857E-02	..428E-02	.520	.260	.152	.761E-01	.657E+02	.329E+02	4
5	2	..167E-01	..837E-02	..388E-02	..194E-02	.750	.375	.315	.158	.177E+01	.889E+02	5
6	3	..362E-02	..121E-02	..144E-01	..479E-02	1.15	.383	.467	.156	.261E+01	.869E+02	6
7	3	.149	.497E-01	.491E-01	.164E+01	.750	.250	.197	.658E-01	3.28	1.09	7
8	4	..316	..204	1.59	.398	1.45	.363	.573	.143	38.5	9.61	8
9	4	..109	..272	2.49	.622	1.29	.323	.508	.127	8.54	2.14	9
10	5	..758E-02	..152E-02	..195	..391E-01	1.26	.251	.624	.849E-01	.250E+01	.499E+02	10
11	5	..208	..416E-01	.239	.479E-01	1.27	.254	.357	.713E-01	.212E+01	.424E+02	11
12	6	..287	..478E-01	.234E-01	.391E+02	1.85	.309	.742	.124	.372E+01	.621E+02	12
13	6	..186	..310E-01	..653E-01	..109E-01	1.10	.183	.245	.408E-01	.147E+01	.246E+02	13
14	7	..194	..277E-01	.465	.665E-01	1.47	.210	.413	.589E-01	.223E+01	.319E+02	14
15	7	..114	..163	135.	19.3	2.43	.347	1.43	.204	24.3	3.47	15
16	8	..162	..202	684.	85.5	3.69	.461	2.62	.328	176.	22.0	16
17	8	..474	..592	138.	17.3	2.55	.319	1.48	.185	24.5	3.06	17
18	9	.192	.213E-01	..1.22	..135	2.48	.276	.821	.912E-01	.344E+01	.383E+02	18
19	9	..231	..257E-01	.854E-01	.949E-02	2.27	.242	.838	.931E-01	.485E+01	.533E+02	19
20	10	..279	..273E-01	..191E-01	..191E-02	2.13	.213	.613	.613E-01	.691E+01	.691E+02	20
21	10	..214	..214E-01	..110	..110E-01	2.56	.256	.786	.786E-01	.714E+01	.714E+02	21
22	11	1.84	.167	..143	..130E-01	3.33	.303	1.30	.118	.612E+01	.556E+02	22
23	11	2.74	.249	268.	24.4	3.94	.358	2.52	.229	42.7	3.89	23
24	12	..9.98	..832	815.	67.9	4.98	.415	3.48	.290	236.	19.7	24
25	12	..6.45	..538	273.	22.8	4.02	.335	2.51	.209	42.0	3.50	25
26	13	..2.26	..174	..402	..309E-01	4.13	.318	2.51	.122	.877E+01	.674E+02	26
27	13	..305	..158E-01	.142	.109E-01	3.32	.256	1.02	.783E-01	.667E+01	.513E+02	27
28	14	.382	.273E-01	..163	..116E-01	3.42	.244	1.09	.776E-01	.579E+01	.414E+02	28
29	14	.138	.983E-02	.563E-01	.402E-02	3.12	.223	.859	.614E-01	.493E+01	.352E+02	29
30	15	..3.78	..252	1.60	.106	5.04	.336	1.97	.132	.134	.890E+02	30
31	15	..5.12	..341	..182	..121E-01	4.79	.319	1.74	.116	.837E-01	.558E+02	31
32	16	..205	..128E-01	..403	..252E-01	3.27	.205	.987	.617E-01	.271E+01	.169E+02	32

INTEGRALS INVOLVING NEGATIVE FREQUENCIES

SUM FREQ SEQ NO	IN SCAN (N)	BR	AR/N	BI	BI/N	BIC	BIC/N	BIC**2	(BIC**2)/N	S(W3)* BIC**2	(S(W3))* BIC**2/N	SUM FREQ SEQ NO
1	31	2.90	.937E-01	-2.03	-.655E-01	7.81	.252	2.73	.882E-01	.201	.649E-02	1
2	30	1.22	.407E-01	-.972	-.324E-01	8.96	.299	3.40	.113	.120	.400E-02	2
3	29	-.176	-.608E-02	.268	.925E-02	8.73	.301	3.29	.114	.126	.433E-02	3
4	28	-.348	-.124E-01	.373	.133E-01	7.53	.269	2.51	.897E-01	.108	.387E-02	4
5	27	-.943	-.349E-01	-.444E-01	-.165E-02	9.01	.334	3.80	.141	.213	.790E-02	5
6	26	-2.42	-.930E-01	-.701	-.270E-01	7.94	.305	2.98	.115	.166	.440E-02	6
7	25	-4.35	-.174	536.	21.5	9.67	.387	6.13	.245	.102	.408	7
8	24	2.11	.479E-01	.160E 04	66.8	11.4	.474	8.18	.341	.549	.229	8
9	23	4.44	.193	528.	23.0	9.44	.411	6.37	.277	.107	.465	9
10	22	2.15	.977E-01	.400	.182E-01	6.41	.291	2.22	.101	.131	.594E-02	10
11	21	-.411	-.198E-01	-.192	-.913E-02	6.54	.311	2.36	.112	.140	.667E-02	11
12	20	-.367	-.183E-01	-.139E-02	-.695E-04	6.69	.335	2.67	.134	.134	.670E-02	12
13	19	.415E-01	.218E-02	.284	.150E-01	4.63	.244	1.31	.690E-01	.789E-01	.415E-02	13
14	18	.468	.260E-01	.190	.105E-01	3.86	.214	.966	.536E-01	.522E-01	.296E-02	14
15	17	2.06	.121	273.	16.1	6.00	.353	3.62	.213	.614	.361	15
16	16	-2.99	-.187	804.	50.2	6.95	.434	4.70	.294	.315	.197	16
17	15	-5.80	-.387	259.	17.3	6.47	.431	4.26	.284	.705	.470	17
18	14	.648E-01	.463E-02	-.736	-.526E-01	4.55	.325	1.76	.126	.738E-01	.527E-02	18
19	13	.218	.167E-01	-.368	-.283E-01	3.57	.274	1.20	.921E-01	.693E-01	.533E-02	19
20	12	.284	.237E-01	-.421	-.351E-01	3.35	.279	1.09	.911E-01	.123	.103E-01	20
21	11	-.544E-02	-.513E-03	-.466	-.423E-01	3.23	.293	1.27	.115	.115	.105E-01	21
22	10	.208	.208E-01	-.235	-.235E-01	2.10	.210	.586	.586E-01	.276E-01	.276E-02	22
23	9	-.598	-.665E-01	.969	.108	2.66	.295	1.23	.136	.208	.231	23
24	8	-1.27	-.159	-8.98	-1.12	2.46	.308	1.21	.151	.821	.103	24
25	7	-.139	-.198E-01	-.272	-.388	2.06	.294	1.04	.149	.175	.249	25
26	6	.511E-02	.852E-03	-.246E-02	-.409E-03	1.36	.226	.374	.623E-01	.206E-01	.344E-02	26
27	5	-.105E-01	-.211E-02	.807E-03	.161E-03	1.22	.244	.332	.663E-01	.217E-01	.435E-02	27
28	4	.470E-02	.118E-02	-.403E-02	-.101E-02	1.02	.254	.305	.762E-01	.162E-01	.406E-02	28
29	3	.779E-02	.260E-02	-.186E-02	-.620E-03	.874	.292	.282	.940E-01	.162E-01	.540E-02	29
30	2	.177E-01	.887E-02	-.299E-03	-.149E-03	.807	.404	.341	.171	.231E-01	.115E-01	30
31	1	.165E-02	.165E-02	-.293E-02	-.293E-02	.217	.217	.470E-01	.470E-01	.226E-02	.226E-02	31

STOP NORMAL PROGRAM COMPLETION

NAME: BIVEC

TYPE: Main Program

PURPOSE: To calculate auto and cross rotary bispectra of two vector series, to write disc files of rotary bicoherences for transmittal to plotting program RBPLOT,⁽¹⁾ and to compute confidence levels for the rotary bicoherences.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BIVEC uses as input an RWDISC⁽²⁾ file of Fourier coefficients having a particular order to be described below. These coefficients are assumed to result from two real data series per vector series, the real series constituting the scalar components. The coefficients are normally the output of program FOURIER,⁽³⁾ which has done any necessary prewhitening, subsampling, overlapping, mean and trend removal, and Hanning.

If L is the piece length and P is the number of pieces resulting from program FOURIER, there are $2 \cdot LP$ words input into BIVEC for each vector data series. Program BIVEC computes approximately $(3/2)(L/2)^2$ values of auto rotary bicoherence, or either $(3/2)(L/2)^2$ or $3(L/2)^2$ values of cross rotary bicoherence, the latter depending on the combination of sources of data. A normalization of rotary bicoherences by amplitude can be specified (see below). Corresponding rotary biphase and rotary biphase error are also computed if desired. In normal use, program BIVEC writes consecutive files of rotary bicoherences on the disc to be used as input to plotting program RBPLOT. The user can instruct the program to list rotary bicoherences and related quantities when the rotary bicoherences fall between limits set by the user. Finally, a listing of partitioning of rotary bicoherences into bins of specified size can be obtained, as well as a list of rotary bicoherences which correspond to specified confidence levels.

COMPUTATIONS PERFORMED:I. Rotary Bispectra

The default definitions of rotary bispectrum, rotary bicoherence, and rotary biphas used by program BIVEC are those given by N. C. G. Yao. (4), (5) From two real series, $u_1(t)$ and $u_2(t)$, construct the complex vector series $\vec{u}(t) = u_1(t) + iu_2(t)$. Let ω be the angular velocity of $\vec{u}(t)$ and let $\sigma = |\omega|$. Further let $A_1(\omega)$, $B_1(\omega)$ be the cosine and sine Fourier coefficients of real series $u_1(t)$ and $A_2(\omega)$, $B_2(\omega)$ be the corresponding quantities for real series $u_2(t)$.

One can represent the vector series $\vec{u}(t)$ by

$$\vec{u}(t) = u_1(t) + iu_2(t) = \sum_{\sigma=0}^{\infty} (U_+(\sigma)e^{i\sigma t} + U_-(\sigma)e^{-i\sigma t})$$

where the coefficients $U_+(\sigma)$ and $U_-(\sigma)$ are called by Yao "rotary Fourier coefficients," and they correspond respectively to positive and negative angular frequencies present in the vector series $\vec{u}(t)$. Yao shows that $U_+(\sigma)$ and $U_-(\sigma)$ are given by:

$$U_+(\sigma) = [A_1(\sigma) + B_2(\sigma)] + i[A_2(\sigma) - B_1(\sigma)]; \quad \omega = \sigma$$

$$U_-(\sigma) = [A_1(\sigma) - B_2(\sigma)] + i[A_2(\sigma) + B_1(\sigma)]; \quad \omega = -\sigma$$

in terms of Fourier coefficients of the component real series.

The rotary spectrum $\langle P(\omega)d\omega \rangle$ is then defined as:

$$\begin{aligned} \langle P(\omega)d\omega \rangle &= \langle U(\omega_1)U^*(\omega_2) \rangle && \text{when } \omega_1 = \omega_2 \\ &= 0 && \text{when } \omega_1 \neq \omega_2 \end{aligned} \quad (1)$$

and the rotary bispectrum $\langle RB(\omega_1, \omega_2)d\omega^2 \rangle$ as

$$\begin{aligned} \langle RB(\omega_1, \omega_2)d\omega^2 \rangle &= \langle U(\omega_1)U(\omega_2)U^*(\omega_3) \rangle && \text{when } \omega_1 + \omega_2 = \omega_3 \\ &&& \text{when } \omega_1 + \omega_2 \neq \omega_3 \end{aligned} \quad (2)$$

where $U(\omega)$ represents $U_+(\sigma)$ or $U_-(\sigma)$ depending on the sign of ω . In the largest frequency domain of interest (see below) there will be six possible interpretations of equation (2) in terms of magnitude σ .

The rotary bicoherence $Rbic(\omega_1, \omega_2)$ is defined as:

$$Rbic(\omega_1, \omega_2) = \frac{|\langle RB(\omega_1, \omega_2) d\omega^2 \rangle|}{[\langle P(\omega_1) \rangle \langle P(\omega_2) \rangle \langle P(\omega_3) \rangle d\omega^3]^{1/2}} \quad (3)$$

and the rotary biphas $R\phi(\omega_1, \omega_2)$ as:

$$R\phi(\omega_1, \omega_2) = \tan^{-1} \frac{\text{Im} \langle RB(\omega_1, \omega_2) d\omega^2 \rangle}{\text{Re} \langle RB(\omega_1, \omega_2) d\omega^2 \rangle} \quad (4)$$

If P is the number of pieces, the averaging procedure indicated in equations (1) and (2) leads to:

$$Rbic(\omega_1, \omega_2) = \frac{P^{1/2} \left| \sum_{k=1}^P RB_k(\omega_1, \omega_2) d\omega^2 \right|}{\left[\sum_{k=1}^P P_k(\omega_1) d\omega \sum_{k=1}^P P_k(\omega_2) d\omega \sum_{k=1}^P P_k(\omega_3) d\omega \right]^{1/2}} \quad (5)$$

where k identifies each piece.

Alternate form of rotary bicoherence

Program BIVEC can compute an alternate form of rotary bicoherence based on the following definition of rotary bispectrum normalized by its amplitude:

$$\langle RB'(\omega_1, \omega_2) d\omega^2 \rangle = \left\langle \frac{RB_k(\omega_1, \omega_2) d\omega^2}{|RB_k(\omega_1, \omega_2) d\omega^2|} \right\rangle \quad (6)$$

where $RB_k(\omega_1, \omega_2) d\omega^2$ is given by equation (2) applied to piece k .

Using this definition, the piece-averaged rotary bicoherence is now:

$$\begin{aligned}
 R_{bic}'(\omega_1, \omega_2) &= |\langle RB'(\omega_1, \omega_2) d\omega^2 \rangle| \\
 &= \frac{1}{P} \left| \sum_{k=1}^P \frac{RB_k(\omega_1, \omega_2) d\omega^2}{|RB_k(\omega_1, \omega_2) d\omega^2|} \right|
 \end{aligned}
 \tag{7}$$

Frequency domain

As stated above, program BIVEC can operate on one or two vector series. Let the first be represented by X and the second by Y. Let the juxtaposition of three symbols from the set {X,Y} refer to a particular choice of sources for Fourier coefficients of frequency $\omega_1, \omega_2, \omega_3$ respectively. It can be shown that there are only four distinct rotary bispectra that can be calculated from the two vector series X and Y:

- 1) auto rotary bispectrum for XXX
- 2) auto rotary bispectrum for YYY
- 3) cross rotary bispectrum for XXY
- 4) cross rotary bispectrum for XYY

It can further be shown that it is not necessary to let the constituent frequencies range over all possible values. The minimum necessary domain for cases 1), 2), 3) is:

$$\begin{aligned}
 \{A\} &= \{(|\omega_1| \geq |\omega_2|) \cap (-L/2 \leq \omega_1 < L/2) \cap \\
 &\quad (-\omega_1 - L/2 \leq \omega_2 \leq -\omega_1 + L/2) \cap (\omega_2 \neq 0) \cap (\omega_2 \neq -\omega_1)\}
 \end{aligned}$$

where $\omega_3 = \omega_1 + \omega_2$, L is the piece length, and ω is expressed in units of cycles per point. Pictorially, this is the unshaded area in figure 1. For case (4), however, it is necessary to approximately double this domain so that the shaded area in figure 1 is also included. The minimum necessary domain is now:

$$\begin{aligned}
 \{A'\} &= \{(-L/2 \leq \omega_1 \leq L/2) \cap (-L/2 \leq \omega_2 \leq L/2) \cap \\
 &\quad (-\omega_1 - L/2 \leq \omega_2 \leq -\omega_1 + L/2) \cap (\omega_2 \neq 0) \cap (\omega_2 \neq -\omega_1)\}
 \end{aligned}$$

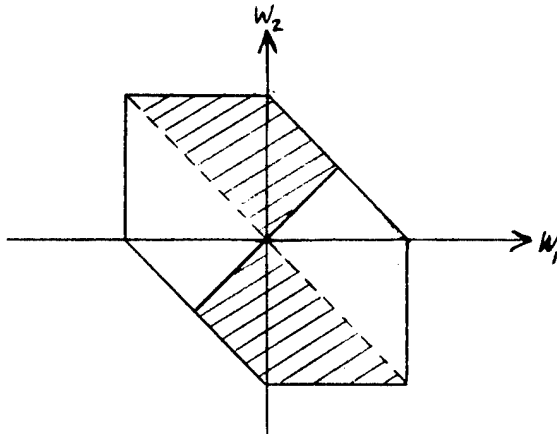


Figure 1

Order of the computations

Program BIVEC can handle two vector data series, each consisting of two scalar series each of length $16K$ words. This means that for 50% overlap and for the limiting case of an infinite number of pieces, each vector series entering the program consists of $64K$ Fourier coefficients. In order to permit computation of the bispectra, and to minimize the number of disc accesses, the following steps are taken:

1) Organization of input file

Let $\{a_i\}$ be the set of all real parts of the i th-frequency Fourier coefficient taken over all pieces, and let $\{b_i\}$ be the corresponding imaginary parts. Then, for a piece length of L , the input file elements have the following order:

$$\{a_0\}, \{a_1\}, \{b_1\}, \{a_2\}, \{b_2\}, \dots, \{a_{\frac{L}{2}-1}\}, \\ \{b_{\frac{L}{2}-1}\}, \{a_{\frac{L}{2}}\}.$$

The elements of $b_{\frac{L}{2}}$ are all zero and do not

appear in the input file. Program FOURIER (3) automatically outputs coefficients in this order. This permits all elements associated with a given frequency to be input by entering the series only once.

2) Organization of frequency domain and computation procedure

Figure 2 is a representation of 1/4 of the maximum domain described above.

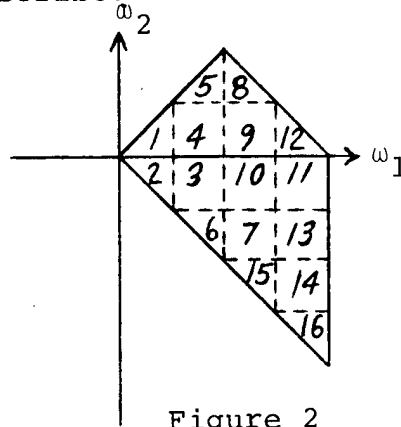


Figure 2

This area is shown divided into 16 sub-domains or "rasters." Program BIVEC calculates rotary bispectra for each raster in the order of numbering in fig. 2. Simultaneous with this calculation, is computation of rotary bispectra for the raster which is symmetric about the origin, having sum frequency ω_3 of opposite sign, and which uses the same set of Fourier coefficients. In this way, there are no more than $LP/4$ elements from each component scalar series in core at any time, where L = piece length, P = number of pieces. The reading of series "quarters" from the disc is a function of each particular combination of source series, an attempt being made to exploit symmetries for maximum efficiency. Within each raster, computations are done from upper left to lower right along diagonal scans of slope (-1) , each scan thus corresponding to a single sum frequency ω_3 . The disc is accessed for the coefficients of frequency ω_3 only when these coefficients are not already available in core.

Computations over the shaded area of fig. 1 are done similarly.

II. Confidence Levels

As each rotary bicoherence is computed, the user may elect to assign it to a bin having a size of his choice. The user may also determine the number of bins by specifying

the upper rotary bicoherence limit to be considered. After this sorting has been done, program BIVEC determines the rotary bicoherences below which specified fractions ("confidence levels") of the total number of rotary bicoherences calculated by the program lie. This is done by calculating the fraction of the total which lies below the upper limit of each bin, finding the two closest such fractions which "bracket" the specified confidence level, and linearly interpolating the two corresponding rotary bicoherences to find the one which corresponds to the specified confidence level. A dump of the distribution is available.

USAGE:

(1) Tailoring load module size

For long series of Fourier coefficients, it is desirable to have large data buffers to keep disc accesses to a minimum. On the other hand these large data buffers can be wasteful for shorter series both in terms of (page-minutes) charges and the penalty in turnaround time. The user is therefore advised to consider recompiling the program to match his needs if he is going to be handling a number of data series of comparable length. To edit the source program prior to recompilation, the following steps should be observed:

1. Copy the source file to a duplicate file, and then work with the duplicate file;
2. Change the DIMENSIONS of W11, W12, W21, W22 to at least $LP/4$, where L = piece length and P is the number of pieces.
3. Make the first dimension of OM exactly the same as in (2).
4. Check that arrays W31 and W32 are DIMENSIONed at least $2P$.
5. Check that SQ1 and SQ2 are DIMENSIONed at least $(L/8, L/8)$, both arrays identically.
6. Check that the initialization of variable ISQD is identical to either dimension in (5), and that ISOM is initialized at exactly the dimensions in (2).
7. Recompile.

(2) Control Cards

Let aaa = account number
 uuu = user number
 fff = input file name (RWDISC, keyed)
 ttii = file names for transmittal to plotting
 program (consecutive files)

Assume the element file is called BIVECR and exists in account aaa. The Control Card portion of the input deck is then as follows:

```

      !JOB aaa,uuu
*    !LIMIT (TIME,10), (CORE,30)
      !ASSIGN F:RAD, (FILE,fff), (DIRECT), (KEYED)
+    !ASSIGN F:1, (FILE,tt11), (OUT), (SAVE)
+    !ASSIGN F:2, (FILE,tt22), (OUT), (SAVE)
++   !ASSIGN F:3, (FILE,tt33), (OUT), (SAVE)
++   !ASSIGN F:4, (FILE,tt44), (OUT), (SAVE)
      !LOAD (EF, (BIVECR)), (UNSAT, (3))
      !RUN
      !DATA
      (DATA CARDS)

```

(3) Data cards

Program BIVEC has NAMELIST input followed by one or two data cards depending on the application. The NAMELIST input must be terminated by an * card whether or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are as follows. In general the switches are to be set to 0 for "off" and 1 for "on".

* Time and core limits will vary widely depending on application. See below.

† Necessary only when storing bicoherences on disc, for transmittal to plotting program. DCB assignments may be changed by NAMELIST parameters ISTORE1 and ISTORE2 respectively.

†† Must be included along with DCB assignments mentioned in footnote †, when doing cross rotary bispectrum for case XYZ and transmitting bicoherences via disc to plotting program. May be changed by NAMELIST parameters ISTORE3 and ISTORE4 respectively.

<u>NAMELIST VARIABLE</u>	<u>MEANING</u>	<u>DEFAULT VALUE</u>
MF	Number of logical files in input RAD file	21
IDATSTART	Starting location of data in each input logical file	5
LISTBI	Switch to list rotary bi-coherences and associated quantities	0
SIGLVL	95% confidence level for rotary bicoherences, above which phase error PHSERR is calculated and printed when LISTBI=1	*
IDUMP	Switch to store rotary bi-coherences in a consecutive file for transmittal to plotting program	1
ISTORE1	DCB assignments for consecutive files of rotary bicoherences created by setting switch IDUMP. Applies always	1
ISTORE2		2
ISTORE3	Like ISTORE1 and ISTORE2, except that these apply only for cross rotary bispectrum in the case XYZ	3
ISTORE4		4
INORM	Sets amplitude normalization described above	0
LABREAD	Switch to read labels for each series in input RWDISC file and to write these labels to output files for use by the plotting program. Each label consists of original buoy file name and variable number. If not set, then EBCDIC blanks will be written to output file	1

* Default is an approximate value. If P = number of pieces, let the number of degrees of freedom EDOF be defined as follows:

$$\text{EDOF} = \begin{cases} 36P^2/(19P-1) & \text{for Hanning, overlapping} \\ 2P & \text{for no Hanning, no overlapping} \end{cases}$$

Then the default value of SIGLVL is

$$\text{SIGLVL} = (6/\text{EDOF})^{1/2}$$

<u>NAMelist VARIABLE</u>	<u>MEANING</u>	<u>DEFAULT VALUE</u>
IFOURIER	Switch to read 132 character label in input file describing processing done by program FOURIER and to write these labels to output files for use by the plotting program. If not set, then EBCDIC blanks will be written to output file	1
KONF	Switch for determining rotary bicoherence confidence levels	0
IHANN	Determines method of calculating phase error when LISTBI $\neq$ 0 (See PHSErr, page 13) IHANN=0 not Hanned and not overlapped IHANN=1 Hanned and overlapped	1
BINSIZE	Bin width for sorting rotary bicoherences. A given rotary bicoherence is placed in bin I, where $I = \text{INT}(\text{rot. bicoh.} / \text{BINSIZE}) + 1$	0.01
RBICLIM	The rotary bicoherence "limit" for sorting which determines the number of bins of size BINSIZE, by determining the integer NBINS obtained by rounding $(\text{RBICLIM} / \text{BINSIZE})$, and then redefining RBICLIM as $(\text{NBINS}) \times (\text{BINSIZE})$	1.5
KB1	Switch to display partitioning of rotary bicoherences	1
KB2	Switch to display fraction of total rotary bicoherences under each bin limit	1

The data cards follow the * card which terminates NAMELIST input. They are all in generalized, free-field format and are as follows:

Card 1 (Mandatory)

<u>Variable No.</u>	<u>Meaning</u>
* { 1	Logical file for 1st scalar component of 1st vector series
2	Logical file for 2nd scalar component of 1st vector series
* { 3	Logical file for 1st scalar component of 2nd vector series
4	Logical file for 2nd scalar component of 2nd vector series
5	Logical file of 1st scalar component for sum frequency series
6	Logical file of 2nd scalar component for sum frequency series
7	Total number of pieces in each series
** 8	Piece length
9 } 10 }	Lower, upper piece sequence numbers inclusive, between which user wants to average
11 } 12 }	Lower, upper limits determining when to print rotary bicoherences when LISTBI≠0. Rotary bicoherences must satisfy: (var. #11) ≤ rotary bicoherence ≤ (var. #12)

Card 2 (Input only when KONF≠0)

<u>Variable No.</u>	<u>Meaning</u>
1 (NCONF)	Number of confidence levels to follow (≤10)
2 (CONF)	NCONF confidence levels expressed as decimal, 0 to 1

-
- * The user must take care to use no more than two distinct vector series or the program will not proceed. Two vector series are distinct unless both 1st and 2nd scalar component series are respectively identical. The program automatically determines the form of rotary bispectrum discussed under "Frequency Domain," as well as the corresponding frequency domain.
- ** Piece length must be evenly divisible by 8, have no prime factor greater than 5, and be no greater than 4000. Appendix 2 is a list of permissible choices.

OUTPUT:(1) Line Printer

Typical line printer output is displayed in Appendix 1. First a dump of NAMELIST parameter values is printed, followed by a facsimile of input data cards. Next, if confidence levels are called for, the "adjusted rotary bicoherence limit" (as determined by the rule given in describing variable RBICLIM in the section on Data Cards above) is printed.

This is followed by a listing of individual rotary bicoherences if called for. There is one rotary bicoherence value per line, plus other output described below. The listing occurs only within the limits given by variables #11 and #12 on Card 1, and the order is the order of computation. Every positive sum frequency considered is followed by the corresponding negative sum frequency. The user is cautioned that for cross rotary bicoherences of form XYY, the total number of rotary bicoherences computed is about $3(L/2)^2$, which is nearly 50000 for a piece length of 256. Without sufficiently narrow limits on the rotary bicoherences actually printed, this can result in a prohibitively large number of output pages.

The items output in each line depend on which form of rotary bicoherence is called for in the program. They are as follows, in order of appearance from left to right on the line:

(A) Default rotary bicoherence (after Yao):

<u>Variable</u>	<u>Meaning</u>
F1 F2 F3	Frequency triplet (sequence numbers) associated with the rotary bicoherence
RBIC	Rotary bicoherence
RBPH	Rotary biphase
RNUM	$ \langle RB(\omega_1, \omega_2) d\omega^2 \rangle $
RBR	$\langle \text{Real} [RB(\omega_1, \omega_2) d\omega^2] \rangle$
RBI	$\langle \text{Imag} [RB(\omega_1, \omega_2) d\omega^2] \rangle$
P1	$\langle P(\omega_1) d\omega \rangle$
P2	$\langle P(\omega_2) d\omega \rangle$
P3	$\langle P(\omega_3) d\omega \rangle$

<u>Variable</u>	<u>Meaning</u>
RDEN	$[\langle P(\omega_1)d\omega \rangle \langle P(\omega_2)d\omega \rangle \langle P(\omega_3)d\omega \rangle]^{1/2}$
PHSERR*	Phase error; depends on IHANN (see page 9) $= \sqrt{2P}$ for IHANN=0, P = No. of pieces $= 57.296 \sin^{-1} [1.96/(RBIC \cdot W)]$ for IHANN=1 default where $W = [36P^2/(19P-1)]^{1/2}$
(B) Alternate rotary bicoherence (normalized by amplitude)	

<u>Variable</u>	<u>Meaning</u>
F1 F2 F3	Frequency triplet (sequence numbers) associated with this rotary bicoherence
RBIC	Rotary bicoherence
RBPH	Rotary biphas
RBR	$\langle \text{Real} [RB'(\omega_1, \omega_2)d\omega^2] \rangle$
RBI	$\langle \text{Imag} [RB'(\omega_1, \omega_2)d\omega^2] \rangle$
PHSERR	Defined as above

After listing individual rotary bicoherences, there is an optional listing of how the rotary bicoherences are partitioned among bins, followed by the total number of rotary bicoherences submitted. The format is displayed in Appendix 1.

Following this there is an optional display of cumulative fractional numbers of rotary bicoherences lying below each upper bin limit. Again, the format is displayed in Appendix 1.

Finally, when a calculation of confidence levels is called for, a summary of the result for each confidence level is displayed as in Appendix 1. This is followed by a statement of the number of rotary bicoherences which happened to fall above the limit used in the program (also stated).

(2) Disc

Normally two or four consecutive disc files are produced by program BIVEC, as explained above, which are used to transmit the computed rotary bicoherences to the plotting programs. These files also contain information about the origin of the data series, the processing history, and the organization of the input file of Fourier coefficients.

* PHSERR is calculated and printed only above 95% confidence level given by SIGLVL (see NAMELIST variable table). In the line printer output shown in Appendix 1, SIGLVL=0.5612.

Each output consecutive disc file is organized as follows:

<u>Words, inclusive</u>	<u>Number of Words</u>	<u>Meaning</u>
1 thru N	N	N rotary bicoherences
N+1 thru N+3	3	Buoy format file name for source of variable 1, Card 1
N+4	1	Buoy format variable number for source of variable 1, Card 1
N+5 thru N+24	20	Like words N+1 thru N+4, for variables 2 thru 6, Card 1
N+25	1	No. of logical files in input RWDISC file (usually 21)
N+26 thru N+58	33	Processing history from program FOURIER

The first task of the plotting program is to construct from each consecutive file of rotary bicoherences a two-dimensional array whose elements occupy the same relative position in the matrix as do the grid points in the final plot. By using the coding which does this the user can access this file directly. The author can be consulted for details.

RESTRICTIONS:

1. The length of the data series which program BIVEC can handle depends on the dimensions of arrays W11, W12, W21, W22 (see "USAGE"), as well as on how the data series was split when processed. There is an upper limit of 16K words on the data series length, or 32K words on the length of the resulting Fourier coefficient series entering BIVEC. By "data series" is meant each component series of each vector series, so that 128K Fourier coefficients can actually be used.

Let

L = maximum length of data series. $2L$ is assumed to be an exact multiple of L , the piece length, for 50% overlap and L is assumed to be an exact multiple of L for no overlap.

L = piece length

D = dimensions of the arrays WIJ

P = no. of pieces

50% overlap

$$L = (4D+L)/2 \quad \text{and} \quad P = 2L/L-1 \quad (8)$$

No overlap

$$L = 4D \quad \text{and} \quad P = L/L \quad (9)$$

In all cases the number of points in the Fourier coefficient series entering BIVEC is PL, and the four WIJ arrays mentioned above must be dimensioned at least PL/4.

2. The first dimension of OM must be exactly the dimension of WIJ arrays mentioned in (1).
3. The arrays W31 and W32 must be dimensioned at least 2P, where P = no. of pieces.
4. The arrays SQI, I=1,2 must be dimensioned at least SQI(M,M) where M = L/8, L = piece length.
5. Variable ISQD must be initialized to either dimension of the arrays SQI(M,M) mentioned in (4). Variable ISOM must be initialized at exactly the dimension of the arrays WIJ mentioned in (1).
6. L must be evenly divisible by 8, have no prime factor greater than 5, and be ≤ 4000 .
7. (No. of confidence levels) ≤ 10 .
8. (No. of bins used in sorting rotary bicoherences) ≤ 200

ERRORS AND DIAGNOSTICS:

<u>Message</u>	<u>Meaning</u>	<u>Action Taken</u>
"LAST PIECE TO AVG. GREATER THAN NO OF PCS"	Self-explanatory	Abort
"NO MORE THAN TWO DIFFERENT VECTOR SERIES ALLOWED"	(Based on choice of logical files in input RWDISC file). Vector series are different unless both component series are the same	Abort
"TWO OUTPUT DCB's HAVE BEEN SET EQUAL CAN'T DO"	Self-explanatory	Abort

"PIECE LENGTH MUST BE FROM FOLLOWING LIST," followed by a list of acceptable values through 4000

Piece length fails to satisfy one of following:
1) evenly divisible by 8
2) no prime factor greater than 5
3) <4000

"NUMBER OF BINS MUST NOT EXCEED 200"

Self-explanatory Abort

STORAGE AND CPU TIME REQUIREMENTS:

As of the date of this report, program BIVEC has not been run extensively on long series. An operating history would be useful to the user to enable him to tailor the size of the program. It is intended to add to the following list as more runs of the program are reported.

TASK	PIECE LENGTH	NO. OF PIECES	DIMENSIONS OF W11,W12,W21,W22	DIMENSIONS OF W31,W32	DIMENSIONS OF SQ1,SQ2	PEAK CORE (PAGE SIZE 512)	CPU TIME (MIN)	CHARGE (CU)
AUTO ROTARY BICOH. SORTING	256	39	2560	100	(35,35)	50	9.38	15.85
AUTO ROTARY BICOH.	256	39	2560	100	(35,35)	50	9.34	15.56

The sorting above was done to determine confidence levels from Gaussian noise. In both cases the same number of rotary bicoherences was calculated. It is anticipated that calculation of cross rotary bicoherences for the case XXY on the above series would take essentially the same time, and for the case XYX would take approximately double the time (and cost), since twice as many rotary bicoherences are calculated.

SUBPROGRAMS REQUIRED: None

PROGRAMMER: Gerard H. Martineau

ORIGINATOR: Melbourne G. Briscoe

DATE: October, 1977

REFERENCES:

- (1) Report on Program RBPLOT, by G. H. Martineau.
- (2) "Handbook for Computer Users," Information Processing Center of W.H.O.I., pp. V-E-1 ff.
- (3) Report on Program FOURIER, by G. H. Martineau.
- (4) "Bispectral and Cross-Bispectral Analysis of Wind and Currents off Oregon Coast," Ph.D. Thesis at Oregon State University, by N. C. G. Yao, June, 1974.
- (5) "Rotary Cross-Bispectra and Energy Transfer Functions between Non-Gaussian Vector Processes I. Development and Example," by N.-C. Yao, Steve Neshyba, and Henry Crew, Journal of Physical Oceanography, Vol. 5, January, 1975, p. 164.

APPENDIX 1
(BIVEC)

17:45 JAN 26, '78 ID=04F4
 JGB 462,1719
 LIMIT (TIME,3),(C9RE,15),(PRDR),(ACCRUNT)
 ASSIGN FIRAD,(FILE,BVECI1),(DIRECT),(KEYED)
 LEAD (EF,(BIVECO5R)),(UNSAT,(3))
 IP1 ASSOCIATED.

*** ALLOCATION SUMMARY ***

PROTECTION LOCATION PAGES

DATA (00) A000 7
 PROCEDURE (01) R000 2
 DCB (10) AEOO 1
 ***** SGN= D4L SIZE=008.1K *****

***** PROTECTION TYPES: 00 DATA 01 PROCEDURE 10 STATIC

SEGH1=0 AC61 SEGH1=1 C5SD SEGH1=2 AFF
 SEGL9=0 A000 SEGL9=1 R000 SEGL9=2 AEOO
 00 SIZE= C62 01 SIZE= 155E 10 SIZE= 200

RUN

IDATSTART = 5

MF = 21

ISTORE1 = 1

ISTORE2 = 2

ISTORE3 = 3

ISTORE4 = 4

BINSIZE = 1.000000E+02

RBICLIM = 1.50000

IHI = 1

KB1 = 1

KB2 = 1

K8NF = 1

LISTBI = 1

IDUMP = 0

LABREAD = 1

IFOURIER = 1

INORM = 0

IMANN = 1

1 2 1 2 1 2 10 64 1 10 .545 .575

4 .90 .95 .99 .999

ADJUSTED RBT BICBH LIMIT, HOLDING BINSIZE AS INPUT 1.50

F1	F2	F3	RBICBH	RBIPHASE	/<RB>/	<REAL(RB)>	<IMAG(RB)>	<P(W1)DW>	<P(W2)DW>	<P(W3)DW>	SQRT(P(RB)<P>)	PHSER
8	2	10	.566	-25	.166E+01	.150E-01	.707E-02	.106E 00	.673E-01	.121E 00	.293E-01	52.5
7	5	12	.551	-33	.321E+01	.270E-01	.173E-01	.144E 00	.155E 00	.152E 00	.583E-01	
-9	7	-2	.546	-99	.240E+01	.380E-02	.237E-01	.108E 00	.144E 00	.124E 00	.403E-01	53.0
14	-2	12	.563	-156	.282E+01	.257E-01	.114E-01	.133E 00	.124E 00	.152E 00	.501E-01	
13	3	16	.550	-49	.270E+01	.177E-01	.204E-01	.148E 00	.132E 00	.124E 00	.491E-01	
-14	-2	-16	.569	173	.161E+01	.160E-01	.197E-02	.706E-01	.124E 00	.917E-01	.284E-01	52.1
11	-10	1	.559	-158	.324E+01	.300E-01	.122E-01	.117E 00	.970E-01	.295E 00	.579E-01	
-17	16	-1	.549	-156	.202E+01	.185E-01	.823E-02	.586E-01	.124E 00	.187E 00	.568E-01	
-19	9	-10	.551	-83	.187E+01	.232E-02	.185E-01	.705E-01	.168E 00	.970E-01	.339E-01	
-21	10	-11	.552	-172	.233E+01	.231E-01	.344E-02	.110E 00	.121E 00	.134E 00	.423E-01	
22	3	25	.561	81	.174E+01	.283E-02	.172E-01	.639E-01	.132E 00	.115E 00	.311E-01	

F1	F2	F3	RBIC0H	RB1PHASE	/<RB>/	<REAL(RB)>	<IMAG(RB)>	<P(W1)DW>	<P(W2)DW>	<P(W3)DW>	SQRT(PRGD<P>)	PHSER
-23	-5	-28	.574	28	.217E-01	.193E-01	.101E-01	.136E 00	.995E-01	.106E 00	.378E-01	51.5
-20	-3	17	.554	-143	.238E-01	-.231E-01	-.173E-01	.219E 00	.121E 00	.102E 00	.321E-01	
-22	1	-21	.569	-15	.369E-01	.357E-01	.932E-02	.129E 00	.295E 00	.110E 00	.549E-01	52.1
26	-6	20	.558	61	.283E-01	.137E-01	.247E-01	.162E 00	.722E-01	.219E 00	.507E-01	
27	-6	21	.558	86	.235E-01	.166E-02	.234E-01	.129E 00	.722E-01	.191E 00	.421E-01	
-30	2	-28	.566	4	.159E-01	.159E-01	.115E-02	.111E 00	.673E-01	.106E 00	.281E-01	52.5
32	-1	31	.547	95	.258E-01	-.211E-02	.258E-01	.574E-01	.187E 00	.208E 00	.472E-01	
-28	-1	-29	.546	-157	.225E-01	-.207E-01	-.868E-02	.106E 00	.187E 00	.855E-01	.411E-01	
-25	12	-13	.563	28	.136E-01	.120E-01	.630E-02	.638E-01	.152E 00	.599E-01	.241E-01	52.8
-26	9	-17	.559	-67	.180E-01	.708E-02	.166E-01	.106E 00	.168E 00	.586E-01	.323E-01	
-30	11	-19	.572	-179	.173E-01	-.173E-01	-.315E-03	.111E 00	.117E 00	.705E-01	.302E-01	51.8
30	-9	21	.572	142	.239E-01	-.189E-01	.147E-01	.847E-01	.108E 00	.191E 00	.418E-01	51.7
28	-19	9	.557	-159	.169E-01	-.158E-01	-.597E-02	.785E-01	.705E-01	.168E 00	.305E-01	
-29	19	-10	.557	33	.137E-01	.164E-01	.108E-01	.855E-01	.150E 00	.970E-01	.353E-01	
-30	20	-10	.553	101	.268E-01	-.520E-02	.263E-01	.111E 00	.219E 00	.970E-01	.485E-01	
31	-29	2	.558	-7	.193E-01	.192E-01	-.240E-02	.208E 00	.855E-01	.673E-01	.346E-01	

NO OF ROT BIC'S

BIN LIMITS

BIN NO

BIN NO	BIN LIMITS	NO OF ROT BIC'S
1	.00	3
2	.01	4
3	.02	9
4	.03	7
5	.04	
6	.05	19
7	.06	10
8	.07	11
9	.08	22
10	.09	16
11	.10	35
12	.11	30
13	.12	35
14	.13	28
15	.14	28
16	.15	40
17	.16	29
18	.17	40
19	.18	35
20	.19	35
21	.20	40
22	.21	47
23	.22	36
24	.23	37
25	.24	47
26	.25	35
27	.26	39
28	.27	29
29	.28	26
30	.29	42
31	.30	47
32	.31	36
33	.32	40
34	.33	31
35	.34	37
36	.35	31
37	.36	32
38	.37	35
39	.38	23
40	.39	34
41	.40	32
42	.41	32
43	.42	20

42	.41	.42	22
43	.42	.43	25
44	.43	.44	27
45	.44	.45	20
46	.45	.46	22
47	.46	.47	12
48	.47	.48	12
49	.48	.49	11
50	.49	.50	6
51	.50	.51	7
52	.51	.52	9
53	.52	.53	14
54	.53	.54	5
55	.54	.55	7
56	.55	.56	13
57	.56	.57	7
58	.57	.58	6
59	.58	.59	9
60	.59	.60	11
61	.60	.61	5
62	.61	.62	2
63	.62	.63	3
64	.63	.64	3
65	.64	.65	2
66	.65	.66	2
67	.66	.67	3
68	.67	.68	3
69	.68	.69	4
70	.69	.70	3
71	.70	.71	1
72	.71	.72	2
73	.72	.73	2
74	.73	.74	1
75	.74	.75	0
76	.75	.76	5
77	.76	.77	2
78	.77	.78	0
79	.78	.79	0
80	.79	.80	2
81	.80	.81	2
82	.81	.82	0
83	.82	.83	0
84	.83	.84	0
85	.84	.85	0
86	.85	.86	0
87	.86	.87	0
88	.87	.88	0
89	.88	.89	0
90	.89	.90	1
91	.90	.91	0
92	.91	.92	0
93	.92	.93	0
94	.93	.94	0
95	.94	.95	0
96	.95	.96	0
97	.96	.97	0
98	.97	.98	0
99	.98	.99	0
100	.99	1.00	0
101	1.00	1.01	0

NO OF RPT BIC'S

BIN NO BIN LIMITS

102	1.01	1.02	0
103	1.02	1.03	0
104	1.03	1.04	0
105	1.04	1.05	0
106	1.05	1.06	0
107	1.06	1.07	0
108	1.07	1.08	0
109	1.08	1.09	0
110	1.09	1.10	0
111	1.10	1.11	1
112	1.11	1.12	0
113	1.12	1.13	0
114	1.13	1.14	0
115	1.14	1.15	0
116	1.15	1.16	0
117	1.16	1.17	0
118	1.17	1.18	0
119	1.18	1.19	0
120	1.19	1.20	0
121	1.20	1.21	0
122	1.21	1.22	0
123	1.22	1.23	0
124	1.23	1.24	0
125	1.24	1.25	0
126	1.25	1.26	0
127	1.26	1.27	0
128	1.27	1.28	0
129	1.28	1.29	0
130	1.29	1.30	0
131	1.30	1.31	0
132	1.31	1.32	0
133	1.32	1.33	0
134	1.33	1.34	0
135	1.34	1.35	0
136	1.35	1.36	0
137	1.36	1.37	0
138	1.37	1.38	0
139	1.38	1.39	0
140	1.39	1.40	0
141	1.40	1.41	0
142	1.41	1.42	0
143	1.42	1.43	0
144	1.43	1.44	0
145	1.44	1.45	0
146	1.45	1.46	0
147	1.46	1.47	0
148	1.47	1.48	0
149	1.48	1.49	0
150	1.49	1.50	0

TOTAL NO OF RPT BIC'S SUBMITTED: 1504

FRACT BELOW
UPPER BIN LIM

BIN NO BIN LIMITS

1	.00	.01	.002
2	.01	.02	.005
3	.02	.03	.011
4	.03	.04	.015

5	.04	.028
6	.05	.035
7	.06	.042
8	.07	.057
9	.08	.067
10	.09	.090
11	.10	.110
12	.11	.134
13	.12	.152
14	.13	.171
15	.14	.197
16	.15	.217
17	.16	.243
18	.17	.267
19	.18	.290
20	.19	.316
21	.20	.348
22	.21	.372
23	.22	.396
24	.23	.428
25	.24	.451
26	.25	.477
27	.26	.496
28	.27	.513
29	.28	.541
30	.29	.572
31	.30	.596
32	.31	.623
33	.32	.648
34	.33	.668
35	.34	.689
36	.35	.713
37	.36	.728
38	.37	.751
39	.38	.772
40	.39	.793
41	.40	.807
42	.41	.821
43	.42	.838
44	.43	.856
45	.44	.869
46	.45	.884
47	.46	.892
48	.47	.900
49	.48	.907
50	.49	.911
51	.50	.916
52	.51	.922
53	.52	.931
54	.53	.934
55	.54	.939
56	.55	.947
57	.56	.952
58	.57	.956
59	.58	.962
60	.59	.969
61	.60	.973
62	.61	.974
63	.62	.976
64	.63	.978

FRACT BELOW
UPPER BIN LIM

BIN LIMITS

BIN NO

65	.64	.65	.979
66	.65	.66	.981
67	.66	.67	.983
68	.67	.68	.985
69	.68	.69	.987
70	.69	.70	.988
71	.70	.71	.989
72	.71	.72	.991
73	.72	.73	.991
74	.73	.74	.991
75	.74	.75	.995
76	.75	.76	.996
77	.76	.77	.996
78	.77	.78	.996
79	.78	.79	.997
80	.79	.80	.999
81	.80	.81	.999
82	.81	.82	.999
83	.82	.83	.999
84	.83	.84	.999
85	.84	.85	.999
86	.85	.86	.999
87	.86	.87	.999
88	.87	.88	.999
89	.88	.89	.999
90	.89	.90	.999
91	.90	.91	.999
92	.91	.92	.999
93	.92	.93	.999
94	.93	.94	.999
95	.94	.95	.999
96	.95	.96	.999
97	.96	.97	.999
98	.97	.98	.999
99	.98	.99	.999
100	.99	1.00	.999
101	1.00	1.01	.999
102	1.01	1.02	.999
103	1.02	1.03	.999
104	1.03	1.04	.999
105	1.04	1.05	.999
106	1.05	1.06	.999
107	1.06	1.07	.999
108	1.07	1.08	.999
109	1.08	1.09	.999
110	1.09	1.10	.999
111	1.10	1.11	1.000
112	1.11	1.12	1.000
113	1.12	1.13	1.000
114	1.13	1.14	1.000
115	1.14	1.15	1.000
116	1.15	1.16	1.000
117	1.16	1.17	1.000
118	1.17	1.18	1.000
119	1.18	1.19	1.000
120	1.19	1.20	1.000
121	1.20	1.21	1.000
122	1.21	1.22	1.000
123	1.22	1.23	1.000
124	1.23	1.24	1.000

125	1.24	1.25	1.000
126	1.25	1.26	1.000
127	1.26	1.27	1.000
128	1.27	1.28	1.000
129	1.28	1.29	1.000
130	1.29	1.30	1.000
131	1.30	1.31	1.000
132	1.31	1.32	1.000
133	1.32	1.33	1.000
134	1.33	1.34	1.000
135	1.34	1.35	1.000
136	1.35	1.36	1.000
137	1.36	1.37	1.000
138	1.37	1.38	1.000
139	1.38	1.39	1.000
140	1.39	1.40	1.000
141	1.40	1.41	1.000
142	1.41	1.42	1.000
143	1.42	1.43	1.000
144	1.43	1.44	1.000
145	1.44	1.45	1.000
146	1.45	1.46	1.000
147	1.46	1.47	1.000
148	1.47	1.48	1.000
149	1.48	1.49	1.000
150	1.49	1.50	1.000

CONFIDENCE LEVEL (PERCENT)	ROTARY BICOHERENCE
90.0	.481
95.0	.565
99.0	.715
99.9	.895

NR. OF ROT BIC'S GREATER THAN SPECIFIED MAXIMUM OF 1.50 IS 0

STOP NORMAL PROGRAM COMPLETION

APPENDIX 2
(BIVEC)

Appendix 2

Permissible Piece Lengths for Program BIV

16	384	1440	3456
24	400	1536	3600
32	432	1600	3848
40	480	1728	3888
48	512	1800	4000
64	576	1920	
72	600	1944	
80	640	2000	
96	648	2048	
120	720	2160	
128	768	2304	
144	800	2400	
160	864	2560	
200	960	2592	
216	1024	2880	
240	1080	2916	
256	1152	3000	
288	1200	3072	
320	1280	3200	
360	1296	3240	

NAME: RBPLOT

TYPE: Main Program

PURPOSE: To construct auto or cross rotary bicoherence contour plots from disc files produced by program BIVEC(1)

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program RBPLOT reads consecutive disc files of auto or cross rotary bicoherences produced by program BIVEC and then makes a contour plot of these rotary bicoherences over an appropriate domain. The program presently exists in two versions, RBPLOT05 which makes separately annotated plots (in separate plot files) each over 1/4 of the maximum domain as explained below, and RBPLOT06 which makes a single plot over the entire domain of interest.

INPUT:

Input consists of two consecutive disc files in the cases of auto rotary bicoherence and cross rotary bicoherence of form XXY, and four such files in the case of cross rotary bicoherence of form XYY. (See reference 1 for explanation of notation.)

OUTPUT:

(1) Line Printer

Sample line printer output appears in Appendix 1. A facsimile of three input cards is printed, followed by a listing of the contour levels from Card 2 under a separate heading, "CONTOUR LEVELS."

(2) Plots

Consider the following region in the $\omega_1 \omega_2$ plane:

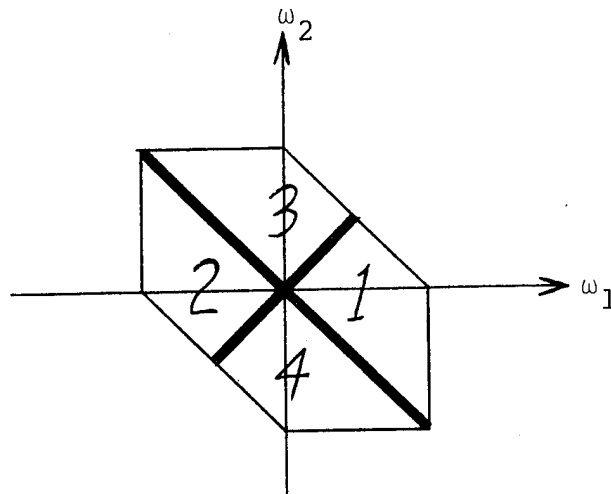


Figure 1

In terms of the notation of the report on program BIVEC (ref. 1), which calculates the rotary bicoherences, the user will recall that auto rotary bicoherences and cross rotary bicoherences of form XXY are calculated over the domains 1 and 2 of figure 1, while cross rotary bicoherences of form XYX require the addition of 3 and 4. As mentioned above, program RBPLOT exists in two versions, RBPLOT05 and RBPLOT06. RBPLOT05 generates separate plot files of each region of figure 1 as appropriate; thus it considers region 1+2, or 1+2+3+4. The plots generated constitute Appendix 2. They are separately annotated, but only the region 1 plot has the data origin, contour levels, and processing history. Program RBPLOT06, on the other hand, generates a single plot similar in appearance to figure 1 and displayed as Appendix 3. Again it consists of regions 1+2, or 1+2+3+4 as appropriate.

The program is capable of producing disc plot files to be used for immediate or deferred Versatec plots, or a plot tape to be used for Calcomp plotting. See the W.H.O.I. "Handbook for Computer Users" for details.

USAGE:

(1) Control cards

```
Let   aaa = account number
      uuu = user number
      ffi = input file names
      ppp = plot file name
```

```

!JOB aaa,uuu
!LIMIT (TIME,t), (CORE,c)
!ASSIGN F:1, (FILE,ff1)
!ASSIGN F:2, (FILE,ff2)
* !ASSIGN F:3, (FILE,ff3)
* !ASSIGN F:4, (FILE,ff4)
** !ASSIGN F:5, (FILE,SCRATCH), (OUTIN)
!LOAD (EF,(RBPLOTR), (PLOTDFER,3)), (UNSAT,(3))
!RUN
!DATA

```

→ (DATA CARDS)

† !PLOTV

Program RBPLOT has NAMELIST input followed by three data cards. The NAMELIST input must be terminated by an * card whether or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are as follows. In general, "switches" are set to 0 for "off," 1 for "on."

* Necessary only if plotting cross rotary bicoherences of form $XY\bar{Y}$.

```

** Necessary only for RBPLOT06, cross rotary bicoherences
   of form XYY.

```

[†]Plot size is set to Versatec paper for RBPLOT05, and 30" Calcomp paper for RBPLOT06.

<u>NAMelist Variable</u>	<u>Meaning</u>	<u>Default Values</u>
ISTORE1,ISTORE2 ISTORE3,ISTORE4	DCB assignments for input consecutive files	1,2,3,4 in order
ISCR	DCB assignment for scratch file	5
NDECPL	Format control for axis annotation: >0: Number of digits to the right of the decimal point which are plotted, after rounding =0: Only integer portion of number and decimal point are plotted, after rounding =-1: Only integer portion of number is plotted, after rounding <-1: NDECPL -1 digits are truncated from the inte- ger portion after rounding	1
WIDTH	Width, in inches, of plot of L/2 points for RBPLOT05, L points for RBPLOT06, where L=piece length	8.75 for RBPLOT05 24.0 for RBPLOT06
CMPPT	Spacing in centimeters be- tween adjacent data (grid) points on plot. If entered by user, this will override any specification of vari- able WIDTH	$WIDTH * 2.54 / (L/2)$ (L=pc. length)
IBUOY	Switch to print informa- tion about data origin on plot, including original buoy format file names and variable numbers	1
IFOURIER	Switch to print processing history on plot (from pro- gram FOURIER)	1

The data cards follow the *card which terminates NAMELIST input. They are all in generalized, free-field format and are as follows:

Card 1

<u>Variable No.</u>	<u>Meaning</u>
1,2,3	This sequence of integers gives the form of auto or rotary cross bicoherence; e.g., rotary cross of form XYZ is 122, and rotary auto is 111 or 222.
LPIECE	Piece length
SAMPSEC	Sample interval in seconds
FREQTIC	* Distance between tic marks on plot in (hr) ⁻¹

Card 2

<u>Variable No.</u>	<u>Meaning</u>
1 (NCONLV)	Number of contour levels to follow (≤ 10)
2 thru (1+NCONLV) }	NCONLV contour levels for plot
(2+NCONLV) thru (1+2·NCONLV) }	
	NCONLV confidence levels (percent) for contour levels

Card 3

-up to 72 characters of identification, which will appear immediately above the plot. The first 36 characters will appear on one line, and the next 36 characters on the next line. The user is responsible for a proper transition of text from one line to the next, occurring between columns 36 and 37.

* Full-scale frequency (Nyquist frequency) is 1800/SAMPSEC. Variable FREQTIC does not have to divide evenly into the full-scale frequency.

RESTRICTIONS:

- (1) Program RBPLOT uses a large two-dimensional buffer, "DATA," which can be thought of as a matrix whose elements correspond one-for-one with the grid points of the plot, and whose rectangular "boundary" completely contains the plot, the non-data points being padded with flags to that effect. The dimensions of DATA therefore govern the maximum piece length which can be plotted. The rule is as follows: If L is the piece length, then array DATA must be dimensioned at least DATA(R,C) where:

$$C \geq 1/2$$

$$R \geq 1.5C + L$$

The "standard" dimensions of DATA are DATA(200,130). If recompilation is necessary, the following three additional steps must be taken:

- a) Array XDATA(C,R) is EQUIVALENCE'd to DATA, and must have dimensions exactly the reverse of those of DATA.
 - b) Variables MD1 and MD2 (immediately following the above EQUIVALENCE), must be initialized to the first and second dimensions of DATA respectively.
 - c) Check that array EXCH is DIMENSIONED the same as the first dimension of DATA.
- (2) The number of contour levels must be ≤ 10 .

SUBPROGRAMS REQUIRED: PLOTDFER, for a Versatec or graphics terminal plot.

STORAGE REQUIREMENT AND TIMING:

In a RBPLOT run in which PLOTDFER was also loaded, and array DATA was dimensioned DATA(200,130), a peak core of 77 512-word pages was used, along with 1.08 minutes of CPU time. This was to plot auto rotary bicoherence.

ERRORS AND DIAGNOSTICS:

None

PROGRAMMER:

Gerard H. Martineau

ORIGINATOR:

Melbourne G. Briscoe

DATE:

October, 1977

REFERENCES:

- (1) Report on program BIVEC, by Gerard H. Martineau,
Woods Hole Oceanographic Institution.

APPENDIX 1
(RBPLOT)

10:20 DEC 02, '77 ID=05B5
 JOB 462,1719,7 • TERMINAL JOB
 LIMIT (CORE,4),(TIME,1),(L9,50),(DB,1)
 PCL
 C RBPLST05LP T9 LP

1 1 1 64 56.25 5
 5 .405 .442 .485 .548 .705 80 85 90 95 99
 4 COUNTER RBT VEC'S AMP SAME AS G N FREQ 3/4,9/16,3/8,3/16 NYQUIST

CNTBUR LEVELS:

.405

.442

.485

.548

.705

APPENDIX 2, PART 1
(RBPLOT)

AUTO ROTARY BICOHERENCE

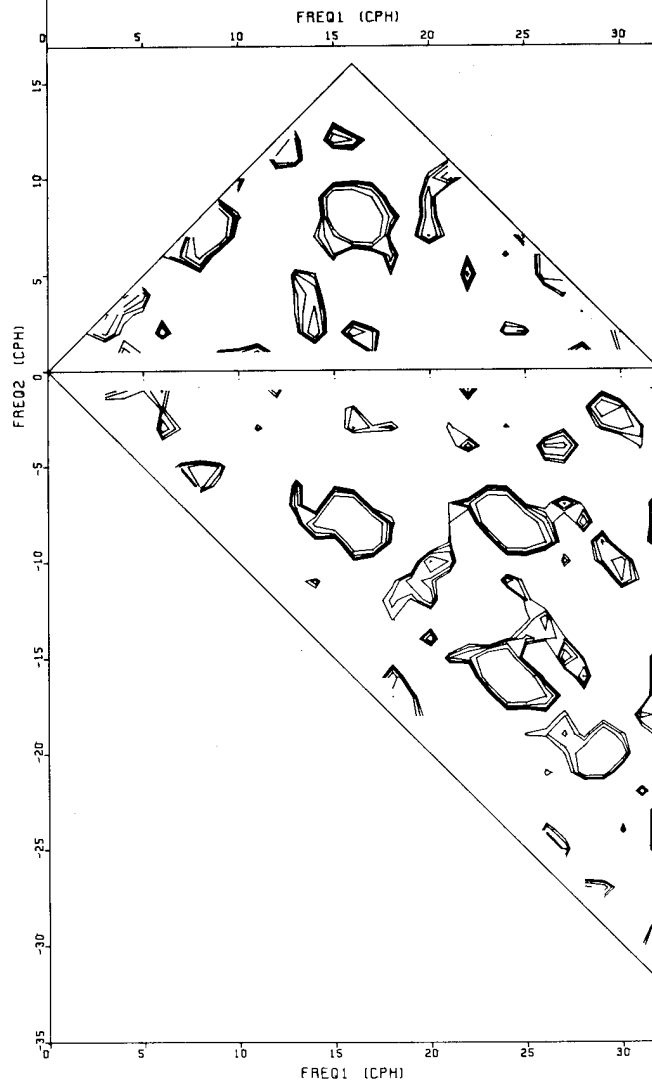
PROCESSING HISTORY

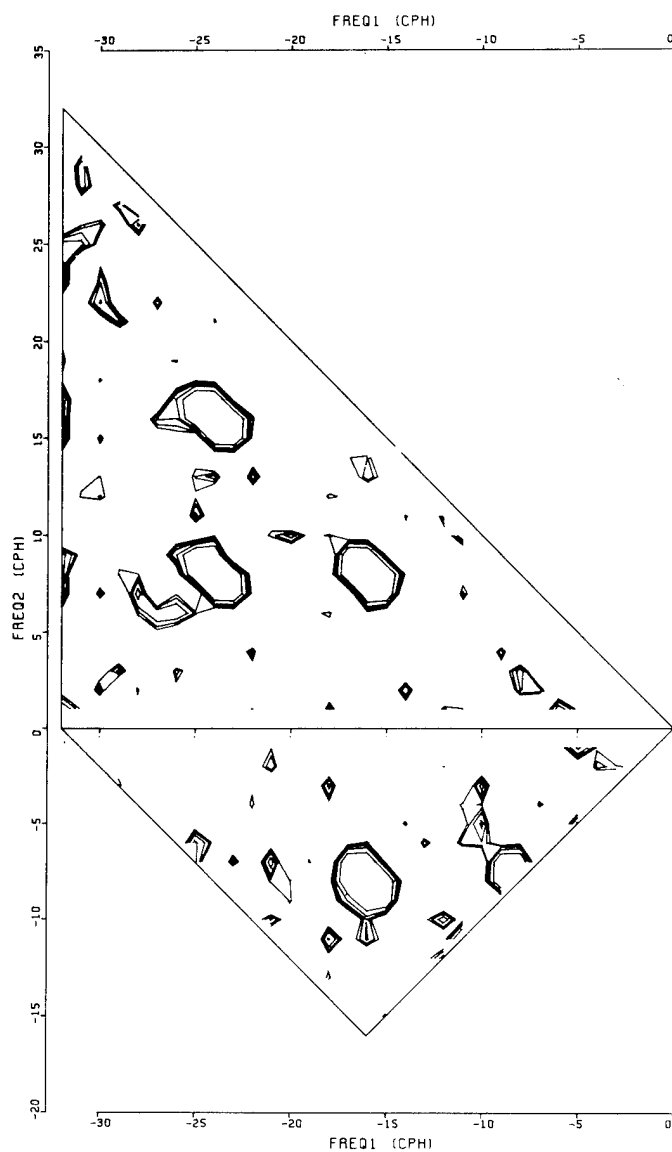
NO LF'S 11=NO WDS DATA 352=PAEM 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 64
 #NO PCS 10=OLAP 1=ISUBT 0=HANN 1 =CREATED 14.11 MAR 08, '78

CONTOUR LEVELS AND PERCENT CONFIDENCE

	0.406	80.0
TIME OF PLOT:	0.436	85.0
14:56 MAR 08, '78	0.481	90.0
	0.565	95.0
	0.715	99.0

3 COUNTER ROT VEC'S S/N=10 HANNED
 FREQ 8.16,24 NYQUIST=32



APPENDIX 2, PART 2
(RBPLLOT)

APPENDIX 3
(RBPLOT)

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

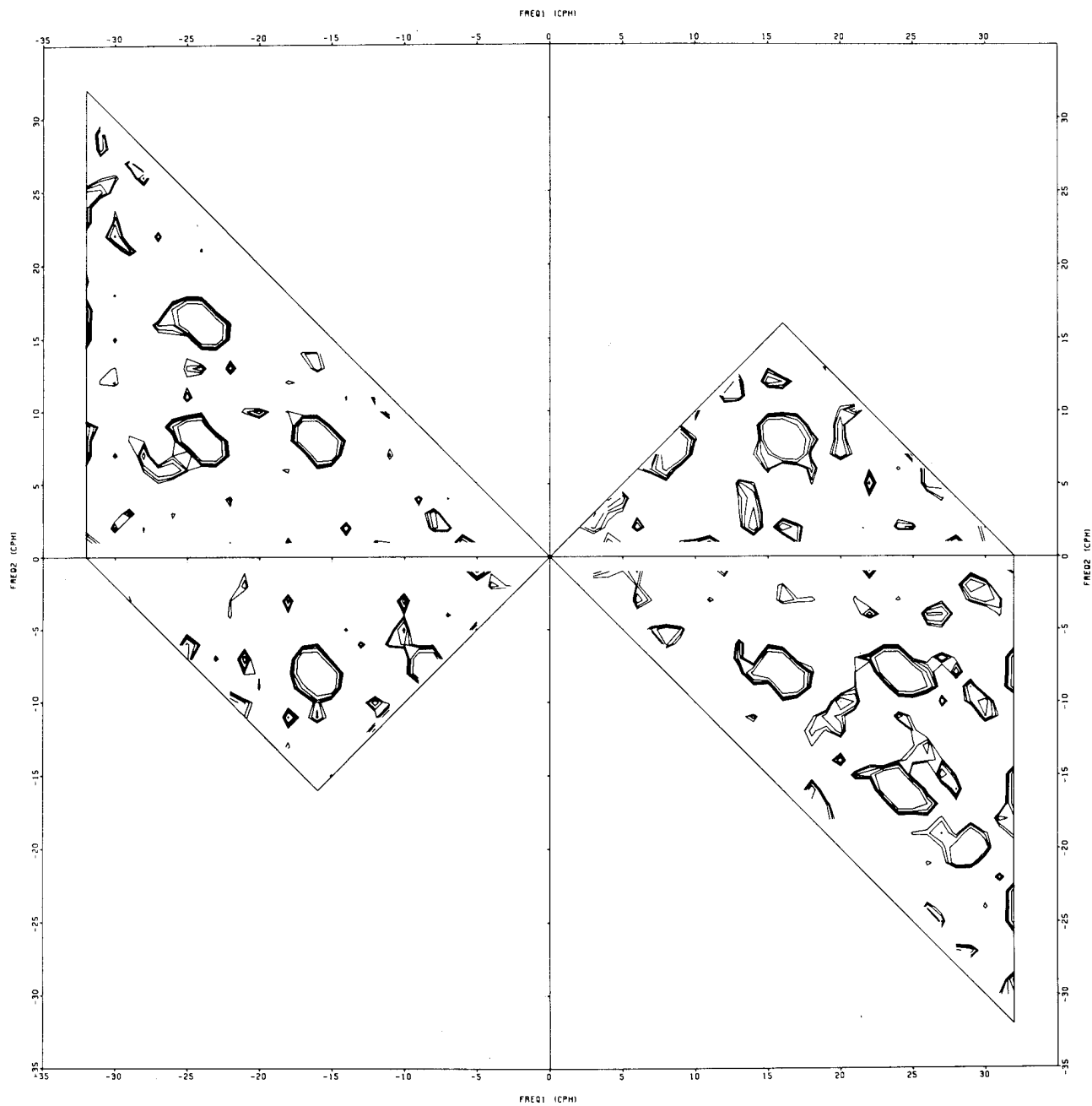
0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 352*PREW 0*
 SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
 OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07, '77

3 COUNTER ROT VEC'S S/N=10 HANNED
 FREQ 8.16,24 NYQUIST=32

TIME OF PLOT:
 15:43 DEC 07, '77



E. FORTRAN IV LISTINGS, ALPHABETICALLY

- PROGRAM BISCAL -

```

1 - C          <<<<< B I S C A L O 4 >>>>>
2 - C CALCULATES BISPECTRA AND CROSS-BISPECTRA FOR ONE OR TWO
3 - C REAL SERIES AND DETERMINES CONFIDENCE LEVELS.GENERATES DISC
4 - C FILE FOR TRANSMITTAL TO PLOTTING PROGRAM.
5 - C PROGRAMMER: G. MARTINEAU 10/13/77
6 - C
7 - C
8 - C          DIMENSION OM1(5400),OM2(5400),SQ(70,70),
9 - C          $W3(300),KBIN(200),ABIN(200),CONF(10),BCL(10),KARD(20),
10 - C          $LABL(33),PHSERR(2)
11 - C          EQUIVALENCE (KBIN,ABIN)
12 - C          DOUBLE PRECISION R1,R2,R3,C1,C2,C3,S1,S2,S3,BR,BC,BMOD,
13 - C          $BRP,BCP
14 - C          NAMELIST IDATSTART,MF,ISTORE,
15 - C          $BINSIZE,BICOHLIM,IHI,KB1,KB2,KONF,LISTBI,IDUMP,INORM,
16 - C          $ISQOUT,IHANN,SIGLVL
17 - C          CALL ABORTSET(7005,1)
18 - C          PI=3.14159265;TWOPI=2.*PI
19 - C          IDUMP=1
20 - C          MF=21;IDATSTART=5
21 - C          ISTORE=1
22 - C          IHI=1
23 - C          M33=0
24 - C          BINSIZE=.01;BICOHLIM=1.5;KB1=KB2=1;KONF=0
25 - C          SIGLVL=-1 ;          PHSERR(2)=0.
26 - C          KOVER=0 ; LISTBI=0
27 - C          MTOT=0
28 - C          INORM=0
29 - C          ISQOUT=0 ; IHANN=1
30 - C          OUTPUT 'ONE',MTOT
31 - C
32 - C          DO 20 I=1,200
33 - C      20 KBIN(I)=0
34 - C
35 - C          ISQD=70
36 - C          INPUT
37 - C          CALL DPAR(NERR,MF,1,1,KSUM,512)
38 - C          OUTPUT IDATSTART,MF,ISTORE,BINSIZE,
39 - C          $BICOHLIM,IHI,KB1,KB2,KONF,LISTBI,IDUMP,INORM,IHANN
40 - C CHECK EXISTENCE OF DCB ASSIGNMENT FOR OUTPUT FILE
41 - C      IF (IDUMP.NE.0) CALL GETDCB(ISTORE,LOCC)
42 - C      READ (105,1010) KARD
43 - C      1010 FORMAT (20A4)
44 - C      WRITE (108,1020) KARD
45 - C      1020 FORMAT (1H ,20A4)
46 - C      DECODE (80,1050,KARD) LFW1,LFW2,LFW1P2,NPIECES,
47 - C      $LPIECE,NPC1,NPC2,CONF1,CONF2
48 - C      1050 FORMAT (9G)
49 - C      IF (KONF.EQ.0) GO TO 23
50 - C      READ (105,1010) KARD
51 - C      WRITE (108,1020) KARD
52 - C      DECODE (80,1200,KARD) NCONF,NCONF,(CONF(I),I=1,NCONF)
53 - C      1200 FORMAT (G,NG)
54 - C      23 CONTINUE
55 - C      IF (KONF.NE.0) NBINS=INT(BICOHLIM/BINSIZE+0.5);
56 - C      $BICOHLIM=NBINS*BINSIZE ;
57 - C      $WRITE (108,1600) BICOHLIM
58 - C      1600 FORMAT ('ADJUSTED BICOHERENCE LIMIT, HOLDING BINSIZE ',

```

```

59 -      $'AS INPUT ',F6.2)
60 -      IF ((LISTBI.NE.0).AND.(INORM.EQ.0)) WRITE (108,1321)
61 - 1321 FORMAT (/ ,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
62 -      $T40,'BIMOD',T52,'BISPECR',T65,'BISPECI',T78,'AUTOF1',
63 -      $T90,'AUTOF2',T101,'AUTOF1F2',T117,'SD',T125,'PHSERR',/)
64 -      IF ((LISTBI.NE.0).AND.(INORM.NE.0)) WRITE (108,1431)
65 - 1431 FORMAT (/ ,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
66 -      $T41,'BISPPR',T55,'BISPPI',T65,'PHSERR',/)
67 -      NPCS=NPC2-NPC1+1
68 -      IF ((NPCS.EQ.1).AND.(IHANN.EQ.1)) IHANN=0 ; OUTPUT
69 -      $'IHANN HAS BEEN SET TO 0. CAN'T OVERLAP WITH ONE PIECE'
70 -      IF (IHANN.EQ.1) EDOF=36.*NPCS*NPCS/(19*NPCS-1)
71 -      IF (IHANN.EQ.0) EDOF=2*NPCS
72 -      IF (SIGLVL.LT.0.) SIGLVL=SQRT(6./EDOF)
73 -      LPHALF=LPIECE/2
74 -      IF (MOD(LPHALF,2).EQ.1)
75 -      $STOP '(PIECE LENGTH)/2 MUST BE EVEN'
76 -      NFHALF=LPHALF/2
77 -      IF (LISTBI.EQ.0) GO TO 25
78 -      25 CONTINUE
79 - C DECIDE WHETHER TO CALCULATE CROSS-BISPECTRA
80 -      NSERIES=2
81 -      IF ((LFW1.NE.LFW2).AND.(LFW1P2.NE.LFW1).AND.
82 -      $(LFW1P2.NE.LFW2)) OUTPUT
83 -      $'CANNOT HAVE MORE THAN TWO INPUT SERIES' ; STOP
84 -      IF ((LFW1.EQ.LFW2).AND.(LFW2.EQ.LFW1P2)) NSERIES=1
85 -      ISQ=1
86 - C CHECK INPUT INFO. AGAINST LABELS
87 - C      OUTPUT 'MK 1',IDATSTART
88 - C
89 - C
90 -      NFREQLO=NFHALF+1
91 -      NFREQHI=NFHALF
92 -      NPTSLO=NPIECES*(2*NFREQLO-1)
93 -      NPTSHI=NPIECES*(2*NFREQHI-1)
94 - C
95 - C
96 -      30 CONTINUE
97 - C
98 - C SET ARRAY SQ TO -999.
99 - C
100 -      IF (IDUMP.EQ.0) GO TO 41
101 - C
102 -      DO 40 I=1,ISQD
103 - C
104 -      DO 40 J=1,ISQD
105 -      40 SQ(I,J)=-999.
106 - C
107 -      41 CONTINUE
108 - C
109 - C      OUTPUT LFW1,IDATSTART,NPTSLO,NPTSHI,ISQ
110 -      IF (ISQ.EQ.1) CALL RDISC(LFW1,IDATSTART,OM1,NPTSLO)
111 - C      OUTPUT 'MK 3'
112 -      IF ((ISQ.NE.3).OR.(NSERIES.NE.2).OR.(LFW1.NE.LFW2))
113 -      $GO TO 46
114 -      DO 43 I=1,NPTSLO
115 -      43 OM2(I)=OM1(I)
116 -      46 CONTINUE
117 -      IF (ISQ.EQ.3) CALL RDISC(LFW1,IDATSTART+NPTSLO,OM1,NPTSHI)
118 - C      OUTPUT 'MK 4'

```

```

119 -      IF ((ISQ.NE.1).OR.(NSERIES.NE.1)) GO TO 75
120 - C
121 - C
122 -      DO 50 I=1,NPTSLO
123 -      50 OM2(I)=OM1(I)
124 - C
125 - C
126 -      GO TO 100
127 -      75 CONTINUE
128 -      IF (NSERIES.EQ.1) GO TO 100
129 -      IF ((ISQ.EQ.1).AND.(LFW1.NE.LFW2))
130 -      $CALL RDISC(LFW2,IDATSTART,OM2,NPTSLO)
131 -      IF ((ISQ.EQ.1).AND.(LFW1.EQ.LFW2))
132 -      $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
133 -      IF ((ISQ.EQ.5).AND.(LFW1.NE.LFW2))
134 -      $CALL RDISC(LFW2,IDATSTART+NPTSLO,OM2,NPTSHI)
135 -      IF ((ISQ.EQ.5).AND.(LFW1.EQ.LFW2))
136 -      $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
137 - C
138 -      100 CONTINUE
139 - C
140 -      OUTPUT 'MK 7'
141 -      CALL RASTER
142 -      NCPT=NFIRST
143 - C      OUTPUT 'MK 8',NCPT
144 -      105 CONTINUE
145 - C      OUTPUT 'MK 9'
146 -      CALL FREQINIT
147 -      110 CONTINUE
148 - C      OUTPUT 'MK 10'
149 -      IF3= IF1+IF2
150 -      IF2A=IABS(IF2)
151 -      IF3A=IABS(IF3)
152 -      IST3=IDATSTART+(2*(IF3A-1)+1)*NPIECES
153 -      NR3=2*NPIECES
154 -      IF (IF3A.EQ.LPHALF) NR3=NPIECES
155 -      IF (IF3A.EQ.0) NR3=NPIECES
156 -      IF (IF3A.GT.LPHALF) STOP 'CHKPT 1; CHK LOGIC'
157 - C READ IN SUM FREQUENCY COEFFICIENTS IF NECESSARY
158 - C      OUTPUT 'MK 101',ISQ
159 -      CALL SUMFREQ
160 - C      OUTPUT 'MK 102',IREAD
161 -      IF (IREAD.EQ.0) GO TO 120
162 - C DON'T READ SUM FREQ IF ALREADY HAVE IN THIS SCAN
163 -      IF (M33.EQ.1) GO TO 160
164 - C      OUTPUT 'MK 11'
165 - C      OUTPUT LFW1P2,IST3,NR3
166 -      CALL RDISC(LFW1P2,IST3,W3,NR3)
167 -      M33=1
168 -      GO TO 160
169 - C 117 OUTPUT 'MK 13'
170 - C
171 -      120 CONTINUE
172 - C      OUTPUT 'MK 131'
173 - C
174 - C SUM FREQ COEFF'S WILL COME FROM OM1 OR OM2
175 -      IF (ISQ.GT.2) GO TO 138
176 -      IST3=(2*IF3A-1)*NPIECES+1
177 -      IF (NSERIES.NE.1) GO TO 128
178 -      DO 125 I=1,NR3

```

```

179 - 125 W3(I)=OM1(IST3+I-1)
180 - GO TO 160
181 - 128 CONTINUE
182 - IF (LFW1.EQ.LFW2) GO TO 133
183 - IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) GO TO 133
184 - DO 130 I=1,NR3
185 - 130 W3(I)=OM1(IST3+I-1)
186 - GO TO 160
187 - 133 CONTINUE
188 - DO 135 I=1,NR3
189 - 135 W3(I)=OM2(IST3+I-1)
190 - GO TO 160
191 - 138 IF (ISQ.GT.3) GO TO 148
192 - IST3=2*(IF3A-NFHALF-1)*NPIECES+1
193 - IF (NSERIES.NE.1) GO TO 143
194 - DO 140 I=1,NR3
195 - 140 W3(I)=OM1(IST3+I-1)
196 - GO TO 160
197 - 143 CONTINUE
198 - IF (LFW1P2.EQ.LFW2) GO TO 148
199 - DO 145 I=1,NR3
200 - 145 W3(I)=OM1(IST3+I-1)
201 - GO TO 160
202 - 148 CONTINUE
203 - IF (ISQ.EQ.5) GO TO 155
204 - IF (LFW1P2.EQ.LFW2) GO TO 152
205 - IST3=2*(IF3A-NFHALF-1)*NPIECES+1
206 - DO 150 I=1,NR3
207 - 150 W3(I)=OM1(IST3+I-1)
208 - GO TO 160
209 - 152 IST3=(2*IF3A-1)*NPIECES+1
210 - DO 154 I=1,NR3
211 - 154 W3(I)=OM2(IST3+I-1)
212 - GO TO 160
213 - 155 CONTINUE
214 - IST3=(2*IF3A-1)*NPIECES+1
215 - IF (LFW1.NE.LFW2) GO TO 160
216 - DO 157 I=1,NR3
217 - 157 W3(I)=OM2(IST3+I-1)
218 - GO TO 160
219 - C
220 - C
221 - 160 CONTINUE
222 - C OUTPUT 'MK 16'
223 - C
224 - IST1=1+(2*(IF1-1)+1)*NPIECES
225 - IF (ISQ.GT.2) IST1=2*(IF1-NFHALF-1)*NPIECES+1
226 - IST2=1+(2*(IF2A-1)+1)*NPIECES
227 - IF (ISQ.EQ.5) IST2=2*(IF2A-NFHALF-1)*NPIECES+1
228 - IF (IF1.EQ.0) IST1=1
229 - IF (IF2A.EQ.0) IST2=1
230 - C CALCULATE BISPECTRAL COEFFICIENTS
231 - BR=BC=S1=S2=S3=0.
232 - ONE=1.
233 - IF (IF2.LT.0) ONE=-1.
234 - C
235 - C
236 - C OUTPUT 'MK 17',IST1,IST2
237 - DO 200 I=NPCL,NPC2
238 - R1=OM1(IST1+I-1)

```

```

239 -      C1=OM1(IST1+NPIECES+I-1)
240 -      IF ((IF1.EQ.0).OR.(IF1.EQ.LPHALF)) C1=0.
241 -      IF ((LFW1.EQ.LFW2).AND.(NSERIES.EQ.2).AND.
242 -      $(ISQ.NE.3).AND.(ISQ.NE.4)) GO TO 180
243 -      R2=OM2(IST2+I-1)
244 -      C2=OM2(IST2+NPIECES+I-1)
245 -      GO TO 190
246 - 180 CONTINUE
247 -      R2=OM1(IST2+I-1) ; C2=OM1(IST2+NPIECES+I-1)
248 - 190 CONTINUE
249 -      IF ((IF2A.EQ.0).OR.(IF2A.EQ.LPHALF)) C2=0.
250 -      R3=W3(I)
251 -      C3=W3(I+NPIECES)
252 -      IF ((IF3A.EQ.0).OR.(IF3A.EQ.LPHALF)) C3=0.
253 -      IF (INORM.EQ.0) GO TO 195
254 -      BRP=      R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
255 -      BCP=      ONE*R1*C2*R3-      R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
256 -      BMOD=DSQRT(BRP**2+BCP**2)
257 -      BR=BR+BRP/BMOD ; BC=BC+BCP/BMOD
258 -      GO TO 200
259 - 195 CONTINUE
260 -      BR=BR+      R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
261 -      BC=BC+ONE*R1*C2*R3-      R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
262 -      S1=S1+R1*R1+C1*C1
263 -      S2=S2+R2*R2+C2*C2
264 -      S3=S3+R3*R3+C3*C3
265 - 200 CONTINUE
266 - C
267 - C
268 -      IF (INORM.EQ.0) GO TO 202
269 -      BICOH=(DSQRT(BR**2+BC**2))/NPCS
270 -      GO TO 204
271 - 202 CONTINUE
272 -      BR=BR/NPCS ; BC=BC/NPCS
273 -      S1=S1/NPCS ; S2=S2/NPCS ; S3=S3/NPCS
274 - C      OUTPUT S1,S2,S3,NPCS
275 -      SD=DSQRT(S1*S2*S3)
276 - C      OUTPUT BR,BC,SD
277 -      BICOHR=BR/SD
278 -      BICOHC=BC/SD
279 -      BICOH=SQRT(BICOHR*BICOHR+BICOHC*BICOHC)
280 - 204 CONTINUE
281 -      BIPHSE=DATAN2(BC,BR)
282 -      BIPHSE=BIPHSE*360./TWOPI
283 - C      OUTPUT 'TWO',MTOT
284 -      IF (KONF.NE.0) CALL CBIN
285 - C      OUTPUT BICOH
286 -      BIMOD=DSQRT(BR*BR+BC*BC)
287 -      IF (LISTBI.EQ.0) GO TO 215
288 -      IF (INORM.NE.0) GO TO 212
289 -      IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
290 -      IF (IHANN.NE.1) GO TO 205
291 -      IF (BICOH.GE.SIGLVL) NPH=1 ;
292 -      $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 206
293 -      NPH=0 ; GO TO 206
294 - 205 CONTINUE
295 -      IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
296 -      $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 206
297 -      NPH=0
298 - 206 CONTINUE

```

```

299 -      WRITE (108,2000) IF1,IF2,IF3,BICOH,BIPHSE,BIMOD,BR,BC,
300 -      $S1,S2,S3,SD,NPH,(PHSERR(I),I=1,NPH)
301 - 2000 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,3(2X,E11.3),
302 -      $3(1X,E11.3),2X,E11.3,3X,N(F5.1))
303 -      GO TO 215
304 - 212 CONTINUE
305 -      IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
306 -      IF (IHANN.NE.1) GO TO 213
307 -      IF (BICOH.GE.SIGLVL) NPH=1 ;
308 -      $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 214
309 -      NPH=0 ; GO TO 214
310 - 213 CONTINUE
311 -      IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
312 -      $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 214
313 -      NPH=0
314 - 214 CONTINUE
315 -      WRITE (108,2100) IF1,IF2,IF3,BICOH,BIPHSE,
316 -      $BR/NPIECES,BC/NPIECES,NPH,(PHSERR(I),I=1,NPH)
317 - 2100 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
318 -      $2(3X,E11.3),3X,N(F5.1))
319 - 215 CONTINUE
320 - C      OUTPUT IF1
321 - C COMMIT BICOH TO STORAGE BUFFER SQ
322 -      IF (IDUMP.EQ.0) GO TO 242
323 -      CALL INDEX
324 -      SQ(NROW,NCOL)=BICOH
325 - 242 CONTINUE
326 - C INCREMENT FREQUENCIES
327 -      IF1=IF1+1 ; IF2=IF2-1
328 - C END OF SCAN?
329 -      CALL BOUNDARY
330 -      GO TO (110,250),IBDY+1
331 - C INCREMENT SCAN
332 - 250 NCPT=NCPT+1 ; M33=0
333 - C      WRITE (108,2222) IF1,ISQ,NCPT
334 - C2222 FORMAT (1H ,'IF1=',I3,2X,'ISQ=',I2,2X,'NCPT=',I3)
335 - C END OF RASTER?
336 -      IF (NCPT.LE.NLAST) GO TO 105
337 - C OUTPUT TO DISC
338 -      IF (IDUMP.EQ.0) GO TO 260
339 -      WRITE (ISTORE) ((SQ(I,J),I=1,NFHALF),J=1,NFHALF)
340 - 260 CONTINUE
341 - C INCREMENT INDEX FOR AREA BEING CONSIDERED (ISQ)
342 -      IF (ISQOUT.NE.0) OUTPUT ISQ
343 -      ISQ=ISQ+1
344 -      IF ((INSERT.EQ.1).AND.(ISQ.EQ.2)) ISQ=ISQ+1 ; GO TO 30
345 -      IF ((INSERT.EQ.1).AND.(ISQ.EQ.4)) GO TO 400
346 -      IF (ISQ.LE.5) GO TO 30
347 - C
348 - 400 CONTINUE
349 - C TRANSFER FOURIER LABELS TO BICOHERENCE FILE IF APPROPRIATE
350 -      IF (IDUMP.EQ.0) GO TO 450
351 -      IF (IDATSTART.NE.5) GO TO 420
352 -      CALL RDISC(LFW1,1,LABL,4)
353 -      WRITE (ISTORE) (LABL(I),I=1,4)
354 -      CALL RDISC(LFW2,1,LABL,4)
355 -      WRITE (ISTORE) (LABL(I),I=1,4)
356 -      CALL RDISC(LFW1P2,1,LABL,4)
357 -      WRITE (ISTORE) (LABL(I),I=1,4)
358 -      WRITE (ISTORE) MF

```

```

359 -      GO TO 440
360 -      420 CONTINUE
361 -      DO 425 I=1,3
362 -      425 LABL(I)=4H
363 -      LABL(4)=0
364 -      DO 430 KZ=1,3
365 -      430 WRITE (ISTORE) (LABL(I),I=1,4)
366 -      WRITE (ISTORE) MF
367 -      440 CONTINUE
368 -      IF (MF.NE.21) GO TO 450
369 -      CALL RDISC(21,1,LABL,33)
370 -      WRITE (ISTORE) LABL
371 -      450 CONTINUE
372 - C      OUTPUT 'THREE',MTOT
373 -      IF (KONF.NE.0) CALL CONFIDENCE
374 - C
375 - C
376 -      GO TO 750
377 -      700 STOP 'ABORTSET TERMINATION'
378 -      750 STOP 'NORMAL PROGRAM COMPLETION'
379 - C
380 - C
381 -      SUBROUTINE BOUNDARY
382 - C      INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO TEST
383 - C      SCAN FOR FREQUENCY LIMITS
384 -      IBDY=1
385 -      GO TO (100,200,300,400,500),ISQ
386 -      100 IF((IF1.LE.NFHALF).AND.(IF2.GT.0)) IBDY=0
387 -      RETURN
388 -      200 IF (IF1.LE.NFHALF) IBDY=0
389 -      RETURN
390 -      300 IF (IF2.GT.0) IBDY=0
391 -      RETURN
392 -      400 IF ((NCPT.LE.NFHALF).AND.(IF2.GE.-NFHALF)) IBDY=0
393 -      IF ((NCPT.GT.NFHALF).AND.(IF1.LE.LPHALF)) IBDY=0
394 -      RETURN
395 -      500 IF (IF1.LE.LPHALF) IBDY=0
396 -      RETURN
397 - C
398 - C
399 -      SUBROUTINE FREQINIT
400 - C      INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO INITIALIZE
401 - C      FREQUENCIES FOR A GIVEN LINE OF SCAN
402 -      GO TO (100,200,300,400,500),ISQ
403 -      100 IF1=NCPT/2 ; IF2=NCPT/2
404 -      IF (MOD(NCPT,2).EQ.1) IF1=IF1+1
405 -      RETURN
406 -      200 IF1=NCPT+1 ; IF2=-1
407 -      RETURN
408 -      300 IF1=NFIRST-1 ; IF2=NCPT-IF1
409 -      RETURN
410 -      400 CONTINUE
411 -      IF (NCPT.LE.NFHALF) IF1=NFHALF+1 ;
412 -      $IF2=-IF1+NCPT
413 -      IF (NCPT.GT.NFHALF) IF1=NCPT+1 ; IF2=-1
414 -      RETURN
415 -      500 IF1=NCPT+NFHALF+1 ; IF2=-NFHALF-1
416 -      RETURN
417 - C
418 - C

```

```

419 -      SUBROUTINE INDEX
420 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO RELATE
421 - C FREQUENCIES TO STORAGE ARRAY INDICES
422 -      GO TO (100,200,300,400,500),ISQ
423 -      100 NROW=IF2 ; NCOL=IF1
424 -      RETURN
425 -      200 NROW=IF2+NFHALF+1 ; NCOL=IF1
426 -      RETURN
427 -      300 NROW=IF2 ; NCOL=IF1-NFHALF
428 -      RETURN
429 -      400 NROW=IF2+NFHALF+1 ; NCOL=IF1-NFHALF
430 -      RETURN
431 -      500 NROW=IF2+2*NFHALF+1 ; NCOL=IF1-NFHALF
432 -      RETURN
433 - C
434 - C
435 -      SUBROUTINE RASTER
436 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO SET UP SCAN
437 - C PARAMETERS
438 -      GO TO (100,200,300,400,500),ISQ
439 -      100 NFIRST=2 ; NLAST=LPHALF
440 -      RETURN
441 -      200 NFIRST=1 ; NLAST=NFHALF-1
442 -      RETURN
443 -      300 NFIRST=NFHALF+2 ; NLAST=LPHALF
444 -      RETURN
445 -      400 NFIRST=1 ; NLAST=LPHALF-1
446 -      RETURN
447 -      500 NFIRST=1 ; NLAST=NFHALF-1
448 -      RETURN
449 - C
450 - C
451 -      SUBROUTINE SUMFREQ
452 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO DETERMINE WHERE
453 - C TO FIND SUM FREQUENCY COEFFICIENTS
454 - C RETURNS IREAD=1 IF READING IS NECESSARY, 0 OTHERWISE
455 -      IREAD=0
456 -      GO TO (100,400,150,200,300),ISQ
457 -      100 IF (IF3A.GT.NFHALF) IREAD=1
458 -      RETURN
459 -      150 CONTINUE
460 -      IF (INSERT.EQ.1) RETURN
461 -      IF (LFW1.EQ.LFW2) IREAD=1 ; RETURN
462 -      IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) IREAD=1 ; RETURN
463 -      RETURN
464 -      200 CONTINUE
465 -      IF ((LFW1P2.EQ.LFW1).AND.(IF3A.LE.NFHALF).AND.
466 -      $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
467 -      IF ((LFW1P2.EQ.LFW2).AND.(IF3A.GT.NFHALF).AND.
468 -      $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
469 -      IF ((LFW1.EQ.LFW2).AND.(INSERT.EQ.2)) IREAD=1 ; RETURN
470 -      RETURN
471 -      300 IF (LFW1.NE.LFW2) IREAD=1
472 -      400 CONTINUE
473 -      RETURN
474 - C
475 - C
476 -      SUBROUTINE CBIN
477 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO DETERMINE
478 - C CONFIDENCE LIMITS FOR BICOHERENCES

```

```

479 - C HAS ENTRY POINTS CBIN TO PLACE GIVEN BICOHERENCE IN
480 - C BIN AND CONFIDENCE TO INITIATE CONFIDENCE LIMIT CALCULATION
481 - C     OUTPUT 'FOUR',MTOT
482 - C     IF (BICOH.LE.BICOH LIM) GO TO 100
483 - C     KOVER=KOVER+1
484 - C     WRITE (108,1000) IF1,IF2,IF3,BICOH
485 - 1000 FORMAT ('BICOH LIM EXCEEDED AT FREQ TRIPLET ',3(I3,X),
486 - $' WITH VALUE ',G11.5)
487 - C     IF (IHI.NE.0) MTOT=MTOT+1
488 - C     OUTPUT 'FIVE',MTOT
489 - C     RETURN
490 - C
491 - 100 CONTINUE
492 - C     NIB=INT(BICOH/BINSIZE)+1
493 - C     IF (NIB.GT.NBINS) GO TO 250
494 - C     KBIN(NIB)=KBIN(NIB)+1
495 - C     MTOT=MTOT+1
496 - C     RETURN
497 - C
498 - 250 CONTINUE
499 - C     WRITE (108,1025) BICOH,IF1,IF2,IF3
500 - 1025 FORMAT ('BICOH OF ',F6.2,' AT FREQ TRIPLET ',3(I3,X),
501 - $' MET NEITHER BIN NOR LIM CRITERIA')
502 - C     MTOT=MTOT+1
503 - C     RETURN
504 - C
505 - C *****
506 - C     ENTRY CONFIDENCE
507 - C *****
508 - C OPTIONAL DISPLAY OF PARTITIONING OF BICOHERENCES
509 - C     IF (KB1.EQ.0) GO TO 320
510 - C     WRITE (108,1050)
511 - 1050 FORMAT (/,T2,'BIN NO',T14,'BIN LIMITS',T32,'NO OF BIC'S'
512 - $,/)
513 - C
514 - C
515 - C     DO 300 I=1,NBINS
516 - C     WRITE (108,1100) I,(I-1)*BINSIZE,I*BINSIZE,KBIN(I)
517 - 1100 FORMAT (T3,I3,T13,2(F4.2,4X),T34,I5)
518 - C     300 CONTINUE
519 - C
520 - C
521 - 320 CONTINUE
522 - C CONVERT NO OF BIC'S IN KBIN TO TOTAL FRACTION AT INDEX VALUE
523 - C     KSUBTOT=0 ; I=0
524 - C     WRITE (108,1125) MTOT
525 - 1125 FORMAT (/, 'TOTAL NO OF BIC'S SUBMITTED: ',I5,/)
526 - C     FTOT=FLOAT(MTOT)
527 - C
528 - 340 CONTINUE
529 - C     I=I+1
530 - C     IF (I.GT.NBINS) GO TO 350
531 - C
532 - C     KSUBTOT=KSUBTOT+KBIN(I)
533 - C     OUTPUT I,KSUBTOT,KBIN(I),FTOT
534 - C     ABIN(I)=FLOAT(KSUBTOT)
535 - C     ABIN(I)=ABIN(I)/FTOT
536 - C     OUTPUT ABIN(I)
537 - C     GO TO 340
538 - C

```

```

539 - 350 CONTINUE
540 - C OPTIONAL DISPLAY OF PARTIAL SUMS AT INDEX VALUE
541 - IF (KB2.EQ.0) GO TO 400
542 - WRITE (108,1150)
543 - 1150 FORMAT (/,T32,'FRACT BELOW',/,T2,'BIN NO',T14,
544 - $'BIN LIMITS',T31,'UPPER BIN LIM',/)
545 - C
546 - C
547 - DO 380 I=1,NBINS
548 - WRITE (108,1200) I,(I-1)*BINSIZE,I*BINSIZE,ABIN(I)
549 - 1200 FORMAT (T3,I3,T13,2(F4.2,4X),T34,F5.3)
550 - 380 CONTINUE
551 - C
552 - C
553 - 400 CONTINUE
554 - C
555 - C
556 - C COMPUTE CONFIDENCE LIMITS
557 - I=0
558 - C
559 - 420 CONTINUE
560 - I=I+1
561 - IF (I.GT.NCONF) GO TO 600
562 - J=0
563 - C
564 - 450 CONTINUE
565 - J=J+1
566 - IF (J.GT.NBINS) WRITE (108,1250) CONF(I) ; BCL(I)=999. ;
567 - $GO TO 420
568 - 1250 FORMAT ('BINS EXHAUSTED FOR CONFIDENCE LEVEL',F6.3)
569 - IF (ABIN(J).GE.CONF(I)) GO TO 470
570 - GO TO 450
571 - C
572 - 470 CONTINUE
573 - C
574 - JH=J
575 - JL=J-1
576 - Y1=JL*BINSIZE
577 - Y2=JH*BINSIZE
578 - X1=0
579 - IF (JL.NE.0) X1=ABIN(JL)
580 - X2=ABIN(JH)
581 - X0=CONF(I)
582 - BCL(I)=Y1+(X0-X1)*(Y2-Y1)/(X2-X1)
583 - GO TO 420
584 - 600 CONTINUE
585 - C
586 - C PRINT CONFIDENCE LEVELS
587 - WRITE (108,1400) NCONF,(CONF(I)*100.,BCL(I),I=1,NCONF)
588 - 1400 FORMAT (/,T4,'CONFIDENCE',/,T6,'LEVEL',T21,'BICOHERENCE',
589 - $/,T5,'(PERCENT)',/,N(T7,F4.1,T24,F5.3,/),/)
590 - WRITE (108,1450) BICOHLIM,KOVER
591 - 1450 FORMAT ('NO. OF BICOHERENCES GREATER THAN SPECIFIED ',
592 - $'MAXIMUM OF ',F5.2,' IS ',I5,/)
593 - RETURN
594 - END

```

- PROGRAM BISUM -

```

1 - C          <<<<<< B I S U M 0 2 >>>>>>
2 - C CALCULATES BISPECTRA AND CROSS-BISPECTRA FOR ONE OR TWO
3 - C REAL SERIES AND INTEGRATES THEM OVER CONSTANT SUM FREQUENCIES
4 - C AND PRINTS FORMATTED SUMMARY
5 - C   PROGRAMMER: GERARD H. MARTINEAU           1/19/78
6 - C   ORIGINATOR: MELBOURNE G. BRISCOE
7 - C   DIMENSION OM1(5400),OM2(5400),SQ(70,70),
8 - C   $W3(300),CONF(10),BCL(10),KARD(20),
9 - C   $LABL(33),PHSERR(2)
10 - C   DIMENSION BRU(130),BIU(130),BICU(130),BIC2U(130),
11 - C   $BCSU(130),BRL(130),BIL(130),BICL(130),BIC2L(130),
12 - C   $BCSL(130)
13 - C   DOUBLE PRECISION R1,R2,R3,C1,C2,C3,S1,S2,S3,BR,BC,BMOD,
14 - C   $BRP,BCP,BM
15 - C   NAMELIST IDATSTART,MF,ISTORE,
16 - C   $BINSIZE,BICOHLIM,IHI,KB1,KB2,LISTBI,IDUMP,INORM,
17 - C   $ISQOUT,IHANN,SIGLVL,KDCB,KDISC
18 - C   CALL ABORTSET(700S,1)
19 - C   PI=3.14159265;TWOPI=2.*PI
20 - C   IDUMP=0
21 - C   KDCB=2 ; KDISC=0
22 - C   MF=21;IDATSTART=5
23 - C   ISTORE=0
24 - C   JDIM=130
25 - C   IHI=1
26 - C   M33=0
27 - C   BINSIZE=.01;BICOHLIM=1.5;KB1=KB2=1;KONF=0
28 - C   SIGLVL=-1 ;           PHSERR(2)=0.
29 - C   KOVER=0 ; LISTBI=0
30 - C   MTOT=0
31 - C   INORM=0
32 - C   ISQOUT=0 ; IHANN=1
33 - C   OUTPUT 'ONE',MTOT
34 - C
35 - C
36 - C INITIALIZE OUTPUT ARRAYS
37 - C   DO 10 I=1,JDIM
38 - C   BRU(I)=BIU(I)=BIC2U(I)=BICU(I)=BCSU(I)=0.
39 - C   BRL(I)=BIL(I)=BIC2L(I)=BICL(I)=BCSL(I)=0.
40 - C   10 CONTINUE
41 - C
42 - C
43 - C   ISQD=70
44 - C   INPUT
45 - C   CALL DPAR(NERR,MF,1,1,KSUM,512)
46 - C   OUTPUT IDATSTART,MF,ISTORE,BINSIZE,
47 - C   $BICOHLIM,IHI,KB1,KB2,LISTBI,IDUMP,INORM,IHANN,
48 - C   $KDISC,KDCB
49 - C CHECK EXISTENCE OF DCB ASSIGNMENT FOR OUTPUT FILE
50 - C   IF (IDUMP.NE.0) CALL GETDCB(ISTORE,LOCC)
51 - C   READ (105,1010) KARD
52 - C   1010 FORMAT (20A4)
53 - C   WRITE (108,1020) KARD
54 - C   1020 FORMAT (1H ,20A4)
55 - C   DECODE (80,1050,KARD) LFW1,LFW2,LFW1P2,NPIECES,
56 - C   $LPIECE,NPC1,NPC2,CONF1,CONF2
57 - C   1050 FORMAT (9G)
58 - C   23 CONTINUE

```

```

59 -      IF ((LISTBI.NE.0).AND.(INORM.EQ.0)) WRITE (108,1321)
60 - 1321 FORMAT (/ ,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
61 -      $T40,'BIMOD',T52,'BISPECR',T65,'BISPECI',T78,'AUTOF1',
62 -      $T90,'AUTOF2',T101,'AUTOF1F2',T117,'SD',T125,'PHSERR',/)
63 -      IF ((LISTBI.NE.0).AND.(INORM.NE.0)) WRITE (108,1431)
64 - 1431 FORMAT (/ ,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
65 -      $T41,'BISPPR',T55,'BISPP1',T65,'PHSERR',/)
66 -      NPCS=NPC2-NPC1+1
67 -      IF ((NPCS.EQ.1).AND.(IHANN.EQ.1)) IHANN=0 ; OUTPUT
68 -      $'IHANN HAS BEEN SET TO 0. CAN'T OVERLAP WITH ONE PIECE'
69 -      IF (IHANN.EQ.1) EDOF=36.*NPCS*NPCS/(19*NPC5-1)
70 -      IF (IHANN.EQ.0) EDOF=2*NPC5
71 -      IF (SIGLVL.LT.0.) SIGLVL=SQRT(6./EDOF)
72 -      LPHALF=LPIECE/2
73 -      IF (MOD(LPHALF,2).EQ.1)
74 -      $STOP '(PIECE LENGTH)/2 MUST BE EVEN'
75 -      NFHALF=LPHALF/2
76 -      IF (LISTBI.EQ.0) GO TO 25
77 - 25 CONTINUE
78 - C DECIDE WHETHER TO CALCULATE CROSS-BISPECTRA
79 -      NSERIES=2
80 -      IF ((LFW1.NE.LFW2).AND.(LFW1P2.NE.LFW1).AND.
81 -      $(LFW1P2.NE.LFW2)) OUTPUT
82 -      $'CANNOT HAVE MORE THAN TWO INPUT SERIES' ; STOP
83 -      IF ((LFW1.EQ.LFW2).AND.(LFW2.EQ.LFW1P2)) NSERIES=1
84 -      ISQ=1
85 - C CHECK INPUT INFO. AGAINST LABELS
86 - C      OUTPUT 'MK 1',IDATSTART
87 - C
88 - C
89 -      NFREQLO=NFHALF+1
90 -      NFREQHI=NFHALF
91 -      NPTSLO=NPIECES*(2*NFREQLO-1)
92 -      NPTSHI=NPIECES*(2*NFREQHI-1)
93 - C
94 - C
95 - 30 CONTINUE
96 - C
97 - C SET ARRAY SQ TO -999.
98 - C
99 -      IF (IDUMP.EQ.0) GO TO 41
100 - C
101 -      DO 40 I=1,ISQD
102 - C
103 -      DO 40 J=1,ISQD
104 - 40 SQ(I,J)=-999.
105 - C
106 - 41 CONTINUE
107 - C
108 - C      OUTPUT LFW1,IDATSTART,NPTSLO,NPTSHI,ISQ
109 -      IF (ISQ.EQ.1) CALL RDISC(LFW1,IDATSTART,OM1,NPTSLO)
110 - C      OUTPUT 'MK 3'
111 -      IF ((ISQ.NE.3).OR.(NSERIES.NE.2).OR.(LFW1.NE.LFW2))
112 -      $GO TO 46
113 -      DO 43 I=1,NPTSLO
114 - 43 OM2(I)=OM1(I)
115 - 46 CONTINUE
116 -      IF (ISQ.EQ.3)CALL RDISC(LFW1,IDATSTART+NPTSLO,OM1,NPTSHI)
117 - C      OUTPUT 'MK 4'
118 -      IF ((ISQ.NE.1).OR.(NSERIES.NE.1)) GO TO 75

```

```

119 - C
120 - C
121 -     DO 50 I=1,NPTSLO
122 -     50 OM2(I)=OM1(I)
123 - C
124 - C
125 -     GO TO 100
126 -     75 CONTINUE
127 -     IF (NSERIES.EQ.1) GO TO 100
128 -     IF ((ISQ.EQ.1).AND.(LFW1.NE.LFW2))
129 -     $CALL RDISC(LFW2,IDATSTART,OM2,NPTSLO)
130 -     IF ((ISQ.EQ.1).AND.(LFW1.EQ.LFW2))
131 -     $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
132 -     IF ((ISQ.EQ.5).AND.(LFW1.NE.LFW2))
133 -     $CALL RDISC(LFW2,IDATSTART+NPTSLO,OM2,NPTSHI)
134 -     IF ((ISQ.EQ.5).AND.(LFW1.EQ.LFW2))
135 -     $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
136 - C
137 -     100 CONTINUE
138 - C
139 - C     OUTPUT 'MK 7'
140 -     CALL RASTER
141 -     NCPT=NFIRST
142 - C     OUTPUT 'MK 8',NCPT
143 -     105 CONTINUE
144 - C     OUTPUT 'MK 9'
145 -     CALL FREQINIT
146 -     110 CONTINUE
147 - C     OUTPUT 'MK 10'
148 -     IF3= IF1+IF2
149 -     IF2A=IABS(IF2)
150 -     IF3A=IABS(IF3)
151 -     IST3=IDATSTART+(2*(IF3A-1)+1)*NPIECES
152 -     NR3=2*NPIECES
153 -     IF (IF3A.EQ.LPHALF) NR3=NPIECES
154 -     IF (IF3A.EQ.0) NR3=NPIECES
155 -     IF (IF3A.GT.LPHALF) STOP 'CHKPT 1; CHK LOGIC'
156 - C READ IN SUM FREQUENCY COEFFICIENTS IF NECESSARY
157 - C     OUTPUT 'MK 101',ISQ
158 -     CALL SUMFREQ
159 - C     OUTPUT 'MK 102',IREAD
160 -     IF (IREAD.EQ.0) GO TO 120
161 - C DON'T READ SUM FREQ IF ALREADY HAVE IN THIS SCAN
162 -     IF (M33.EQ.1) GO TO 160
163 - C     OUTPUT 'MK 11'
164 - C     OUTPUT LFW1P2,IST3,NR3
165 -     CALL RDISC(LFW1P2,IST3,W3,NR3)
166 -     M33=1
167 -     GO TO 160
168 - C 117 OUTPUT 'MK 13'
169 - C
170 -     120 CONTINUE
171 - C     OUTPUT 'MK 131'
172 - C
173 - C SUM FREQ COEFF'S WILL COME FROM OM1 OR OM2
174 -     IF (ISQ.GT.2) GO TO 138
175 -     IST3=(2*IF3A-1)*NPIECES+1
176 -     IF (NSERIES.NE.1) GO TO 128
177 -     DO 125 I=1,NR3
178 -     125 W3(I)=OM1(IST3+I-1)

```

```

179 -      GO TO 160
180 - 128 CONTINUE
181 -      IF (LFW1.EQ.LFW2) GO TO 133
182 -      IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) GO TO 133
183 -      DO 130 I=1,NR3
184 - 130 W3(I)=OM1(IST3+I-1)
185 -      GO TO 160
186 - 133 CONTINUE
187 -      DO 135 I=1,NR3
188 - 135 W3(I)=OM2(IST3+I-1)
189 -      GO TO 160
190 - 138 IF (ISQ.GT.3) GO TO 148
191 -      IST3=2*(IF3A-NFHALF-1)*NPIECES+1
192 -      IF (NSERIES.NE.1) GO TO 143
193 -      DO 140 I=1,NR3
194 - 140 W3(I)=OM1(IST3+I-1)
195 -      GO TO 160
196 - 143 CONTINUE
197 -      IF (LFW1P2.EQ.LFW2) GO TO 148
198 -      DO 145 I=1,NR3
199 - 145 W3(I)=OM1(IST3+I-1)
200 -      GO TO 160
201 - 148 CONTINUE
202 -      IF (ISQ.EQ.5) GO TO 155
203 -      IF (LFW1P2.EQ.LFW2) GO TO 152
204 -      IST3=2*(IF3A-NFHALF-1)*NPIECES+1
205 -      DO 150 I=1,NR3
206 - 150 W3(I)=OM1(IST3+I-1)
207 -      GO TO 160
208 - 152 IST3=(2*IF3A-1)*NPIECES+1
209 -      DO 154 I=1,NR3
210 - 154 W3(I)=OM2(IST3+I-1)
211 -      GO TO 160
212 - 155 CONTINUE
213 -      IST3=(2*IF3A-1)*NPIECES+1
214 -      IF (LFW1.NE.LFW2) GO TO 160
215 -      DO 157 I=1,NR3
216 - 157 W3(I)=OM2(IST3+I-1)
217 -      GO TO 160
218 - C
219 - C
220 - 160 CONTINUE
221 - C      OUTPUT 'MK 16'
222 - C
223 -      IST1=1+(2*(IF1-1)+1)*NPIECES
224 -      IF (ISQ.GT.2) IST1=2*(IF1-NFHALF-1)*NPIECES+1
225 -      IST2=1+(2*(IF2A-1)+1)*NPIECES
226 -      IF (ISQ.EQ.5) IST2=2*(IF2A-NFHALF-1)*NPIECES+1
227 -      IF (IF1.EQ.0) IST1=1
228 -      IF (IF2A.EQ.0) IST2=1
229 - C CALCULATE BISPECTRAL COEFFICIENTS
230 -      BR=BC=S1=S2=S3=0.
231 -      ONE=1.
232 -      IF (IF2.LT.0) ONE=-1.
233 - C
234 - C
235 - C      OUTPUT 'MK 17',IST1,IST2
236 -      DO 200 I=NPCL,NPC2
237 -      R1=OM1(IST1+I-1)
238 -      C1=OM1(IST1+NPIECES+I-1)

```

```

239 -      IF ((IF1.EQ.0).OR.(IF1.EQ.LPHALF)) C1=0.
240 -      IF ((LFW1.EQ.LFW2).AND.(NSERIES.EQ.2).AND.
241 -      $(ISQ.NE.3).AND.(ISQ.NE.4)) GO TO 180
242 -      R2=UM2(IST2+I-1)
243 -      C2=OM2(IST2+NPIECES+I-1)
244 -      GO TO 190
245 - 180 CONTINUE
246 -      R2=OM1(IST2+I-1) ; C2=OM1(IST2+NPIECES+I-1)
247 - 190 CONTINUE
248 -      IF ((IF2A.EQ.0).OR.(IF2A.EQ.LPHALF)) C2=0.
249 -      R3=W3(I)
250 -      C3=W3(I+NPIECES)
251 -      IF ((IF3A.EQ.0).OR.(IF3A.EQ.LPHALF)) C3=0.
252 -      IF (INORM.EQ.0) GO TO 195
253 -      BRP=      R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
254 -      BCP=  ONE*R1*C2*R3-      R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
255 -      BMOD=DSQRT(BRP**2+BCP**2)
256 -      BR=BR+BRP/BMOD ; BC=BC+BCP/BMOD
257 -      GO TO 200
258 - 195 CONTINUE
259 -      BR=BR+      R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
260 -      BC=BC+ONE*R1*C2*R3-      R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
261 -      S1=S1+R1*R1+C1*C1
262 -      S2=S2+R2*R2+C2*C2
263 -      S3=S3+R3*R3+C3*C3
264 - 200 CONTINUE
265 - C
266 - C
267 -      IF (INORM.NE.0) GO TO 204
268 - 202 CONTINUE
269 -      BR=BR/NPCS ; BC=BC/NPCS
270 -      S1=S1/NPCS ; S2=S2/NPCS ; S3=S3/NPCS
271 - C      OUTPUT S1,S2,S3,NPCS
272 -      SD=DSQRT(S1*S2*S3)
273 - C      OUTPUT BR,BC,SD
274 -      BICOHR=BR/SD
275 -      BICOHC=BC/SD
276 - 204 CONTINUE
277 -      BIPHSE=DATAN2(BC,BR)
278 -      BIPHSE=BIPHSE*360./TWOPI
279 -      IF (INORM.EQ.0) GO TO 206
280 -      BIMOD=DSQRT(BR**2+BC**2)
281 -      BICOH=BIMOD/NPCS
282 -      IF (IF2.LT.0) GO TO 205
283 -      BRU(IF3A)=BRU(IF3A)+BR/NPCS
284 -      BIU(IF3A)=BIU(IF3A)+BC/NPCS
285 -      BICU(IF3A)=BICU(IF3A)+BICOH
286 -      BIC2U(IF3A)=BIC2U(IF3A)+BICOH**2
287 -      BCSU(IF3A)=0.
288 -      GO TO 208
289 - 205 CONTINUE
290 -      BRL(IF3A)=BRL(IF3A)+BR/NPCS
291 -      BIL(IF3A)=BIL(IF3A)+BC/NPCS
292 -      BICL(IF3A)=BICL(IF3A)+BICOH
293 -      BIC2L(IF3A)=BIC2L(IF3A)+BICOH**2
294 -      BCSL(IF3A)=0.
295 -      GO TO 208
296 - 206 CONTINUE
297 -      BCM=BICOH+R*BICOHR+BICOHC*BICOHC
298 -      BICOH=SQRT(BCM)

```

```

299 - C      OUTPUT 'TWO',MTOT
300 - C      OUTPUT BICOH
301 -      BM=BR*BR+BC*BC
302 -      BIMOD=DSQRT(BM)
303 -      IF (IF2.LT.0) GO TO 207
304 -      BRU(IF3A)=BRU(IF3A)+BR
305 -      BIU(IF3A)=BIU(IF3A)+BC
306 -      BICU(IF3A)=BICU(IF3A)+BICOH
307 -      BIC2U(IF3A)=BIC2U(IF3A)+BCM
308 -      BCSU(IF3A)=BCSU(IF3A)+BCM*S3
309 -      GO TO 208
310 - 207 CONTINUE
311 -      BRL(IF3A)=BRL(IF3A)+BR
312 -      BIL(IF3A)=BIL(IF3A)+BC
313 -      BICL(IF3A)=BICL(IF3A)+BICOH
314 -      BIC2L(IF3A)=BIC2L(IF3A)+BCM
315 -      BCSL(IF3A)=BCSL(IF3A)+BCM*S3
316 - 208 CONTINUE
317 -      IF (LISTBI.EQ.0) GO TO 215
318 -      IF (INORM.NE.0) GO TO 212
319 -      IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
320 -      IF (IHANN.NE.1) GO TO 210
321 -      IF (BICOH.GE.SIGLVL) NPH=1 ;
322 -      $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 211
323 -      NPH=0 ; GO TO 211
324 - 210 CONTINUE
325 -      IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
326 -      $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 211
327 -      NPH=0
328 - 211 CONTINUE
329 -      WRITE (108,2000) IF1,IF2,IF3,BICOH,BIPHSE,BIMOD,BR,BC,
330 -      $S1,S2,S3,SD,NPH,(PHSERR(1),I=1,NPH)
331 - 2000 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,3(2X,E11.3),
332 -      $3(1X,E11.3),2X,E11.3,3X,N(F5.1))
333 -      GO TO 215
334 - 212 CONTINUE
335 -      IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
336 -      IF (IHANN.NE.1) GO TO 213
337 -      IF (BICOH.GE.SIGLVL) NPH=1 ;
338 -      $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 214
339 -      NPH=0 ; GO TO 214
340 - 213 CONTINUE
341 -      IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
342 -      $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 214
343 -      NPH=0
344 - 214 CONTINUE
345 -      WRITE (108,2100) IF1,IF2,IF3,BICOH,BIPHSE,
346 -      $BR/NPIECES,BC/NPIECES,NPH,(PHSERR(1),I=1,NPH)
347 - 2100 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
348 -      $2(3X,E11.3),3X,N(F5.1))
349 - 215 CONTINUE
350 - C      OUTPUT IF1
351 - C      COMMIT BICOH TO STORAGE BUFFER SQ
352 -      IF (IDUMP.EQ.0) GO TO 242
353 -      CALL INDEX
354 -      SQ(NROW,NCOL)=BICOH
355 - 242 CONTINUE
356 - C      INCREMENT FREQUENCIES
357 -      IF1=IF1+1 ; IF2=IF2-1
358 - C      END OF SCAN?

```

```

359 -      CALL BOUNDARY
360 -      GO TO (110,250),IBDY+1
361 - C INCREMENT SCAN
362 -      250 NCPT=NCPT+1 ; M33=0
363 - C      WRITE (108,2222) IF1,ISQ,NCPT
364 - C2222 FORMAT (1H ,'IF1=',I3,2X,'ISQ=',I2,2X,'NCPT=',I3)
365 - C END OF RASTER?
366 -      IF (NCPT.LE.NLAST) GO TO 105
367 - C OUTPUT TO DISC
368 -      IF (IDUMP.EQ.0) GO TO 260
369 -      WRITE (ISTORE) ((SQ(I,J),I=1,NFHALF),J=1,NFHALF)
370 -      260 CONTINUE
371 - C INCREMENT INDEX FOR AREA BEING CONSIDERED (ISQ)
372 -      IF (ISQOUT.NE.0) OUTPUT ISQ
373 -      ISQ=ISQ+1
374 -      IF ((INSEKIES.EQ.1).AND.(ISQ.EQ.2)) ISQ=ISQ+1 ; GO TO 30
375 -      IF ((INSERIES.EQ.1).AND.(ISQ.EQ.4)) GO TO 400
376 -      IF (ISQ.LE.5) GO TO 30
377 - C
378 -      400 CONTINUE
379 - C TRANSFER FOURIER LABELS TO BICOHERENCE FILE IF APPROPRIATE
380 -      IF (IDUMP.EQ.0) GO TO 450
381 -      IF (IDATSTART.NE.5) GO TO 420
382 -      CALL RDISC(LFW1,1,LABL,4)
383 -      WRITE (ISTORE) (LABL(I),I=1,4)
384 -      CALL RDISC(LFW2,1,LABL,4)
385 -      WRITE (ISTORE) (LABL(I),I=1,4)
386 -      CALL RDISC(LFW1P2,1,LABL,4)
387 -      WRITE (ISTORE) (LABL(I),I=1,4)
388 -      WRITE (ISTORE) MF
389 -      GO TO 440
390 -      420 CONTINUE
391 -      DO 425 I=1,3
392 -      425 LABL(I)=4H
393 -      LABL(4)=0
394 -      DO 430 KZ=1,3
395 -      430 WRITE (ISTORE) (LABL(I),I=1,4)
396 -      WRITE (ISTORE) MF
397 -      440 CONTINUE
398 -      IF (MF.NE.21) GO TO 450
399 -      CALL RDISC(21,1,LABL,33)
400 -      WRITE (ISTORE) LABL
401 -      450 CONTINUE
402 - C
403 -      CALL PRINTOUT
404 - C
405 -      GO TO 750
406 -      700 STOP 'ABORTSET TERMINATION'
407 -      750 STOP 'NORMAL PROGRAM COMPLETION'
408 - C
409 - C
410 -      SUBROUTINE BOUNDARY
411 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO TEST
412 - C SCAN FOR FREQUENCY LIMITS
413 -      IBDY=1
414 -      GO TO (100,200,300,400,500),ISQ
415 -      100 IF((IF1.LE.NFHALF).AND.(IF2.GT.0)) IBDY=0
416 -      RETURN
417 -      200 IF (IF1.LE.NFHALF) IBDY=0
418 -      RETURN

```

```

419 - 300 IF (IF2.GT.0) IBDY=0
420 - RETURN
421 - 400 IF ((NCPT.LE.NFHALF).AND.(IF2.GE.-NFHALF)) IBDY=0
422 - IF ((NCPT.GT.NFHALF).AND.(IF1.LE.LPHALF)) IBDY=0
423 - RETURN
424 - 500 IF (IF1.LE.LPHALF) IBDY=0
425 - RETURN
426 - C
427 - C
428 - SUBROUTINE FREQUINIT
429 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO INITIALIZE
430 - C FREQUENCIES FOR A GIVEN LINE OF SCAN
431 - GO TO (100,200,300,400,500),ISQ
432 - 100 IF1=NCPT/2 ; IF2=NCPT/2
433 - IF (MOD(NCPT,2).EQ.1) IF1=IF1+1
434 - RETURN
435 - 200 IF1=NCPT+1 ; IF2=-1
436 - RETURN
437 - 300 IF1=NFIRST-1 ; IF2=NCPT-IF1
438 - RETURN
439 - 400 CONTINUE
440 - IF (NCPT.LE.NFHALF) IF1=NFHALF+1 ;
441 - IF2=-IF1+NCPT
442 - IF (NCPT.GT.NFHALF) IF1=NCPT+1 ; IF2=-1
443 - RETURN
444 - 500 IF1=NCPT+NFHALF+1 ; IF2=-NFHALF-1
445 - RETURN
446 - C
447 - C
448 - SUBROUTINE INDEX
449 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO RELATE
450 - C FREQUENCIES TO STORAGE ARRAY INDICES
451 - GO TO (100,200,300,400,500),ISQ
452 - 100 NROW=IF2 ; NCOL=IF1
453 - RETURN
454 - 200 NROW=IF2+NFHALF+1 ; NCOL=IF1
455 - RETURN
456 - 300 NROW=IF2 ; NCOL=IF1-NFHALF
457 - RETURN
458 - 400 NROW=IF2+NFHALF+1 ; NCOL=IF1-NFHALF
459 - RETURN
460 - 500 NROW=IF2+2*NFHALF+1 ; NCOL=IF1-NFHALF
461 - RETURN
462 - C
463 - C
464 - SUBROUTINE RASTER
465 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO SET UP SCAN
466 - C PARAMETERS
467 - GO TO (100,200,300,400,500),ISQ
468 - 100 NFIRST=2 ; NLAST=LPHALF
469 - RETURN
470 - 200 NFIRST=1 ; NLAST=NFHALF-1
471 - RETURN
472 - 300 NFIRST=NFHALF+2 ; NLAST=LPHALF
473 - RETURN
474 - 400 NFIRST=1 ; NLAST=LPHALF-1
475 - RETURN
476 - 500 NFIRST=1 ; NLAST=NFHALF-1
477 - RETURN
478 - C

```

```

479 - C
480 -     SUBROUTINE SUMFREQ
481 - C INTERNAL SUBROUTINE FOR PROGRAM BISCALO3 TO DETERMINE WHERE
482 - C TO FIND SUM FREQUENCY COEFFICIENTS
483 - C RETURNS IREAD=1 IF READING IS NECESSARY, 0 OTHERWISE
484 -     IREAD=0
485 -     GO TO (100,400,150,200,300),ISQ
486 -     100 IF (IF3A.GT.NFHALF) IREAD=1
487 -     RETURN
488 -     150 CONTINUE
489 -     IF (NSERIES.EQ.1) RETURN
490 -     IF (LFW1.EQ.LFW2) IREAD=1 ; RETURN
491 -     IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) IREAD=1 ; RETURN
492 -     RETURN
493 -     200 CONTINUE
494 -     IF ((LFW1P2.EQ.LFW1).AND.(IF3A.LE.NFHALF).AND.
495 -     $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
496 -     IF ((LFW1P2.EQ.LFW2).AND.(IF3A.GT.NFHALF).AND.
497 -     $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
498 -     IF ((LFW1.EQ.LFW2).AND.(NSERIES.EQ.2)) IREAD=1 ; RETURN
499 -     RETURN
500 -     300 IF (LFW1.NE.LFW2) IREAD=1
501 -     400 CONTINUE
502 -     RETURN
503 - C
504 - C
505 -     SUBROUTINE PRINTOUT
506 - C INTERNAL SUBROUTINE FOR PROGRAM BISUM01 TO PRINT BISPECTRAL
507 - C INTEGRALS
508 -     IF (KDISC.EQ.0) GO TO 70
509 -     IF (NSERIES.EQ.1) GO TO 50
510 -     WRITE (KDCB,5000) 2*(LPHALF-1)
511 -     5000 FORMAT (' CROSS SPECTRUM *** TOTAL NUMBER OF SUM ',
512 -     $'FREQ'S: ',I4,' *** EACH BINARY RECORD EQUALS ONE LINE '
513 -     $,'OF PRINTED OUTPUT')
514 -     GO TO 70
515 -     50 WRITE (KDCB,6000) LPHALF-1
516 -     6000 FORMAT (' AUTO SPECTRUM *** TOTAL NUMBER OF SUM ',
517 -     $'FREQ'S: ',I4,' *** EACH BINARY RECORD EQUALS ONE LINE '
518 -     $,'OF PRINTED OUTPUT')
519 -     70 CONTINUE
520 -     NLINES=0
521 -     WRITE (108,1000)
522 -     1000 FORMAT (1H1,T26,'I N T E G R A L S   O V E R   ',
523 -     $'P O S I T I V E   F R E Q U E N C I E S   A L O N E',/)
524 -     NLINES=NLINES+2
525 -     WRITE (108,1200)
526 -     1200 FORMAT (X,'SUM   NO IN',T127,'SUM',/,X,'FREQ   SCAN',
527 -     $T18,'BR',T26,'BR/N',T39,'BI',T48,'BI/N',T60,'BIC',T70,
528 -     $'BIC/N',T82,'BIC**2 (BIC**2)/N',T104,'S(W3)* (S(W3)*',
529 -     $T127,'FREQ',/, 'SEQ NO (N)',T106,
530 -     $'BIC**2 BIC**2)/N SEQ NO',/)
531 -     NLINES=NLINES+4
532 - C
533 - C
534 -     DO 100 I=2,LPHALF
535 -     NLINES=NLINES+1
536 -     IF (NLINES.GT.60) NLINES=6 ; WRITE (108,2000) ;
537 -     $WRITE (108,1200)
538 -     2000 FORMAT (1H1)

```

```

539 -      KP=I/2
540 -      WRITE (108,1500) I,KP,BRU(I),BRU(I)/KP,BIU(I),BIU(I)/KP,
541 -      $BICU(I),BICU(I)/KP,BIC2U(I),BIC2U(I)/KP,
542 -      $BCSU(I),BCSU(I)/KP,I
543 - 1500 FORMAT (X,I4,T9,I3,T14,3(G9.3,X,G9.3,3X),X,
544 -      $2(G9.3,X,G9.3,4X),I4)
545 -      IF (KDISC.NE.0)
546 -      $WRITE (KDCB)      I,KP,BRU(I),BRU(I)/KP,BIU(I),BIU(I)/KP,
547 -      $BICU(I),BICU(I)/KP,BIC2U(I),BIC2U(I)/KP,
548 -      $BCSU(I),BCSU(I)/KP,I
549 - 100 CONTINUE
550 - C
551 - C
552 -      IF (NSERIES.EQ.1) GO TO 300
553 -      NLINES=0
554 -      WRITE (108,4000)
555 - 4000 FORMAT(1H1,T27,'I N T E G R A L S   I N V O L V I N G   '
556 -      $,'N E G A T I V E   F R E Q U E N C I E S',/)
557 -      NLINES=NLINES+2
558 -      WRITE (108,1200)
559 -      NLINES=NLINES+4
560 - C
561 - C
562 -      DO 200 I=1,LPHALF-1
563 -      NLINES=NLINES+1
564 -      IF (NLINES.GT.60) NLINES=6 ; WRITE (108,2000) ;
565 -      $WRITE (108,1200)
566 -      KP=LPHALF-I
567 -      WRITE (108,1500) I,KP,BRL(I),BRL(I)/KP,BIL(I),BIL(I)/KP,
568 -      $BICL(I),BICL(I)/KP,BIC2L(I),BIC2L(I)/KP,
569 -      $BCSL(I),BCSL(I)/KP,I
570 -      IF (KDISC.NE.0)
571 -      $WRITE (KDCB)      I,KP,BRL(I),BRL(I)/KP,BIL(I),BIL(I)/KP,
572 -      $BICL(I),BICL(I)/KP,BIC2L(I),BIC2L(I)/KP,
573 -      $BCSL(I),BCSL(I)/KP,I
574 - 200 CONTINUE
575 - C
576 - C
577 - 300 CONTINUE
578 -      RETURN
579 -      END

```

- PROGRAM BIVEC -

```

1 - C                <<<<<< B I V E C O 5 >>>>>>
2 - C CALCULATES ROTARY AUTO-BISPECTRUM AND ROTARY CROSS-BISPECTRUM
3 - C FOR ONE OR TWO VECTOR SERIES AND DETERMINES CONFIDENCE LEVELS
4 - C FOR BICOHERENCES.
5 - C GENERATES TWO OR FOUR DISC FILES FOR TRANSMITTAL TO
6 - C PLOTTING PROGRAM.
7 - C PROGRAMMER:  GERARD H. MARTINEAU
8 - C ORIGINATOR:  MELBOURNE G. BRISCOE
9 - C DATE       :  OCT 17, 1977
10 - C*****
11 -     DIMENSION W11(2560),W12(2560),W21(2560),W22(2560),
12 -     $OM(2560,4),SQ1(35,35),SQ2(35,35),W31(80),W32(80),
13 -     $KBIN(200),ABIN(200),CONF(10),BCL(10),KARD(20),LW(6)
14 -     DIMENSION LABL(33),PHSERR(2)
15 -     EQUIVALENCE (W11,OM),(W12,OM(1,2)),(W21,OM(1,3)),
16 -     $(W22,OM(1,4)),(KBIN,ABIN),
17 -     $(LW(1),LW11),(LW(2),LW12),(LW(3),LW21),(LW(4),LW22),
18 -     $(LW(5),LW31),(LW(6),LW32)
19 -     DOUBLE PRECISION A11,B11,A12,B12,A21,B21,A22,B22,A31,B31,
20 -     $A32,B32,P1U,P2U,P3U,P1V,P2V,P3V,
21 -     $RNUMU,RNUMV,RDENU,RDENV,RBUR,RBUI,RBVR,RBVI,
22 -     $UIR,U1I,U2R,U2I,U3R,U3I,V1R,V1I,V2R,V2I,V3R,V3I,RNPCS,
23 -     $RMODU,RMODV,RBURT,RBUI,T,RBVRT,RBVIT,ANPCS
24 -     COMMON /PRIME/ NPRIME(65)
25 -     DATA NPRIME/
26 -     $ 16, 24, 32, 40, 48, 64, 72, 80, 96, 120,
27 -     $ 128, 144, 160, 200, 216, 240, 256, 288, 320, 360,
28 -     $ 384, 400, 432, 480, 512, 576, 600, 640, 648, 720,
29 -     $ 768, 800, 864, 960,1024,1080,1152,1200,1280,1296,
30 -     $1440,1536,1600,1728,1800,1920,1944,2000,2048,2160,
31 -     $2304,2400,2560,2592,2880,2916,3000,3072,3200,3240,
32 -     $3456,3600,3848,3888,4000/
33 -     NAMELIST IDATSTART,MF,ISTORE1,ISTORE2,BINSIZE,RBICLIM,
34 -     $IHI,KB1,KB2,KONF,LISTBI,IDUMP,LABREAD,IFOURIER,INORM,
35 -     $ISTORE3,ISTORE4,ISQOUT,IHANN,SIGLVL
36 -     ISQD=35 ; ISOM=2560
37 -     PI=3.14159265 ; TWOPI=2.*PI
38 -     DTR=PI/180. ; RTD=180./PI
39 -     ISQOUT=0 ; IHANN=1
40 -     SIGLVL=-1 ; PHSERR(2)=0
41 -     CALL ABORTSET(700S,1)
42 -     CALL BIVCTRL
43 -     IF ((LISTBI.NE.0).AND.(INORM.EQ.0)) WRITE (108,1321)
44 - 1321 FORMAT (/ ,T3,'F1',T9,'F2',T15,'F3',T20,'RBICOH RBIPHASE'
45 -     $,T39,'/ <RB>/',T51,'<REAL(RB)>',T64,'<IMAG(RB)>',T76,
46 -     $,'<P(W1)DW>',T88,'<P(W2)DW>',T100,'<P(W3)DW>',T111,
47 -     $,'<SQR(T PROD<P>)>',T125,'PHSERR',/)
48 -     IF ((LISTBI.NE.0).AND.(INORM.NE.0)) WRITE (108,1431)
49 - 1431 FORMAT (/ ,T3,'F1',T9,'F2',T15,'F3',T20,'RBICOH RBIPHASE'
50 -     $,T38,'<REAL(RB'')>',T52,'<IMAG(RB'')>',T65,'PHSERR',/)
51 -     NPCS=NPC2-NPC1+1
52 -     IF ((NPCS.EQ.1).AND.(IHANN.EQ.1)) IHANN=0 ; OUTPUT
53 -     $'IHANN HAS BEEN SET TO 0. CAN'T OVERLAP WITH ONE PIECE'
54 -     IF (IHANN.EQ.1) EDOF=36.*NPCS*NPCS/(19*NPCS-1)
55 -     IF (IHANN.EQ.0) EDOF=2*NPCS
56 -     IF (SIGLVL.LT.0) SIGLVL=SQR(6./EDOF)
57 -     LPATHE=LPIECE/8 ; LPQU=LPIECE/4 ; LPHALF=LPIECE/2
58 -     LP38=3*LPATHE ; IQND=2*NPIECES*LPATHE

```

```

59 -      IQU1=IDATSTART+NPIECES ; IQU2=IQU1+IQNO
60 -      IQU3=IQU1+2*IQNO ; IQU4=IQU1+3*IQNO
61 - C START OF LOOP FOR /W2/>/W1/
62 -      KPASS=1
63 -      50 CONTINUE
64 - C INITIALIZE RASTER AT ZERO
65 -      ISQ=0
66 - C
67 - C
68 - C STARTING POINT FOR RASTER LOOP:
69 -      100 CONTINUE
70 - C OUTPUT TO DISC
71 -      IF ((IDUMP.EQ.0).OR.(ISQ.EQ.0)) GO TO 102
72 -      KST1=ISTORE1 ; KST2=ISTORE2
73 -      IF (KPASS.EQ.2) KST1=ISTORE3 ; KST2=ISTORE4
74 -      WRITE (KST1)      ((SQ1(I,J),I=1,LPATHE),J=1,LPATHE)
75 -      WRITE (KST2)      ((SQ2(I,J),I=1,LPATHE),J=1,LPATHE)
76 -      102 CONTINUE
77 - C INCREMENT RASTER
78 -      ISQ=ISQ+1
79 -      IF (ISQOUT.NE.0) OUTPUT ISQ
80 - C IS RASTER LIMIT EXCEEDED?
81 -      IF (ISQ.GT.16) GO TO 300
82 - C SET STORAGE ARRAYS SQ1,SQ2 TO -999. IF DUMPING
83 -      IF (IDUMP.EQ.0) GO TO 110
84 - C
85 - C
86 -      DO 105 I=1,ISQD
87 - C
88 -      DO 105 J=1,ISQD
89 -      SQ1(I,J)=-999.
90 -      105 SQ2(I,J)=-999.
91 - C
92 - C
93 -      110 CONTINUE
94 -      CALL FREQBIAS12
95 -      IF ((ISQ.NE.11).AND.(ISQ.NE.16)) GO TO 125
96 - C
97 - C
98 -      DO 120 I=1,ISOM
99 -      W11(I)=0.
100 -      120 W12(I)=0.
101 - C
102 - C
103 -      125 IF ((ISQ.NE.16).OR.(ISW.NE.3)) GO TO 135
104 - C
105 - C
106 -      DO 130 I=1,ISOM
107 -      W21(I)=0.
108 -      130 W22(I)=0.
109 - C
110 - C
111 -      135 CONTINUE
112 -      IF (KPASS.EQ.1) GO TO 137
113 -      LWP=LW11 ; LWQ=LW12
114 -      LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ
115 -      137 CALL READW1W2
116 -      IF (KPASS.EQ.1) GO TO 138
117 -      LWP=LW11 ; LWQ=LW12
118 -      LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ

```

```

119 - X      WRITE (108,1717) ((OM(I,J),I=1,4),J=1,4)
120 -      138 CALL RASTER
121 - C      INITIALIZE SCAN (SUM) FREQUENCY
122 -      NCPT=IFIRST-1
123 - C
124 - C
125 - C      STARTING POINT FOR SCAN LOOP:
126 -      140 CONTINUE
127 - C      INCREMENT SCAN
128 -      NCPT=NCPT+1
129 - X      OUTPUT NCPT
130 - C      CHECK FOR END OF RASTER
131 -      IF (NCPT.GT.ILAST) GO TO 100
132 -      IF (KPASS.EQ.1) GO TO 145
133 -      LWP=LW11 ; LWQ=LW12
134 -      LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ
135 -      145 CALL READW3
136 -      IF (KPASS.EQ.1) GO TO 150
137 -      LWP=LW11 ; LWQ=LW12
138 -      LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ
139 - X      WRITE (108,7777) (W31(N),N=1,2),(W32(N),N=1,2)
140 - X7777 FORMAT (X,2(2F2.0,4X))
141 -      150 CALL FREQINIT
142 - C      INITIALIZE FREQUENCIES 1,2
143 -      IF1=IF1-1 ; IF2=IF2+1
144 - C
145 - C
146 - C
147 - C      STARTING POINT FOR FREQ. 1,2 LOOP (GRID POINT LOOP)
148 -      155 CONTINUE
149 - C      INCREMENT FREQUENCIES 1,2
150 -      IF1=IF1+1 ; IF2=IF2-1
151 -      IF2ABS=ABS(IF2)
152 - C      CHECK FOR END OF SCAN
153 -      CALL BOUNDARY
154 -      IF (IBDY.EQ.1) GO TO 140
155 - C      BEGIN CALCULATIONS
156 -      RBUR=RBUI=RBVR=RBVI=0.
157 -      P1U=P2U=P3U=P1V=P2V=P3V=0.
158 - C
159 - C
160 -      DO 200 I=NPC1,NPC2
161 -      IF3=IF1+IF2
162 -      IA1=(IF1-IFB1-1)*2*NPIECES+I
163 -      IB1=IA1+NPIECES
164 -      IA2=(IF2ABS-IFB2-1)*2*NPIECES+I
165 -      IB2=IA2+NPIECES
166 -      A11=W11(IA1) ; B11=W11(IB1)
167 -      A12=W12(IA1) ; B12=W12(IB1)
168 -      IF (ISW.EQ.3) GO TO 170
169 -      GO TO (160,160,170,170,160,160,170,170,170,170,170,170,
170 -      $170,170,160,160),ISQ
171 -      160 A21=W11(IA2) ; B21=W11(IB2)
172 -      A22=W12(IA2) ; B22=W12(IB2)
173 -      GO TO 180
174 -      170 A21=W21(IA2) ; B21=W21(IB2)
175 -      A22=W22(IA2) ; B22=W22(IB2)
176 -      180 A31=W31(I) ; B31=W31(NPIECES+I)
177 -      A32=W32(I) ; B32=W32(NPIECES+I)
178 - X      WRITE (108,1717) A11,A12,B11,B12,A21,A22,B21,B22,

```

```

179 - X      $A31,A32,B31,B32
180 - X1717 FORMAT (X,3(4F4.0,4X))
181 -      U1R=A11+B12 ; U1I=A12-B11
182 -      U3R=A31+B32 ; U3I=-A32+B31
183 -      V1R=A11-B12 ; V1I=A12+B11
184 -      V3R=A31-B32 ; V3I=-A32-B31
185 -      IF (IF2.LT.0) GO TO 185
186 -      U2R=A21+B22 ; U2I=A22-B21
187 -      V2R=A21-B22 ; V2I=A22+B21
188 -      P2U=P2U+U2R**2+U2I**2 ; P2V=P2V+V2R**2+V2I**2
189 -      GO TO 190
190 -      185 U2R=A21-B22 ; U2I=A22+B21
191 -      V2R=A21+B22 ; V2I=A22-B21
192 -      P2U=P2U+U2R**2+U2I**2 ; P2V=P2V+V2R**2+V2I**2
193 -      190 CONTINUE
194 -      IF (INORM.EQ.0) GO TO 195
195 -      RBURT=U1R*U2R*U3R-U1I*U2I*U3R-U1R*U2I*U3I-U1I*U2R*U3I
196 -      RBUIT=U1R*U2I*U3R+U1I*U2R*U3R+U1R*U2R*U3I-U1I*U2I*U3I
197 -      RBVRT=V1R*V2R*V3R-V1I*V2I*V3R-V1R*V2I*V3I-V1I*V2R*V3I
198 -      RBVIT=V1R*V2I*V3R+V1I*V2R*V3R+V1R*V2R*V3I-V1I*V2I*V3I
199 -      RMODU=DSQRT(RBURT**2+RBUIT**2)
200 -      RBUR=RBUR+RBURT/RMODU ; RBUI=RBUI+RBUIT/RMODU
201 -      RMODV=DSQRT(RBVRT**2+RBVIT**2)
202 -      RBVR=RBVR+RBVRT/RMODV ; RBVI=RBVI+RBVIT/RMODV
203 -      GO TO 200
204 -      195 CONTINUE
205 -      P1U=P1U+U1R**2+U1I**2 ; P3U=P3U+U3R**2+U3I**2
206 -      P1V=P1V+V1R**2+V1I**2 ; P3V=P3V+V3R**2+V3I**2
207 -      RBUR=RBUR+U1R*U2R*U3R-U1I*U2I*U3R-U1R*U2I*U3I-U1I*U2R*U3I
208 -      RBUI=RBUI+U1R*U2I*U3R+U1I*U2R*U3R+U1R*U2R*U3I-U1I*U2I*U3I
209 -      RBVR=RBVR+V1R*V2R*V3R-V1I*V2I*V3R-V1R*V2I*V3I-V1I*V2R*V3I
210 -      RBVI=RBVI+V1R*V2I*V3R+V1I*V2R*V3R+V1R*V2R*V3I-V1I*V2I*V3I
211 -      200 CONTINUE
212 - C
213 - C
214 -      IF (INORM.EQ.0) GO TO 202
215 -      RBICU=(DSQRT(RBUR**2+RBUI**2))/ANPCS
216 -      RBICV=(DSQRT(RBVR**2+RBVI**2))/ANPCS
217 -      GO TO 204
218 -      202 CONTINUE
219 -      RNUMU=DSQRT(RBUR**2+RBUI**2) ; RDENU=DSQRT(P1U*P2U*P3U)
220 -      RBICU=RNPCS*RNUMU/RDENU
221 -      RNUMV=DSQRT(RBVR**2+RBVI**2) ; RDENV=DSQRT(P1V*P2V*P3V)
222 -      RBICV=RNPCS*RNUMV/RDENV
223 -      204 CONTINUE
224 -      RBPHU=DATAN(RBUI, RBUR)
225 -      RBPHV=DATAN(RBVI, RBVR)
226 -      RBPHU=RTD*RBPHU ; RBPHV=RTD*RBPHV
227 - C SORT ROT BIC'S IF CALLED FOR
228 -      IF (KONF.NE.0) CALL CBIN(RBICU,1) ; CALL CBIN(RBICV,2)
229 - C PRINT ROT BIC'S WITHIN SPECIFIED LIMITS
230 -      IF (LISTBI.EQ.0) GO TO 219
231 -      JF1=IF1 ; JF2=IF2
232 -      IF (KPASS.EQ.2) JF1=IF2 ; JF2=IF1
233 -      IF (INORM.NE.0) GO TO 212
234 -      IF ((RBICU.LE.CONF1).OR.(RBICU.GT.CONF2)) GO TO 208
235 -      IF (IHANN.NE.1) GO TO 205
236 -      IF (RBICU.GE.SIGLVL) NPH=1 ;
237 -      $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICU) ; GO TO 206
238 -      NPH=0 ; GO TO 206

```

```

239 - 205 CONTINUE
240 -   IF ((IHANN.EQ.0).AND.(RBICU.GE.SIGLVL)) NPH=1 ;
241 -   $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 206
242 -   NPH=0
243 - 206 CONTINUE
244 -   WRITE (108,2000) JF1,JF2,IF3,RBICU,RBPHU,RNUMU/NPIECES,
245 -   $RBUR/NPIECES,RBUI/NPIECES,P1U/NPIECES,P2U/NPIECES,
246 -   $P3U/NPIECES,RDENU/NPIECES**1.5,NPH,(PHSERR(I),I=1,NPH)
247 - 2000 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,3(2X,E11.3)
248 -   $,3(1X,E11.3),2X,E11.3,3X,N(F5.1))
249 - 208 CONTINUE
250 -   IF ((RBICV.LE.CONF1).OR.(RBICV.GT.CONF2)) GO TO 219
251 -   IF (IHANN.NE.1) GO TO 209
252 -   IF (RBICV.GE.SIGLVL) NPH=1 ;
253 -   $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICV) ; GO TO 210
254 -   NPH=0 ; GO TO 210
255 - 209 CONTINUE
256 -   IF ((IHANN.EQ.0).AND.(RBICV.GE.SIGLVL)) NPH=1 ;
257 -   $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 210
258 -   NPH=0
259 - 210 CONTINUE
260 -   WRITE (108,2000) -JF1,-JF2,-IF3,RBICV,RBPHV,RNUMV/NPIECES
261 -   $,RBVR/NPIECES,RBVI/NPIECES,P1V/NPIECES,P2V/NPIECES,
262 -   $P3V/NPIECES,RDENV/NPIECES**1.5,NPH,(PHSERR(I),I=1,NPH)
263 -   GO TO 219
264 - 212 CONTINUE
265 -   IF ((RBICU.LE.CONF1).OR.(RBICU.GT.CONF2)) GO TO 216
266 -   IF (IHANN.NE.1) GO TO 213
267 -   IF (RBICU.GE.SIGLVL) NPH=1 ;
268 -   $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICU) ; GO TO 214
269 -   NPH=0 ; GO TO 214
270 - 213 CONTINUE
271 -   IF ((IHANN.EQ.0).AND.(RBICU.GE.SIGLVL)) NPH=1 ;
272 -   $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 214
273 -   NPH=0.
274 - 214 CONTINUE
275 -   WRITE (108,2100) JF1,JF2,IF3,RBICU,RBPHU,RBUR/NPIECES,
276 -   $RBUI/NPIECES,NPH,(PHSERR(I),I=1,NPH)
277 - 2100 FORMAT(X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
278 -   $2(3X,E11.3),3X,N(F5.1))
279 - 216 CONTINUE
280 -   IF ((RBICV.LE.CONF1).OR.(RBICV.GT.CONF2)) GO TO 219
281 -   IF (IHANN.NE.1) GO TO 217
282 -   IF (RBICV.GE.SIGLVL) NPH=1 ;
283 -   $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICV) ; GO TO 218
284 -   NPH=0 ; GO TO 218
285 - 217 CONTINUE
286 -   IF ((IHANN.EQ.0).AND.(RBICV.GE.SIGLVL)) NPH=1 ;
287 -   $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 218
288 -   NPH=0
289 - 218 CONTINUE
290 -   WRITE (108,2200) -JF1,-JF2,-IF3,RBICV,RBPHV,RBVR/NPIECES,
291 -   $RBVI/NPIECES,NPH,(PHSERR(I),I=1,NPH)
292 - 2200 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
293 -   $2(3X,E11.3),3X,N(F5.1))
294 - 219 CONTINUE
295 - C COMMIT RBICU TO SQ1 AND RBICV TO SQ2
296 -   IF (IDUMP.EQ.0) GO TO 240
297 -   NCI=IF1-IFB1
298 -   IF (IF2.LT.0) NR1=IF2+1+LPATHE+IFB2 ; GO TO 220

```

```

299 -      NR1=IF2-IFB2
300 -      220 SQ1(NR1,NC1)=RBICU
301 -      NC2=IFB1+LPATHE-IF1+1
302 -      IF (IF2.LT.0) NR2=-IFB2-IF2 ; GO TO 230
303 -      NR2=IFB2+LPATHE-IF2+1
304 -      230 SQ2(NR2,NC2)=RBICV
305 -      240 CONTINUE
306 - C
307 - C END OF FREQ. 1,2 LOOP (GRID POINT LOOP)
308 -      GO TO 155
309 - C
310 - C
311 -      300 CONTINUE
312 - C OUTPUT APPROPRIATE LABELS TO DISC
313 -      IF (IDUMP.EQ.0) GO TO 400
314 -      IF (LABREAD.EQ.0) GO TO 360
315 - C
316 -      DO 350 I=1,6
317 -      350 CALL RDISC(LW(I),1,LABL(4*I-3),4)
318 - C
319 -      360 CONTINUE
320 - C
321 - C
322 -      KST1=ISTORE1 ; KST2=ISTORE2
323 -      IF (KPASS.EQ.2) KST1=ISTORE3 ; KST2=ISTORE4
324 -      DO 370 I=1,6
325 -      WRITE (KST1)      (LABL(K),K=4*I-3,4*I)
326 -      WRITE (KST2)      (LABL(K),K=4*I-3,4*I)
327 -      370 CONTINUE
328 - C
329 - C
330 -      WRITE (KST1)      MF
331 -      WRITE (KST2)      MF
332 - C
333 -      DO 380 I=1,33
334 -      380 LABL(I)=4H
335 -      IF ((MF.NE.21).OR.(IFOURIER.EQ.0)) GO TO 390
336 - C
337 -      CALL RDISC(21,1,LABL,33)
338 -      390 CONTINUE
339 -      WRITE (KST1)      LABL
340 -      WRITE (KST2)      LABL
341 -      400 CONTINUE
342 - C END OF LOOP FOR /W2/ > /W1/
343 -      IF ((ISW.EQ.3).AND.(KPASS.EQ.1)) KPASS=2 ; GO TO 50
344 - C DETERMINE CONFIDENCE LEVELS IF CALLED FOR
345 -      IF (KONF.NE.0) CALL CONFIDENCE
346 - C
347 -      GO TO 800
348 -      700 STOP 'ABORTSET TERMINATION'
349 - C
350 -      800 STOP 'NORMAL PROGRAM COMPLETION'
351 - C*****
352 - C*****
353 -      SUBROUTINE BIVCTRL
354 - C INTERNAL SUBROUTINE FOR PROGRAM BIVECO1 TO INITIALIZE,
355 - C AND READ AND CHECK INPUT PARAMETERS.
356 -      IDUMP=1
357 -      MF=21 ; IDATSTART=5
358 -      ISTORE1=1 ; ISTORE2=2 ; ISTORE3=3 ; ISTORE4=4

```

```

359 -      IHI=1
360 -      LABREAD=1 ; IFOURIER=1
361 -      BINSIZE=0.01 ; RBICLIM=1.5 ; KB1=KB2=1 ; KONF=0
362 -      KOVER=0 ; LISTBI=0
363 -      MTOT=0 ; LIMPCS=150
364 -      INORM=0
365 - C
366 - C
367 -      DO 20 I=1,200
368 - 20 KBIN(I)=0
369 - C
370 - C
371 -      DO 40 I=1,33
372 - 40 LABL(I)=4H
373 - C
374 - C
375 -      INPUT
376 -      OUTPUT IDATSTART,MF,ISTORE1,ISTORE2,ISTORE3,ISTORE4,
377 - $BINSIZE,RBICLIM,IHI,KB1,KB2,KONF,LISTBI,IDUMP,LABREAD,
378 - $IFOURIER,INORM,IHANN
379 -      CALL DPAR(NERR,MF,1,1,KSUM,512)
380 -      READ (105,1000) KARD
381 - 1000 FORMAT (20A4)
382 -      WRITE (108,1100) KARD
383 - 1100 FORMAT (X,20A4)
384 -      DECODE (80,1200,KARD) LW11,LW12,LW21,LW22,LW31,LW32,
385 - $NPIECES,LPIECE,NPC1,NPC2,CONF1,CONF2
386 - 1200 FORMAT (12G)
387 -      IF (NPC2.GT.NPIECES) OUTPUT NPC2,NPIECES,
388 - $'LAST PIECE TO AVG GREATER THAN NO OF PCS' ; STOP
389 -      ANPCS=DFLOAT(NPIECES)
390 -      RNPCS=DSQRT(ANPCS)
391 -      IF ((LW11.NE.LW21).OR.(LW12.NE.LW22)) GO TO 30
392 -      IF ((LW31.EQ.LW11).AND.(LW32.EQ.LW12)) ISW=1 ; GO TO 50
393 -      ISW=2 ; GO TO 50
394 - 30 CONTINUE
395 -      IF (((LW31.EQ.LW11).AND.(LW32.EQ.LW12)).OR.
396 - $((LW31.EQ.LW21).AND.(LW32.EQ.LW22))) ISW=3 ; GO TO 50
397 -      OUTPUT LW11,LW12,LW21,LW22,LW31,LW32,
398 - $'NO MORE THAN TWO DIFFERENT VECTOR SERIES ALLOWED' ; STOP
399 - 50 CONTINUE
400 - C CHECK FOR EXISTENCE OF DCB ASSIGNMENTS FOR OUTPUT FILES
401 -      IF (IDUMP.EQ.0) GO TO 80
402 -      N1=ISTORE1 ; N2=ISTORE2 ; N3=ISTORE3 ; N4=ISTORE4
403 -      IF (ISW.EQ.3) GO TO 60
404 -      IF (N1.NE.N2) GO TO 70
405 -      OUTPUT ISTORE1,ISTORE2 ; GO TO 65
406 - 60 CONTINUE
407 -      IF ((N1.NE.N2).AND.(N1.NE.N3).AND.(N1.NE.N4).AND.
408 - $(N2.NE.N3).AND.(N2.NE.N4).AND.(N3.NE.N4)) GO TO 70
409 -      OUTPUT ISTORE1,ISTORE2,ISTORE3,ISTORE4
410 - 65 OUTPUT 'TWO OUTPUT DCB'S HAVE BEEN SET EQUAL CAN'T DO.'
411 - $ ; STOP
412 - 70 CONTINUE
413 -      CALL GETDCB(ISTORE1,LOCC)
414 -      CALL GETDCB(ISTORE2,LOCC)
415 -      IF (ISW.NE.3) GO TO 80
416 -      CALL GETDCB(ISTORE3,LOCC)
417 -      CALL GETDCB(ISTORE4,LOCC)
418 - 80 CONTINUE

```

```

419 - C
420 - C
421 -     DO 100 I=1,65
422 -     IF (LPIECE.EQ.NPRIME(I)) GO TO 150
423 - 100 CONTINUE
424 - C
425 - C
426 -     WRITE (108,1250) NPRIME ; STOP
427 - 1250 FORMAT ('PIECE LENGTH MUST BE FROM FOLLOWING LIST:',/,
428 -     $(10(X,I7)))
429 - 150 CONTINUE
430 -     IF (NPIECES.GT.LIMPCS) WRITE (108,1275) ; STOP
431 - 1275 FORMAT ('NUMBER OF PIECES MUST NOT EXCEED 150')
432 -     IF (KONF.EQ.0) GO TO 300
433 -     READ (105,1000) KARD
434 -     WRITE (108,1100) KARD
435 -     DECODE (80,3000,KARD) NCONF,NCONF,(CONF(I),I=1,NCONF)
436 - 3000 FORMAT (G,NG)
437 -     NBINS=INT(RBICLIM/BINSIZE+0.5)
438 -     IF (NBINS.GT.200) WRITE (108,3050) ; STOP
439 - 3050 FORMAT ('NUMBER OF BINS MUST NOT EXCEED 200')
440 -     RBICLIM=NBINS*BINSIZE
441 -     WRITE (108,4000) RBICLIM
442 - 4000 FORMAT ('ADJUSTED ROT BICOH LIMIT, HOLDING BINSIZE ',
443 -     $'AS INPUT ',F6.2)
444 - 300 CONTINUE
445 -     RETURN
446 - C*****
447 - C*****
448 -     SUBROUTINE FREQBIAS12
449 - C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01 TO
450 - C RETURN FREQUENCY BIASES FB1,FB2
451 -     GO TO (10,10,30,30,50,50,70,70,90,90,110,110,130,140,
452 -     $150,160),ISQ
453 -     10 IFB1=IFB2=0. ; RETURN
454 -     30 IFB1=LPATHE ; IFB2=0. ; RETURN
455 -     50 IFB1=LPATHE ; IFB2=LPATHE ; RETURN
456 -     70 IFB1=LPQU ; IFB2=LPATHE ; RETURN
457 -     90 IFB1=LPQU ; IFB2=0. ; RETURN
458 -     110 IFB1=LP38 ; IFB2=0. ; RETURN
459 -     130 IFB1=LP38 ; IFB2=LPATHE ; RETURN
460 -     140 IFB1=LP38 ; IFB2=LPQU ; RETURN
461 -     150 IFB1=LPQU ; IFB2=LPQU ; RETURN
462 -     160 IFB1=LP38 ; IFB2=LP38 ; RETURN
463 - C*****
464 - C*****
465 -     SUBROUTINE READW1W2
466 - C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01
467 - C TO FILL BUFFERS FOR FREQUENCIES 1,2
468 -     GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
469 -     $150,160),ISQ
470 - C
471 -     10 DO 13 I=1,2
472 -     13 CALL RDISC(LW(I),IQU1,OM(1,I),IQNO)
473 - C
474 -     DO 18 I=1,2
475 -     GO TO (14,15,16),ISW
476 -     14 CALL RDISC(LW(I),IQU2,OM(1,I+2),IQNO) ; GO TO 18
477 -     15 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO) ; GO TO 18
478 -     16 CALL RDISC(LW(I+2),IQU1,OM(1,I+2),IQNO)

```

```

479 -      18 CONTINUE
480 - C
481 -      20 CONTINUE
482 -          IF (ISW.NE.2) GO TO 26
483 - C
484 -          DO 24 I=1,2
485 -      24 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO)
486 - C
487 -      26 RETURN
488 - C
489 -      30 DO 35 I=1,2
490 -          CALL RDISC(LW(I),IQU2,OM(1,I),IQNO)
491 -          IF (ISW.NE.3) CALL RDISC(LW(I+2),IQU1,OM(1,I+2),IQNO)
492 -      35 CONTINUE
493 - C
494 -      40 RETURN
495 - C
496 -      50 DO 58 I=1,2
497 -          GO TO (54,55,56),ISW
498 -      54 CALL RDISC(LW(I),IQU3,OM(1,I+2),IQNO) ; GO TO 58
499 -      55 CALL RDISC(LW(I+4),IQU3,OM(1,I+2),IQNO) ; GO TO 58
500 -      56 CALL RDISC(LW(I+2),IQU2,OM(1,I+2),IQNO)
501 -      58 CONTINUE
502 - C
503 -          RETURN
504 -      60 IF (ISW.EQ.3) RETURN
505 - C
506 -          DO 68 I=1,2
507 -          GO TO (64,65),ISW
508 -      64 CALL RDISC(LW(I),IQU1,OM(1,I+2),IQNO) ; GO TO 68
509 -      65 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO)
510 -      68 CONTINUE
511 - C
512 -          RETURN
513 - C
514 -      70 DO 73 I=1,2
515 -      73 CALL RDISC(LW(I),IQU3,OM(1,I),IQNO)
516 - C
517 -          IF (ISW.EQ.3) RETURN
518 - C
519 -          DO 78 I=1,2
520 -          CALL RDISC(LW(I+2),IQU2,OM(1,I+2),IQNO)
521 -      78 CONTINUE
522 - C
523 -      80 RETURN
524 - C
525 -      90 DO 95 I=3,4
526 -      95 CALL RDISC(LW(I),IQU1,OM(1,I),IQNO)
527 - C
528 -      100 RETURN
529 - C
530 -      110 DO 115 I=1,2
531 -      115 CALL RDISC(LW(I),IQU4,OM(1,I),IQNO-NPIECES)
532 - C
533 -      120 RETURN
534 - C
535 -      130 DO 135 I=3,4
536 -      135 CALL RDISC(LW(I),IQU2,OM(1,I),IQNO)
537 - C
538 -          RETURN

```

```

539 - C
540 - 140 DO 145 I=3,4
541 - 145 CALL RDISC(LW(I),IQU3,OM(1,I),IQNO)
542 - C
543 - RETURN
544 - C
545 - 150 DO 153 I=1,2
546 - 153 CALL RDISC(LW(I),IQU3,OM(1,I),IQNO)
547 - C
548 - IF (ISW.EQ.3) RETURN
549 - C
550 - DO 158 I=1,2
551 - GO TO (154,155),ISW
552 - 154 CALL RDISC(LW(I),IQU1,OM(1,I+2),IQNO) ; GO TO 158
553 - 155 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO)
554 - 158 CONTINUE
555 - C
556 - RETURN
557 - C
558 - 160 DO 163 I=1,2
559 - 163 CALL RDISC(LW(I),IQU4,OM(1,I),IQNO-NPIECES)
560 - C
561 - IF (ISW.NE.3) RETURN
562 - C
563 - DO 168 I=3,4
564 - 168 CALL RDISC(LW(I),IQU4,OM(1,I),IQNO-NPIECES)
565 - C
566 - RETURN
567 - C*****
568 - C*****
569 - SUBROUTINE RASTER
570 - C INTERNAL SUBROUTINE FOR PROGRAM BIVECO1 TO
571 - C DETERMINE 1ST, LAST SUM FREQ'S FOR THIS RASTER
572 - GO TO (10,20,30,40,50,20,30,80,90,100,110,120,100,30,
573 - $20,20),ISQ
574 - 10 IF (KPASS.EQ.1) GO TO 15
575 - IFIRST=3 ; ILAST=LPQU-1 ; RETURN
576 - 15 IFIRST=2 ; ILAST=LPQU ; RETURN
577 - 20 IFIRST=1 ; ILAST=LPATHE-1 ; RETURN
578 - 30 IFIRST=1 ; ILAST=LPQU-1 ; RETURN
579 - 40 IFIRST=LPATHE+2 ; ILAST=LP38 ; RETURN
580 - 50 IF (KPASS.EQ.1) GO TO 55
581 - IFIRST=LPQU+3 ; ILAST=LPHALF-1 ; RETURN
582 - 55 IFIRST=LPQU+2 ; ILAST=LPHALF ; RETURN
583 - 80 IFIRST=LP38+2 ; ILAST=LPHALF ; RETURN
584 - 90 IFIRST=LPQU+2 ; ILAST=LPHALF ; RETURN
585 - 100 IFIRST=LPATHE+1 ; ILAST=LP38-1 ; RETURN
586 - 110 IFIRST=LPQU+1 ; ILAST=LPHALF-1 ; RETURN
587 - 120 IFIRST=LP38+2 ; ILAST=LPHALF ; RETURN
588 - C*****
589 - C*****
590 - SUBROUTINE READW3
591 - C INTERNAL SUBROUTINE FOR PROGRAM BIVECO1 TO FILL
592 - C BUFFERS W31,W32 FROM CORE OR DISC APPROPRIATELY
593 - NRD=2*NPIECES
594 - GO TO (10,20,30,40,50,60,70,500,90,100,110,120,130,500,
595 - $60,60),ISQ
596 - 10 IF (NCPT.LE.LPATHE) GO TO 20
597 - IF (ISW.EQ.1) KOL=3 ; MA=2 ; GO TO 300
598 - IF ((ISW.NE.2).OR.(NCPT.NE.LPATHE+1)) GO TO 15

```

```

599 - C
600 -      DO 12 I=1,2
601 - 12 CALL RDISC(LW(I+4),IQU2,OM(1,I+2),IQNO)
602 - C
603 - 15 IF (ISW.NE.2) GO TO 18
604 -      KOL=3 ; MA=2 ; GO TO 300
605 - 18 GO TO 500
606 - 20 CONTINUE
607 -      IF (ISW.EQ.1) KOL=1 ; MA=1 ; GO TO 300
608 -      IF ((ISW.EQ.2).OR.(LW31.EQ.LW21)) KOL=3 ; MA=1 ; GO TO 300
609 -      KOL=1 ; MA=1 ; GO TO 300
610 - 30 IF (ISW.EQ.2) GO TO 500
611 -      IF (NCPT.GT.LPATHE) GO TO 35
612 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW21)) KOL=3 ; MA=1 ; GO TO 300
613 -      GO TO 500
614 - 35 CONTINUE
615 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=2 ; GO TO 300
616 -      GO TO 500
617 - 40 IF ((ISW.EQ.2).OR.(NCPT.GT.LPQU)) GO TO 500
618 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=2 ; GO TO 300
619 -      GO TO 500
620 - 50 IF (ISW.EQ.3) GO TO 500
621 -      IF (NCPT.NE.LP38+1) GO TO 58
622 - C
623 -      DO 51 I=1,ISOM
624 -      W21(I)=0.
625 - 51 W22(I)=0.
626 - C
627 - C
628 -      DO 56 I=1,2
629 -      GO TO (52,54),ISW
630 - 52 CALL RDISC(LW(I),IQU4,OM(1,I+2),IQNO-NPIECES) ; GO TO 56
631 - 54 CALL RDISC(LW(I+4),IQU4,OM(1,I+2),IQNO-NPIECES)
632 - 56 CONTINUE
633 - C
634 - 58 CONTINUE
635 -      IF (NCPT.LE.LP38) KOL=3 ; MA=3 ; GO TO 300
636 -      KOL=3 ; MA=4 ; GO TO 300
637 - 60 IF (ISW.EQ.3) GO TO 500
638 -      KOL=3 ; MA=1 ; GO TO 300
639 - 70 IF ((ISW.EQ.2).OR.(NCPT.LE.LPATHE)) GO TO 500
640 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW21)) KOL=3 ; MA=2 ; GO TO 300
641 -      GO TO 500
642 - 90 IF ((ISW.EQ.2).OR.(NCPT.GT.LP38)) GO TO 500
643 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=3 ; GO TO 300
644 -      GO TO 500
645 - 100 IF ((ISW.EQ.2).OR.(NCPT.LE.LPQU)) GO TO 500
646 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=3 ; GO TO 300
647 -      GO TO 500
648 - 110 IF ((ISW.EQ.2).OR.(NCPT.LE.LP38)) GO TO 500
649 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=4 ; GO TO 300
650 -      GO TO 500
651 - 120 CONTINUE
652 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=4 ; GO TO 300
653 -      GO TO 500
654 - 130 IF ((ISW.EQ.2).OR.(NCPT.GT.LPQU)) GO TO 500
655 -      IF ((ISW.EQ.1).OR.(LW31.EQ.LW21)) KOL=3 ; MA=2 ; GO TO 300
656 -      GO TO 500
657 - 300 LOC=1+NRD*(NCPT-(MA-1)*LPATHE-1)
658 - C

```

```

659 - C
660 -      DO 350 I=1,NRD
661 -          W31(I)=OM(LOC+I-1,KOL)
662 -      350 W32(I)=OM(LOC+I-1,KOL+1)
663 - C
664 - C
665 -      RETURN
666 -      500 KREL=IDATSTART+NPIECES*(2*NCPT-1)
667 -          CALL RDISC(LW31,KREL,W31,NRD)
668 -          CALL RDISC(LW32,KREL,W32,NRD)
669 -      RETURN
670 - C*****
671 - C*****
672 -      SUBROUTINE FREQINIT
673 - C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01 TO INITIALIZE
674 - C FREQUENCIES IF1,IF2 FOR A GIVEN SCAN.
675 -      GO TO (10,20,30,40,10,60,70,80,90,100,110,120,130,140,
676 -          $150,160),ISQ
677 -      10 IF (KPASS.EQ.1) GO TO 15
678 -          IF1=NCPT/2+1 ; IF2=NCPT/2
679 -          IF (MOD(NCPT,2).EQ.0) IF2=NCPT/2-1
680 -      RETURN
681 -      15 IF1=NCPT/2 ; IF2=NCPT/2
682 -          IF (MOD(NCPT,2).EQ.1) IF1=IF1+1
683 -      RETURN
684 -      20 IF2=-1
685 -      25 IF1=-IF2+NCPT ; RETURN
686 -      30 IF1C=LPATHE+1 ; IF2C=-1 ; IDIAG=LPATHE
687 -      35 CONTINUE
688 -          IF (NCPT.LE.IDIAG) IF1=IF1C ; IF2=-IF1+NCPT ; RETURN
689 -          IF1=-IF2C+NCPT ; IF2=IF2C ; RETURN
690 -      40 IF1C=LPATHE+1 ; IF2C=LPATHE ; IDIAG=LPQU+1 ; GO TO 35
691 -      60 IF2=-LPATHE-1 ; GO TO 25
692 -      70 IF1C=LPQU+1 ; IF2C=-LPATHE-1 ; IDIAG=LPATHE ; GO TO 35
693 -      80 IF1=LPQU+1
694 -      85 IF2=-IF1+NCPT ; RETURN
695 -      90 IF1C=LPQU+1 ; IF2C=LPATHE ; IDIAG=LP38+1 ; GO TO 35
696 -      100 IF1C=LPQU+1 ; IF2C=-1 ; IDIAG=LPQU ; GO TO 35
697 -      110 IF1C=LP38+1 ; IF2C=-1 ; IDIAG=LP38 ; GO TO 35
698 -      120 IF1=LP38+1 ; GO TO 85
699 -      130 IF1C=LP38+1 ; IF2C=-LPATHE-1 ; IDIAG=LPQU ; GO TO 35
700 -      140 IF1C=LP38+1 ; IF2C=-LPQU-1 ; IDIAG=LPATHE ; GO TO 35
701 -      150 IF2=-LPQU-1 ; GO TO 25
702 -      160 IF2=-LP38-1 ; GO TO 25
703 - C*****
704 - C*****
705 -      SUBROUTINE BOUNDARY
706 - C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01
707 - C TO TEST FOR END OF SCAN
708 -      IBDY=0
709 -      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
710 -          $150,160),ISQ
711 -      10 IDIAG=LPATHE ; IM1=LPATHE ; IM2=1 ; GO TO 500
712 -      20 IM1=LPATHE ; GO TO 300
713 -      30 IDIAG=LPATHE ; IM1=LPQU ; IM2=-LPATHE ; GO TO 500
714 -      40 IDIAG=LPQU ; IM1=LPQU ; IM2=1 ; GO TO 500
715 -      50 IDIAG=LP38 ; IM1=LPQU ; IM2=LPATHE+1 ; GO TO 500
716 -      60 IM1=LPQU ; GO TO 300
717 -      70 IDIAG=LPATHE ; IM1=LP38 ; IM2=-LPQU ; GO TO 500
718 -      80 IDIAG=LPHALF ; IM2=LPATHE+1 ; GO TO 500

```

```

719 -      90 IDIAG=LP38 ; IM1=LP38 ; IM2=1 ; GO TO 500
720 -     100 IDIAG=LPQU ; IM1=LP38 ; IM2=-LPATHE ; GO TO 500
721 -     110 IDIAG=LP38 ; IM1=LPHALF ; IM2=-LPATHE ; GO TO 500
722 -     120 IDIAG=LPHALF ; IM2=1 ; GO TO 500
723 -     130 IDIAG=LPQU ; IM1=LPHALF ; IM2=-LPQU ; GO TO 500
724 -     140 IDIAG=LPATHE ; IM1=LPHALF ; IM2=-LP38 ; GO TO 500
725 -     150 IM1=LP38 ; GO TO 300
726 -     160 IM1=LPHALF ; GO TO 300
727 - C
728 -     300 IF (IF1.GT.IM1) IBDY=1
729 -         RETURN
730 -     500 IF (NCPT.GT.IDIAG) GO TO 550
731 -         IF (IF2.LT.IM2) IBDY=1
732 -         RETURN
733 -     550 IF (IF1.GT.IM1) IBDY=1
734 -         RETURN
735 - C*****
736 - C*****
737 -     SUBROUTINE CBIN(BICOH,IPASS)
738 - C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO DETERMINE
739 - C CONFIDENCE LIMITS FOR BICOHERENCES
740 - C HAS ENTRY POINTS CBIN TO PLACE GIVEN BICOHERENCE IN
741 - C BIN AND CONFIDENCE TO INITIATE CONFIDENCE LIMIT CALCULATION
742 -     IF (BICOH.LE.RBICLIM) GO TO 100
743 -     KOVER=KOVER+1
744 -     IF (IPASS.EQ.1) WRITE (108,1000) IF1,IF2,IF3,BICOH
745 -     IF (IPASS.EQ.2) WRITE (108,1000) -IF1,-IF2,-IF3,BICOH
746 - 1000 FORMAT('ROT BICOH LIM EXCEEDED AT FREQ TRIPLET ',3(I3,X),
747 - $' WITH VALUE ',G11.5)
748 -     IF (IHI.NE.0) MTOT=MTOT+1
749 -     RETURN
750 - C
751 -     100 CONTINUE
752 -     NIB=INT(BICOH/BINSIZE)+1
753 -     IF (NIB.GT.NBINS) GO TO 250
754 -     KBIN(NIB)=KBIN(NIB)+1
755 -     MTOT=MTOT+1
756 -     RETURN
757 - C
758 -     250 CONTINUE
759 -     IF (IPASS.EQ.1) WRITE (108,1025) BICOH,IF1,IF2,IF3
760 -     IF (IPASS.EQ.2) WRITE (108,1025) BICOH,-IF1,-IF2,-IF3
761 - 1025 FORMAT ('ROT BICOH OF ',F6.2,' AT FREQ TRIPLET ',3(I3,X),
762 - $' MET NEITHER BIN NOR LIM CRITERIA')
763 -     MTOT=MTOT+1
764 -     RETURN
765 - C
766 - C*****
767 -     ENTRY CONFIDENCE
768 - C*****
769 - C OPTIONAL DISPLAY OF PARTITIONING OF BICOHERENCES
770 -     IF (KB1.EQ.0) GO TO 320
771 -     WRITE (108,1050)
772 - 1050 FORMAT (/ ,T2,'BIN NO',T14,'BIN LIMITS',T34,
773 - $'NO OF ROT BIC'S',/)
774 - C
775 - C
776 -     DO 300 I=1,NBINS
777 -     WRITE (108,1100) I,(I-1)*BINSIZE,I*BINSIZE,KBIN(I)
778 - 1100 FORMAT (T3,I3,T13,2(F4.2,4X),T34,I5)

```

```

779 - 300 CONTINUE
780 - C
781 - C
782 - 320 CONTINUE
783 - C CONVERT NO OF ROT BIC'S IN KBIN TO TOTAL FRACTION
784 - C AT INDEX VALUE
785 -     KSUBTOT=0 ; I=0
786 -     WRITE (108,1125) MTOT
787 - 1125 FORMAT (/, 'TOTAL NO OF ROT BIC'S SUBMITTED: ', I5, /)
788 -     FTOT=FLOAT(MTOT)
789 - C
790 - 340 CONTINUE
791 -     I=I+1
792 -     IF (I.GT.NBINS) GO TO 350
793 - C
794 -     KSUBTOT=KSUBTOT+KBIN(I)
795 -     ABIN(I)=FLOAT(KSUBTOT)
796 -     ABIN(I)=ABIN(I)/FTOT
797 -     GO TO 340
798 - C
799 - 350 CONTINUE
800 - C OPTIONAL DISPLAY OF PARTIAL SUMS AT INDEX VALUE
801 -     IF (KB2.EQ.0) GO TO 400
802 -     WRITE (108,1150)
803 - 1150 FORMAT (/, T32, 'FRACT BELOW', /, T2, 'BIN NO', T14,
804 -     '$BIN LIMITS', T31, 'UPPER BIN LIM', /)
805 - C
806 - C
807 -     DO 380 I=1, NBINS
808 -     WRITE (108,1200) I, (I-1)*BINSIZE, I*BINSIZE, ABIN(I)
809 - 1200 FORMAT (T3, I3, T13, 2(F4.2, 4X), T34, F5.3)
810 - 380 CONTINUE
811 - C
812 - C
813 - 400 CONTINUE
814 - C
815 - C
816 - C COMPUTE CONFIDENCE LIMITS
817 -     I=0
818 - C
819 - 420 CONTINUE
820 -     I=I+1
821 -     IF (I.GT.NCONF) GO TO 600
822 -     J=0
823 - C
824 - 450 CONTINUE
825 -     J=J+1
826 -     IF (J.GT.NBINS) WRITE (108,1250) CONF(I) ; BCL(I)=999. ;
827 -     $GO TO 420
828 - 1250 FORMAT ('BINS EXHAUSTED FOR CONFIDENCE LEVEL', F6.3)
829 -     IF (ABIN(J).GE.CONF(I)) GO TO 470
830 -     GO TO 450
831 - C
832 - 470 CONTINUE
833 - C
834 -     JH=J
835 -     JL=J-1
836 -     Y1=JL*BINSIZE
837 -     Y2=JH*BINSIZE
838 -     X1=0

```

```
839 -      IF (JL.NE.0) X1=ABIN(JL)
840 -      X2=ABIN(JH)
841 -      X0=CONF(I)
842 -      BCL(I)=Y1+(X0-X1)*(Y2-Y1)/(X2-X1)
843 -      GO TO 420
844 - 600 CONTINUE
845 - C
846 - C PRINT CONFIDENCE LEVELS
847 -      WRITE (108,1400) NCONF, (CONF(I)*100.,BCL(I),I=1,NCONF)
848 - 1400 FORMAT (/ ,T4,'CONFIDENCE',/,T6,'LEVEL',T21,'ROTARY ',
849 - $ 'BICOHERENCE',/,T5,'(PERCENT)',/,N(T7,F4.1,T24,F5.3,/),/)
850 -      WRITE (108,1450) RBICLIM,KOVER
851 - 1450 FORMAT ('NO. OF ROT BIC'S    GREATER THAN SPECIFIED ',
852 - $ 'MAXIMUM OF ',F5.2,' IS ',I5,/)
853 -      RETURN
854 -      END
```

- PROGRAM BPLOT -

```

1 - C          ***** B P L O T *****
2 - C PROGRAM TO PLOT FILE OF BICOHERENCES WRITTEN BY
3 - C PROGRAM BISCAL
4 - C PROGRAMMER: GERARD H. MARTINEAU
5 - C ORIGINATOR: MELBOURNE G. BRISCOE
6 - C          DATE: JULY, 1977
7 -          DIMENSION DATA(200,130),IBUF(1000),CONLV(15),
8 -          $KARD(20),IDENT1(9),IDENT2(9),ICLK(4),LABL(33),CONFLV(15),
9 -          $INAMES(9),IVAR(3)
10 -          MD1=200 ; MD2=130
11 -          NAMELIST ISTORE,NDECPL,CMPPT,IHISTORY,WIDTH
12 -          ISTORE=NDECPL=1 ; CMPPT=-1. ; WIDTH=8.75
13 -          IHISTORY=1
14 -          INPUT
15 -          READ (105,1111) KARD
16 - 1111 FORMAT (20A4)
17 -          WRITE (108,1122) KARD
18 - 1122 FORMAT (X,20A4)
19 -          DECODE (80,3333,KARD) NSERIES,LPIECE
20 - 3333 FORMAT (2G)
21 -          LPHALF=LPIECE/2 ; NFHALF=LPHALF/2
22 -          IF (CMPPT.LT.0) CMPPT=WIDTH*2.54/LPHALF
23 -          READ (105,1111) KARD
24 -          WRITE (108,1122) KARD
25 -          DECODE (80,2222,KARD) SAMPSEC,FREQTIC,NCONLV,
26 -          $NCONLV,(CONLV(I),I=1,NCONLV),
27 -          $NCONLV,(CONFLV(I),I=1,NCONLV)
28 - 2222 FORMAT (3G,NG,NG)
29 -          READ (105,1111) KARD
30 -          WRITE (108,1122) KARD
31 -          DECODE (80,4444,KARD) IDENT1,IDENT2
32 - 4444 FORMAT (9A4,9A4)
33 - C
34 - C
35 - C READ BICOHERENCES INTO ARRAY DATA(200,130)
36 - C FIRST SET ARRAY DATA TO -999.
37 - C
38 - C
39 - C          OUTPUT MD2,MD1,NFHALF,LPHALF
40 - C          DO 50 J=1,MD2
41 - C
42 - C          DO 50 I=1,MD1
43 - C          50 DATA(I,J)=-999.
44 - C
45 - C          OUTPUT NSERIES,LPHALF,NFHALF,ISTORE
46 - C
47 - C          IF (NSERIES.EQ.1) GO TO 60
48 - C          READ (ISTORE) ((DATA(I,J),I=LPHALF+2,LPHALF+NFHALF+1),
49 - C          $J=1,NFHALF)
50 - C          OUTPUT 'B11'
51 - C          READ (ISTORE) ((DATA(I,J),I=NFHALF+1,LPHALF),J=1,NFHALF)
52 - C          OUTPUT 'B22'
53 - C          READ (ISTORE) ((DATA(I,J),I=LPHALF+2,LPHALF+NFHALF+1),
54 - C          $J=NFHALF+1,LPHALF)
55 - C          OUTPUT 'B33'
56 - C          READ (ISTORE) ((DATA(I,J),I=NFHALF+1,LPHALF),
57 - C          $J=NFHALF+1,LPHALF)
58 - C          OUTPUT 'B44'

```

```

59 -      READ (ISTORE) ((DATA(I,J),I=1,NFHALF),J=NFHALF+1,LPHALF)
60 - C      OUTPUT 'B2'
61 -      GO TO 80
62 -      60 CONTINUE
63 - C      OUTPUT 'B3'
64 -      READ (ISTORE) ((DATA(I,J),I=1,NFHALF),J=1,NFHALF)
65 -      READ (ISTORE) ((DATA(I,J),I=1,NFHALF),J=NFHALF+1,LPHALF)
66 -      80 CONTINUE
67 -      IF (IHISTORY.EQ.0) GO TO 85
68 - C READ FOURIER LABELS
69 -      READ (ISTORE) (INAMES(I),I=1,3),IVAR(1)
70 -      READ (ISTORE) (INAMES(I),I=4,6),IVAR(2)
71 -      READ (ISTORE) (INAMES(I),I=7,9),IVAR(3)
72 -      READ (ISTORE) MF
73 -      IF (MF.NE.21) GO TO 85
74 -      READ (ISTORE) LABL
75 -      85 CONTINUE
76 - C      DO 90 K=1,11,10
77 - C      WRITE (108,1111) ((DATA(I,J),J=K,K+9),I=1,30)
78 - C1111 FORMAT (10(X,F6.2))
79 - C      90 CONTINUE
80 - C      OUTPUT 'B4'
81 -      CALL PLOTS(IBUF,-1000)
82 - C      OUTPUT 'B5'
83 -      CALL PLOT(0.,0.5,-3)
84 - C      OUTPUT 'B6'
85 -      WRITE (108,1000) (CONLV(I),I=1,NCONLV)
86 -      1000 FORMAT (/, 'CONTOUR LEVELS:', /, (1H ,5G13.3, /))
87 -      CALL DIMWH(DATA,MD1,MD2)
88 -      CALL FLAGZ(-999.0)
89 -      CALL NOLABL
90 -      DEL=CMPT/2.54
91 -      AXLEN=LPHALF*DEL
92 -      SAMPHR=SAMPSEC/3600.
93 -      DFTOT=1./(2.*SAMPHR)
94 -      DISTIC=AXLEN*FREQTIC/DFTOT
95 -      AYLEN=(INT(AXLEN/DISTIC)+1)*DISTIC
96 -      AYHLEN=AXLEN/2.+0.25
97 -      FIRSTV=-(AYLEN/DISTIC)*FREQTIC
98 - C      OUTPUT 'B7'
99 -      IF (NSERIES.EQ.1) GO TO 120
100 - C X-AXIS WITH TIC MARKS ONLY, STARTING AT ORIGIN
101 - C      OUTPUT AXLEN,AYLEN,AYHLEN,FREQTIC,DISTIC,NDECPL,FIRSTV
102 -      CALL AXWJS(AXLEN,AXLEN,' ',-1,AXLEN,-90.,0.,1.,DISTIC,
103 -      $-1,0,0,.12,.7,.0001,.14)
104 - C X-AXIS STARTING ABOVE MAX Y VALUE
105 - C      OUTPUT 'B71',AXLEN+AYHLEN,AXLEN,FREQTIC,DISTIC,NDECPL
106 -      CALL AXDRAW(AXLEN+AYHLEN,AXLEN,'FREQ1 (CPH)',11,
107 -      $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
108 - C X-AXIS STARTING BELOW MIN Y VALUE
109 - C      OUTPUT 'B72',AXLEN-AYLEN,AXLEN,FREQTIC,DISTIC,NDECPL
110 -      CALL AXDRAW(AXLEN-AYLEN,AXLEN,'FREQ1 (CPH)',-11,
111 -      $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
112 - C Y-AXIS FOR TWO SERIES
113 - C      OUTPUT 'B73',AXLEN-AYLEN,AXLEN,AYLEN+AYHLEN,FIRSTV,DISTIC
114 -      CALL AXDRAW(AXLEN-AYLEN,AXLEN,'FREQ2 (CPH)',11,
115 -      $AYLEN+AYHLEN,0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
116 -      GO TO 150
117 -      120 CONTINUE
118 - C X-AXIS FOR ONE SERIES

```

```

119 -      CALL AXDRAW(0.,AXLEN,'FREQ1 (CPH)',-11,AXLEN,-90.,0.,
120 -      $FREQTIC,DISTIC,NDEC PL)
121 - C Y-AXIS FOR ONE SERIES
122 -      CALL AXDRAW(0.,AXLEN,'FREQ2 (CPH)',11,AYHLEN,0.,0.,
123 -      $FREQTIC,DISTIC,NDEC PL)
124 -      150 CONTINUE
125 - C PLOT IDENTIFICATION
126 -      IF (NSERIES.EQ.1) GO TO 160
127 -      EXPLOT=AXLEN+AYHLEN
128 -      GO TO 165
129 -      160 EXPLOT=AYHLEN
130 -      165 WYEPLLOT=AXLEN
131 -      IF (NSERIES.EQ.1) GO TO 170
132 -      CALL SYMBOL(EXPLOT+7.0,WYEPLLOT-1.8,0.315,
133 -      $'CROSS BICOHERENCE',-90.,17)
134 -      GO TO 175
135 -      170 CALL SYMBOL(EXPLOT+7.0,WYEPLLOT-2.0,0.315,
136 -      $'AUTO BICOHERENCE',-90.,16)
137 -      175 CONTINUE
138 -      IF (IHISTORY.EQ.0) GO TO 179
139 -      CALL SYMBOL(EXPLOT+6.5,WYEPLLOT-1.6,0.175,
140 -      $'ORIGINAL FILE NAMES:',-90.,20)
141 -      CALL SYMBOL(EXPLOT+6.5,WYEPLLOT-5.7,0.175,
142 -      $'VARIABLE NUMBERS:',-90.,17)
143 -      CALL SYMBOL(EXPLOT+6.2,WYEPLLOT-0.5,0.14,'FREQUENCY 1:',
144 -      $-90.,12)
145 -      CALL SYMBOL(EXPLOT+5.9,WYEPLLOT-0.5,0.14,'FREQUENCY 2:',
146 -      $-90.,12)
147 -      CALL SYMBOL(EXPLOT+5.6,WYEPLLOT-0.5,0.14,'FREQUENCY 3:',
148 -      $-90.,12)
149 -      IF (MF.NE.21) GO TO 177
150 -      CALL SYMBOL(EXPLOT+5.1,WYEPLLOT-2.9,0.175,
151 -      $'PROCESSING HISTORY',-90.,18)
152 -      177 CONTINUE
153 -      CALL SYMBOL(EXPLOT+6.2,WYEPLLOT-2.5,0.14,INAMES,-90.,12)
154 -      CALL SYMBOL(EXPLOT+5.9,WYEPLLOT-2.5,.14,INAMES(4),-90.,12)
155 -      CALL SYMBOL(EXPLOT+5.6,WYEPLLOT-2.5,.14,INAMES(7),-90.,12)
156 -      FVAR=FLOAT(IVAR(1))
157 -      CALL NUMBER(EXPLOT+6.2,WYEPLLOT-7.1,0.14,FVAR,-90.,-1)
158 -      FVAR=FLOAT(IVAR(2))
159 -      CALL NUMBER(EXPLOT+5.9,WYEPLLOT-7.1,0.14,FVAR,-90.,-1)
160 -      FVAR=FLOAT(IVAR(3))
161 -      CALL NUMBER(EXPLOT+5.6,WYEPLLOT-7.1,0.14,FVAR,-90.,-1)
162 -      IF (MF.NE.21) GO TO 179
163 -      CALL SYMBOL(EXPLOT+4.8,WYEPLLOT-0.7,0.105,LABL,-90.,72)
164 -      CALL SYMBOL(EXPLOT+4.5,WYEPLLOT-1.3,.105,LABL(19),-90.,60)
165 -      179 CONTINUE
166 -      CALL SYMBOL(EXPLOT+1.15,WYEPLLOT-1.0,0.210,IDENT1,-90.,36)
167 -      CALL SYMBOL(EXPLOT+0.85,WYEPLLOT-1.0,0.210,IDENT2,-90.,36)
168 -      CALL TODAY(ICLK)
169 -      CALL SYMBOL(EXPLOT+3.0,WYEPLLOT-0.6,0.175,'TIME OF PLOT:',
170 -      $-90.,13)
171 -      CALL SYMBOL(EXPLOT+2.7,WYEPLLOT-0.3,0.175,ICLK,-90.,16)
172 -      CALL SYMBOL(EXPLOT+3.8,WYEPLLOT-3.8,0.140,
173 -      $'CONTOUR LEVELS AND PERCENT CONFIDENCE',-90.,37)
174 -      BIASX=BIASY=0.
175 - C
176 - C
177 -      IF (NCONLV.GT.10) NCONLV=10
178 -      DO 185 I=1,NCONLV

```

```

179 -      IF (I.EQ.6)      BIASX=0. ; BIASY=BIASY-3.0
180 -      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLT-4.0+BIASY,0.210,
181 -      $CONLV(I),-90.,3)
182 -      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLT-5.4+BIASY,0.210,
183 -      $CONFLV(I),-90.,1)
184 -      BIASX=BIASX-0.4
185 -      185 CONTINUE
186 - C
187 - C
188 - C DRAW BOUNDARIES
189 -      CALL PLOT(0.,0.,3)
190 - C      OUTPUT 'B8'
191 -      IF (NSERIES.EQ.1) GO TO 200
192 -      CALL PLOT(AXLEN,AXLEN,2)
193 -      CALL PLOT(1.5*AXLEN,AXLEN/2.,2)
194 -      CALL PLOT(AXLEN,0.,2)
195 -      CALL PLOT(0.,0.,2)
196 -      GO TO 205
197 -      200 CONTINUE
198 -      CALL PLOT(0.,AXLEN,3)
199 -      CALL PLOT(AXLEN/2.,AXLEN/2.,2)
200 -      CALL PLOT(0.,0.,2)
201 -      205 CONTINUE
202 - C CALLS TO SYMBOL AND NUMBER
203 -      GO TO (210,220),NSERIES
204 -      210 CALL GRID(DEL,AXLEN-DEL,DEL,-DEL)
205 - C      OUTPUT DEL,NFHALF,LPHALF,'ONE'
206 -      CALL WHCNTR(DATA,1,NFHALF,1,LPHALF,NCONLV,CONLV)
207 -      CALL PLOT(0.,0.,3)
208 -      CALL PLOT(1.5*AXLEN,-0.5,999)
209 -      GO TO 250
210 -      220 CALL GRID(0.,AXLEN-DEL,DEL,-DEL)
211 - C      OUTPUT DEL,NFHALF,LPHALF,'TWO'
212 -      CALL WHCNTR(DATA,1,LPHALF+NFHALF+1,1,LPHALF,NCONLV,CONLV)
213 -      CALL PLOT(0.,0.,3)
214 -      CALL PLOT(2.5*AXLEN,-0.5,999)
215 -      250 CONTINUE
216 -      STOP 'NORMAL PROGRAM COMPLETION'
217 -      END

```

- PROGRAM FOURIER -

```

1 - C      <<< FOURIER >>>
2 - C PROGRAM TO CREATE RWDISC FILE OF FOURIER COEFFICIENTS
3 - C FROM RWDISC FILE OF SERIES
4 - C INPUT FILE HAS 21 LOGICAL FILES, THE LAST BEING
5 - C IMAGE OF ORIGINAL TAPDIS FILE.
6 - C PROGRAMMER: G. MARTINEAU VERSION 02 1/12/78
7 -      COMMON WK(33000),
8 -      $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
9 -      $ICLK(4),LABL(33),IOVERLAP,IHANN,
10 -     $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
11 -     $R2,ERRY,ERRAO,ERRAL,ISIGN,ITAPER,MFFT,
12 -     $NFILES,ISUBT,MF,IFIRST,ILAST
13 -     DIMENSION X(450),Y(450)
14 -     EQUIVALENCE (NSSMP(1),NSUBSAMP)
15 -     CALL CTRL
16 - X     OUTPUT 'F1'
17 -     CALL DPAR(NERR,MF,1,1,KSUM,512)
18 - C
19 - C
20 -     DO 500 LF=IFIRST,ILAST
21 - X     OUTPUT 'F2'
22 -     CALL INPUT(LF)
23 - X     OUTPUT 'F3'
24 -     CALL KOEFF(X,Y,LPIECE)
25 - X     OUTPUT 'F4'
26 -     CALL OUTOMEGA(LF)
27 - X     OUTPUT 'F5'
28 -     500 CONTINUE
29 - C
30 - C
31 -     CALL TODAY(ICLK)
32 -     CALL LABEL
33 - X     OUTPUT 'F6'
34 -     CALL CDISC
35 -     STOP 'PROCESSING COMPLETED'
36 - C
37 - C
38 -     SUBROUTINE LABEL
39 - C INTERNAL SUBROUTINE FOR PROGRAM FOURIER TO CREATE
40 - C LABEL AND OUTPUT TO LOG. FILE 21 AND WRITE
41 - C ON LINE PRINTER
42 -     NDATL=NPIECES*LPIECE
43 -     NFILES=ILAST-IFIRST+1
44 -     ENCODE (132,1000,LABL,NOCH) NFILES,NPIN,IPREW,ISUBSAMP,
45 -     $NSUBSAMP,LPIECE,NPIECES,IOVERLAP,ISUBT,IHANN,ICLK
46 -     1000 FORMAT ('NO LF'S ',I2,'*NO WDS DATA ',I5,'*PREW ',I1,
47 -     $'*SUBSAMP ',I1,'*NSUBSAMP ',I3,'*PC SIZE ',I5,'*NO PCS ',
48 -     $I3,'*OLAP ',I1,'*ISUBT ',I1,'*HANN ',I1,' *CREATED ',4A4)
49 -     IF (MF.NE.21) GO TO 150
50 - C WRITE L.F. 21 ON DISC
51 -     CALL WDISC(21,1,LABL,33)
52 -     150 CONTINUE
53 - C WRITE LABEL TO LINE PRINTER
54 -     WRITE(108,2000) LABL
55 -     2000 FORMAT(/,'LABEL AS WRITTEN ON L.F. 21 IF MF=21:',
56 -     $/,1H ,33A4,/)
57 -     RETURN
58 -     END

```

```

59 - C*****
60 - C*****
61 - SUBROUTINE LINREG(NPTS,X,Y,A0,A1,R2,ERRY,ERRA0,ERRA1)
62 - C*****
63 - C
64 - C LINEAR REGRESSION OF Y ON X
65 - C
66 - C DOUBLE PRECISION SX,SY,SX2,SY2,SXY,SXX,SY,Y,A0,A1
67 - C DIMENSION X(1),Y(1),TEMP(450)
68 - C CALCULATE SUMS OF SQUARES, ETC.
69 - C CALL SUMM(NPTS,X,SX)
70 - C CALL SUMM(NPTS,Y,SY)
71 - C DO 10 I=1,NPTS
72 - 10 TEMP(I)=X(I)*X(I)
73 - C CALL SUMM(NPTS,TEMP,SX2)
74 - C DO 20 I=1,NPTS
75 - 20 TEMP(I)=Y(I)*Y(I)
76 - C CALL SUMM(NPTS,TEMP,SY2)
77 - C DO 30 I=1,NPTS
78 - 30 TEMP(I)=X(I)*Y(I)
79 - C CALL SUMM(NPTS,TEMP,SXY)
80 - C SXX=SX2-SX*SX/NPTS
81 - C SY=SY2-SY*SY/NPTS
82 - C SSXY=SXY-SX*SY/NPTS
83 - C CALCULATE REGRESSION LINE Y=A0+A1*X
84 - C A1=SSXY/SXX
85 - C A0=SY/NPTS-A1*SX/NPTS
86 - C CALCULATE CORRELATION COEFFICIENT
87 - C R2=SSXY*SSXY/SXX/SY
88 - C CALCULATE STANDARD ERRORS OF ESTIMATE
89 - C ERRY=DSQRT((SY2-A0*SY-A1*SXY)/(NPTS-2))
90 - C ERRA0=ERRY*DSQRT(SX2/NPTS/SXX)
91 - C ERRA1=ERRY/DSQRT(SXX)
92 - C RETURN
93 - C END
94 - C*****
95 - C*****
96 - C SUBROUTINE TRENDR(NPTS,X,A0,A1)
97 - C*****
98 - C
99 - C REMOVES LINEAR REGRESSION LINE FOUND BY LINREG
100 - C
101 - C DOUBLE PRECISION A0,A1
102 - C DIMENSION X(1)
103 - C DO 10 I=1,NPTS
104 - 10 X(I)=X(I)-A0-A1*I
105 - C RETURN
106 - C END
107 - C*****
108 - C*****
109 - C SUBROUTINE SUMM(N,X,SUMX)
110 - C*****
111 - C
112 - C DOUBLE PRECISION SUM1,SUMX
113 - C DIMENSION X(1)
114 - C SUM1=0
115 - C DO 10 I=1,N
116 - 10 SUM1=SUM1+X(I)
117 - C SUMX=SUM1
118 - C RETURN

```

```

119 -      END
120 - C*****
121 - C*****
122 -      SUBROUTINE CTRL
123 - C FOR PROGRAM FOURIER, TO INITIALIZE,
124 - C AND READ IN CONTROL PARAMETERS
125 -      COMMON WK(33000),
126 -      $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
127 -      $ICLK(4),LABL(33),IOVERLAP,IHANN,
128 -      $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
129 -      $R2,ERRY,ERRAO,ERRAL,ISIGN,ITAPER,MFFT,
130 -      $NFILES,ISUBT,MF,IFIRST,ILAST
131 -      DIMENSION KFFT(84)
132 -      DATA KFFT/
133 -      $ 12, 16, 20, 24, 32, 36, 40, 48, 60, 64,
134 -      $ 72, 80, 96, 100, 108, 120, 128, 144, 160, 180,
135 -      $ 192, 200, 216, 240, 256, 288, 300, 320, 324, 360,
136 -      $ 384, 400, 432, 480, 500, 512, 540, 576, 600, 640,
137 -      $ 648, 720, 768, 800, 864, 900, 960, 972, 1000, 1024,
138 -      $ 1080, 1152, 1200, 1280, 1296, 1440, 1500, 1536, 1600, 1620,
139 -      $ 1728, 1800, 1920, 1944, 2000, 2048, 2160, 2304, 2400, 2500,
140 -      $ 2560, 2592, 2700, 2880, 2916, 3000, 3072, 3200, 3240, 3456,
141 -      $ 3600, 3840, 3888, 4000/
142 -      EQUIVALENCE (NSSMP(1),NSUBSAMP)
143 -      IPREW=ISUBSAMP=NSUBSAMP=ISUBT=0
144 -      MF=21
145 -      IOVERLAP=IHANN=1
146 -      NAMELIST MF
147 -      READ(105,1000) IFIRST,ILAST,NDO2,IPREW,ISUBSAMP,
148 -      $ISUBSAMP,(NSSMP(1), I=1,ISUBSAMP),LPIECE,IOVERLAP,ISUBT,
149 -      $IHANN
150 -      1000 FORMAT (5G,N(G),4G)
151 -      DO 90 I=1,84
152 -      IF (LPIECE.EQ.KFFT(I)) GO TO 200
153 -      90 CONTINUE
154 -      WRITE (108,2000) LPIECE,KFFT ; STOP
155 -      2000 FORMAT ('PIECE LENGTH OF ',I4,' INVALID.MUST BE ',
156 -      $'FROM THE FOLLOWING SET:',/(X,25I5))
157 -      200 CONTINUE
158 -      NPIN=NDO2
159 -      IF (ISUBSAMP.EQ.0) GO TO 100
160 - C NSUBSAMP MUST BE ODD FOR TRIANGULAR WEIGHTING
161 -      IF(MOD(NSUBSAMP,2) .EQ. 0) WRITE(108,1111);
162 -      $OUTPUT NSUBSAMP;STOP
163 -      1111 FORMAT('SUBSAMPLE LENGTH MUST BE ODD')
164 -      NDO2 = NDO2/NSUBSAMP
165 -      100 CONTINUE
166 -      INPUT
167 -      IF ((IHANN.EQ.0).AND.(IOVERLAP.EQ.1)) IOVERLAP=0 ;
168 -      $OUTPUT 'OVERLAP POSSIBLE ONLY IF HANN. IOVERLAP SET TO 0'
169 -      IF ((IOVERLAP.EQ.0).AND.((LPIECE.GT.450).OR.
170 -      $(LPIECE.LT.NDO2/150))) OUTPUT LPIECE,
171 -      $'PIECE LENGTH DOES NOT SATISFY RESTRICTION FOR NO OVERLAP
172 -      $'; STOP
173 -      IF ((IOVERLAP.NE.0).AND.((LPIECE.GT.450).OR.
174 -      $(LPIECE.LT.NDO2/75))) OUTPUT LPIECE,
175 -      $'PIECE LENGTH DOES NOT SATISFY RESTRICTION FOR OVERLAP';
176 -      $STOP
177 -      IF (MF.GT.21) OUTPUT 'MF CANNOT EXCEED 21' ; STOP
178 -      IF (((MF.LT.21).AND.(ILAST.GT.MF)).OR.

```

```

179 -      $((MF.EQ.21).AND.(ILAST.GT.20))) WRITE (108,2222)
180 - 2222 FORMAT ('LAST LOG. FILE CANNOT EXCEED MF OR 20')
181 -      RETURN
182 -      END
183 - C*****
184 - C*****
185 -      SUBROUTINE INPUT(LF)
186 - C FOR PROGRAM FOURIER, TO INPUT AND PREPARE
187 - C AN RWDISC LOG. FILE FOR THE TRANSFORM.
188 - C DOES PREWHITENING,SAMPLING AND SPLITTING
189 - C INTO PIECES
190 -      COMMON WK(33000),
191 -      $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
192 -      $ICLK(4),LABL(33),IOVERLAP,IHANN,
193 -      $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
194 -      $R2,ERRY,ERRAO,ERRAL,ISIGN,ITAPER,MFFT,
195 -      $NFILES,ISUBT,MF,IFIRST,ILAST
196 -      EQUIVALENCE (NSSMP(1),NSUBSAMP)
197 - X      OUTPUT 'I1'
198 - C READ IN DATA FROM RWDISC FILE, STARTING AFTER LABELS
199 -      CALL RDISC(LF,5,WK,NPIN)
200 -      IF(IPREW .EQ. 1) CALL PREWHITE
201 -      IF(ISUBSAMP .EQ. 0) GO TO 150
202 - C
203 - C
204 - C DO TRIANGULAR WEIGHTING
205 - C
206 -      DO 100 I=1,ND02
207 -      I1 = 1+(I-1)*NSUBSAMP
208 -      MT = NSUBSAMP/2+1
209 -      ASUM = 0.
210 - C
211 -      DO 80 J=1,NSUBSAMP
212 -      80 ASUM = ASUM+((MT-ABS(MT-J))/(MT*MT))*WK(I1+J-1)
213 - C
214 -      WK(I) = ASUM
215 - 100 CONTINUE
216 - C
217 - C
218 - 150 CONTINUE
219 - X      OUTPUT 'I2'
220 - C
221 -      IF (IOVERLAP.EQ.0) GO TO 300
222 - C
223 -      NDIM = 2*ND02
224 -      NPIECES = 2*ND02/LPIECE-1
225 - C CALCULATE LENGTH OF TOTAL PIECE TO BE MOVED
226 -      NMOVE = NPIECES/2
227 -      LPHALF = LPIECE/2
228 -      IF(NMOVE .EQ. 0) MPT(1) = 1; RETURN
229 -      NMLP = NMOVE*LPIECE
230 - C FILL ARRAY OF INDICES
231 - C
232 - C
233 -      DO 200 I=1,NPIECES
234 -      MPT(I) = (1+(I/2)*LPIECE)*MOD(I,2)+
235 -      $(1+(I/2-1)*LPIECE+ND02)*MOD(I+1,2)
236 - 200 CONTINUE
237 - C
238 - C

```

```

239 - C ORGANIZE ARRAY WK
240 - C
241 - C
242 - X      OUTPUT 'I3'
243 -      DO 250 I=1,NMLP
244 -      WK(NDO2+I) = WK(LPHALF +I)
245 -      250 CONTINUE
246 -      GO TO 800
247 - C
248 - C NO SPLITTING INTO PIECES
249 -      300 CONTINUE
250 -      NDIM=NDO2
251 -      NPIECES=NDO2/LPIECE
252 - C FILL ARRAY OF INDICES
253 - C
254 - C
255 -      DO 400 I=1,NPIECES
256 -      MPT(I)=1+(I-1)*LPIECE
257 -      400 CONTINUE
258 - C
259 - C
260 -      800 CONTINUE
261 - C
262 - C
263 -      RETURN
264 -      SUBROUTINE PREWHITE
265 - C INTERNAL SUBROUTINE FOR SUBROUTINE INPUT
266 - C TO PREWHITEN DATA SERIES
267 - C
268 - C
269 - X      OUTPUT 'I4'
270 -      DO 100 I=1,NDO2-1
271 -      100 WK(I) = WK(I+1)-WK(I)
272 - C
273 - C
274 -      WK(NDO2) = WK(NDO2-1)
275 -      RETURN
276 -      END
277 - C*****
278 - C*****
279 -      SUBROUTINE KOEFF(X,Y,LPIECE)
280 - C FOR PROGRAM FOURIER, TO DO FFT ON EACH PIECE
281 - C AND REPLACE N POINTS OF PIECE BY
282 - C COEFFICIENTS A0,A1,...A(N/2),B1,B2,...B(N/2-1)
283 - C ALSO SUBTRACTS MEAN OR LINEAR TREND, AND FILTERS
284 -      COMMON WK(33000),
285 -      $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
286 -      $ICLK(4),LABL(33),IDOVERLAP,IHANN,
287 -      $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPCDUM,
288 -      $R2,ERRY,ERRA0,ERRA1,ISIGN,ITAPER,MFFT,
289 -      $NFILES,ISUBT,MF,IFIRST,ILAST
290 - X      OUTPUT NDO2,NDIM,IPREW,ISUBSAMP,NPIECES,NFILES,ISUBT
291 - X      OUTPUT MFFT,ISIGN,ITAPER
292 -      EQUIVALENCE (NSSMP(1),NSUBSAMP)
293 -      DIMENSION X(LPIECE),Y(LPIECE)
294 -      DOUBLE PRECISION A0,A1
295 - X      OUTPUT LPIECE
296 -      LPHALF = LPIECE/2
297 - C
298 - C

```

```

299 -      DO 800 I=1,NPIECES
300 -      M=MPT(I)
301 -      AMEAN(I) = 0.
302 - C  PULL PIECE OUT OF WORKING ARRAY
303 - C
304 -      DO 40 J=1,LPIECE
305 -      X(J) = WK(M+J-1)
306 -      40 Y(J) = J
307 - X  WRITE (108,1111) I,MPT(I),(X(J),J=1,LPIECE)
308 - X1111 FORMAT ('DATA PIECE',2X,'PIECE NO. ',I2,10X,'MPT(I)=' ,
309 - X      $I5,/,6(1H ,5G13.5,/),(1H ,2G13.5,//))
310 - C
311 -      DO 80 J=1,LPIECE
312 -      80 AMEAN(I) = AMEAN(I) + X(J)
313 -      AMEAN(I) = AMEAN(I)/LPIECE
314 -      GO TO (400,200,100,85),ISUBT + 1
315 -      85 CONTINUE
316 - C  SUBTRACT MEAN OF EACH PIECE
317 - C
318 -      DO 90 J=1,LPIECE
319 -      90 X(J) = X(J) - AMEAN(I)
320 - C
321 - X  OUTPUT 'K2'
322 -      GO TO 400
323 -      100 CONTINUE
324 - C  CALCULATE TREND BY LINEAR REGRESSION
325 - C  AND REMOVE IT
326 -      CALL LINREG(LPIECE,Y,X,A0,A1,R2,ERRY,ERRA0,ERRA1)
327 -      CALL TRENDR(LPIECE,X,A0,A1)
328 -      GO TO 400
329 -      200 CONTINUE
330 - C  TREND REMOVAL BASED ON ENDPOINTS
331 - X  OUTPUT 'K3'
332 -      AA = (X(LPIECE)-X(1))/(LPIECE-1)
333 -      BB = (LPIECE*X(1)-X(LPIECE))/(LPIECE-1)
334 - C
335 -      DO 300 J=1,LPIECE
336 -      X(J) = X(J)-AA*J-BB
337 -      300 CONTINUE
338 - C
339 -      400 CONTINUE
340 - C  DO FOURIER TRANSFORM
341 - X  OUTPUT 'K4',LPIECE
342 -      INDCTR=1
343 -      DO 450 J=1,LPIECE
344 -      450 Y(J)=0.
345 -      CALL HARM1(X,Y,INDCTR,LPIECE)
346 - C
347 -      IF (IHANN.EQ.0) GO TO 685
348 -      C=SQRT(8./3.)
349 - C
350 -      DO 500 K=1,LPIECE
351 -      500 X(K)=C*X(K)
352 - C
353 -      550 CONTINUE
354 -      SAV1=X(1)
355 - C
356 -      DO 650 K=1,LPHALF-1
357 -      650 X(K)=0.5*X(K+1)-0.25*(X(K)+X(K+2))
358 - C

```

```

359 -      DO 655 K=1,LPHALF-1
360 -      655 X(1+LPHALF-K)=X(LPHALF-K)
361 - C
362 -      X(1)=SAV1
363 -      SAV1=X(LPHALF+2)
364 - C
365 -      DO 670 K=LPHALF+2,LPIECE-2
366 -      670 X(K)=0.5*X(K+1)-0.25*(X(K)+X(K+2))
367 - C
368 -      DO 675 K=1,LPHALF-3
369 -      675 X(LPIECE-K)=X(LPIECE-K-1)
370 - C
371 -      X(LPHALF+2)=SAV1
372 -      685 CONTINUE
373 - C RESTORE PIECE TO WORKING ARRAY
374 - C
375 - X      OUTPUT M
376 -      DO 700 J=1,LPIECE
377 -      700 WK(M+J-1) = X(J)
378 - C
379 - X      WRITE (108,2222) I,MPT(I),(X(J),J=1,LPIECE)
380 - X2222 FORMAT ('COEFFICIENTS',2X,'PIECE NO. ',I2,10X,'MPT(I)=' ,
381 - X      $I5,/,6(1H ,5G13.5,/),(1H ,2G13.5,/) )
382 -      800 CONTINUE
383 - C
384 - C
385 -      RETURN
386 -      END
387 - C*****
388 - C*****
389 -      SUBROUTINE OUTOMEGA(LF)
390 - C FOR PROGRAM FOURIER TO OUTPUT COEFFICIENTS TO RWDISC
391 - C FILE IN ORDER OF INCREASING FREQUENCY AND
392 - C ORDER A,B AT A GIVEN FREQUENCY, I.E.:
393 - C A(0),A(1),B(1),...,A(N/2-1),B(N/2-1),A(N/2)
394 -      COMMON WK(33000),
395 -      $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
396 -      $ICLK(4),LABL(33),IOVERLAP,IHANN,
397 -      $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
398 -      $R2,ERRY,ERRAO,ERRAL,ISIGN,ITAPER,MFFT,
399 -      $NFILES,ISUBT,MF,IFIRST,ILAST
400 -      EQUIVALENCE (NSSMP(1),NSUBSAMP)
401 -      DIMENSION BUF(2048)
402 -      KOEFF1=1;LPHALF=LPIECE/2;KA=0;KB=LPHALF+1;IAB=-1;M=1
403 -      IF (NPIECES.EQ.1) GO TO 200
404 -      NCOEFF=2048/NPIECES
405 -      KOEFF2=NCOEFF
406 -      IF (NCOEFF.GE.LPIECE) KOEFF2=LPIECE
407 - C FILL BUF FROM KOEFF1 THRU KOEFF2
408 -      50 CONTINUE
409 -      M=1
410 - C
411 - C
412 -      DO 100 I=KOEFF1,KOEFF2
413 -      IAB=-IAB
414 -      IF (I.EQ.2) IAB=-IAB
415 -      IF (IAB.EQ.1) KA=KA+1 ; KUSE=KA
416 -      IF (IAB.EQ.-1) KB=KB+1 ; KUSE=KB
417 - X      IF (KUSE.GT.LPIECE) STOP 'CHK LOGIC 0000'
418 - C

```

```
419 -      DO 80 J=1,NPIECES
420 -      LOC=MPT(J)-1
421 -      BUF(M)=WK(LOC+KUSE)
422 -      M=M+1
423 -      80 CONTINUE
424 - C
425 -      100 CONTINUE
426 - C
427 - C
428 - C WRITE TO DISC
429 -      IWR=(KOEFF1-1)*NPIECES+1
430 -      NWR=(KOEFF2-KOEFF1+1)*NPIECES
431 -      CALL WDISC(LF,4+IWR,BUF,NWR)
432 - C RETURN IF HAVE WRITTEN LPIECE COEFFICIENTS
433 -      IF (KOEFF2.EQ.LPIECE) RETURN
434 - X      IF (KOEFF2.GT.LPIECE) STOP 'CHK LOGIC 1111'
435 -      KOEFF1=KOEFF2+1
436 -      KOEFF2=KOEFF2+NCOEFF
437 -      IF (KOEFF2.GE.LPIECE) KOEFF2=LPIECE
438 -      GO TO 50
439 - X 150 STOP 'CHK LOGIC 2222'
440 -      200 RETURN
441 -      END
```

- PROGRAM FRAGTAP -

```

1 - C          <<<<<< F R A G T A P >>>>>>
2 - C PROGRAM TO READ TAPDIS FILES AND CREATE CONSECUTIVE FILES
3 - C FOR LATER TRANSMITTAL TO NON-TAPDIS RWDISC FILE
4 - C PROGRAMMER: G. MARTINEAU VERSION 01 3/9/77
5 -     DIMENSION FILE(18000),IFILE(2500),JDCB(25),NSEQ(25),
6 -     $NV(25),ILF(25,4),JFILES(25),NCYC(25),IFILNAM(3),
7 -     $ISVAR(4),ILAB(9)
8 -     EQUIVALENCE (FILE,IFILE),(ILAB,IFILNAM),(ILAB(4),INVAR),
9 -     $(ILAB(5),ICYCLES),(ILAB(6),ISVAR)
10 -     MFILE=7
11 -     CALL DPAR(NERR,MFILE,1,1,KSUM)
12 -     WRITE (108,1000)
13 - 1000 FORMAT ('INPUT RECORDS FOLLOW (IN GENERALIZED FORMAT)')
14 -     READ (105,1500) NF,NF,{JDCB(I),I=1,NF},LDCB
15 - 1500 FORMAT (G,NG,G)
16 -     WRITE (108,1500) NF,NF,{JDCB(I),I=1,NF},LDCB
17 - C
18 - C
19 -     DO 100 I=1,NF
20 -     READ (105,2000) NSEQ(I),NV(I),NV(I),(ILF(I,K),K=1,NV(I))
21 - 2000 FORMAT (2G,NG)
22 -     WRITE (108,2200) NSEQ(I),NV(I),NV(I),(ILF(I,K),K=1,NV(I))
23 - 2200 FORMAT(1H ,2G,NG)
24 - 100 CONTINUE
25 - C
26 - C
27 - C CALCULATE STARTING LOCATIONS OF BLOCKS IN TAPDIS FILE
28 -     JFILES(1)=1
29 -     CALL RDISC(1,100,NOFILES,1)
30 - X     OUTPUT NOFILES
31 -     IF (NERR.NE.0) OUTPUT NERR ; STOP 2
32 -     LENGTH=100*NOFILES
33 -     CALL RDISC(1,1,IFILE,LENGTH)
34 - X     OUTPUT LENGTH
35 -     IF (NERR.NE.0) OUTPUT NERR ; STOP 3
36 - C
37 - C
38 -     DO 180 I=1,NOFILES
39 - 180 NCYC(I)=IFILE(100*(I-1)+9)
40 - C
41 -     DO 200 I=2,NOFILES
42 - 200 JFILES(I)=JFILES(I-1)+IFILE(100*(I-2)+9)
43 - C
44 - C
45 -     WRITE (108,2500) (I,JFILES(I),I=1,NOFILES)
46 - 2500 FORMAT (/ ,T4,'SEQUENTIAL',T21,'REL LOCATION',/,
47 -     $T2,'INPUT FILE READ',T22,'IN TAPDIS',/,T5,'BY TAPDIS',
48 -     $T21,'FILE (KREL)',/, (T8,I2,T23,I6))
49 -     WRITE (108,2700)
50 - 2700 FORMAT (/ ,T5,'OUTPUT FILE SUMMARY IN ORDER OF ACCESS',/,
51 -     $T2,'OUTPUT',T29,'BUOY TAPE VAR'S',T50,'TOTAL WORDS',/,
52 -     $T3,'DCB',T13,'TAPE FILE',T28,'ACCESSED THIS FILE',T50,
53 -     $'IN OUTPUT FILE',/)
54 - C
55 - C
56 -     DO 240 I=1,NF
57 -     IXDCB=JDCB(I)
58 -     KALL=NSEQ(I)

```

```

59 -      ICYCLES=NCYC(KALL)
60 -      INVAR=NV(I)
61 - C
62 -      DO 230 N=1,3
63 - 230 IFILNAM(N)=IFILE(100*(KALL-1)+N)
64 - C
65 -      DO 235 N=1,INVAR
66 - 235 ISVAR(N)=IFILE(100*(KALL-1)+4+N)
67 - C
68 -      WRITE (108,3000) IXDCB,IFILNAM,INVAR,(ISVAR(N),N=1,INVAR)
69 -      $,ICYCLES+9
70 - 3000 FORMAT (T2,I3,T12,3A4,T29,N(I2,2X),T52,I5)
71 - X      OUTPUT I,IXDCB
72 -      CALL BUFFER OUT(IXDCB,1,ILAB,9,ISTAT)
73 -      IF (ISTAT.NE.2) OUTPUT ISTAT; STOP 3
74 - 240 CONTINUE
75 - C
76 - C
77 - C WRITE LABELS TO BINARY SEQ. FILE
78 -      CALL BUFFER OUT(LDCB,1,IFILE,LENGTH,ISTAT)
79 - X      OUTPUT 'LABELS OUTPUT'
80 -      IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 1
81 - C READ AND WRITE TAPDIS FILES
82 - C
83 - C
84 -      DO 300 I=1,NF
85 -      IXDCB=JDCB(I)
86 -      KALL=NSEQ(I)
87 -      ICYCLES=NCYC(KALL)
88 -      KSTART=JFILES(KALL)
89 - C
90 -      DO 250 J=1,NV(I)
91 -      IVAR=ILF(I,J)
92 - X      OUTPUT IVAR,KSTART,ICYCLES
93 -      CALL RDISC(IVAR,KSTART,FILE,ICYCLES)
94 -      IF (NERR.NE.0) OUTPUT NERR ; STOP 4
95 - X      OUTPUT I,IXDCB,ICYCLES
96 -      CALL BUFFER OUT(IXDCB,1,FILE,ICYCLES,ISTAT)
97 -      IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 4
98 - 250 CONTINUE
99 - C
100 - 300 CONTINUE
101 - C
102 - C
103 -      STOP 'OUTPUT COMPLETED'
104 -      END

```

- PROGRAM GENRAN -

```

1 - C          <<<< G E N R A N O 4 >>>>
2 - C PROGRAM TO GENERATE RWDISC FILE OF KNOWN SIGNALS.
3 - C USES IPC PROGRAM NORAN FOR PSEUDO-RANDOM NOISE.
4 - C PROGRAMMER: G. MARTINEAU      8/2/77
5 -          DOUBLE PRECISION T,PI,TWOPI,DFLS,DFLTH,FL,DFMS,DFMTH
6 -          DOUBLE PRECISION
7 -          $W11(20),W12(20),W21(20),W22(20),W31(20),W32(20),
8 -          $W41(20),W42(20),
9 -          $T11(20),T12(20),T21(20),T22(20),T31(20),T32(20),
10 -         $T41(20),T42(20),
11 -         $AH1(20),AH2(20),AH3(20),AH4(20),AH5(20),AH6(20),
12 -         $TH1(20),TH2(20),TH3(20),TH4(20),TH5(20),TH6(20)
13 -         DIMENSION KHARM(20),XS1(20),XS2(20),XS3(20),XS4(20),
14 -         $BMEAN(20),SDE(20),IXA(20)
15 -         DIMENSION LABL(4)
16 -         COMMON B1(2560),KG(20),KNG(20),XG(20),XNG(2,20),FUND(20),
17 -         $AH1,AH2,AH3,AH4,AH5,AH6,
18 -         $TH1,TH2,TH3,TH4,TH5,TH6,
19 -         $IDIM,IFL,TWOPI,LENGTH
20 -         NAMELIST IG,ING,AMEAN,STDEV,AG,ANG,AS1,AS2,AS3,AS4,IHARM,
21 -         $F11,F12,F21,F22,F31,F32,P11,P12,P21,P22,P31,P32,KSAME,IX,
22 -         $F41,F42,P41,P42,
23 -         $ASQ,ACUBE,HA1,HA2,HA3,HA4,HA5,HA6,FF,
24 -         $HP1,HP2,HP3,HP4,HP5,HP6,
25 -         $NOLIST,IDATSTART
26 -         IDIM=2560 ; MODFLG=0
27 -         IHARM=0
28 -         IDATSTART=5
29 -         NOLIST=1
30 -         PI=3.141592653589793
31 -         TWOPI=2.*PI
32 -         DTR=PI/180.
33 -         RTD=180./PI
34 -         LABL(1)=4HGENR
35 -         LABL(2)=4HANOU
36 -         LABL(3)=4HTPUT
37 -         LABL(4)=0
38 -         CALL INGEN
39 -         CALL OPAR(NERR,MF,1,1,KSUM,512)
40 -         NBLKS=LENGTH/IDIM
41 -         MMOD=LENGTH-IDIM*NBLKS
42 -         NBSAVE=NBLKS
43 -         IDIMSAVE=IDIM
44 - C CALL NORAN
45 -         IF (NSERIES.GT.MF) OUTPUT
46 -         $'NO. OF SERIES CAN'T BE GREATER THAN NO. OF LOG. FILES';
47 -         $STOP
48 -         IF ((MF.EQ.21).AND.(NSERIES.EQ.21)) NSERIES=20 ; OUTPUT
49 -         $'NO. OF SERIES HAS BEEN SET TO 20'
50 - C
51 - C
52 - C
53 -         DO 400 IFL=1,NSERIES
54 -         LZ=-1
55 -         CALL WDISC(IFL,1,LABL,4)
56 -         IF (NBLKS.EQ.0) GO TO 375
57 - 305 CONTINUE
58 - C

```

```

59 - C
60 -     KV=IXA(IFL)
61 -     DO 370 J=1,NBLKS
62 - C INITIALIZE WORKING ARRAYS
63 - C
64 -     DO 310 I=1,IDIM
65 - 310 B1(I)=0.
66 - C
67 -     IF ((KG(IFL).EQ.0).AND.(KNG(IFL).EQ.0)) GO TO 350
68 -     IF (MOD(IDIM,2).EQ.1) GO TO 320
69 -     IHALF=IDIM/2
70 - C
71 -     DO 315 K=1,IHALF
72 - 315 CALL NORAN(1,KV,BMEAN(IFL),SDE(IFL),B1(K),B1(K+IHALF))
73 -     GO TO 330
74 - 320 CONTINUE
75 -     IHALF=IDIM/2
76 - C
77 -     DO 325 K=1,IHALF+1
78 - 325 CALL NORAN(1,KV,BMEAN(IFL),SDE(IFL),B1(K),B1(K+IHALF+1))
79 - C
80 - 330 CONTINUE
81 - C SUBROUTINE NOISE ADDS GAUSSIAN PLUS NON-G. NOISE
82 -     CALL NOISE
83 - 350 CONTINUE
84 - C
85 - C
86 -     IF (KHARM(IFL).EQ.0) GO TO 355
87 - C SUBROUTINE HARMONICS ADDS 1ST THRU 6TH HARMONICS
88 -     CALL HARMONICS
89 - 355 CONTINUE
90 - C
91 -     DO 360 K=1,IDIM
92 -     LZ=LZ+1
93 -     DFLS=DFLCAT(LZ) ; DFLTH=DFLOAT(LENGTH)
94 -     T=TWOPI*DFLS/DFLTH
95 -     N=IFL
96 -     B1(K)=B1(K)+
97 -     $XS1(N)*DSIN(W11(N)*T+T11(N))*DSIN(W12(N)*T+T12(N))+
98 -     $XS2(N)*DSIN(W21(N)*T+T21(N))*DSIN(W22(N)*T+T22(N))+
99 -     $XS3(N)*DSIN(W31(N)*T+T31(N))*DSIN(W32(N)*T+T32(N))+
100 -     $XS4(N)*DSIN(W41(N)*T+T41(N))*DSIN(W42(N)*T+T42(N))
101 - 360 CONTINUE
102 - C
103 -     IF (MODFLG.EQ.1) GO TO 365
104 -     CALL WDISC(IFL,IDATSTART+(J-1)*IDIM,B1,IDIM)
105 -     GO TO 370
106 - 365 CALL WDISC(IFL,IDATSTART+NBSAVE*IDIMSAVE,B1,IDIM)
107 - 370 CONTINUE
108 - C
109 - C
110 - 375 CONTINUE
111 -     IF (MMOD.EQ.0) GO TO 390
112 -     NBLKS=1
113 -     MODFLG=1
114 -     IDIM=MMOD
115 -     MMOD=0
116 -     GO TO 305
117 - 390 CONTINUE
118 -     NBLKS=NBSAVE

```

```

119 -      IDIM=IDIMSAVE
120 -      MODFLG=0
121 -      MMOD=LENGTH-IDIM*NBLKS
122 -      400 CONTINUE
123 - C
124 - C
125 - C
126 -      CALL CDISC
127 -      STOP
128 - C
129 - C
130 - C
131 - C
132 -      SUBROUTINE INGEN
133 - C  INTERNAL SUBROUTINE FOR PROGRAM GENRAN02 TO INITIALIZE
134 - C  AND INPUT PROGRAM VARIABLES
135 -      IDEC=0
136 -      WRITE (108,1000)
137 -      1000 FORMAT ('INPUT LENGTH, NO. OF SERIES, NO. OF LOG. FILES')
138 -      READ (105,1500) LENGTH,NSERIES,MF
139 -      1500 FORMAT (3G)
140 -      WRITE (108,2000)
141 -      2000 FORMAT ('YOU WILL BE ASKED TO INPUT NAMELIST VARIABLES ',
142 -      $'FOR EACH SERIES.',/,
143 -      $'IG,ING,IHARM (VALUE 1 OR 0) ARE SWITCHES FOR GAUSS., ',
144 -      $'NON-G. NOISE, AND ',/,
145 -      $T6,'HARMONIC SERIES ** DEFAULTS 0,0,0',/,
146 -      $'GAUSSIAN NOISE:',/,
147 -      $' AG,AMEAN,STDEV ARE AMPLITUDE,MEAN,ST DEV. DEFAULTS ',
148 -      $'1,0,1',/,
149 -      $' "SEED" FOR SUBROUTINE NORAN IS IX. DEFAULT 314159',/,
150 -      $'NON-GAUSSIAN NOISE:',/,
151 -      $' FORM IS ASQ*X**2+ACUBE*X**3 WHERE X IS GAUSS. NOISE ',
152 -      $/, ' DEFAULTS ARE ASQ=ACUBE=1',/,
153 -      $'HARMONIC SERIES:',/,
154 -      $' SIX TERMS.AMPLITUDES ARE HA1 THRU HA6. FUND FREQ. IS ',
155 -      $' FF.',/,
156 -      $' PHASES HP1 THRU HP6 IN DEGREES. DEFAULTS ALL 0',/,
157 -      $'SUM OF SIN*SIN TERMS:',/,
158 -      $' FOUR TERMS. AMPLITUDES ARE AS1,AS2,AS3,AS4. FREQ''S, '
159 -      $,'PHASES (IN DEGREES)',/,
160 -      $' ARE FIJ,PIJ WHERE I=1,2,3,4 REFERS TO SIN*SIN TERM ',/
161 -      $,' WITH AMPL. AS1,AS2,AS3,AS4 AND J=1,2 REFERS TO 1ST OR'
162 -      $,' 2ND FACTOR ',/,
163 -      $' IN A GIVEN SIN*SIN TERM. ** DEFAULTS ALL 0.',/,
164 -      $'INPUT KSAME=(SERIES NO.) TO DUPLICATE ALL PARAMETERS ',
165 -      $/, 'FROM A PREVIOUSLY INPUT SIGNAL.')
166 -      50 CONTINUE
167 - C
168 - C
169 -      DO 100 JJ=1,NSERIES
170 -      WRITE (108,2500) JJ
171 -      2500 FORMAT (/,'ENTER SIGNAL PARAMETERS FOR SERIES ',I2,' :')
172 -      KSAME=0
173 -      IF (IDEC.EQ.1) GO TO 60
174 -      IG=ING=IHARM=0
175 -      AG=ANG=1.;AS1=AS2=AS3=AS4=0.;AMEAN=0.;STDEV=1.
176 -      ASQ=ACUBE=1.
177 -      IX=314159
178 -      HA1=HA2=HA3=HA4=HA5=HA6=0.

```

```

179 -      F11=F12=F21=F22=F31=F32=P11=P12=P21=P22=P31=P32=0.
180 -      F41=F42=P41=P42=0.
181 -      FF=HP1=HP2=HP3=HP4=HP5=HP6=0.
182 -      GO TO 70
183 - 60 CONTINUE
184 -      IX=IXA(JJ)
185 -      IG=KG(JJ); ING=KNG(JJ); AMEAN=BMEAN(JJ); STDEV=SDE(JJ)
186 -      IHARM=KHARM(JJ)
187 -      AG=XG(JJ); AS1=XS1(JJ); AS2=XS2(JJ); AS3=XS3(JJ)
188 -      AS4=XS4(JJ)
189 -      ASQ=XNG(1,JJ); ACUBE=XNG(2,JJ)
190 -      HA1=AH1(JJ); HA2=AH2(JJ); HA3=AH3(JJ); HA4=AH4(JJ);
191 -      HA5=AH5(JJ); HA6=AH6(JJ); FF=FUND(JJ)
192 -      HP1=TH1(JJ)/DTR; HP2=TH2(JJ)/DTR; HP3=TH3(JJ)/DTR
193 -      HP4=TH4(JJ)/DTR; HP5=TH5(JJ)/DTR; HP6=TH6(JJ)/DTR
194 -      F11=W11(JJ); F12=W12(JJ); F21=W21(JJ); F22=W22(JJ)
195 -      F31=W31(JJ); F32=W32(JJ)
196 -      F41=W41(JJ); F42=W42(JJ)
197 -      P11=T11(JJ)/DTR; P12=T12(JJ)/DTR; P21=T21(JJ)/DTR
198 -      P22=T22(JJ)/DTR; P31=T31(JJ)/DTR; P32=T32(JJ)/DTR
199 -      P41=T41(JJ)/DTR; P42=T42(JJ)/DTR
200 - 70 CONTINUE
201 -      INPUT
202 -      IF ((KSAME.NE.0).AND.(KSAME.GT.JJ))
203 - $OUTPUT 'KSAME TOO LARGE',KSAME,JJ ; STOP
204 -      IF (KSAME.EQ.0) GO TO 75
205 -      KK=KSAME
206 -      IXA(JJ)=IXA(KK)
207 -      KG(JJ)=KG(KK); KNG(JJ)=KNG(KK); BMEAN(JJ)=BMEAN(KK)
208 -      KHARM(JJ)=KHARM(KK)
209 -      SDE(JJ)=SDE(KK) ; XG(JJ)=XG(KK)
210 -      XNG(1,JJ)=XNG(1,KK) ; XNG(2,JJ)=XNG(2,KK)
211 -      XS1(JJ)=XS1(KK); XS2(JJ)=XS2(KK); XS3(JJ)=XS3(KK)
212 -      XS4(JJ)=XS4(KK)
213 -      W11(JJ)=W11(KK); W12(JJ)=W12(KK); W21(JJ)=W21(KK)
214 -      W22(JJ)=W22(KK); W31(JJ)=W31(KK); W32(JJ)=W32(KK)
215 -      W41(JJ)=W41(KK) ; W42(JJ)=W42(KK)
216 -      T11(JJ)=T11(KK); T12(JJ)=T12(KK); T21(JJ)=T21(KK)
217 -      T22(JJ)=T22(KK); T31(JJ)=T31(KK); T32(JJ)=T32(KK)
218 -      T41(JJ)=T41(KK) ; T42(JJ)=T42(KK)
219 -      AH1(JJ)=AH1(KK); AH2(JJ)=AH2(KK); AH3(JJ)=AH3(KK)
220 -      AH4(JJ)=AH4(KK); AH5(JJ)=AH5(KK); AH6(JJ)=AH6(KK)
221 -      FUND(JJ)=FUND(KK)
222 -      TH1(JJ)=TH1(KK); TH2(JJ)=TH2(KK); TH3(JJ)=TH3(KK)
223 -      TH4(JJ)=TH4(KK); TH5(JJ)=TH5(KK); TH6(JJ)=TH6(KK)
224 -      GO TO 100
225 - 75 CONTINUE
226 -      IXA(JJ)=IX
227 -      KG(JJ)=IG; KNG(JJ)=ING; BMEAN(JJ)=AMEAN; SDE(JJ)=STDEV
228 -      KHARM(JJ)=IHARM
229 -      XG(JJ)=AG; XS1(JJ)=AS1; XS2(JJ)=AS2; XS3(JJ)=AS3
230 -      XS4(JJ)=AS4
231 -      XNG(1,JJ)=ASQ; XNG(2,JJ)=ACUBE
232 -      AH1(JJ)=HA1; AH2(JJ)=HA2; AH3(JJ)=HA3; AH4(JJ)=HA4;
233 -      AH5(JJ)=HA5; AH6(JJ)=HA6; FUND(JJ)=FF
234 -      TH1(JJ)=HP1*DTR; TH2(JJ)=HP2*DTR; TH3(JJ)=HP3*DTR
235 -      TH4(JJ)=HP4*DTR; TH5(JJ)=HP5*DTR; TH6(JJ)=HP6*DTR
236 -      W11(JJ)=F11; W12(JJ)=F12; W21(JJ)=F21; W22(JJ)=F22
237 -      W31(JJ)=F31; W32(JJ)=F32
238 -      W41(JJ)=F41 ; W42(JJ)=F42

```

```

239 -      T11(JJ)=P11*DTR;T12(JJ)=P12*DTR;T21(JJ)=P21*DTR
240 -      T22(JJ)=P22*DTR;T31(JJ)=P31*DTR;T32(JJ)=P32*DTR
241 -      T41(JJ)=P41*DTR ; T42(JJ)=P42*DTR
242 -      100 CONTINUE
243 - C
244 - C
245 -      IF (NOLIST.EQ.0) GO TO 170
246 -      WRITE (108,3000)
247 -      3000 FORMAT (/, 'PARAMETERS FOR ALL SERIES TO BE CREATED ',
248 -      $ 'FOLLOW: ', /)
249 - C
250 - C
251 -      DO 150 JJ=1, NSERIES
252 -      WRITE (108,3200) JJ
253 -      3200 FORMAT (/, 'S E R I E S ', I2, ' :')
254 -      WRITE (108,3500) KG(JJ), BMEAN(JJ), SDE(JJ), IXA(JJ),
255 -      $ KNG(JJ), XNG(1, JJ),
256 -      $ XNG(2, JJ), KHARM(JJ), FUND(JJ), AH1(JJ), AH2(JJ), AH3(JJ),
257 -      $ AH4(JJ), AH5(JJ), AH6(JJ), TH1(JJ)*RTD, TH2(JJ)*RTD,
258 -      $ TH3(JJ)*RTD, TH4(JJ)*RTD, TH5(JJ)*RTD, TH6(JJ)*RTD,
259 -      $ XS1(JJ), XS2(JJ), XS3(JJ), XS4(JJ),
260 -      $ W11(JJ), T11(JJ)*RTD, W12(JJ), T12(JJ)*RTD,
261 -      $ W21(JJ), T21(JJ)*RTD, W22(JJ), T22(JJ)*RTD,
262 -      $ W31(JJ), T31(JJ)*RTD, W32(JJ), T32(JJ)*RTD,
263 -      $ W41(JJ), T41(JJ)*RTD, W42(JJ), T42(JJ)*RTD
264 -      3500 FORMAT ('GAUSSIAN NOISE: IG=', I, ' AMEAN=', G8.2, ' STDEV=',
265 -      $ G8.2, ' SEED=', I10, /,
266 -      $ 'NON-G. NOISE: ING=', I, ' ASQ=', G8.2, ' ACUBE=', G8.2, /,
267 -      $ 'HARMONIC SERIES: IHARM=', I, ' FF=', G8.2, /,
268 -      $ 'HA1=', G8.2, 'HA2=', G8.2, 'HA3=', G8.2, 'HA4=', G8.2, 'HA5=',
269 -      $ G8.2, 'HA6=', G8.2, /,
270 -      $ 'HP1=', G8.2, 'HP2=', G8.2, 'HP3=', G8.2, 'HP4=', G8.2, 'HP5=',
271 -      $ G8.2, 'HP6=', G8.2, /,
272 -      $ 'SUM OF SIN*SIN TERMS: ', /, ' AS1=', G8.2, ' AS2=', G8.2,
273 -      $ ' AS3=', G8.2, ' AS4=', G8.2, /,
274 -      $ ' F11=', G9.3, 5X, 'P11=', F6.1, 8X, 'F12=', G9.3, 5X, 'P12=', F6.1
275 -      $, /,
276 -      $ ' F21=', G9.3, 5X, 'P21=', F6.1, 8X, 'F22=', G9.3, 5X, 'P22=', F6.1
277 -      $, /,
278 -      $ ' F31=', G9.3, 5X, 'P31=', F6.1, 8X, 'F32=', G9.3, 5X, 'P32=', F6.1
279 -      $, /,
280 -      $ ' F41=', G9.3, 5X, 'P41=', F6.1, 8X, 'F42=', G9.3, 5X, 'P42=', F6.1
281 -      $)
282 -      150 CONTINUE
283 - C
284 - C
285 -      170 CONTINUE
286 -      WRITE (108,3700)
287 -      3700 FORMAT ('DO YOU WISH TO RE-ENTER INPUT LOOP?', /,
288 -      $ 'VARIABLES WILL REMAIN AS SET UNLESS CHANGED.', /,
289 -      $ 'ANSWER YES OR NO.')
290 -      READ (105,4000) NOYES
291 -      4000 FORMAT (A4)
292 -      IF ((NOYES.NE.4HYES ).AND.(NOYES.NE.4HNO )) GO TO 170
293 -      IF (NOYES.EQ.4HYES ) IDEC=1 ; GO TO 50
294 -      OUTPUT 'INPUT COMPLETED'
295 -      RETURN
296 -      END

```

```

1 -      SUBROUTINE HARMONICS
2 - C FOR PROGRAM GENRAN03 TO GENERATE HARMONICS
3 -      DOUBLE PRECISION T,PI,TWOPI,DFLS,DFLTH,FL,DFMS,DFMTH
4 -      DOUBLE PRECISION
5 -      $AH1(20),AH2(20),AH3(20),AH4(20),AH5(20),AH6(20),
6 -      $TH1(20),TH2(20),TH3(20),TH4(20),TH5(20),TH6(20)
7 -      COMMON B1(2560),KG(20),KNG(20),XG(20),XNG(2,20),FUND(20),
8 -      $AH1,AH2,AH3,AH4,AH5,AH6,
9 -      $TH1,TH2,TH3,TH4,TH5,TH6,
10 -      $IDIM,IFL,TWOPI,LENGTH,MZ
11 - C
12 -      DO 200 K=1,IDIM
13 -      MZ=MZ+1
14 -      DFMS=DFLOAT(MZ) ; DFMTH=DFLOAT(LENGTH)
15 -      T=TWOPI*DFMS/DFMTH
16 -      N=IFL ; FL=FUND(IFL)
17 -      B1(K)=B1(K)+
18 -      $AH1(N)*DSIN( FL*T+TH1(N))+AH2(N)*DSIN( 2.*FL*T+TH2(N))+
19 -      $AH3(N)*DSIN(3.*FL*T+TH3(N))+AH4(N)*DSIN(4.*FL*T+TH4(N))+
20 -      $AH5(N)*DSIN(5.*FL*T+TH5(N))+AH6(N)*DSIN(6.*FL*T+TH6(N))
21 - 200 CONTINUE
22 - C
23 -      RETURN
24 -      END

```

```

1 -      SUBROUTINE NOISE
2 - C FOR PROGRAM GENRAN03 TO GENERATE GAUSSIAN AND NON-G. NOISE
3 -      DOUBLE PRECISION T,PI,TWOPI,DFLS,DFLTH,FL,DFMS,DFMTH
4 -      DOUBLE PRECISION
5 -      $AH1(20),AH2(20),AH3(20),AH4(20),AH5(20),AH6(20),
6 -      $TH1(20),TH2(20),TH3(20),TH4(20),TH5(20),TH6(20)
7 -      COMMON B1(2560),KG(20),KNG(20),XG(20),XNG(2,20),FUND(20),
8 -      $AH1,AH2,AH3,AH4,AH5,AH6,
9 -      $TH1,TH2,TH3,TH4,TH5,TH6,
10 -     $IDIM,IFL,TWOPI,LENGTH,MZ
11 -     IF ((KG(IFL).EQ.0).OR.(KNG(IFL).EQ.0)) GO TO 250
12 - C
13 -     DO 200 K=1,IDIM
14 -     200 B1(K)=XG(IFL)*B1(K)+XNG(1,IFL)*B1(K)**2+
15 -     $XNG(2,IFL)*B1(K)**3
16 - C
17 -     GO TO 500
18 -     250 CONTINUE
19 -     IF (KG(IFL).EQ.0) GO TO 350
20 - C
21 -     DO 300 K=1,IDIM
22 -     300 B1(K)=XG(IFL)*B1(K)
23 - C
24 -     GO TO 500
25 -     350 CONTINUE
26 - C
27 -     DO 400 K=1,IDIM
28 -     400 B1(K)=XNG(1,IFL)*B1(K)**2+XNG(2,IFL)*B1(K)**3
29 - C
30 -     500 CONTINUE
31 -     RETURN
32 -     END

```

- PROGRAM ORDAT -

```

1 - C          <<<<<<  O R D A T S  >>>>>>
2 - C PROGRAM TO READ CONSECUTIVE FILES PRODUCED BY FRAGTAP
3 - C AND TO CONSTRUCT LARGE RWDISC FILE
4 - C PROGRAMMER:  G. MARTINEAU   VERSION 01   3/9/77
5 - C          DIMENSION FILE(18000),IFILE(2500),ILF(4),NAMDCB(4),
6 - C          $IBUOYVAR(4)
7 - C          EQUIVALENCE (FILE,IFILE)
8 - C          MFILE=21
9 - C          CALL DPAR(NERR,MFILE,1,1,KSUM,512)
10 - C          NWDS=18000
11 - C          READ (105,1000) NDCB,LDCB
12 - C          1000 FORMAT (2G)
13 - C
14 - C
15 - C          DO 600 I=1,NDCB
16 - C          READ (105,2000) IDCBC,NSERIES,NSERIES,(ILF(J),J=1,NSERIES)
17 - C          2000 FORMAT (2G,NG)
18 - C READ IN DATA
19 - C          CALL BUFFER IN(IDCBC,1,FILE,NWDS,ISTAT,INWDS)
20 - X          OUTPUT IDCBC,(IFILE(K),K=1,9)
21 - C          IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 1
22 - C          IF (INWDS.NE.9) OUTPUT INWDS,IDCB ; STOP 10
23 - C          NUMVAR=IFILE(4)
24 - C          JCYC=IFILE(5)
25 - X          OUTPUT JCYC
26 - C          IF (NUMVAR.NE.NSERIES) OUTPUT IDCBC,NUMVAR,NSERIES ;
27 - C          $STOP 'DISCREPANCY IN NO. OF VARIABLES'
28 - C
29 - C
30 - C
31 - X          OUTPUT NUMVAR
32 - C          DO 150 N=1,NSERIES
33 - C          150 IBUOYVAR(N)=IFILE(N+5)
34 - C
35 - C          DO 200 K=1,NSERIES
36 - C          LF=ILF(K)
37 - C          CALL BUFFER IN(IDCBC,1,FILE(10),NWDS,ISTAT,INWDS)
38 - C          IF (INWDS.NE.JCYC) OUTPUT N,INWDS,JCYC ; STOP 50
39 - C
40 - C          DO 170 N=1,3
41 - C          170 IFILE(N+5)=IFILE(N)
42 - C
43 - C          IFILE(9)=IBUOYVAR(K)
44 - X          OUTPUT LF,JCYC,INWDS
45 - X          WRITE (108,222) K,(IFILE(M),M=6,9),
46 - X          $(FILE(M),M=10,24)
47 - X222 FORMAT (1H , 'K=',11,2X,3A4,3X,18,/, (5G14.4))
48 - C          CALL WDISC(LF,1,FILE(6),JCYC+4)
49 - C          200 CONTINUE
50 - C
51 - C          WRITE (108,3000) IDCBC,(IFILE(J),J=6,8),NSERIES,JCYC,
52 - C          $NSERIES,(ILF(J),J=1,NSERIES),NSERIES,
53 - C          $(IBUOYVAR(J),J=1,NSERIES)
54 - C          3000 FORMAT (/, 'OUTPUT SUMMARY FOR DCB ',I2,/,T16,
55 - C          $'BUOY FILE: ',3A4,/,T12,'NO. OF SERIES: ',I1,/,T4,
56 - C          $'DATA WDS. EACH SERIES: ',G,/,T4,
57 - C          $'WRITTEN TO LOG. FILES: ',N(I2,2X),/,T11,
58 - C          $'BUOY VAR. NOS.: ',N(I2,2X),//)

```

```
59 - 600 CONTINUE
60 - C
61 - C
62 - C CONSTRUCT LABEL FILE
63 -     CALL BUFFER IN(LDCB,1,IFILE,NWDS,ISTAT,INWDS)
64 -     IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 2
65 -     CALL WDISC(21,1,IFILE,INWDS)
66 -     WRITE (108,4000) INWDS
67 - 4000 FORMAT (/ ,1H ,I4, ' WORDS WRITTEN IN LABEL L. F. NO. 21' )
68 -     CALL CDISC
69 -     STOP 'NORMAL END'
70 -     END
```

- PROGRAM RBPLOT05 -

```

1 - C          <<<<<<< R B P L O T O 5 >>>>>>>
2 - C PROGRAM TO PLOT ROTARY BICOHERENCES GENERATED BY PROGRAM
3 - C BIVECO5 AND TRANSMITTED BY CONSECUTIVE DISC FILES.
4 - C PROGRAMMER:  GERARD H. MARTINEAU
5 - C ORIGINATOR:  MELBOURNE G. BRISCOE
6 - C      DATE:  OCT. 18, 1977
7 - C*****
8 -      DIMENSION DATA(200,130),IBUF(1000),CONLV(15),KARD(20),
9 -      $IDENT1(9),IDENT2(9),ICLK(4),LABL(33),CONFLV(15),
10 -     $INAMES(18),IVAR(6),EXCH(200)
11 -     MD1=200 ; MD2=130
12 -     IPASS=1
13 -     NAMELIST ISTORE1,ISTORE2,NDECPL,CMPPT,IBUOY,IFOURIER,
14 -     $ISTORE3,ISTORE4,WIDTH
15 -     NDECPL=IBUOY=IFOURIER=1
16 -     ISTORE1=1 ; ISTORE2=2 ; ISTORE3=3 ; ISTORE4=4
17 -     CMPPT=-1. ; WIDTH=8.75
18 -     ISW=1 ; NVEC=2 ; KPASS=1
19 -     INPUT
20 -     READ (105,1111) KARD
21 - 1111 FORMAT (20A4)
22 -     WRITE (108,1122) KARD
23 - 1122 FORMAT (X,20A4)
24 -     DECODE (80,3333,KARD) IW1,IW2,IW3,LPIECE,SAMPSEC,FREQTIC
25 - 3333 FORMAT (6G)
26 -     IF ((IW1.EQ.IW2).AND.(IW1.EQ.IW3).AND.(IW2.EQ.IW3))
27 -     $NVEC=1
28 -     IF ((NVEC.EQ.2).AND.(IW1.NE.IW2)) ISW=3
29 -     LPHALF=LPIECE/2 ; LPATHE=LPIECE/8
30 -     IF (CMPPT.LT.0.) CMPPT=WIDTH*2.54/LPHALF
31 -     LPQU=LPIECE/4 ; LP38=LPATHE*3
32 -     LP58=LPATHE*5 ; LP34=LPATHE*6
33 -     READ (105,1111) KARD
34 -     WRITE (108,1122) KARD
35 -     DECODE (80,2222,KARD) NCONLV,NCONLV,(CONLV(I),I=1,NCONLV)
36 -     $,NCONLV,(CONFLV(I),I=1,NCONLV)
37 - 2222 FORMAT (G,NG,NG)
38 -     READ (105,1111) KARD
39 -     WRITE (108,1122) KARD
40 -     DECODE (80,4444,KARD) IDENT1,IDENT2
41 - 4444 FORMAT (9A4,9A4)
42 - C
43 - C
44 - C      30 CONTINUE
45 - C READ ROTARY BICOHERENCES INTO ARRAY DATA(200,130)
46 - C FIRST SET ARRAY DATA TO -999.
47 - C
48 - C
49 - C      DO 50 J=1,MD2
50 - C
51 - C      DO 50 I=1,MD1
52 - C      50 DATA(I,J)=-999.
53 - C
54 - C
55 - C      DO 60 ISQ=1,16
56 - C      CALL BIAS(IPASS)
57 - X      OUTPUT IBIASR,IBIASC
58 - C      IF((IPASS.EQ.1).AND.(KPASS.EQ.1))ISTORED=ISTORE1;GO TO 55

```

```

59 -      IF((IPASS.EQ.2).AND.(KPASS.EQ.1))ISTORED=ISTORE2;GO TO 55
60 -      IF((IPASS.EQ.1).AND.(KPASS.EQ.2))ISTORED=ISTORE3;GO TO 55
61 -      IF((IPASS.EQ.2).AND.(KPASS.EQ.2))ISTORED=ISTORE4;GO TO 55
62 -      55 READ (ISTORED) ((DATA(IBIASR+I,IBIASC+J),I=1,LPATHE),
63 -      $J=1,LPATHE)
64 - C      WRITE (108,5353) ((DATA(IBIASR+I,IBIASC+J),I=1,LPATHE),
65 - C      $J=1,LPATHE)
66 - C5353 FORMAT (X,5G14.4)
67 -      60 CONTINUE
68 - C
69 - C
70 - C REARRANGE DATA MATRIX IF NECESSARY
71 -      IF (KPASS.NE.2) GO TO 67
72 - C
73 - C
74 -      DO 66 J=1,LPHALF/2
75 - C
76 -      DO 61 I=1,MD1
77 -      61 EXCH(I)=DATA(I,J)
78 - C
79 -      DO 63 I=1,MD1
80 -      63 DATA(I,J)=DATA(I,LPHALF+1-J)
81 - C
82 -      DO 65 I=1,MD1
83 -      65 DATA(I,LPHALF+1-J)=EXCH(I)
84 - C
85 -      66 CONTINUE
86 - C
87 - C
88 -      67 CONTINUE
89 -      IF ((IPASS.EQ.2).AND.(KPASS.EQ.1)) GO TO 360
90 -      IF ((IPASS.EQ.1).AND.(KPASS.EQ.2)) GO TO 190
91 -      IF ((IPASS.EQ.2).AND.(KPASS.EQ.2)) GO TO 195
92 - C READ FOURIER LABELS
93 - C
94 - C
95 -      68 DO 70 I=1,6
96 -      READ (ISTORE1) (INAMES(J),J=3*I-2,3*I),IVAR(I)
97 -      70 CONTINUE
98 - C
99 - C
100 -      READ (ISTORE1) MF
101 -      READ (ISTORE1) LABL
102 -      CALL PLOTS(IBUF,-1000)
103 -      CALL PLOT(0.,0.5,-3)
104 -      WRITE (108,1000) (CONLV(I),I=1,NCONLV)
105 -      1000 FORMAT (/, 'CONTOUR LEVELS:',/, (1H ,5G13.3,/))
106 -      CALL DIMWH(DATA,MD1,MD2)
107 -      CALL FLAGZ(-999.0)
108 -      CALL NOLABL
109 -      DEL=CMPT/2.54
110 -      AXLEN=LPHALF*DEL
111 -      SAMPHR=SAMPSEC/3600.
112 -      DFTOT=1./(2.*SAMPHR)
113 -      DISTIC=AXLEN*FREQTIC/DFTOT
114 -      AYLEN=(INT(AXLEN/DISTIC)+1)*DISTIC
115 -      AYHLEN=AXLEN/2.+0.25
116 -      FIRSTV=-(AYLEN/DISTIC)*FREQTIC
117 - C PLOT IDENTIFICATION
118 -      EXPLOT=AXLEN+AYHLEN

```

```

119 -      WYEPLT=AXLEN
120 -      IF (NVEC.EQ.1) GO TO 100
121 -      CALL SYMBOL(EXPLOT+8.3,WYEPLT-0.8,0.315,
122 -      $'ROTARY CROSS BICOHERENCE',-90.,24)
123 -      GO TO 120
124 - 100 CALL SYMBOL(EXPLOT+8.3,WYEPLT-0.9,0.315,
125 -      $'ROTARY AUTO BICOHERENCE',-90.,23)
126 - 120 CONTINUE
127 -      IF (IBUOY.EQ.0) GO TO 125
128 -      CALL SYMBOL(EXPLOT+7.8,WYEPLT-1.7,0.210,
129 -      $'COMPONENTS OF VECTOR SERIES',-90.,27)
130 -      CALL SYMBOL(EXPLOT+7.4,WYEPLT-1.6,0.175,
131 -      $'ORIGINAL FILE NAMES:',-90.,20)
132 -      CALL SYMBOL(EXPLOT+7.4,WYEPLT-5.7,0.175,
133 -      $'VARIABLE NUMBERS:',-90.,17)
134 -      CALL SYMBOL(EXPLOT+7.1,WYEPLT-0.5,0.14,'FREQUENCY 1:',
135 -      $-90.,12)
136 -      CALL SYMBOL(EXPLOT+6.5,WYEPLT-0.5,0.14,'FREQUENCY 2:',
137 -      $-90.,12)
138 -      CALL SYMBOL(EXPLOT+5.9,WYEPLT-0.5,0.14,'FREQUENCY 3:',
139 -      $-90.,12)
140 - 125 CONTINUE
141 -      IF ((MF.NE.21).OR.(IFOURIER.EQ.0)) GO TO 130
142 -      CALL SYMBOL(EXPLOT+5.1,WYEPLT-2.9,0.175,
143 -      $'PROCESSING HISTORY',-90.,18)
144 -      CALL SYMBOL(EXPLOT+4.8,WYEPLT-0.7,0.105,LABL,-90.,72)
145 -      CALL SYMBOL(EXPLOT+4.5,WYEPLT-1.3,.105,LABL(19),-90.,60)
146 - 130 CONTINUE
147 - C
148 - C
149 -      IF (IBUOY.EQ.0) GO TO 150
150 -      DO 135 I=1,3
151 -      CALL SYMBOL(EXPLOT+7.1-0.6*(I-1),WYEPLT-2.5,0.14,
152 -      $INAMES(6*I-5),-90.,12)
153 -      CALL SYMBOL(EXPLOT+6.85-0.6*(I-1),WYEPLT-2.5,0.14,
154 -      $INAMES(6*I-2),-90.,12)
155 -      FVAR=FLOAT(IVAR(2*I-1))
156 -      CALL NUMBER(EXPLOT+7.1-0.6*(I-1),WYEPLT-7.1,0.14,FVAR,
157 -      $-90.,-1)
158 -      FVAR=FLOAT(IVAR(2*I))
159 - 135 CALL NUMBER(EXPLOT+6.85-0.6*(I-1),WYEPLT-7.1,0.14,FVAR,
160 -      $-90.,-1)
161 - C
162 - C
163 - 150 CONTINUE
164 -      CALL SYMBOL(EXPLOT+1.15,WYEPLT-1.0,0.210,IDENT1,-90.,36)
165 -      CALL SYMBOL(EXPLOT+0.85,WYEPLT-1.0,0.210,IDENT2,-90.,36)
166 -      CALL TODAY(ICLK)
167 -      CALL SYMBOL(EXPLOT+3.0,WYEPLT-0.6,0.175,'TIME OF PLOT:',
168 -      $-90.,13)
169 -      CALL SYMBOL(EXPLOT+2.7,WYEPLT-0.3,0.175,ICLK,-90.,16)
170 -      CALL SYMBOL(EXPLOT+3.8,WYEPLT-3.8,0.140,
171 -      $'CONTOUR LEVELS AND PERCENT CONFIDENCE',-90.,37)
172 -      BIASX=BIASY=0.
173 - C
174 - C
175 -      IF (NCONLV.GT.10) NCONLV=10
176 -      DO 185 I=1,NCONLV
177 -      IF (I.EQ.6)          BIASX=0. ; BIASY=BIASY-3.0
178 -      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLT-4.0+BIASY,0.210,

```

```

179 -      $CONLV(1),-90.,3)
180 -      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLT-5.4+BIASY,0.210,
181 -      $CONFLV(1),-90.,1)
182 -      BIASX=BIASX-0.4
183 -      185 CONTINUE
184 -      190 IF (KPASS.EQ.1) GO TO 200
185 - C ESTABLISH NEW PLOT ORIGIN
186 -      CALL PLOT(0.,-0.25,-3)
187 - C Y-AXIS WITH TIC MARKS ONLY, STARTING AT W1=0
188 -      CALL AXWJS(AYLLEN,0.,' ',1,AXLEN,90.,0.,1.,DISTIC,-1,0,0,
189 -      $0.12,.7,.0001,.14)
190 - C Y-AXIS STARTING AT MAX W1
191 -      CALL AXWJS(AYLLEN+AYHLEN,0.,'FREQ2 (CPH)',-11,AXLEN,90.,
192 -      $0.,FREQTIC,DISTIC,NDECPL,0,0)
193 - C Y-AXIS STARTING AT MIN W1
194 -      CALL AXWJS(0.,0.,'FREQ2 (CPH)',11,AXLEN,90.,0.,FREQTIC,
195 -      $DISTIC,NDECPL,0,0)
196 - C X-AXIS
197 -      FIRSTV=-(AYLLEN/DISTIC)*FREQTIC
198 -      CALL AXDRAW(0.,AXLEN+0.2,'FREQ1 (CPH)',11,AYHLEN+AYLLEN,
199 -      $0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
200 - C DRAW BOUNDARIES
201 -      CALL PLOT(AYLLEN,0.,3)
202 -      CALL PLOT(AYLLEN-AXLEN,AXLEN,2)
203 -      CALL PLOT(AYLLEN,AXLEN,2)
204 -      CALL PLOT(AYLLEN+AXLEN/2.,AXLEN/2.,2)
205 -      CALL PLOT(AYLLEN,0.,2)
206 -      CALL GRID(AYLLEN-AXLEN,AXLEN,DEL,-DEL)
207 -      CALL WHCNTR(DATA,1,LPQU+LPHALF+1,1,LPHALF,NCONLV,CONLV)
208 -      CALL PLOT(0.,0.,999)
209 -      192 CONTINUE
210 - C BEGIN PROCEDURE FOR PLOTTING NEGATIVE SUM FREQUENCIES
211 -      IPASS=2
212 -      GO TO 30
213 -      195 CONTINUE
214 - C ESTABLISH NEW PLOT ORIGIN
215 -      CALL PLOT(0.,0.6,-3)
216 - C Y-AXIS WITH TIC MARKS ONLY, STARTING AT W1=0
217 -      CALL AXWJS(AYHLEN,AXLEN,' ',-1,AXLEN,-90.,0.,1.,DISTIC,
218 -      $-1,0,0,0.12,.7,.0001,.14)
219 - C Y-AXIS STARTING AT MAX W1
220 -      FIRSTV=0.
221 -      CALL AXWJS(AYHLEN+AYLLEN,AXLEN,'FREQ2 (CPH)',11,AXLEN,
222 -      $-90.,FIRSTV,FREQTIC,DISTIC,NDECPL,0,0)
223 - C Y-AXIS STARTING AT MIN W1
224 -      CALL AXWJS(0.,AXLEN,'FREQ2 (CPH)',-11,AXLEN,-90.,FIRSTV,
225 -      $FREQTIC,DISTIC,NDECPL,0,0)
226 - C X-AXIS
227 -      FIRSTV=-(AYHLEN/DISTIC)*FREQTIC
228 -      CALL AXDRAW(0.,-0.2,'FREQ1 (CPH)',-11,AYHLEN+AYLLEN,
229 -      $0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
230 - C DRAW BOUNDARIES
231 -      CALL PLOT(AYHLEN,0.,3)
232 -      CALL PLOT(AYHLEN-AXLEN/2.,AXLEN/2.,2)
233 -      CALL PLOT(AYHLEN,AXLEN,2)
234 -      CALL PLOT(AYHLEN+AXLEN,0.,2)
235 -      CALL PLOT(AYHLEN,0.,2)
236 -      CALL GRID(AYHLEN-AXLEN/2.,AXLEN-DEL,DEL,-DEL)
237 -      CALL WHCNTR(DATA,1,LPQU+LPHALF+1,1,LPHALF,
238 -      $NCONLV,CNVLV)

```

```

239 -      CALL PLOT(0.,0.,3)
240 -      GO TO 400
241 - C
242 - C
243 -      200 CONTINUE
244 - C X-AXIS WITH TIC MARKS ONLY, STARTING AT ORIGIN
245 -      CALL AXWJS(AXLEN,AXLEN,' ', -1,AXLEN,-90.,0.,1.,DISTIC,
246 -      $-1,0,0.,.12,.7,.0001,.14)
247 - C X-AXIS STARTING ABOVE MAX Y VALUE
248 -      CALL AXDRAW(AXLEN+AYHLEN,AXLEN,'FREQ1 (CPH)',11,
249 -      $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
250 - C X-AXIS STARTING BELOW MIN Y VALUE
251 -      CALL AXDRAW(AXLEN-AYLLEN,AXLEN,'FREQ1 (CPH)',-11,
252 -      $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
253 - C Y-AXIS FOR TWO SERIES
254 -      CALL AXDRAW(AXLEN-AYLLEN,AXLEN,'FREQ2 (CPH)',11,
255 -      $2*AYLLEN,0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
256 - C DRAW BOUNDARIES
257 -      CALL PLOT(0.,0.,3)
258 -      CALL PLOT(AXLEN,AXLEN,2)
259 -      CALL PLOT(1.5*AXLEN,AXLEN/2.,2)
260 -      CALL PLOT(AXLEN,0.,2)
261 -      CALL PLOT(0.,0.,2)
262 -      220 CALL GRID(0.,AXLEN-DEL,DEL,-DEL)
263 -      CALL WHCNTR(DATA,1,LPHALF+LPQU+1,1,LPHALF,NCONLV,CONLV)
264 -      230 CALL PLOT(0.,0.,999)
265 -      250 CONTINUE
266 - C BEGIN PROCEDURE FOR PLOTTING NEGATIVE SUM FREQUENCIES
267 -      IPASS=2
268 -      GO TO 30
269 -      360 CONTINUE
270 -      AYHLEN=(INT(AXLEN/(2.*DISTIC))+1)*DISTIC
271 - C ESTABLISH NEW PLOT ORIGIN
272 -      CALL PLOT(0.,0.,-3)
273 - C X-AXIS WITH TIC MARKS ONLY, STARTING AT W2=0
274 -      CALL AXWJS(AYHLEN,0.,' ',1,AXLEN,90.,0.,1.,DISTIC,-1,
275 -      $0,0,.12,.7,.0001,.14)
276 -      YINT=AINT(AXLEN/DISTIC)
277 -      FIRSTV=-FREQTIC*YINT
278 -      YLTH=DISTIC*YINT
279 - C X-AXIS STARTING AT MAX W2
280 -      CALL AXDRAW(AYHLEN+AYLLEN,YLTH,'FREQ1 (CPH)',11,YLTH,
281 -      $-90.,FIRSTV,FREQTIC,DISTIC,NDECPL)
282 - C X-AXIS STARTING AT MIN W2
283 -      CALL AXDRAW(0.,YLTH,'FREQ1 (CPH)',-11,YLTH,-90.,FIRSTV,
284 -      $FREQTIC,DISTIC,NDECPL)
285 - C Y-AXIS
286 -      FIRSTV=-(AYHLEN/DISTIC)*FREQTIC
287 -      CALL AXDRAW(0.,AXLEN+0.2,'FREQ2 (CPH)',11,AYHLEN+AYLLEN,
288 -      $0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
289 - C DRAW BOUNDARIES
290 -      CALL PLOT(AYHLEN,0.,3)
291 -      CALL PLOT(AYHLEN-AXLEN/2.,AXLEN/2.,2)
292 -      CALL PLOT(AYHLEN,AXLEN,2)
293 -      CALL PLOT(AYHLEN+AXLEN,AXLEN,2)
294 -      CALL PLOT(AYHLEN,0.,2)
295 -      CALL GRID(AYHLEN-AXLEN/2.,AXLEN,DEL,-DEL)
296 -      CALL WHCNTR(DATA,1,LPQU+LPHALF+1,1,LPHALF,NCONLV,CONLV)
297 -      400 CALL PLOT(0.,-.25,999)
298 - C PLOT POSITIVE SUM FREQUENCIES FOR /W2/>/W1/ IF ISW=3

```

```

299 -      IF ((ISW.EQ.3).AND.(KPASS.EQ.1)) KPASS=2 ; IPASS=1 ;
300 -      $ GO TO 30
301 -      STOP 'NORMAL PROGRAM COMPLETION'
302 - C*****
303 - C*****
304 -      SUBROUTINE BIAS(IP)
305 - C INTERNAL SUBROUTINE FOR PROGRAM RBPLOT
306 -      4 GO TO (6,8),IP
307 -      6 GO TO (10,15,15,10,20,25,25,20,10,15,15,10,25,30,30,35),
308 -      $ISQ
309 -      8 GO TO (30,40,40,30,35,50,50,35,30,40,40,30,50,60,60,20),
310 -      $ISQ
311 -      10 CALL ASN(IBIASR,1) ; GO TO 65
312 -      15 CALL ASN(IBIASR,2) ; GO TO 65
313 -      20 CALL ASN(IBIASR,3) ; GO TO 65
314 -      25 CALL ASN(IBIASR,4) ; GO TO 65
315 -      30 CALL ASN(IBIASR,5) ; GO TO 65
316 -      35 CALL ASN(IBIASR,6) ; GO TO 65
317 -      40 CALL ASN(IBIASR,7) ; GO TO 65
318 -      50 CALL ASN(IBIASR,8) ; GO TO 65
319 -      60 CALL ASN(IBIASR,9) ; GO TO 65
320 -      65 GO TO (67,68),IP
321 -      67 GO TO (70,70,75,75,75,75,80,80,80,80,85,85,85,85,80,85),
322 -      $ISQ
323 -      68 GO TO (85,85,80,80,80,80,75,75,75,75,70,70,70,70,75,70),
324 -      $ISQ
325 -      70 CALL ASN(IBIASC,6) ; RETURN
326 -      75 CALL ASN(IBIASC,5) ; RETURN
327 -      80 CALL ASN(IBIASC,4) ; RETURN
328 -      85 CALL ASN(IBIASC,2) ; RETURN
329 -      90 RETURN
330 - C*****
331 -      SUBROUTINE ASN(IB,IS)
332 -      5 GO TO (10,20,30,40,50,60,70,80,90),IS
333 -      10 IB=LPHALF+1 ; RETURN
334 -      20 IB=LP38 ; RETURN
335 -      30 IB=LP58+1 ; RETURN
336 -      40 IB=LPQU ; RETURN
337 -      50 IB=LPTHE ; RETURN
338 -      60 IB=0 ; RETURN
339 -      70 IB=LPQU+1 ; RETURN
340 -      80 IB=LP38+1 ; RETURN
341 -      90 IB=LPHALF+1 ; RETURN
342 -      95 RETURN
343 -      END

```

- PROGRAM RBPLOT06 -

```

1 - C          <<<<<< R B P L O T O 6 >>>>>>
2 - C PROGRAM TO PLOT ROTARY BICOHERENCES GENERATED BY PROGRAM
3 - C BIVEC05 AND TRANSMITTED BY CONSECUTIVE DISC FILES.
4 - C CREATES SINGLE 30 IN CALCOMP PLOT OVER ENTIRE DOMAIN
5 - C PROGRAMMER:  GERARD H. MARTINEAU
6 - C ORIGINATOR:  MELBOURNE G. BRISCOE
7 - C      DATE:   NOV. 3, 1977
8 - C *****
9 -      DIMENSION DATA(200,130),IBUF(1000),CONLV(15),KARD(20),
10 -      $IDENT1(9),IDENT2(9),ICLK(4),LABL(33),CONFLV(15),
11 -      $INAMES(18),IVAR(6),EXCH(200),XDATA(130,200),IFREQ(3)
12 -      EQUIVALENCE (DATA,XDATA)
13 -      MD1=200 ; MD2=130
14 -      IPASS=1
15 -      NAMELIST ISTORE1,ISTORE2,NDECPL,CMPPT,IBUOY,IFOURIER,
16 -      $ISTORE3,ISTORE4,ISCR,KOMPACT,WIDTH
17 -      NDECPL=IBUOY=IFOURIER=1
18 -      ISTORE1=1 ; ISTORE2=2 ; ISTORE3=3 ; ISTORE4=4 ; ISCR=5
19 -      CMPPT=-1. ; WIDTH=24.
20 -      ISW=1 ; NVEC=2 ; KPASS=1 ; KOMPACT=0
21 -      INPUT
22 -      READ (105,1111) KARD
23 - 1111 FORMAT (20A4)
24 -      WRITE (108,1122) KARD
25 - 1122 FORMAT (X,20A4)
26 -      DECODE (80,3333,KARD) IW1,IW2,IW3,LPIECE,SAMPSEC,FREQTIC
27 - 3333 FORMAT (6G)
28 -      IF (CMPPT.LT.0.) CMPPT=WIDTH*2.54/LPIECE
29 -      IF ((IW1.EQ.IW2).AND.(IW1.EQ.IW3).AND.(IW2.EQ.IW3))
30 -      $NVEC=1
31 -      IF ((NVEC.EQ.2).AND.(IW1.NE.IW2)) ISW=3
32 - C CHECK FOR EXISTENCE OF DCB ASSIGNMENTS
33 -      CALL GETDCB(ISTORE1,LOCC)
34 -      CALL GETDCB(ISTORE2,LOCC)
35 -      IF (ISW.NE.3) GO TO 10
36 -      CALL GETDCB(ISTORE3,LOCC)
37 -      CALL GETDCB(ISTORE4,LOCC)
38 -      CALL GETDCB(ISCR,LOCC)
39 - 10 CONTINUE
40 -      LPHALF=LPIECE/2 ; LPATHE=LPIECE/8
41 -      LPQU=LPIECE/4 ; LP38=LPATHE*3
42 -      LP58=LPATHE*5 ; LP34=LPATHE*6
43 -      READ (105,1111) KARD
44 -      WRITE (108,1122) KARD
45 -      DECODE (80,2222,KARD) NCONLV,NCONLV,(CONLV(I),I=1,NCONLV)
46 -      $,NCONLV,(CONFLV(I),I=1,NCONLV)
47 - 2222 FORMAT (G,NG,NG)
48 -      READ (105,1111) KARD
49 -      WRITE (108,1122) KARD
50 -      DECODE (80,4444,KARD) IDENT1,IDENT2
51 - 4444 FORMAT (9A4,9A4)
52 - C
53 - C
54 -      30 CONTINUE
55 - C READ ROTARY BICOHERENCES INTO ARRAY DATA(200,130)
56 - C FIRST SET ARRAY DATA TO -999.
57 - C
58 - C

```

```

59 -      DO 50 J=1,MD2
60 - C
61 -      DO 50 I=1,MD1
62 -      50 DATA(I,J)=-999.
63 - C
64 - C
65 -      DO 60 ISQ=1,16
66 -      CALL BIAS(IPASS)
67 - X      OUTPUT IBIASR,IBIASC
68 -      IF((IPASS.EQ.1).AND.(KPASS.EQ.1))ISTORED=ISTORE1;GO TO 55
69 -      IF((IPASS.EQ.2).AND.(KPASS.EQ.1))ISTORED=ISTORE2;GO TO 55
70 -      IF((IPASS.EQ.1).AND.(KPASS.EQ.2))ISTORED=ISTORE3;GO TO 55
71 -      IF((IPASS.EQ.2).AND.(KPASS.EQ.2))ISTORED=ISTORE4;GO TO 55
72 -      55 READ (ISTORED) ((DATA(IBIASR+I,IBIASC+J),I=1,LPATHE),
73 -      $J=1,LPATHE)
74 -      60 CONTINUE
75 - C
76 - C
77 - C REARRANGE DATA MATRIX IF NECESSARY
78 -      IF (KPASS.EQ.1) GO TO 80
79 - C
80 - C
81 -      DO 66 J=1,LPHALF/2
82 - C
83 -      DO 61 I=1,MD1
84 -      61 EXCH(I)=DATA(I,J)
85 - C
86 -      DO 63 I=1,MD1
87 -      63 DATA(I,J)=DATA(I,LPHALF+1-J)
88 - C
89 -      DO 65 I=1,MD1
90 -      65 DATA(I,LPHALF+1-J)=EXCH(I)
91 - C
92 -      66 CONTINUE
93 - C
94 - C
95 -      67 CONTINUE
96 -      IF (IPASS.EQ.2) REWIND ISCR
97 -      WRITE (ISCR) ((DATA(I,J),I=1,LP34+1),J=LPHALF,1,-1)
98 - C
99 - C
100 -      DO 75 J=1,MD2
101 - C
102 -      DO 75 I=1,MD1
103 -      75 DATA(I,J)=-999.
104 - C
105 - C
106 -      REWIND ISCR
107 -      IF (IPASS.EQ.1)
108 -      $READ (ISCR) ((XDATA(I,J),J=1,LP34+1),I=2,LPHALF+1)
109 -      IF (IPASS.EQ.2)
110 -      $READ (ISCR) ((XDATA(I,J),J=1,LP34+1),I=1,LPHALF)
111 -      80 CONTINUE
112 -      IF ((IPASS.EQ.2).AND.(KPASS.EQ.1)) GO TO 200
113 -      IF ((IPASS.EQ.1).AND.(KPASS.EQ.2)) GO TO 300
114 -      IF ((IPASS.EQ.2).AND.(KPASS.EQ.2)) GO TO 400
115 - C INITIALIZING PLOT CALLS
116 -      CALL PLOTS(IBUF,-1000)
117 -      CALL PLOT(0.,0.,-3)
118 - C GENERAL SETUP FOR WHCNTR

```

```

119 -      CALL DIMWH(DATA,MD1,MD2)
120 -      CALL FLAGZ(-999.0)
121 -      CALL NOLABL
122 -      DEL=CMPT/2.54
123 -      AXL=LPHALF*DEL
124 -      SAMPHR=SAMPSEC/3600.
125 -      DFTOT=1./(2.*SAMPHR)
126 -      DISTIC=AXL*FREQTIC/DFTOT
127 -      TRYLEN=AXL/DISTIC
128 -      ALEN=(1+INT(TRYLEN))*DISTIC
129 -      IF (ABS(ALEN-TRYLEN).LE.0.1) ALEN=ALEN+DISTIC
130 -      AXLEN=2.*ALEN
131 -      FIRSTV=-(ALEN/DISTIC)*FREQTIC
132 - C PLOT TITLE AND LABELS
133 -      EXPLOT=AXLEN+1.
134 -      WYEPLT=28.
135 -      IF (NVEC.EQ.1) GO TO 100
136 -      CALL SYMBOL(EXPLOT+4.3,WYEPLT-7.9,0.420,
137 -      $'CROSS ROTARY BICOHERENCE',-90.,24)
138 -      GO TO 120
139 -      100 CALL SYMBOL(EXPLOT+4.3,WYEPLT-8.2,0.420,
140 -      $'AUTO ROTARY BICOHERENCE',-90.,23)
141 -      120 CONTINUE
142 - C
143 - C
144 -      DO 130 I=1,6
145 -      READ (ISTORE1) (INAMES(J),J=3*I-2,3*I),IVAR(I)
146 -      130 CONTINUE
147 - C
148 - C
149 -      READ (ISTORE1) MF
150 -      READ (ISTORE1) LABL
151 -      IF (IBODY.EQ.0) GO TO 150
152 -      CALL SYMBOL(EXPLOT+3.4,WYEPLT-1.2,0.210,
153 -      $'COMPONENTS OF VECTOR SERIES',-90.,27)
154 -      CALL SYMBOL(EXPLOT+3.0,WYEPLT-1.1,0.175,
155 -      $'ORIGINAL FILE NAMES',-90.,20)
156 -      CALL SYMBOL(EXPLOT+3.0,WYEPLT-5.2,0.175,
157 -      $'VARIABLE NUMBERS',-90.,17)
158 -      IFREQ(1)=4HFREQ ; IFREQ(2)=4HUENC
159 - C
160 - C
161 -      DO 140 I=1,3
162 -      IF (I.EQ.1) IFREQ(3)=4HY 1: ; GO TO 137
163 -      IF (I.EQ.2) IFREQ(3)=4HY 2: ; GO TO 137
164 -      IFREQ(3)=4HY 3:
165 -      137 CALL SYMBOL(EXPLOT+2.7-0.6*(I-1),WYEPLT,0.14,IFREQ,
166 -      $-90.,12)
167 -      CALL SYMBOL(EXPLOT+2.7-0.6*(I-1),WYEPLT-2.0,0.14,
168 -      $INAMES(6*I-5),-90.,12)
169 -      CALL SYMBOL(EXPLOT+2.45-0.6*(I-1),WYEPLT-2.0,0.14,
170 -      $INAMES(6*I-2),-90.,12)
171 -      140 CONTINUE
172 - C
173 - C
174 -      DO 145 I=1,3
175 -      FVAR=FLOAT(IVAR(2*I-1))
176 -      CALL NUMBER(EXPLOT+2.7-0.6*(I-1),WYEPLT-6.6,0.14,FVAR,
177 -      $-90.,-1)
178 -      CALL NUMBER(EXPLOT+2.45-0.6*(I-1),WYEPLT-6.6,0.14,FVAR,

```

```

179 -      $-90.,-1)
180 -      145 CONTINUE
181 - C
182 - C
183 -      150 CONTINUE
184 -      CALL SYMBOL(EXPLOT+3.4,WYEPLT-9.8,0.175,
185 -      $'CONTOUR LEVELS AND PERCENT CONFIDENCE',-90.,37)
186 -      BIASX=BIASY=0.
187 -      IF (NCONLV.GT.10) NCONLV=10
188 - C
189 - C
190 -      DO 160 I=1,NCONLV
191 -      IF (I.EQ.6) BIASX=0. ; BIASY=BIASY-3.0
192 -      CALL NUMBER(EXPLOT+2.9+BIASX,WYEPLT-11.9+BIASY,0.210,
193 -      $CONLV(I),-90.,3)
194 -      CALL NUMBER(EXPLOT+2.9+BIASX,WYEPLT-13.3+BIASY,0.210,
195 -      $CONFLV(I),-90.,3)
196 -      BIASX=BIASX-0.4
197 -      160 CONTINUE
198 - C
199 - C
200 -      CALL SYMBOL(EXPLOT+0.4,WYEPLT-9.2,0.210,IDENT1,-90.,36)
201 -      CALL SYMBOL(EXPLOT,WYEPLT-9.2,0.210,IDENT2,-90.,36)
202 -      IF ((MF.NE.21).OR.(IFOURIER.EQ.0)) GO TO 170
203 -      CALL SYMBOL(EXPLOT+3.4,WYEPLT-21.4,0.175,
204 -      $'PROCESSING HISTORY',-90.,18)
205 -      CALL SYMBOL(EXPLOT+2.9,WYEPLT-18.8,0.175,LABL,-90.,36)
206 -      CALL SYMBOL(EXPLOT+2.4,WYEPLT-18.8,0.175,LABL(10),-90.,
207 -      $48)
208 -      CALL SYMBOL(EXPLOT+1.9,WYEPLT-18.8,0.175,LABL(22),-90.,
209 -      $48)
210 -      170 CONTINUE
211 -      CALL TODAY(ICLK)
212 -      CALL SYMBOL(EXPLOT+1.0,WYEPLT-22.0,0.175,
213 -      $'TIME OF PLOT:',-90.,13)
214 -      CALL SYMBOL(EXPLOT+0.7,WYEPLT-21.6,0.175,ICLK,-90.,16)
215 - C LABELLING COMPLETED. ESTABLISH NEW ORIGIN.
216 -      CALL PLOT(0.,14.5*AXL,-3)
217 - C BOUNDARIES OF DOMAIN, PLUS X AND Y AXES (UNLABELLED)
218 -      CALL PLOT(AXL,-AXL,3)
219 -      CALL PLOT(0.5*AXL,-0.5*AXL,2)
220 -      CALL PLOT(AXL,0.,2)
221 -      CALL PLOT(2.*AXL,0.,2)
222 -      CALL PLOT(0.,-2.*AXL,2)
223 -      CALL PLOT(AXL,-2.*AXL,2)
224 -      CALL PLOT(1.5*AXL,-1.5*AXL,2)
225 -      CALL PLOT(AXL,-AXL,2)
226 -      OFFSET=ALN-AXL
227 -      CALL AXWJS(AXL,OFFSET,' ',-1,AXLEN,-90.,0.,1.,DISTIC,
228 -      $-1,0,0,.12,.7,.0001,.14)
229 -      CALL AXWJS(-OFFSET,-AXL,' ',1,AXLEN,0.,0.,1.,DISTIC,
230 -      $-1,0,0,.12,.7,.0001,.14)
231 -      IF (ISW.NE.3) GO TO 180
232 -      CALL PLOT(2.*AXL,0.,3)
233 -      CALL PLOT(2.*AXL,-AXL,2)
234 -      CALL PLOT(1.5*AXL,-1.5*AXL,2)
235 -      CALL PLOT(0.,-2.*AXL,3)
236 -      CALL PLOT(0.,-AXL,2)
237 -      CALL PLOT(0.5*AXL,-0.5*AXL,2)
238 -      180 CONTINUE

```

```

239 - C LABELLED BOUNDARIES OF PLOT
240 -     CALL AXWJS(-OFFSET,OFFSET,'FREQ2 (CPH)',11,AXLEN,0.,
241 -     $FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,0.14)
242 -     CALL AXWJS(2.*AXL+OFFSET,OFFSET,'FREQ1 (CPH)',11,AXLEN,
243 -     $-90.,FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,0.14)
244 -     CALL AXWJS(-OFFSET,OFFSET,'FREQ1 (CPH)',-11,AXLEN,-90.,
245 -     $FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,0.14)
246 -     CALL AXWJS(-OFFSET,-2.*AXL-OFFSET,'FREQ2 (CPH)',-11,
247 -     $AXLEN,0.,FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,
248 -     $0.14)
249 - C BEGIN CONTOURING
250 - C POSITIVE SUM FREQUENCIES FOR /W1/ > /W2/
251 -     CALL GRID(0.,-AXL-DEL,DEL,-DEL)
252 -     CALL WHCNTR(DATA,1,LP34+1,1,LPHALF,NCONLV,CONLV)
253 -     IPASS=2
254 -     GO TO 30
255 -     200 CONTINUE
256 - C NEGATIVE SUM FREQUENCIES FOR /W1/ > /W2/
257 -     CALL GRID(0.5*AXL,0.,DEL,-DEL)
258 -     CALL WHCNTR(DATA,1,LP34+1,1,LPHALF,NCONLV,CONLV)
259 -     IPASS=1 ; KPASS=2
260 -     IF (ISW.NE.3) GO TO 500
261 -     GO TO 30
262 -     300 CONTINUE
263 - C POSITIVE SUM FREQUENCIES FOR /W1/ < /W2/
264 -     CALL DIMWH(XDATA,MD2,MD1)
265 -     CALL GRID(AXL,0.,DEL,-DEL)
266 -     CALL WHCNTR(XDATA,1,LPHALF+1,1,LP34+1,NCONLV,CONLV)
267 -     IPASS=2
268 -     GO TO 30
269 -     400 CONTINUE
270 - C NEGATIVE SUM FREQUENCIES FOR /W1/ < /W2/
271 -     CALL GRID(0.,-0.5*AXL,DEL,-DEL)
272 -     CALL WHCNTR(XDATA,1,LPHALF+1,1,LP34+1,NCONLV,CONLV)
273 -     500 CONTINUE
274 -     CALL PLOT(0.,0.,999)
275 -     STOP 'NORMAL PROGRAM COMPLETION'
276 - C*****
277 - C*****
278 -     SUBROUTINE BIAS(IP)
279 - C INTERNAL SUBROUTINE FOR PROGRAM RBPLOT
280 -     4 GO TO (6,8),IP
281 -     6 GO TO (10,15,15,10,20,25,25,20,10,15,15,10,25,30,30,35),
282 -     $ISQ
283 -     8 GO TO (30,40,40,30,35,50,50,35,30,40,40,30,50,60,60,20),
284 -     $ISQ
285 -     10 CALL ASN(IBIASR,1) ; GO TO 65
286 -     15 CALL ASN(IBIASR,2) ; GO TO 65
287 -     20 CALL ASN(IBIASR,3) ; GO TO 65
288 -     25 CALL ASN(IBIASR,4) ; GO TO 65
289 -     30 CALL ASN(IBIASR,5) ; GO TO 65
290 -     35 CALL ASN(IBIASR,6) ; GO TO 65
291 -     40 CALL ASN(IBIASR,7) ; GO TO 65
292 -     50 CALL ASN(IBIASR,8) ; GO TO 65
293 -     60 CALL ASN(IBIASR,9) ; GO TO 65
294 -     65 GO TO (67,68),IP
295 -     67 GO TO (70,70,75,75,75,75,80,80,80,80,85,85,85,85,80,85),
296 -     $ISQ
297 -     68 GO TO (85,85,80,80,80,80,75,75,75,75,70,70,70,70,75,70),
298 -     $ISQ

```

```

299 -      70 CALL ASN(IBIASC,6) ; RETURN
300 -      75 CALL ASN(IBIASC,5) ; RETURN
301 -      80 CALL ASN(IBIASC,4) ; RETURN
302 -      85 CALL ASN(IBIASC,2) ; RETURN
303 -      90 RETURN
304 - C*****
305 -      SUBROUTINE ASN(IB,IS)
306 -          5 GO TO (10,20,30,40,50,60,70,80,90), IS
307 -      10 IB=LPHALF+1 ; RETURN
308 -      20 IB=LP38 ; RETURN
309 -      30 IB=LP58+1 ; RETURN
310 -      40 IB=LPQU ; RETURN
311 -      50 IB=LPATHE ; RETURN
312 -      60 IB=0 ; RETURN
313 -      70 IB=LPQU+1 ; RETURN
314 -      80 IB=LP38+1 ; RETURN
315 -      90 IB=LPHALF+1 ; RETURN
316 -      95 RETURN
317 -      END

```

BIBLIOGRAPHIC DATA SHEET	1. Report No. WHOI-77-75	2.	3. Recipient's Accession No.
	4. Title and Subtitle AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA		5. Report Date April 1978
7. Author(s) Gerard H. Martineau and Melbourne G. Briscoe		8. Performing Organization Rept. No.	
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, MA 02543		10. Project/Task/Work Unit No.	
		11. Contract/Grant No. OCE76-14739	
12. Sponsoring Organization Name and Address National Science Foundation		13. Type of Report & Period Covered Technical	
		14.	
15. Supplementary Notes			
16. Abstracts This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \omega_1 \pm \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, <u>there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.</u>			
17. Key Words and Document Analysis. 17a. Descriptors 1. Bispectra 2. Time series analysis 3. Rotary spectra			
17b. Identifiers/Open-Ended Terms			
17c. COSATI Field/Group			
18. Availability Statement		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 235
		20. Security Class (This Page) UNCLASSIFIED	22. Price

Woods Hole Oceanographic Institution WHOI-77-75 AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA by Gerard H. Martineau and Melbourne G. Briscoe. 235 pages. April 1978. Prepared for the National Science Foundation under Grant OCE76-14739. This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_1 = \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.	1. Bispectra 2. Time series analysis 3. Rotary spectra I. Martineau, Gerard H. II. Briscoe, Melbourne G. III. OCE76-14739 This card is UNCLASSIFIED	Woods Hole Oceanographic Institution WHOI-77-75 AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA by Gerard H. Martineau and Melbourne G. Briscoe. 235 pages. April 1978. Prepared for the National Science Foundation under Grant OCE76-14739. This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_1 = \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.	1. Bispectra 2. Time series analysis 3. Rotary spectra I. Martineau, Gerard H. II. Briscoe, Melbourne G. III. OCE76-14739 This card is UNCLASSIFIED	Woods Hole Oceanographic Institution WHOI-77-75 AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA by Gerard H. Martineau and Melbourne G. Briscoe. 235 pages. April 1978. Prepared for the National Science Foundation under Grant OCE76-14739. This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_1 = \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.	1. Bispectra 2. Time series analysis 3. Rotary spectra I. Martineau, Gerard H. II. Briscoe, Melbourne G. III. OCE76-14739 This card is UNCLASSIFIED
--	--	--	--	--	--