A TRIDENT SCHOLAR PROJECT REPORT

NO. 81

MATHEMATICAL-STATISTICAL AND DIGITAL COMPUTER ANALYSIS OF TIME SERIES DATA

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND
1976

This document has been approved for public release and sale; its distribution is unlimited.
Mathematical-Statistical and Digital Computer Analysis of Time Series Data

A Trident Scholar Project Report

by

Midshipman Steven Rader, l/c
U. S. Naval Academy
Annapolis, Maryland

John S. Kalme
Assoc. Professor John S. Kalme
Mathematics Department

Accepted for Trident Scholar Committee

Chairman

Date
MATHEMATICAL-STATISTICAL AND DIGITAL COMPUTER ANALYSIS OF TIME SERIES DATA

The purpose of this study is to develop a series of computer programs for use in analyzing time series data (waves), such as EEG readings. Programs were used to produce reliable estimates of correlations, spectra, cross spectra, and partial coherences of multi-channel random processes. The software package was written to be easily adaptable to different sampling rates, amounts of data, and numbers of channels. Provisions for digital prefiltering of data, detrending, and smoothing (using a number of lag windows) were also included. Techniques for estimation of spectra by fitting single- and multi-channel autoregressive schemes to sampled data were also applied and found to yield results consistent with the other methods. All programs were written in FORTRAN and run on the USNA/DTSS computer system.
PREFACE

This study was undertaken as part of a Trident Scholar Research project. It is the result of two semesters of study during the academic year 1975-76. The many hours which my advisor, Assoc. Prof. John S. Kalme, spent contributing help and guidance are sincerely appreciated. I would also like to thank Assoc. Prof. Karel Montor for supplying EEG data, and Maj. David A. Wright (CAF) for his assistance in digitizing the data.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>PREFACE</td>
<td>2</td>
</tr>
<tr>
<td>CHAPTER 1</td>
<td>4</td>
</tr>
<tr>
<td>Elementary Definitions</td>
<td>5</td>
</tr>
<tr>
<td>Discrete Implementation</td>
<td>8</td>
</tr>
<tr>
<td>Smoothing</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER 2</td>
<td>13</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>31</td>
</tr>
<tr>
<td>APPENDIX A. Program Listings</td>
<td>32</td>
</tr>
<tr>
<td>APPENDIX B. Program Outputs</td>
<td>109</td>
</tr>
</tbody>
</table>
CHAPTER 1
I. Elementary Definitions

A sample space $\Omega$ is the set of all possible outcomes of an experiment. Each possible outcome $\omega$ is called an elementary event. A (non-elementary) event $A$ in $\Omega$ is any subset of $\Omega$, any collection of elementary events. A probability measure $P$ defined on $\Omega$ is a rule which to each event $A$ in $\Omega$ assigns a real number $P(A)$ (called the probability of $A$) such that the following conditions are satisfied:

1. $P(\emptyset) = 0$
2. $P(\Omega) = 1$
3. If $A_i$ are pairwise disjoint events, then $P(\bigcup A_i) = \sum P(A_i)$

A random variable $X$ is a function defined on a sample space $\Omega$, which assigns a real or complex value to each elementary event $\omega$. The expectation $E(X)$ of the random variable is defined as

$$\int_{\Omega} X(\omega) \, dP(\omega) \quad \text{provided} \quad \int_{\Omega} |X(\omega)| \, dP(\omega) < \infty$$

Both integrations are performed in a Lebesgue sense.

A random process is a function of two variables, $t$ and $\omega$, where $t$ is a real number or integer, and $\omega$ is an elementary event in the sample space $\Omega$. Thus, $X(t, \omega)$ is a random process if $t$ is allowed to vary over an interval, but $X(t_0, \omega)$ for a fixed $t_0$ is a random variable. Usually the second argument is omitted when expressing a random variable: $X(t, \omega)$ is written as $X(t)$.

A random process is said to be stationary in the wide sense if $E(X(t)X(t + v))$ depends only upon $v$, not $t$. This expectation is called the correlation function of $X$ and is denoted by $R_X(v)$.

If $Y(t)$ is another stationary process defined on the same
sample space, and $E(X(t)Y(t+v))$ depends only on $v$, then we say that $X$ and $Y$ are jointly stationary. This expectation is denoted by $R_{XY}(v)$ and is called the cross correlation of the two random processes $X$ and $Y$.

The Fourier integral of the autocorrelation function of $X$ is called the power spectral density of $X$, and is evaluated by this expression:

$$S_X(f) = \int_{-\infty}^{\infty} R_X(\nu) e^{2\pi i \nu f} d\nu$$

The power spectral density of a process indicates the amount of energy the process contains in any frequency interval. For example, EEG waveforms have much of their energy concentrated near 10 Hz; if such a wave were passed through a 10 Hz bandpass filter it would lose relatively little power. Thus, we would expect its spectral density function to have a peak about 10 Hz.

Similarly, the Fourier integral of the cross correlation of $X$ and $Y$ is called the cross spectral density of $X$ and $Y$ and is denoted by $S_{XY}(f)$:

$$S_{XY}(f) = \int_{-\infty}^{\infty} R_{XY}(\nu) e^{2\pi i \nu f} d\nu$$

The cross spectral density represents the amount of power shared by the two processes at any frequency. For example, the cross spectral density of the input and the output of a bandpass filter would have a very high cross spectral density over the frequencies passed by the filter.

Of course, the autocorrelation and cross correlation functions can be recovered from the spectral density and cross spectral density functions, respectively, by using the inverse Fourier
integrals.

If the processes $X$ and $Y$ are complex-valued, the definitions of autocorrelation and cross correlation are modified to use the conjugate of the first term:

$$R_X(n) = E \left( \bar{X}(\tau) \cdot X(\tau + n) \right)$$

$$R_{XY}(n) = E \left( \bar{X}(\tau) \cdot Y(\tau + n) \right)$$

The coherence between $X$ and $Y$ is a normalized function of the density functions:

$$\gamma_{XY}^2(\phi) = \frac{|S_{XY}(\phi)|^2}{S_x(\phi) \cdot S_y(\phi)}$$

The partial coherence of $X$ and $Y$ after the effects of a third time series $Z$ have been removed is found using this formidable-looking expression:

$$\gamma_{XZ}^2(\phi) = \frac{\left|S_{XZ}(\phi) \cdot S_{YZ}(\phi) \right|^2}{\left[ S_x(\phi) - \frac{|S_{xz}(\phi)|^2}{S_z(\phi)} \right]\left[ S_y(\phi) - \frac{|S_{yz}(\phi)|^2}{S_z(\phi)} \right]}$$
II. Discrete Implementation

Suppose we have \( N \) observed values of two processes \( X \) and \( Y \). We thus have \( X(t) \) and \( Y(t) \) defined only for \( t=0,1,2,...N-1 \). We then define the correlation functions in terms of an average rather than an expectation:

\[
R_X(\nu) = \frac{1}{N} \sum_{t=0}^{N-1} X(t)X(t+\nu) \quad R_{XY}(\nu) = \frac{1}{N} \sum_{t=0}^{N-1} X(t)Y(t+\nu)
\]

For most purposes it is necessary to determine correlations only for values of \( \nu \) less than or equal to a certain limit \( L \). (Usually \( L \) is less than 10% or 20% of the number of observed data points \( N \).) For purposes of computation, extend the sequences \( X \) and \( Y \) to length \( N' \) (which must be greater than \( N + L \)) by appending zeros to the end; call these new sequences \( X' \) and \( Y' \). Thus,

\[
X'(t)=X(t) \quad \text{and} \quad Y'(t)=Y(t) \quad \text{for} \quad t=0,1,...N-1
\]

but \( X'(t)=0 \) and \( Y'(t)=0 \) for \( t=N,N+1,...N'-1 \)

If the fast Fourier transform (FFT) is to be used, \( N' \) must be a power of two.

Let \( \hat{X}'(\omega) \) and \( \hat{Y}'(\omega) \) be the Fourier transforms of \( X' \) and \( Y' \):

\[
\hat{X}'(\omega) = \sum_{t=0}^{N'-1} X'(t)e^{-i\frac{2\pi}{N'} t\omega}\quad \hat{Y}'(\omega) = \sum_{t=0}^{N'-1} Y'(t)e^{-i\frac{2\pi}{N'} t\omega}
\]

Now form a new sequence in which each term is the product of the complex conjugate of the corresponding term of \( \hat{X}' \) and the corresponding term of \( \hat{Y}' \); then take the inverse Fourier transform of this sequence.
When we insert the above expressions for \(X'(u)\) and \(Y'(u)\), and manipulate the (finite) summations, we obtain the following:

\[
C(n) = \frac{1}{N} \sum_{\mu=0}^{N'-1} \left\{ \sum_{x=0}^{N' - 1} X'(x) e^{-2\pi i \mu (x - N' - v)/N'} \right\} \left\{ \sum_{\alpha=0}^{N' - 1} Y'(\alpha) e^{2\pi i \mu (\alpha - v)/N'} \right\} e^{-2\pi i \mu n/N'}
\]

It is easily verified that the trigonometric expression in the brackets is equal to zero unless \((s-t-v)\) is zero or an integral multiple (positive or negative) of \(N'\), in which case it is equal to one. Because \(s\) and \(t\) are restricted to the range 0 to \(N'-1\), only two sets of \((s,t)\) pairs meet this criterion.

If \(s-t-v = -N'\), then \(t = N' - v + s\), and \(t\) must be greater than \(N' - v\).

We are restricting \(v\) to be less than \(L\), and \(N'\) exceeds \(N+L\), so \(N' - v\) will be greater than \(N\). However, \(X'(t) = 0\) if \(t\) is greater than or equal to \(N\), because \(X'\) is only an extension of \(X\) beyond \(N-1\).

Thus, this set of \((s,t)\) pairs contributes nothing to the sum.

If \(s-t-v = 0\), then \(s = t + v\) and \(t\) is restricted to the range 0 to \(N-v-1\). Our expression then reduces to

\[
C(n) = \sum_{x=0}^{N'-v-1} X'(x) Y'(x + v)
\]

Again, because of the way in which the original sequences were extended, \(Y'(t+v)\) is nonzero only if \(t+v\) is less than \(N\), only if
t is less than or equal to N-v-1. Therefore, N-v-1 may be taken as the upper limit of summation and we have the following relation:

\[ C(N) = \sum_{k=0}^{N-1} \overline{X(k)} Y(k+N) = N \cdot R_{XY}(N) \]

Thus, by performing only three Fourier transforms we can obtain all values of the correlation function which interest us simultaneously. One more transform produces the cross spectral density function.

Observe that by substituting X and X' for Y and Y' the autocorrelation and power spectral density functions of X would have been obtained. By using the fast Fourier transform to perform the above calculations, estimates of spectra may be very efficiently generated on a digital computer. This method was realized in the subroutine CROSS, which can produce either cross correlations or autocorrelations.
III. Smoothing

When a correlation function $R(v)$ (in this section $R$ can be either an autocorrelation or a cross correlation) is Fourier transformed to produce a spectral density function, the explicit relation between $R$ and $S$ is

$$
\hat{S}(f) = \mathcal{F}[R(v)] + \frac{L^2}{2} \left[ \mathcal{R}(\omega) \cos\left( \frac{\pi \omega}{f_c} \right) + \mathcal{R}(L) \cos\left( \frac{\pi L f}{f_c} \right) \right]
$$

where $h$ is the sampling interval, and $f_c$ is the Nyquist frequency, $1/2h$, which is the highest frequency which can be unambiguously determined with a given sampling rate. The $\hat{S}$ indicates that this is a raw estimate of the density.

Unfortunately, the above expression is not a consistent estimate in the sense of mean square convergence. Its variance does not go to zero as the number of sample points increases, and a graph of raw spectral estimates will oscillate wildly about the true values of spectral density.

To improve the spectral estimates, it is necessary to first "smooth" or average the spectral values. A very simple but effective method is to replace each value of the spectral estimate with a weighted average of the original and neighboring values:

$$
\tilde{S}(0) = 0.5 \hat{S}(0) + 0.5 \hat{S}(1)
$$

$$
\tilde{S}(k) = 0.25 \hat{S}(k-1) + 0.5 \hat{S}(k) + 0.25 \hat{S}(k+1) \text{ for } k=1,2,...,L-1
$$

$$
\tilde{S}(L) = 0.5 \hat{S}(L-1) + 0.5 \hat{S}(L)
$$

This smoothing method can be implemented by applying a "window" to the correlation function:

$$
R'(v) = R(v) \cdot D(v/L)
$$

where $D$ is a weighing function defined as
This is known as a Tukey window. Another possible window, which results in a different degree of smoothing, is the modified Tukey window

\[ D(u) = \frac{1}{2} (1 + \cos \pi u), \quad |u| \leq 1 \]

\[ D(u) = 0.54 + 0.46 \cos \pi u. \]
We consider random processes \( \{ X_j(t) \} \), \( 1 \leq j \leq n \), \(-\infty < t < \infty \), that is, for each \( t \), \( X_j(t) \) is a random variable, where all \( X_j(t) \) are defined on the same sample space. We assume \( E( X_i(s) X_j(t+s) ) = R_{X_i X_j}(t) \) does not depend on \( s \), where \( i \) can equal \( j \). Then

\[
R_{X_i X_j}(t) = \int_{-\infty}^{\infty} e^{i2\pi tf} S_{X_i X_j}(f) \, df
\]

If \( i=j \), \( R_{X_i X_i}(t) \) is called autocorrelation function of \( X_i \), and \( S_{X_i X_i}(f) \) is the power spectral density of \( X_i \). If \( i \neq j \), \( R_{X_i X_j}(t) \) is called the cross-correlation function of \( X_i \) and \( X_j \), and \( S_{X_i X_j}(f) \) is the cross-spectral density function.

Assume \( E( X_j(t) ) = 0 \) for all \( j \) and \( t \).

Also,

\[
S_{X_i X_j}(f) = \int_{-\infty}^{\infty} e^{-i2\pi ft} R_{X_i X_j}(t) \, dt
\]

There exists a random spectral representation of the \( X_j(t) \):

\[
X_j(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda} \, dZ_{X_j}(\lambda)
\]

where

\[
Z_{X_j}(\lambda) - Z_{X_j}(\lambda) = \text{e.i.m.} \int_{-T}^{T} \frac{e^{-i2\pi \lambda t} - e^{-i2\pi \lambda t}}{-i2\pi t} X_j(t) \, dt
\]
The $Z_{X_j}(\lambda)$ are processes with orthogonal increments and

$$E\{\int_{-\infty}^{\infty} f(\lambda) \, dZ_{X_i}(\lambda) \cdot \int_{-\infty}^{\infty} g(\lambda) \, dZ_{X_j}(\lambda)\} =$$

$$= \int_{-\infty}^{\infty} f(\lambda) \, g(\lambda) \, S_{X_iX_j}(\lambda) \, d\lambda$$

where $i$ can equal $j$.

$Z_{X_j}(\lambda)$ forms a random spectral measure. Let us consider the physical significance of $S_{XX}(\omega)$.

Consider a linear time invariant filter with input a stationary process $X(t)$ and output $Y(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda} H(\lambda) \, dZ_X(\lambda)$.

$H(\lambda)$ is called the transfer function. Then

$$R_{YY}(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda} |H(\lambda)|^2 \, S_{XX}(\lambda) \, d\lambda$$

and

$$S_{YY}(\omega) = |H(\omega)|^2 \, S_{XX}(\omega).$$

Also,

$$S_{YY}(\omega) \bigg/ S_{XX}(\omega) = H(\omega)$$

Hence if we can get estimates $\hat{S}_{XY}(\omega)$ for $S_{XY}(\omega)$ and $\hat{S}_{XX}(\omega)$ of $S_{XX}(\omega)$, we can get an estimate $\hat{H}(\omega)$ of $H(\omega)$:

$$\hat{H}(\omega) = \frac{\hat{S}_{XY}(\omega)}{\hat{S}_{XX}(\omega)}.$$  Also,  $|\hat{H}(\omega)|^2 = \frac{\hat{S}_{YY}(\omega)}{\hat{S}_{XX}(\omega)}$

is an estimate of the square of the gain $|H(\omega)|^2$.

Most often we do not have explicit expressions for $H(\omega)$. 
Take for example a system such as a ship which acts like a black box. The waves $X(t)$ act as an input forcing function. The ship processes $X(t)$ in some way and responds by pitching as an output $Y(t)$.

(Sometimes we can write

$$Y(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda} H(\lambda) dZ_x(\lambda) = \int_{-\infty}^{\infty} h(\lambda - \mu) X(\mu) d\mu$$

where

$$H(t) = \int_{-\infty}^{\infty} h(\mu) e^{-i2\pi t\mu} d\mu.$$

$h(\mu)$ is called impulse response).

We subject the ship model to waves whose spectral distribution ranges over the frequencies it will actually encounter and see how the ship pitches. If the ship encounters waves of its natural pitching frequency, the ship will pitch badly. In this case the design must be adjusted to bring the natural pitching frequency to some frequency at which the waves normally encountered have little energy. The captain could also be warned of the sea conditions under which he will have to alter course or speed to avoid dangerous resonant pitching. As another example consider an RC filter:
This is a low pass filter. It passes low frequencies and attenuates high frequencies.

For the filter

\[
E_o(t) = \int_{-\infty}^{\infty} e^{i2\pi tf} \left( \frac{1}{1 + i2\pi RC\lambda} \right) dZ_{E_i}(\lambda)
\]

\[
S_{E_oE_o}(f) = \left( \frac{1}{1 + 4\pi^2 R^2 C^2 f^2} \right) S_{E_iE_i}(f)
\]

\[
|H(f)|^2 = \frac{4\pi^2 R^2 C^2 f^2}{1 + 4\pi^2 R^2 C^2 f^2}
\]
This is a high pass filter. It passes high frequencies, but attenuates low frequencies. One might wonder how we can estimate spectra for continuous parameter or analog signals by sampling at discrete time points and by using the digital computer recover the spectra. In most applications $S_{xx}(f) \neq 0$ for $|f| > W$ for some $W$.

For high-quality speech

$$S_{xx}(f) \neq 0, \quad 100 < f < 10,000 \text{ Hz}$$

For multichannel telephony

$$S_{xx}(f) \neq 0, \quad 300 < f < 3400 \text{ Hz}$$

For high-quality music

$$S_{xx}(f) \neq 0, \quad 30 < f < (10-15) \times \text{kHz}$$

For EEG

$$S_{xx}(f) \equiv 0, \quad |f| > 50 \text{ Hz}$$

The following Sampling Theorem holds:

If a function of time $x(t)$ contains no frequency components higher than $W$ hertz, the time function can be completely specified by determining the ordinates at a series of points spaced $\frac{1}{2W}$ seconds apart. Reconstitution of the original time function, i.e., the signal wave form is possible if the sample pulses are passed through a suitable low pass filter. This is important in time division multiplex systems.

If $S_{xx}(f) \equiv 0$ for $|f| > W$, then
\[ X(t) = \sum_{n=-\infty}^{\infty} X\left(\frac{n}{2W}\right) \frac{\sin \frac{2\pi W(t - \frac{n}{2W})}{2\pi W(t - \frac{n}{2W})}}{2\pi W(t - \frac{n}{2W})} \]

Let the discrete time series \( X_i(t) \), \( t = 0, \pm 1, \pm 2, \ldots \)
be obtained by sampling a continuous parameter time series \( Y_i(t) \) at time intervals of length \( h \): \( X_i(t) = Y_i(t \cdot h) \)
Similarly define \( X_j(t) \) for \( t = 0, \pm 1, \ldots \) .

Let
\[ R_{X_iX_j}(t) = R_{Y_iY_j}(t \cdot h) \]

This sampling can be obtained by using a PDP8-E minicomputer, which incorporates an analog-to-digital converter.

Then
\[ R_{X_iX_j}(t) = \int_{-\frac{1}{2h}}^{\frac{1}{2h}} S_{X_iX_j}(f) e^{i2\pi ft} df \]

where
\[ S_{X_iX_j}(f) = \sum_{n=-\infty}^{\infty} S_{Y_iY_j}(f + \frac{n}{h}) , \quad -\frac{1}{2h} \leq f \leq \frac{1}{2h} \]

Thus in order that \( S_{X_iX_j}(f) = S_{Y_iY_j}(f) \)
we must have \( S_{Y_iY_j}(f) = 0 \) for \( |f| > \frac{1}{2h} \).

Otherwise we would get aliasing and frequencies above
will be folded back. (One observes this in old westerns
where the wheels rotate backwards when the stagecoach
starts and slows down). Thus the Nyquist folding frequency
\[ f_c = W = \frac{1}{2h} \]

For EEG we sample at 10 msec intervals, or to get a
power of 2, we sample at \( \frac{1}{64} \) or \( \frac{1}{128} \) sec. intervals for a
period of 10 sec. or less.
Let \( X(t), X(z), \ldots, X(N) \) be samples taken at time intervals of length \( h \). Assume detrending has been performed. We shall discuss the estimation of \( S_{x_i x_j}(f) \).

The sample cross-covariance \( \hat{R}_{x_i x_j}(v) \) of lag \( v \) between \( X_i(\cdot) \) and \( X_j(\cdot) \) is defined to be
\[
\hat{R}_{x_i x_j}(v) = \frac{1}{N} \sum_{t=1}^{N-v} X_i(t)X_j(t+v), \quad v = 0, 1, 2, \ldots, (N-1)
\]
\[
\hat{R}_{x_i x_j}(-v) = \hat{R}_{x_j x_i}(v)
\]
The \( \hat{R}_{x_i x_j}(v) \) are computed by using the FFT.

The sample cross-spectral density function or cross periodogram between \( X_i(\cdot) \) and \( X_j(\cdot) \) is given by
\[
I_{x_i x_j}(f) = \frac{h}{N} \left( \sum_{s=1}^{N} X_i(s)e^{i\frac{2\pi f s}{N}} \right) \left( \sum_{t=1}^{N} X_j(t)e^{-i\frac{2\pi f t}{N}} \right) = h \sum_{k=-(N-1)}^{N-1} \hat{R}_{x_i x_j}(k)e^{-i2\pi f k h}, \quad -\frac{1}{2h} \leq f \leq \frac{1}{2h}
\]
\[
S_{x_i x_j}(f) = h \sum_{\ell=-\infty}^{\infty} R_{x_i x_j}(\ell)e^{-i2\pi f \ell h}
\]
\[
E(I_{x_i x_i}(f)) = E\left( \frac{1}{N} \sum_{t=1}^{N} x_i(t) e^{-i 2\pi f t h} \right) = \frac{h}{N} \int_{-\frac{1}{2}h}^{\frac{1}{2}h} \frac{\sin^2(N\pi(f-\lambda)h)}{\sin^2(\pi(f-\lambda)h)} S_{x_i x_i}(\lambda) d\lambda \to S_{x_i x_i}(f), \text{ as } N \to \infty.
\]

But if for example \(x_i(t)\) is Gaussian we have approximately

\[
P\left[ I_{x_i x_i}(f) > x \right] \approx e^{-\frac{1}{S_{x_i x_i}(f)} x^2}
\]

or

\[
\frac{2 I_{x_i x_i}(f)}{S_{x_i x_i}(f)} \sim \chi^2_2,
\]

\[
\text{VAR} (I_{x_i x_i}(f)) \approx S_{x_i x_i}(f)^2
\]

The periodogram itself is not a good estimate of the spectrum. It is not a consistent estimate in the sense of mean square convergence. Its variance does not go to zero as \(N\) increases. To get a good estimate of \(S_{x_i x_i}(f)\) we must "smooth" the periodogram.

Take for the moment \(h = \frac{1}{2\pi}\)

\[
\hat{S}_{x_i x_i}(\lambda) = \frac{1}{\pi} \int_{-\pi}^{\pi} W(\lambda - \alpha) I_{x_i x_i}(\alpha) d\alpha
\]
\[ E(\hat{S}_{xi xi}(\lambda)) \rightarrow \int_{-\pi}^{\pi} W(\lambda - \alpha) S_{xi xi}(\alpha) d\alpha \approx S_{xi xi}(\lambda) \]

if \( \int_{-\pi}^{\pi} W(\alpha) d\alpha = 1 \)

\[ \text{N.VAR}(\hat{S}_{xi xi}(\lambda)) = 2\pi \int_{-\pi}^{\pi} W^{2}(\lambda - \alpha) S_{xi xi}^{2}(\alpha) d\alpha + o(1) \]

\[ \text{VAR}(\hat{S}_{xi xi}(\lambda)) \approx \frac{2\pi}{N} S_{xi xi}^{2}(\lambda) \left( \int_{-\pi}^{\pi} W^{2}(\alpha) d\alpha \right) \]

Let \( M < N \). Choose \( \frac{M}{N} \approx 0.1 \) or 0.2

We use estimates

\[ \hat{S}_{xi xi}(f) = h \sum_{|m| \leq M} K\left(\frac{v}{M}\right) \hat{R}_{xi xi}(v) e^{-i2\pi f vh} \]

\[ K(-u) = K(u) \quad ; \quad K(0) = 1 \quad , \quad K(u) = 0 \quad \text{for} \quad |u| > 1 \]

\[ \text{VAR}(\hat{S}_{xi xi}(f)) \approx \frac{M}{N} \cdot I \cdot S_{xi xi}(f) \]

\[ I = \int_{-1}^{1} K^{2}(u) du \]

Approximate confidence intervals for the \( S_{xi xi}(f) \) can be found by using the fact that the random variable

\[ \frac{\nu \hat{S}_{xi xi}(f)}{S_{xi xi}(f)} \approx \chi^{2}_\nu \]

where \( \nu = \frac{2N}{IM} \)

We choose points \( f_{K} = \frac{Kf_{c}}{\delta M} \) for \( K = 0, 1, 2, \ldots, M \) to estimate the spectra.

\[ f_{c} = \frac{1}{2h} \]

Then

\[ \hat{S}_{xi xi}\left(\frac{v}{M} f_{c}\right) = h \sum_{|m| \leq M} K\left(\frac{v}{M}\right) \hat{R}_{xi xi}(v) e^{-i\frac{2\pi}{2M} vK} \]
The computations for \( \hat{R}_{x_i x_j}(\nu) \) and the sums in \\
\( \hat{S}_{x_i x_j}(\nu \lambda) \) are performed by efficient use of the FFT. 
We used the lag window \\
\[
K(\omega) = \begin{cases} 
1 - 6|\omega|^2 + 6|\omega|^3, & |\omega| \leq \frac{1}{2} \\
2(1-|\omega|)^3, & \frac{1}{2} \leq \omega \leq 1 \\
0, & \omega > 1
\end{cases}
\]

Then \( I = 0.539, \ \nu = \frac{2}{I} \frac{N}{M} = 3.71 \ \frac{N}{M} \) 

The program MULSPECT estimates the spectra by using the above lag window.

Another method I used involves the use of a modified periodogram and cross-periodograms using cosine taper. Then the modified periodograms are averaged over neighboring points.

A modified periodogram is of the form 
\[
I^*(f) = \frac{h}{NU} \left| \sum_{j=1}^{N} W_n(j) X(j) e^{-i 2\pi j f U} \right|^2
\]

\[
U = \frac{1}{N} \sum_{j=1}^{N} W_n^2(j)
\]

We have written programs which involve new methods of spectral estimation, namely the fitting of autoregressive schemes to given time series.

The program AUTOREG estimates spectra by fitting autoregressive schemes to time series. We solve for \( a_1, a_2, \ldots, a_m \) (with \( a_0 = 1 \)) the following system of equations
\[
\sum_{s=0}^{m} a_s \hat{R}_{xx}(t-s) = 0, \quad t = 1, 2, \ldots, m
\]

\[
\hat{R}_{xx}(-t) = \hat{R}_{xx}(t) = \frac{1}{N} \sum_{j=1}^{N-t} X(j)X(j+t)
\]

The \( \hat{R}_{xx}(t) \) are computed by using subroutine CROSS. The \( a_1, a_2, \ldots, a_m \) are computed by using subroutine LEVNSN.

Let \( \hat{\sigma}^2 = \sum_{k=0}^{m} a_k \hat{R}_{xx}(-k) \)

The spectral estimate is given by

\[
\hat{S}_{xx}(f) = h\hat{\sigma}^2 \frac{1}{\left| \sum_{k=0}^{m} a_k e^{-i2\pi k f h} \right|^2}
\]

Pick \( B = 2^L \geq m \)

Evaluate \( \hat{S}_{xx}(f) \) at

\[
f = \frac{j}{B} f_c = \frac{j}{B} \frac{1}{2h} \quad \text{for} \quad j = 0, 1, 2, \ldots, B
\]

Let

\[
a_{m+1} = a_{m+2} = \ldots = a_{2B-1} = 0
\]

\[
\hat{S}_{xx}(\frac{j}{B} f_c) = h\hat{\sigma}^2 \frac{1}{\left| \sum_{k=0}^{2B-1} a_k e^{-i \frac{2\pi k j}{2B h}} \right|^2}
\]
The program SPCFCLTK estimates spectra by averaging modified periodograms.

The program MAUTOREG estimates multichannel spectra and cross-spectra by fitting multidimensional autoregressive schemes to the multichannel time series.

Let

\[ \hat{R}(v) = \left[ \begin{array}{ccc} \hat{R}_{11}(v) & \cdots & \hat{R}_{1n}(v) \\ \vdots & \ddots & \vdots \\ \hat{R}_{n1}(v) & \cdots & \hat{R}_{nn}(v) \end{array} \right] \]

\[ \hat{R}(-v) = \hat{R}^T(v) \]

\[ \hat{R}_{jk}(v) = \frac{1}{N} \sum_{t=1}^{N-v} X_j(t)X_k(t+v) \]

\[ \hat{S}_{\chi}(f) = \left[ \begin{array}{ccc} \hat{S}_{11}(f) & \cdots & \hat{S}_{1n}(f) \\ \vdots & \ddots & \vdots \\ \hat{S}_{n1}(f) & \cdots & \hat{S}_{nn}(f) \end{array} \right] \]

where \( \hat{S}_{jk}(f) \) is an estimate of \( S_{x_jx_k}(f) \).

Let \( \hat{A}(o) = I_n = \left[ \begin{array}{ccc} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{array} \right], \ n \times n \) identity matrix.
Let \( \hat{A}(j) \) for \( j = 1, 2, \ldots, m \) be \( n \times n \) matrices which are solutions of the system of matrix equations

\[
\sum_{j=0}^{m} \hat{A}(j) \hat{R}(j-k) = 0
\]

for \( k = 1, 2, \ldots, m \) 

( \( \mathbf{O} = n \times n \) zero matrix)

Let \( \hat{V}_m = \sum_{j=0}^{m} \hat{A}(j) \hat{R}(j) \)

Then we estimate the spectral density matrix by

\[
\hat{S}_x(f) = h \left[ \sum_{j=0}^{m} \hat{A}(j) e^{i 2\pi f j h} \right]^{-1} \hat{V}_m \left( \sum_{j=0}^{m} \hat{A}(j) e^{-i 2\pi f j h} \right)^{-1}
\]

Let \( B = 2^k \), \( f_c = \frac{f}{2B} \)

We evaluate \( \hat{S}_x(f) \) at points \( f = \frac{k}{B} f_c \) for \( k = 0, 1, \ldots, B \)

Let \( \hat{A}(j) = 0 \) for \( j = m+1, m+2, \ldots, 2B-1 \)

Then

\[
\hat{S}_x(k f_c) = h \left[ \sum_{j=0}^{2B-1} \hat{A}(j) e^{i \frac{2\pi f_c}{2B} j} \right]^{-1} \hat{V}_m \left( \sum_{j=0}^{2B-1} \hat{A}(j) e^{-i \frac{2\pi f_c}{2B} j} \right)^{-1}
\]

All calculations are performed using FFT.
The elements of the matrix $\hat{R}(\nu)$ are computed by subroutine MAC, outputted in multiplexed form. The matrices $\hat{A}(1), \hat{A}(2), \ldots, \hat{A}(m)$ are computed by the subroutine MULLEV.

Several of the programs involve simulation of time series with specified spectral densities. We can simulate EEG or any time series with almost any spectra. Let \( \{X(n)\} \) be a sequence of independent observations from \( N(0,1) \) (white noise).

Let \( Y(t) = \sum a_n X(t + n) \) (digital filter).

Then
\[
S_{YY}(f) = \left| \sum a_n e^{i2\pi nf} \right|^2 S_{XX}(f)
\]
\[
S_{XX}(f) = h \quad \text{for} \quad -\frac{1}{2h} \leq f \leq \frac{1}{2h}
\]
\[
S_{YY}(f) = \left| \sum a_n e^{i2\pi nf} \right|^2 h
\]

If we take \( \frac{1}{n+L} \sum a_n e^{i2\pi nf} \) as the \( L \)-th partial sum of the Fourier series of the function \( \sqrt{\frac{1}{h}} S_{YY}(f) \) where \( S_{YY}(\cdot) \) is a given function, \( Y(\cdot) \) will have spectral density \( S_{YY}(\cdot) \).

Each \( X(n) \) can be obtained by taking
\[
X(n) = \left( \sum_{j=1}^{\infty} R_j \right) - \zeta
\]
where \( R_1, R_2, \ldots, R_{12} \) are random numbers from the computer.

We took \( h = \frac{1}{24} \)

Simulation was useful for testing programs.
We define coherence for time series $X$ and $Y$

$$\gamma_{xy}^2(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f) \cdot S_{yy}(f)}$$

Partial coherence of $X(\cdot)$ and $Y(\cdot)$ when the effects of $Z(\cdot)$ are removed

$$\gamma_{xy,z}(f) = \frac{|S_{xy}(f) - \frac{S_{xz}(f) \cdot S_{yz}(f)}{S_{zz}(f)}|^2}{\left[ S_{xx}(f) - \frac{|S_{xz}(f)|^2}{S_{zz}(f)} \right] \left[ S_{yy}(f) - \frac{|S_{yz}(f)|^2}{S_{zz}(f)} \right]}$$

Let $v = 2n$ be the effective degrees of freedom.

Let $\gamma^2(f)$ be an estimate of $\gamma^*(f)$

Let $\hat{\gamma}(f) = \sqrt{\hat{\gamma}^2(f)}$, $\gamma(f) = \sqrt{\gamma^*(f)}$

Let $\tanh^{-1}(z) = \frac{1}{2} \ln \frac{1 + z}{1 - z}$, $|z| < 1$

Then $\tanh^{-1}(\hat{\gamma}(f))$, where $\hat{\gamma}(f) = \hat{\gamma}_{xy}(f)$, has approximately a normal distribution:

$$\tanh^{-1}(\hat{\gamma}(f)) \sim N(\tanh^{-1}(\gamma(f)) + \frac{1}{2(n-1)}, \frac{1}{2(n-1)})$$

provided $n > 20$, $0.4 \leq \gamma^*(f) \leq 0.95$

If $f \neq 0$, $\frac{1}{2n}$, and if $\gamma(f) = 0$

then

$$\frac{\hat{\gamma}^2(f)}{(\alpha-1) \left( 1 - \hat{\gamma}^2(f) \right)} = F_{2, 2(n-1)}$$
where $F_{m_1,m_2}$ is a random variable having an F distribution with degrees of freedom $m_1$ and $m_2$. This can be used to test the null hypothesis $H_0: \gamma(f) = 0$ against the alternative $H_1: \gamma(f) > 0$. If $f \neq 0$, $\hat{\gamma}_{xy,z}(f)$ has the same distribution as $\hat{\gamma}_{xy}(f)$.

Spectra and cross-spectra and partial coherences can be used to localize brain tumors and epileptogenic foci in the brain.

Suppose we want to test whether $Z$ drives $X$ and $Y$. ($Z$ might be an epileptogenic focus).

Assume $\gamma_{xv}(f), \gamma_{xz}(f), \gamma_{yv}(f)$,

$S_{xx}(f), S_{yy}(f), S_{zz}(f)$ are significantly different from zero over a certain frequency range.

Assume $\gamma_{xz}^2(f) \neq 0, \gamma_{yz}^2(f) \neq 0$, but $\gamma_{xz}^2(f) = 0$.

Then we would suspect that $Z$ drives $X$ and $Y$.

Suppose we want to test whether $Z$ drives $X_i, X_j, \ldots, X_m$.

Apply the above analysis to all possible subsets of the recordings taken three at a time. Suppose all the spectra $S_{x_i x_j}(f), S_{zz}(f), \gamma_{x_i x_j}^2(f), \gamma_{x_i z}(f), S_{x_i x}(f), \gamma_{x_i x}^2(f), i \neq j$, are nonzero for all $i, j$, but $\gamma_{x_i x_j}^2(f) = 0$ for $i \neq j$. 
Then we would suspect strongly that $Z$ drives $X_1, X_2, \ldots, X_n$. The previously described partial coherence spectral analysis can be extended to a large number of data channels to test whether a linear combination of channels drives other channels.

The multichannel coherence spectra and partial coherence spectra are computed by the program SPCTBGTK. The program also plots the partial coherence spectra as well as the coherence spectra.
Bibliography


Doob, J. L., Stochastic Processes, John Wiley & Sons, Inc., 1953, Chaps: 9, 10, 11


APPENDIX A

PROGRAM LISTINGS
MULSPECT

100 * DIMENSION X(NS*LX),R1(LR*NS*NS),W(M),F(M),S(M*NS*NS)
110 * DIMENSION C(M1*NS*NS),TLAG(2*LR-1),Z(2*LR-1),TLAGA(LR)
120 * RS=NUMBER OF CHANNELS
130 * LX=NUMBER OF DATA POINTS FROM EACH CHANNEL
140 * M1=M+1=LENGTH OF TIME LAG
150 * LR=MAXIMUM DESIRED TIME LAG LE. LX
160 * L=SMALLEST INTEGER SUCH THAT LX<2**L
170 * L=MAXIMUM LAG, M=2**((N-1)
180 * N IS DEFINED SO THAT 2*M=2**N
190 * H=LENGTH OF SAMPLING INTERVAL
200 * LNXS=LX*NS
210 * LRNS=LR*NS*NS
220 * MN=NS+NS
230 * DIMENSION X(320,2),R1(50,2,2),W(32),F(33),S(142),C(33,2,2),TLAG(99)
240 & Z(99),TLAGA(50)
250 CHARACTER CH(2),"1","2"
260 DATA NS,LX,M,M1,LR,L,N,LNXS,LRNSNS,IN/2,320,32,33,50,9,6,640,200,2/
270 LIBRARY "REMA", "NLOGN", "CROSS", "MPARZ", "MACOR", "COQUAD", "MOVE"
280 & "NORAG", "COHERE", "OLDPLO", "GFSORT", "BIG", "SMALL", "PLOTTIR"
290 H=1./64.
300 OPENFILE 2,"NTIDAT","NUMERIC"
310 READ(2)X
315 CALL MPARZ(M,N)
320 DO 1 J=1,NS
330 CALL REMAX(LX,X(1,J))
340 CALL MACOR(NS,LX,X,LR,R1,LNXS,LRNSNS,L)
350 CALL COQUAD(NS,M,N,H,R1,S,M1,LR)
360 CALL COHERE(A1,NS,S,C)
370 DO 7 J=1,"1"
380 7 F(J)=J-1
390 DO 500 J=1,NS-1
400 DO 8 K=J+1,NS
410 WRITE(0,300)CH(J),CH(K)
420 300 FORMAT( COHERENCE FOR CHANNELS ',A1', AND ',A1')
430 CALL OLDPLO(1,J,K),F,M)
450 100 FORMAT(1H ,2(F6.2,4X,E9.2,6X)/)
460 WRITE(0,102)
470 102 FORMAT(5(1H ,/))
480 500 CONTINUE
490 DO 105 J=1,NS
500 WRITE(0,107)CH(J)
510 107 FORMAT( AUTOSPECTRA FOR CHANNEL ',A1')
520 CALL OLDPLO(C(1,J,J),F,M)
540 105 WRITE(0,102)
550 DO 700 I=1,NS-1
560 DO 700 J=I+1,NS
570 WRITE(0,901)CH(I),CH(J)
580 901 FORMAT( CROSS CORRELATION BETWEEN CHANNELS ',A1', AND ',A1')
590 DO 108 I=1,LR-1
600 TLAG(L)=L-LR
MULSPECT (continued)

610 108 Z(L)=RI(LR-L+1,J,J)
620 DO 109 L=LR,LR+LR-1
630 TLAG(L)=L-LR
640 109 Z(L)=RI(L-LR+1,J,J)
650 CALL PLOTR(Z,TLAG,LR+LR-1)
660 WRITE(0,102)
670 700 CONTINUE
680 DO 701 K=1,LR
690 701 TLAGA(K)=K-1
700 DO 801 J=1,NS
710 WRITE(0,501)CH(J)
720 501 FORMAT(’ AUTO CORRELATION FOR CHANNEL ’,A1)
730 CALL PLOTR(R1(I,J,J),TLAGA,LR)
740 DO 1 WRITE(0,102)
750 PRINT,X
760 STOP
770 END
**SPECTGTK**

100 * NS = EFFECTIVE NUMBER OF SCANS READ IN
110 * NV = NUMBER OF CHANNELS
120 * NR = NUMBER OF FREQUENCY RANDS (A POWER OF 2)
140 * NSCANS = THE LEAST POWER OF 2 ≥ NS
150 * SR = SAMPLING RATE = 1/N
160 * X = INPUT SERIES (ARRAY)
170 * DIMP = NV * (NV + 1) * (NR + 1)
190 DIMENSION X(1024,4), P(660)
195 REAL F(33), S(33, 4, 4, 4), SP(33)
200 COMPLEX S(33, 4, 4), S113, S223, S123, S112, S332, S132, S221, S331, S231
210 CHARACTER CH(4) = "1", "2", "3", "4"
220 DATA NS, NV, NP, SR, PI/800, 4, 32, 64, 0, 3.14159265/
225 LIBRARY "CCAF"
230 LIBRARY "FAST", "TRANS", "OLDPL0", "GFSORT", "BIG", "SMALL"
240 OPEN 2, "CH_DAT", "NUMERIC"
250 READ(2) ((X(J, I), J = 1, NS), I = 1, NV)
260 * DETREND EACH SERIES BY SUBTRACTING FROM EACH SERIES
270 * ITS LEAST SQUARES LINEAR REGRESSION LINE
280 FNS = NS
290 TBAR = 0.5 * (FNS + 1.)
300 TSUMSO = (FNS * (FNS + 1.) * (FNS + FNS + 1.)) / 6.
310 DO 76 I2 = 1, NV
320 SUM = 0.
330 CRSPRO = 0.
340 DO 77 II = 1, NS
350 SUM = SUM + X(II, I2)
360 77 CRSPRO = CRSPRO + FLOAT(II) * X(II, I2)
370 FMEAN = SUM / FNS
380 BETA = (CRSPRO - FNS * TBAR * FMEAN) / (TSUMSO - FNS * TBAR * TBAR)
390 DO 76 I2 = 1, NV
400 76 X(II, I2) = X(II, I2) - FMEAN - BETA * (II - TBAR)
430 * INDX = EACH SERIES WITH A COSINE TAPER
440 INDX = 10
450 R = NS/10
460 DO 80 II = 1, I2
470 80 IF(II.EQ.INDX) GO TO 80
480 FIN = II - 0.5
490 TINT = 0.5 * (1.0 - COS(PI * FIN / R))
500 I3 = NS + 1 - I2
510 DO 80 I2 = 1, NV
520 X(I2, I2) = INDYX * X(I2, I2)
530 80 X(I3, I2) = INDYX * X(I3, I2)
550 LOG2NS = 0
560 NSCANS = 1
570 54 IF(NS .LE. NSCANS) GO TO 55
580 LOG2NS = LOG2NS + 1
590 NSCANS = NSCANS * NSCANS
600 GO TO 54
620 55 IF(NS .EQ. NSCANS) GO TO 74
SPCTGTK (continued)

630 * ZERO OUT THE LAST PART OF EACH SERIES IF THE NUMBER OF SCANS
640 * IS NOT A POWER OF 2
650 11=11BEGIN,NSCANS
660 DO 75 11=11BEGIN,NSCANS
670 DO 75 12=1,NV
680 75 X(I1,I2)=0.
690 74 CONTINUE
700 IF(MOD(NV,2))70,82,70
710 * IF THE NUMBER OF SERIES IS ODD, FILL A DUMMY SERIES WITH ZEROS
720 70 NVI=NV+1
730 DO 83 II=1,NV
740 83 X(I1,NVI)=0.0
750 GO TO 85
760 82 NVI=NV
770 85 CONTINUE
780 ID1P=NVI*(NVI+1)*(NB+1)
790 CALL TRANS(P,ID1P,X,NSCANS,NVI,NR,LOG2NS)
800 * CROSS SPECTRAL ESTIMATES ARE IN ARRAY P
810 * THE CROSS SPECTRAL ESTIMATES IN ARRAY P ARE SCALED BY MULTIPLYING
820 * BY CI
830 WNDPwr=FNS-1.25*R
840 FSCANS=NSCANS
850 FNR=NR
860 FD=FSCANS/(FNR+FNH)
870 CI=0.25/(SR*(FD+1.)*WNDPwr)
880 IROWSP=NB+NR+2
890 ICOLSP=(NVI*(NVI+1))/2
900 ISIZEP=IROWSP*ICOLSP
910 DO 95 II=1,ISIZEP
920 95 P(I1)=CI*P(I1)
925 NR=NR+1
930 90 DO 200 J=1,NV
935 1K=2*NB1*(NVI*(J-1)-((J-1)*(J-2))/2-J)
940 DO 201 K=J,NV
945 1JK=1K+2*NB1*K
950 DO 200 1=1,NB1
960 200 S(I,J,K)=COMPLEX(P(IJK+I-1),(-1.0)*P(IJK+I+1))
970 DO 201 J=1,NV-1
980 DO 201 K=J+1,NV
990 DO 201 1=1,NR1
1000 DO 201 1=1,NR1
1005 CSS=CCAB(S(I,J,K))**CCAB(S(I,K,K))
1010 IF(CSS<1.0E-07)17,18,18
1020 17 S(I,K,J)=(0.0,0.0)
1030 GO TO 201
1040 18 S(I,K,J)=COMPLEX(CCAB(S(I,J,K))**2/CSS,0.0)
1050 201 CONTINUE
1060 DO 202 J1=1,NV-2
1070 DO 202 J2=J1+1,NV-1
1080 DO 202 J3=J2+1,NV
1090 202 I=I,NB1
SPCTBGTK (continued)

1095 RES=REAL(S(I,J3,J3))
1100 S113=S(I,J1,J1)*(1.0-S(I,J3,J3))
1101 S223=S(I,J2,J2)*(1.0-S(I,J3,J2))
1110 IF(RES-1.0E-07) 901,902,902
1111 901 S123=S(I,J1,J2)
1112 GO TO 903
1120 902 S123=S(I,J1,J2)-S(I,J1,J3)*CONJG(S(I,J2,J3))/RES
1125 GO TO 202
1126 601 W2(I,J1,J2,J3)=0.0
1127 GO TO 202
1130 602 W2(I,J1,J2,J3)=(CCAB(S123)**2)/(CCAB(S112)*CCAB(S332))
1135 RES=REAL(S(I,J1,J1))
1140 S112=S(I,J1,J1)*(1.0-S(I,J2,J1))
1145 S332=S(I,J3,J3)*(1.0-S(I,J3,J2))
1150 IF(RES-1.0E-07) 1001,1002,1002
1151 1001 S132=S(I,J1,J3)
1152 GO TO 1003
1153 1002 S132=S(I,J1,J3)-S(I,J2,J3)*S(I,J1,J3)/RES
1154 GO TO 202
1155 1101 S231=S(I,J2,J3)
1156 GO TO 1103
1157 1102 S231=S(I,J2,J3)-S(I,J1,J3)*CONJG(S(I,J1,J2))/RES
1160 1103 IF(CCAB(S221)*CCAB(S332)-1.0E-07) 701,702,702
1165 701 W2(I,J1,J3,J2)=0.0
1166 GO TO 202
1167 702 W2(I,J1,J3,J2)=(CCAB(S132)**2)/(CCAB(S112)*CCAB(S332))
1172 RES=REAL(S(I,J1,J1))
1175 S221=S(I,J1,J1)*(1.0-S(I,J2,J1))
1178 S331=S(I,J3,J3)*(1.0-S(I,J3,J1))
1180 IF(RES-1.0E-07) 1101,1102,1102
1183 1101 S231=S(I,J2,J3)
1184 GO TO 1103
1190 1102 S231=S(I,J2,J3)-S(I,J1,J3)*CONJG(S(I,J1,J2))/RES
1191 1201 N1=1,N131=1
1192 1202 N1=1,N1
1193 1203 CONTINUE
1194 1204 DO 7 J=1,NB1
1195 7 F(J)=J-1
1196 DO 500 J=1,NV-1
1197 500 DO 500 K=J+1,NV
1198 WRITE(0,300)CH(J),CH(K)
1199 300 FORMAT(' l', coherence for channels ',A1,' and ',A1')
1200 319 CONTINUE
1201 319 SP(I)=REAL(S(I,K,J))
1204 CALL OLDPL0(SP,F,NB)
1205 WRITE(0,102)
1206 102 FORMAT(25(IH,/) )
1208 DO 500 L=1,NV
1209 IF((J-L)*(K-L)) 418,500,418
1210 418 DO 512 I=1,NR1

37
SPCTBGIK (continued)

1288  SP(I)=n2(I,J,K,L)
1293  512 CONTINUE
1295  WRITE(0,301) CH(J),CH(K),CH(L)
1300  301 FORMAT(' PARTIAL COHERENCE BETWEEN CHANNELS ',A1,' AND ',
1310  &A1,/', 'AFTER THE INFLUENCE OF CHANNEL ',A1,' HAS BEEN REMOVED')
1320  CALL OLDPL0(SP,F,NI)
1330  WRITE(0,102)
1335  500 CONTINUE
1340  DO 417 J=1,NV
1350  WRITE(0,317)CH(J)
1360  317 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
1365  IX=2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2)-1
1370  DO 416 I=1,NB1
1390  416 SP(I)=P(IX+I+1)
1410  CALL OLDPL0(SP,I,F,NI)
1420  WRITE(0,102)
1430  STOP
1440  END
SPCTC.LTK

100  *  NS=EFFECTIVE NUMBER OF CHANNELS READ IN
110  *  NV=NUMBER OF CHANNELS
120  *  NR=NUMBER OF FREQUENCY BANDS (A POWER OF 2)
140  *  JSCANS IS ALSO A POWER OF 2
150  *  SR=SAMPLING RATE=1/H
160  *  X=INPUT SERIES(ARRAY)
170  *  P=ARRAY FOR STORING CROSS SPECTRA
180  DIMENSION X(1024,2),P(198),S(132),C(132),F(33)
185  DIMENSION SI(33,2,2)
190  CHARACTER CH(2)="/1","2"/
200  EQUIVALENCE (S,C)
201  EQUIVALENCE (S,SI)
210  DATA NS,NV,NR,JSCANS,SR,P/320,2,32,1024,64.,3,14159265/
215  LIBRARY "FAST","TRANS","MOVE","NORMAG","COHERE","PLOT","GSORT"
216 &,"RIG","SMALL"
220  OPENFILE 2,"NTIDAT","NUMERIC"
230  READ(2)((X(J,I),J=1,NS),I=1,NV)
240  *  DEIREND THE SERIES BY SUBTRACTING FROM EACH SERIES ITS
250  *  LEAST SQUARES LINEAR REGRESSION LINE
260  FNS=NS
270  TPAR=0.5*(FNS+1.0)
280  TSUMSQ=(FNS*(FNS+1.0)*(FNS+FNS+1.0))/6.0
290  DO 76 12=1,NV
300  SUM=0.0
310  CRSPRO=0.0
320  DO 77 11=1,NS
330  SUM=SUM+X(I1,I2)
340  77  CRSPRO =CRSPRO+FLOAT(I1)*X(I1,I2)
350  FMEAN=SUM/FNS
360  FETA=(CRSPRO=FNS*TRAR*FMEAN)/(TSUMSQ=FNS*TRAR*TRAR)
370  DO 78 11=1,NS
380  78  FREG=FMEAN+FETA*(FLOAT(I1)-TRAR)
390  78  X(I1,I2)=X(I1,I2)-FREG
400  76  CONTINUE
410  *  WINDOW EACH SERIES WITH A COSINE TAPER
420  IR=NS/10
430  R=IR
440  DO 79 11=1,IR
450  79  FII=I1
455  FINF=FIT=0.5
460  WINDOW=0.5*(1.0-COS(PFINT/R))
470  I3=NS+1-I1
480  DO 80 12=1,NV
490  80  X(I1,I2)=WINDOW*X(I1,I2)
500  80  X(I3,I2)=WINDOW*X(I3,I2)
510  79  CONTINUE
520  LOG2NS=0
530  NSCANS=1
540  54  IF(NS.LE.NSCANS)GO TO 55
550  LOG2NS=LOG2NS+1

39
SPCTCLIK (continued)

560 NSCANS=NSCANS+NSCANS
570 GO TO 54
580 55 IF(NS.EQ.0,NSCANS)GO TO 74
590 * ZERO OUT THE LAST PART OF EACH SERIES IF THE NUMBER OF
600 * SCANS IS NOT A POWER OF 2
610 IREON=NS+1
620 DO 75 I=1,IREON,NSCANS
630 DO 75 I2=1,NV
640 75 X(I1,I2)=0.0
650 74 CONTINUE
660 IF(WOD(NV,2))70,82,70
670 * IF THE NUMBER OF SERIES IS ODD, FILL A DUMMY SERIES WITH ZEROS
680 70 NVI=NV:
690 80 83 I1=1,NSCANS
700 83 X(I1,NVI)=0.0
710 80 GO TO 85
720 82 NVI=NV
730 85 CONTINUE
740 IDIMP=NVI*(NVI+1)*(NP+1)
750 CALL TRANS(P,DIIMP,X,NSCANS,NVI,NB,LOG2NS)
760 * CROSS SPECTRUM ESTIMATES ARE IN ARRAY P
770 * THE CROSS SPECTRUM ESTIMATES IN ARRAY P ARE SCALED BY
780 * MULTIPLYING BY C
790 *NDPWR=FNS-1.25/R
800 FSCANS=NSCANS
810 FNP=NB
820 FD=FSCANS/(FNB+FNB)
830 CI=0.25/(SR*(FD+1.0)*NDPWR)
840 IROWSP=NR+NB+2
850 ICOLSP=(NVI*(NVI+1))/2
860 ISIZEP=IROWSP*ICOLSP
870 95 I1=1,ISIZEP
880 95 P(I1)=C1*P(I1)
890 90 NR1=NR+1
900 DO 1000 J1=1,NVI
910 DO 1000 K=J1,NVI
920 DO 1000 I=1,NB1
940 IF(J-K).LT.99,1000,1000
960 1000 CONTINUE
970 CALL COHERE(NR1,NVI,S,C)
980 DO 7 J1=1,NBI
990 7 F(J1)=J1
1000 WRITE(0,300)CH(J),CH(K)
1010 300 FORMAT(* COHENCE FOR CHANNELS ',A1,' AND ',A1)
1020 CALL PLUTOCC(1,NR1*(J-1)+NR1*NVI*(K-1)),F,NB)
1040 100 FORMAT(1H ,F5.0,4X,E9.2/)

40
SPCCLTK (continued)

1050 WRITE(0,102)
1060 102 FORMAT(5(1H,/,))
1070 500 CONTINUE
1080 DO 105 J=1,NV
1090 WRITE(0,107)CH(J)
1100 107 FORMAT( ' AUTOSPECTRA FOR CHANNEL ',A1)
1110 CALL PLOT(C(1+NB1*(J-1)+NB1*NVI*(J-1)),F,NB)
1120 105 CONTINUE
1130 STOP
1140 END
AUTO REG

100 LIBRARY "NLOCM", "CROSS", "REMAV", "LEVSN", "FFT", "CCAB", "OLDPLO", "GFST" 
101 & "PIG", "SMALL"
110 * LR=2*NR
120 DIMENSION X(320), R1(64), A(64), G(33), F(33)
130 COMPLEX AC(64)
140 * 2**LL IS THE SMALLEST POWER OF 2 WITH LX.LE.2**LL
150 DATA LX, NR, LR, LL, L1/320, 32, 64, 6, 6/
160 DATA H/0.015625/
170 OPEN FILE 3, "NPUR", "NUMERIC"
180 READ(3)(X(I), XX, I=1, 320)
190 CALL REMAV(LX, X)
200 CALL CROSS(LX, X, LR, R1, LL)
210 CALL LEVSN(LR, R1, A, S, M)
220 NR1=NR+1
230 M1=M+1
240 DO 2 L=1, M1
250 2 AC(L)=COMPLEX(A(L), 0.0)
260 M2=M1+1
265 NB2=NR+NP2
270 DO 11 L=M2, NB2
280 11 AC(L)=(0.0, 0.0)
290 * 2*NB=2**L1
300 CALL FFT(AC, L1)
310 DO 3 K=1, NB1
320 3 G(K)=(H*S)/ABS(AC(K))**2
330 DO 1 J=1, 33
340 1 F(J)=J-1
350 CALL OLDPLO(G, F, NP)
370 100 FORMAT(1HO, F5.2, 4X, E9.2/)
380 END
MAUTOREG

100 LIBRARY "CROSS", "MOVE", "NORMAG", "REMAV", "OLDPLO"
101 &."GFSORT", "RIG", "SMALL", "RZERO", "ZERO", "COHERE", "MAINV"
102 &."MOVEC", "BRAINY", "MATMUL", "FFT"
103 &."MULLEV"
110 * LR=2*NR, LXNS=LX*NS
120 * LM=SMALLEST INTEGER SUCH THAT LX.LE.2**LM
125 DIMENSION BB(2047,2)
130 DIMENSION X(1024,2), R1(2,2,64), A(2,2,64), AP(2,2,64), R(2,2,64)
131 &., B(2,2,64), VA(2,2), VB(2,2), DA(2,2), DR(2,2), CA(2,2), CB(2,2)
132 &., S(33,2,2), C(33,2,2), F(33)
140 COMPLEX AC(64), Sl(2,2,33), CDR(2,2), ST(2,2)
141 &., CVA(2,2)
150 EQUIVALENCE (S,C)
160 EQUIVALENCE (NS,NV)
170 DATA LX, LXNS/1024,2048/
180 DATA NS, NR, LR, LI, LM/2, 32, 64, 6, 10/
190 OPEN FILE 2, "NPUB", "NUMERIC"
200 READ (2) (PPR(J, I), J=1, 2047)
210 OPEN FILE 3, "NCH2", "NUMERIC"
220 READ (3) (BPRJ, J, I) = 1, 2047)
222 DO 199 J = 1, 2
224 DO 199 J = 1, 1024
226 199 X(J, I) = PPR(1+2*(J-1), I)
230 DO 7 J = 1, NS
240 7 CALL REMAV(LX, X(1, J))
250 CALL MAC(NS, LX, X, LR, R1, LXNS, LM)
260 CALL MULLEV(NS, LR, R1, A, AP, B, RP, VA, VB, DA, DR, CA, CB, M)
270 DO 5 I = 1, NS
280 DO 5 J = 1, NS
290 5 CVA(I, J) = CMPLX(VA(I, J), 0.0)
- * NR1=NR+1
310 DO 1 I = 1, NS
320 DO 1 J = 1, NS
330 1 M1=M+1
340 DO 2 L = 1, M1
350 2 AC(L) = CMPLX(A(I, J, L), 0.0)
360 M2=M1+1
365 NR2=NR+NR
370 DO 11 L=M2, NR2
380 11 AC(L) = (0.0, 0.0, 0.0)
390 * 2*NR2=2**L1
400 CALL FFT(AC(L1))
410 DO 1 K = 1, NBI
420 1 SI(K, I, K) = AC(K)
430 DO 4 K = 1, NBI
440 CALL MAINV(NS, SI(I, K), CDB)
450 DO 3 I = 1, NS
460 DO 3 J = 1, NS
470 3 SI(I, J, K) = CMPLX(CDR(J, I))

43
MAUTOREG (continued)

480 CALL MATMUL(NS,CDR,CVA,ST)
490 CALL MATMUL(NS,ST,SI(1,1,K),CDR)
500 DO 4 I=1,NS
510 DO 4 J=1,NS
520 4 SI(I,J,K)=CDR(I,J)
530 DO 10 I=1,NS
540 DO 10 J=1,NS
550 DO 10 K=1,NBI
560 S(K,I,J)=REAL(SI(I,J,K))
570 IF(J-I)9,10,10
580 9 S(K,J,I)=-AIMAG(SI(I,J,K))
590 10 CONTINUE
600 CALL COHERE(NR1,NS,S,C)
610 DO 97 J=1,NBI
620 97 F(J)=J-1
630 DO 500 3=1,NV-1
640 DO 500 K=3+1,NV
650 WRITE(0,300)CH(J),CH(K)
660 300 FORMAT(' COHERENCE FOR CHANNELS ',AI,' AND ',AI)
670 CALL OLDPLP(C(I,J,K),F,NB)
680 104 FORMAT(9H,5F5.0,4X,E9.2/)
690 WRITE(0,102)
700 102 FORMAT(25(1H ,/))
710 500 CONTINUE
720 DO 105 J=1,NV
730 WRITE(0,102)
740 107 FORMAT(' AUTO SPECTRA FOR CHANNEL ',AI)
750 CALL OLDPLP(C(I,J,J),F,NB)
760 105 CONTINUE
770 STOP
1, ** END

1200 SUBROUTINE ESCAL(N,A,B)
1210 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
1220 DIMENSION A(N,N),R(N,N)
1230 DO 5 I=1,N
1240 DO 5 J=1,N
1250 5 R(I,J)=0.0
1260 B(I,1)=1./A(I,1)
1270 IF(N.EQ.1)RETURN
1280 DO 40 M=2,N
1290 K=M-1
1300 EK=A(M,M)
1310 DO 10 J=1,K
1320 DO 10 I=1,K
1330 10 ED=EK-A(M,I)*R(I,J)*A(J,M)
1340 B(M,M)=1./EK
1350 DO 30 I=1,K
1360 DO 20 J=1,K
1370 20 R(I,M)=B(I,M)-S(I,J)*A(J,M)/EK
MAUTREG (continued)

1380 30 \( R(A, I) = R(I, M) \)
1390 DO 40 I=1,K
1400 DO 40 J=1,K
1410 40 \( R(I, J) = R(I, J) + R(I, M) \times R(M, J) \times EK \)
1420 RETURN
1430 END
1500 SUBROUTINE FAPDEJ(N,A,AINV,DET,ADJUG,P)
1510 LIBRARY "CCAR"
1520 DIMENSION A(N,N), AINV(N,N), ADJUG(N,N), P(N)
1530 COMPLEX A, AINV, DET, ADJUG, P
1540 COMPLEX PN
1550 NN=N*N
1560 CALL MOVEC(NN,A,AINV)
1570 DO 4 K=1,N
1580 4 P(K)=(0.0,0.0)
1590 DO 2 I=1,N
1600 2 P(K)=P(K)+AINV(I,I)
1610 P(K)=P(K)/FLOAT(K)
1620 IF(K.EQ.N)GO TO 5
1630 CALL MOVEC(N*N, AINV, ADJUG)
1640 DO 3 I=1,N
1650 3 ADJUG(I,I)=AINV(I,I)-P(K)
1660 CALL BRATNY(N,N,I,A,N,N,I,ADJUG,AINV)
1670 5 CALL MOVEC(N*N, ADJUG, AINV)
1680 E30=1.0E-30
1690 IF(CCAR(P(N)).LT.E30)GO TO 7
1700 DO 6 I=1,N
1710 DO 6 J=I,N
1720 P(N)=P(N)
1730 6 AINV(I,J)=AINV(I,J)/PN
1740 7 DET=P(N)
1750 IF(MOD(N,2).EQ.1)RETURN
1760 DET=-DET
1770 DO 8 I=1,N
1780 DO 8 J=I,N
1790 8 ADJUG(I,J)=-ADJUG(I,J)
1800 RETURN
1810 END
1900 SUBROUTINE ESCALD(N,A,B,DET)
1910 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
1920 * ALSO COMPUTES DETERMINANT OF A
1930 DIMENSION A(N,N), B(N,N)
1940 DO 5 I=1,N
1950 DO 5 J=I,N
1960 5 R(I,J)=0.0
1970 R(I,I)=A(I,I)
1980 DET=A(I,I)
1990 IF(N.EQ.1)RETURN
2000 DO 40 M=2,N
2010 K=M-1
MAUTOREG (continued)

2020 EK = A(M, I) * R(I, J) * A(J, M)
2030 DO 30 I = 1, K
2040 DO 20 J = 1, K
2050 EK = EK - A(M, I) * R(I, J) * A(J, M)
2060 DET = DET * EK
2070 DO 10 3 = 1, 3
10 EK = EK - A(M, I) * B(I, J) * A(J, M)
2080 EK = EK / DET
2090 DO 30 I = 1, K
30 P(I, M) = B(I, M) - R(I, J) * A(J, M) / EK
2100 DO 40 J = 1, K
40 R(I, J) = B(I, J) + R(I, M) * B(M, J) / EK
2110 RETURN
2120 END

SUBROUTINE STMEO(M, N, A, B, C)
2200 NMAX = LARGEST VALUE OF N TO BE PROCESSED
2210 * NONDUMMY DIMENSION S(NMAX, NMAX)
2220 FOR EXAMPLE, IF NMAX = 4 THEN
2230 DIMENSION S(4, 4)
2240 DIMENSION A(M, N), B(N, N), C(M, N)
2250 CALL MOVE(N*N, B, S)
2260 CALL ESCALD(S, R)
2270 DO 1 I = 1, M
1 DO 2 J = 1, N
2 A(I, J) = A(I, J)
3 DO 4 K = 1, N
4 A(I, J) = A(I, J) + C(I, K) * B(K, J)
5 CALL MOVE(N*N, S, B)
2300 RETURN
2350 END

SUBROUTINE SIMEOD(M, N, A, B, C)
2400 NMAX = LARGEST VALUE OF N TO BE PROCESSED
2410 * NONDUMMY DIMENSION A(NMAX, NMAX)
2420 FOR EXAMPLE, IF NMAX = 4 THEN
2430 DIMENSION A(4, 4)
2440 DIMENSION A(M, N), B(N, N), C(M, N)
2450 CALL MOVE(N*N, B, S)
2460 CALL ESCALD(N, S, R)
2470 DO 1 I = 1, M
1 DO 2 J = 1, N
2 A(I, J) = A(I, J)
3 DO 4 K = 1, N
4 A(I, J) = A(I, J) + C(I, J) * B(K, J)
5 CALL MOVE(N*N, S, B)
2500 RETURN
2550 END

SUBROUTINE MAC(NS, LX, X, LR, RI, LXNS, L)
2600 * SUBROUTINE MAC COMPUTES THE MULTICHANNEL AUTOCORRELATION OF
2610 * THE NS CHANNEL TIME SERIES X
MAUTOREG (continued)

2630 * LX=LENGTH OF EACH TIME SERIES IN EACH CHANNEL
2640 * NS=NUMBER OF CHANNELS
2650 * LR=DESIRED LENGTH OF CORRELATION, LR.LE.2**L
2660 * L IS SMALLEST INTEGER SUCH THAT LX.LE.2**L
2670 * LXNS=LX*NS
2680 * NDUMMAY DIMENSION S(LF), WHERE LF.GE.ANTICIPATED LR
2690 DIMENSION X(LXNS),R1(NS,NS,LR)
2700 DIMENSION S(64)
2710 DO 1 I=1,NS
2720 J1=I+(I-1)*LX
2730 DO 1 J=1,NS
2740 J1=J+(J-1)*LX
2750 CALL CROSS(LX,X(I1),X(J1),LR,S,L)
2760 DO 1 K=1,LR
2770 R1(J1,K)=S(K)
2780 RETURN
2790 END
BIG

100 FUNCTION BIG(A,M)
110 DIMENSION A(M)
120 R=A(1)
130 DO 1 K=2,M
140 IF(A(K)-R)1,1,2
150 2 R=A(K)
160 1 CONTINUE
165 BIG=R
170 RETURN
180 END
SUBROUTINE BRAINY(NRA, NCA, LA, A, NRB, NCR, LB, B, C, LC)

COMPLEX A, B, C

LC = LA + LR - 1

CALL ZERO(NRA*NCR*LC, C)

DO 1 I = 1, LA
1  DO 1 J = 1, LR
    DO 1 K = I + J - 1

DO 1 M = 1, NRA
1  DO 1 N = 1, NCB
2  DO 1 L = 1, NCA

C(M, N, K) = C(M, N, K) + A(M, L, I) * B(L, N, J)

RETURN

END
CCAB

100 FUNCTION CCAB(X)
110 COMPLEX X
120 CCAB=(REAL(X)**2+AIMAG(X)**2)**0.5
130 RETURN
140 END
COHERE

100 SUBROUTINE COHERE(M1,NS,S,C)
110 DIMENSION S(M1,NS,NS),C(M1,NS,NS)
121 * EQUIVALENCE (S,C) IS ALLOWED
130 * SUBROUTINE COHERE COMPUTES THE MAGNITUDE AND PHASE ANGEL OF
140 * THE COHERENCY, AS WELL AS AUTOSPECTRA, EACH SCALED TO HAVE ITS LARGEST
150 * VALUE UNITY
160 DO 10 JP=2,NS
170 J=JP-1
180 DO 10 K=JP,NS
190 DO 10 J=1,NS
195 IF(S(I,J,J)*S(I,K,K).EQ.0.0)GO TO 10
205 IF(ABS(S(I,J,K)).LT.1.0E-07)GO TO 101
210 PH=ATAN2(S(I,J,K),S(I,J,K))
220 102 C(I,J,K)=C0
230 10 C(I,K,J)=180.*PH/3.14159265
240 DO 20 J=1,NS
250 CALL MOVE(M1,S(I,J,J),C(I,J,J))
260 20 CALL NORMAG(M1,C(I,J,J))
270 RETURN
280 101 PH=SIGN(1.5707963,S(I,K,J))
290 60 TO 102
300 END
COQUAD

100 SUBROUTINE COQUAD(H, NS, M, N, RI, S, MI, LR)
110 * SUBROUTINE COQUAD COMPUTES THE MATRIX OF EMPIRICAL AUTOSPECTRA,
120 * COSPECTRA, AND QUADRATURE SPECTRA FROM THE MULTI-CHANNEL
130 * AUTO CORRELATION FUNCTION
140 * NS=NUMBER OF TIME SERIES OR CHANNELS
150 * M=2**(N-1), MI=M+1=TIME LENGTH OF CORRELATION
160 * W(I)=1, W(I)=0
170 * DIMENSION C(NM), WHERE NM IS NONDUMMY DIMENSION >=TWICE
180 * THE MAXIMUM LAG M
190 DIMENSION w(I), RI(LR, NS, NS), S(MI, NS, NS)
200 COMPLEX C(100)
210 DO 20 J=1, NS
220 DO 20 K=J, NS
230 DO 10 I=1, M
240 EVEN=RI(I, J, K)+RI(I, K, J)
250 ODD=RI(I, J, K)-RI(I, K, J)
260 RI(I, K, J)=(I)*ODD
270 10 RI(I, J, K)=RI(I, J, K)*0.5
280 DO 40 J=1, NS
290 DO 40 K=J, NS
300 DO 1 I=1, M
310 C(I)=CMPLX(RI(I, J, K), RI(I, K, J))
320 2 DO 2 I=M1+M
330 2 C(I)=(0., 0.)
340 CALL NLOG(N, C, -1., M+4)
350 S(I, J, K)=H*REAL(C(I))
360 DO 3 I=2, MI
370 3 S(I, J, K)=H*(REAL(C(I))+REAL(C(M1+M1-I)))
380 IF(J-K)7, 40, 40
390 7 DO 4: I=2, M1
400 4 J=I, K=J
410 4 CONTINUE
420 * S(I, J, K) IS THE COSPECTRAL DENSITY OF THE JTH AND KTH
430 * CHANNEL EVALUATED AT LAG I-1 IF K>J, AND EQUAL TO
440 * QUAD SPECTRAL DENSITY EVALUATED AT LAG I-1 IF K<J, FOR J=1, M1
450 RETURN
500 END
SUBROUTINE CROSS(LENX,X,Y,LR,RI,L)

* DIMENSION XX(NMAX),YY(NMAX)

* COMPLEX CX(NMAX),CY(NMAX),CONMAX

* NMAX IS A NONUNARY DIMENSION=2**(L+1), WHERE 2**L IS THE

* SMALLEST POWER OF 2 SUCH THAT LENX<=2**L

DIMENSION X(LENX),Y(LENX),RI(LR)

COMPLEX CX(2048),CY(2048),CR(2048)

EQUIVALENC (CX,C)

* LR <= LENX

LIBRARY "NLOGN"

L2=2**(L+1)

DO 1 J=1,LENX

CJ=CMPLX(X(J),0.0)

1 CONTINUE

DO 2 J=LENX+1,L2

CJ=CMPLX(Y(J),0.0)

2 CONTINUE

CALL NLOGN(L1,CX,-1.0,L2)

CALL NLOGN(L1,CY,-1.0,L2)

DO 4 J=1,L2

C(J)=CONJG(CX(J))*CY(J)

4 CONTINUE

CALL NLOGN(L1,CR,1.0,L2)

DO 5 J=1,LR

RI(J)=REAL(C(J))/FLOAT(LENX)

5 CONTINUE

* RI(J) = THE CROSS CORRELATION OF X AND Y EVALUATED AT

* LAG(J-1), J=1,2,...,M+1

RETURN

END
DETRPEN

100 SUBROUTINE DETREN(NS,NV,SCANS)
110  NS=NUMBER OF SCANS
120  THE FOLLOWING IS USED TO DETREN D A TIME SERIES
130  BY SUBTRACTING ITS LEAST SQUARES REGRESSION LINE
140  FROM EACH CHANNEL OF AN NV CHANNEL TIME SERIES
150 DIMENSION SCANS(NS,NV)
160  FNS=NS
170  TBAR=0.5*(FNS+1.0)
180  TSUMSQ=(FNS*(FNS+1.0)*(2.0*FNS+1.0))/6.0
190  DO 76  I2=1,NV
200   SUM=0.0
210  CRSPRO=0.0
220  DO 77  I1=1,NS
230   SUM=SUM+SCANS(I1,I2)
240  CRSPRO=CRSPRO+FLOAT(I1)*SCANS(I1,I2)
250  77 CONTINUE
260  FMEAN=SUM/FNS
270  BETA=(CRSPRO-FNS*TBAR*FMEAN)/(TSUMSQ-FNS*TBAR*TBAR)
280  DO 78  I1=1,NS
290   FREG=FMEAN+BETA*(FLOAT(I1)-TBAR)
300  SCANS(I1,I2)=SCANS(I1,I2)-FREG
310  78 CONTINUE
320  76 CONTINUE
330  RETURN
340 END

54
ESCAL

100 SUBROUTINE ESCAL(N,A,B)
110 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
120 DIMENSION A(N,N),B(N,N)
130 DO 5 I=1,N
140 DO 5 J=1,N
150 5 R(I,J)=0.0
160 R(1,1)=1./A(1,1)
170 IF(N,F0.1)RETURN
180 DO 40 M=2,N
190 K=M-1
200 EK=A(M,M)
210 DO 10 J=1,K
220 DO 10 I=1,K
230 10 ED=EK-A(I,I)*R(I,J)*A(J,M)
240 R(I,M)=1./EK
250 DO 30 I=1,K
260 DO 20 J=1,K
270 20 R(I,M)=R(I,M)-R(I,J)*A(J,M)/EK
280 30 R(M,I)=R(I,M)
290 DO 40 I=1,K
300 DO 40 J=1,K
310 40 R(I,J)=R(I,J)+R(I,M)*R(M,J)*EK
320 RETURN
330 END
ESCALD

100 SUBROUTINE ESCLAD(N, A, B, DET)
110 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
120 * ALSO COMPUTES DETERMINANT OF A
130 DIMENSION A(N,N), B(N,N)

140 DO 5 I=1,N
150 DO 5 J=1,N
160 B(I,J)=0.0
170 R(I,I)=1./A(I,I)
180 DET=A(I,I)
190 IF(N.EQ.1) RETURN
200 DO 40 M=2,N
210 K=M-1
220 EK=A(K,K)
230 DO 10 I=1,K
240 DO 10 J=1,J
250 10 EK=EK-A(M,J)*B(I,J)*A(J,M)
260 DET=DET*EK
270 R(M,N)=1./EK
280 DO 30 I=1,K
290 DO 30 J=1,K
300 20 R(I,M)=R(I,M)-B(I,J)*A(J,M)/EK
310 30 R(M,I)=R(I,M)
320 DO 40 I=1,K
330 DO 40 J=1,K
340 40 R(I,J)=B(I,J)*R(I,M)*B(M,J)*EK
350 RETURN
360 END
FADDEJ

2400 SUBROUTINE FADDEJ(N,A,AINV,DET,ADJUG,P)
2410 LIBRARY "CCAB"
2420 DIMENSION A(N,N),AINV(N,N),ADJUG(N,N),P(N)
2430 COMPLEX A,AINV,DET,ADJUG,P
2440 COMPLEX PN
2450 NN=4*N
2460 CALL MOVEC(N1,A,AINV)
2470 DO 4 K=1,N
2480 P(K)=(0.0,0.0)
2490 DO 2 I=1,N
2500 2 P(K)=P(K)+AINV(I,I)
2510 P(K)=P(K)/FLOAT(K)
2520 IF(K.EQ.N)GO TO 5
2530 CALL MOVEC(N*N,AINV,ADJUG)
2540 DO 3 J=1,N
2550 3 ADJUG(I,J)=AINV(I,J)-P(K)
2560 4 CALL BRAINV(N,N,1,A,N,N,1,ADJUG,AINV)
2570 5 CALL MOVEC(N*N,ADJUG,AINV)
2580 E30=1.0E-30
2590 IF(CCAB(P(N)).LT.E30)GO TO 7
2600 DO 6 I=1,N
2610 DO 6 J=1,N
2620 PN=P(N)
2630 6 AINV(I,J)=AINV(I,J)/PN
2640 7 DET=P(N)
2650 IF(NOD(N,2).EQ.1)RETURN
2660 DET=-DET
2670 DO 8 I=1,N
2680 DO 8 J=1,N
2690 8 ADJUG(I,J)=-ADJUG(I,J)
2700 RETURN
2710 END
SUBROUTINE FAST(XI,YI,M,N)
110 * SUBROUTINE FAST OBTAINS FINITE FOURIER TRANSFORMS OF THE PAIR OF
120 * SERIES X AND Y
130 DIMENSION X(1024), Y(1024), XI(N), YI(N)
133 DO 19 J=1,N
134 XI(J)=XI(J)
135 19 Y(J)=YI(J)
140 N=2**M
150 DO 1 L=1,M
160 IMAX=2**(J-L)
170 JDELT=2*IMAX
180 ARG=6.2831853/FLOAT(JDELT)
190 C=COS(ARG)
200 S=SIN(ARG)
210 U=1.0
220 V=0.0
230 DO 1 I=1,IMAX
240 DO 2 J=1,N,JDELT
250 K=J+JDELT
260 XJ=X(J)+X(K)
270 YJ=Y(J)+Y(K)
280 XK=X(J)-X(K)
290 YK=Y(J)-Y(K)
300 X(K)=U*XK-V*YK
310 Y(K)=U*YK+V*XK
320 X(J)=XJ
330 2 Y(J)=YJ
340 1 UT=C*U-S*V
350 UT=C*U+S*V
360 V=UT+S*U
370 1 U=UT
380 J=1
390 NT=N/2
400 IMAX=N-1
410 DO 3 I=1,IMAX
420 IF(I.GE.J) GO TO 5
430 XI(I)=XI(J)
440 3 XI(I)=XI(I)
450 XI(I)=XI
460 YI=YT
470 Y(J)=Y(I)
480 Y(J)=YT
490 K=NT
500 4 IF(K.GE.J) GO TO 3
510 J=K
520 K=K/2
530 GO TO 4
540 3 J=J+K
542 DO 21 J=1,N
544 XI(J)=XI(J)
546 21 YI(J)=YI(J)
FAST (continued)

550 RETURN
560 END
FFT

90 SUBROUTINE FFT(X,M)
100   COMPLEX X(1024), U, W, T
110   N=2**M
120   NV2=N/2
130   NM1=N-1
140   J=1
150   DO 7 I=1,NM1
160   IF(I,GE,J) GO TO 5
170   T=X(J)
180   X(J)=X(I)
190   X(I)=T
200   K=NV2
210   IF(K,GE,J) GO TO 7
220   J= J-K
230   K=K/2
240   GO TO 6
250   J=J+K
260   PI=3.14159265358979
270   DO 20 L=1,M
280   LE=2**L
290   LE1=LE/2
300   U=(1.0,0.0)
310   K=COMPLEX(COS(PI/FLOAT(LE1)), (+1.0)*SIN(PI/FLOAT(LE1))))
320   DO 20 J=1,LE1
330   DO 10 I=J,N,LE
340   IP=I+LE1
350   T=X(IP)*U
360   X(IP)=X(I)-T
370   10 X(I)=X(I)+T
380   20 U=U*W
390   RETURN
400   END
FILTER

100 SUBROUTINE FILTER(LX,NWT,AN,WT,X,LY,Y,LN2N)
110 DIMENSION AN(NWT),WT(LX),X(LX),Y(LY)
120 * LY.LE.LX-NWT+1
130 * NWT=2*L+1, 2**LN2N.GE.LX
140 DO 1 J=1,NWT
150 1 W(J)=AA(J)
160 DO 2 J=NWT+1,LX
170 2 W(J)=0.0
180 CALL CROSS(LX,WT,X,LY,Y,LN2N)
190 FN=LX
200 DO 3 J=1,LY
210 3 Y(J)=FN*Y(J)
220 RETURN
230 END
SUBROUTINE FTFLIN(A,NW1,NB2,AF,M,NP1)
10 DIMEsITION A(NW1),AT(NP1)
20 COMPLEX AF(NR2)
30 NR2=2*NB2=2*NM.GE.NW1, NP1=NB+1
40 AF(1)=CMPLX(A(1),0.0)
50 DO 1 J=2,NW1
60 1 AF(J)=CMPLX(2*AF(J),0.0)
70 DO 2 J=NW1+1,NB2
80 2 AF(J)=(0,0,0,0)
90 CALL FRT(AF,M)
100 DO 3 J=1,NP1
110 3 AT(J)=REAL(AF(J))
120 RETURN
130 END
GENFLT

100 SUBROUTINE GENFLT(N,NM1,X,Y,L,O,L2)
110 DIMENSION A(NM1),O(L2),X(L),Y(L)
120 DATA N/6,2831853/
130 * NM1=NM+1, L2.GE.L1
140 LI=L+1
150 DO 5 K=1,L
160 5 X(K)=H*K*X(K)
170 DO 1 I=2,L
180 1 Q(I)=Y(I-1)/Y(I-1)/(X(I)-X(I-1))
190 Q(I)=0.
200 Q(L1)=0.
210 DO 2 I=2,NM1
220 T=r*FL(!AT(I-1)
230 TT=T*T
240 A(I)=T
250 DO 3 J=2,L1
260 3 A(J)=A(J)*0(J-1)-0(J))*COS(T*X(J-1))/TT
270 2 A(I)=(A(I)+(Y(L)*SIN(T*X(L)))/Y(I)*SIN(T*X(L)))/T)*2
280 T=2,0*(Y(L)*X(L)-Y(I)*X(I))
290 DO 4 J=2,L
300 4 T=(Y(J)-Y(J-1))*(X(J)+X(J-1))
310 A(I)=T
320 RETURN
330 END
GENWTS

100 SUBROUTINE GENWTS(ID,L,WTS,NW2)
110* NW2=2*L+1
120 DIMENSION WTS(NW2)
130 DATA PI/3.1415927/
140 IF(L.EQ.0) GO TO 21
150 FX=L/10-2
160 FKK=FK+0.5
170 FL=L
180 DO 22 IS=1,L
190 S=IS
200 INDEX1=(L+1)+IS
210 HS=(1.0+COS(PI*S/FL))/(4.0*FL)
220 WTS(INDEX1)=HS*SIN(PI*FKK*S/FL)/SIN(PI*S/(2.0*FL))
230 DO 23 IS=1,L
240 INDEX1=L+1-IS
250 INDEX2=L+1+IS
260 WTS(INDEX1)=WTS(INDEX2)
270 WTS(L+1)=FKK/FL
280 RETURN
290 21 WTS(1)=1.0
300 RETURN
310 END
GFSORT

100 SUBROUTINE GFSORT(G,F,N)
110 DIMENSION G(M),F(M)
120 N=,
130 20 $N=N/2$
140 IF $N=30,40,30$
150 30 $K=M-N$
160 J=1
170 40 $I=J$
180 40 $L=J+N$
190 IF $G(I)-G(L))50,60,60$
200 50 $E=G(I)$
210 $G(I)=G(L)$
220 $G(L)=E$
230 $E=F(L)$
240 $F(I)=F(L)$
250 $F(L)=E$
260 $I=I-N$
270 IF $I-1)60,49,49$
280 50 $J=J+1$
290 IF $(J-K)41,41,20$
300 40 RETURN
310 END

65
SUBROUTINE LEVNSN(LR,R,A,S,M)
DIMENSION R(LR),A(LR)
V=R(1)
D=R(2)
A(1)=1.0
B=0
IF(LR.EQ.1)RETURN
DO 4 L=2,LR
A(L)=-D/V
S=V
IF(L.EQ.2)GO TO 2
L1=(L-2)/2
L2=L1+1
J=2,L2
HOLD=A(J)
K=L-J+1
A(J)=A(J)+A(L)*A(K)
A(K)=A(K)+A(L)*HOLD
IF(L1.EQ.L-2)GO TO 2
A(L2+1)=A(L2+1)+A(L)*A(L2+1)
V=V+A(L)*L
A=M+1
PRINT,I,V
IF>((S-V)/V -0.05)GO TO 5,6
IF(A.GE.15)RETURN
IF(L.EQ.LR)RETURN
D=0.0
DO 4 I=1,L
K=L-1+2
D=D+A(I)*R(K)
4 RETURN
END
MAC

1200 SUBROUTINE MAC(NS,LX,X,LR,RI,LXNS,L)
1210 * SUBROUTINE MAC COMPUTES THE MULTICHANNEL AUTOCORRELATION OF
1220 * THE NS CHANNEL TIME SERIES X
1230 * LX=LENGTH OF EACH TIME SERIES IN EACH CHANNEL
1240 * NS=NUMBER OF CHANNELS
1250 * LR=DESIRED LENGTH OF CORRELATION, LR.LE.2**L
1260 * L I: SMALLEST INTEGER SUCH THAT LX.LE.2**L
1270 * LXNS=LX*NS
1280 * NONDUMMY DIMENSION S(LP), WHERE LP.GE.ANTICIPATED LR
1290 DIMENSION X(LXNS),RI(NS,NS,LR)
1300 DIMENSION S(64)
1310 DO I=1,NS
1320 I1=I+(I-1)*LX
1330 DO J=1,NS
1340 J1=J+(J-1)*LX
1350 CALL CROSS(LX,X(I1),X(J1),LR,S,L)
1360 DO K=1,LR
1370 RI(J,I,K)=S(K)
1380 RETURN
1390 END
MACOR

100 SUBROUTINE MACOR(NS,LX,X,LR,RI,LXNS,LRNSNS,L)
110 * SUBROUTINE MACOR COMPUTES THE MULTI CHANNEL AUTO CORRELATION
120 * OF THE NS-CHANNEL TIME SERIES X
130 * LX= LENGTH OF EACH TIME SERIES IN EACH CHANNEL
140 * NS=NUMBER OF CHANNELS
150 * LR=DESIZED LENGTH OF CORRELATION, LR .LE. (LX-1)
160 * L IS THE SMALLEST INTEGER SUCH THAT LX.LE.2**L
170 * LXNS=LX*NS, LRNSNS=LR*NS*NS
180 DIMENSION X(LXNS),R1(LRNSNS)
190 DO 1 I=1,NS
200 1 J=1+(I-1)*LX
210 DO 1 J=1,NS
220 J=1+(J-1)*LX
230 J=1+LX*(I-1)+LR*NS*(J-1)
240 1 CALL CROSS(LX,X(I),X(J),LR,RI(I,J),L)
250 RETURN
260 END
MAINV

1500 SUBROUTINE MAINV(N, A, R)
1510 COMPLEX A, R, DET, ADJUG, P
1520 DIMENSION ADJUG(4,4), P(4)
1530 DIMENSION AN, RN, PN, N
1540 CALL FAPDEJ(N, A, R, DET, ADJUG, P)
1550 RETURN
1560 END
SUBROUTINE MATMUL(N,A,B,C)
COMPLEX A(N,N),B(N,N),C(N,N)
DO 1 I=1,N
DO 2 J=1,N
C(I,J)=(0,0)
DO 3 K=1,N
1 C(I,J)=C(I,J)+A(I,K)*B(K,J)
RETURN
END
MOVE

100 SUBROUTINE MOVE(LX,X,Y)
110 DIMENSION X(LX),Y(LX)
120 DO 1 I=1,LX
130 Y(I)=X(I)
140 RETURN
150 END
SUBROUTINE MOVEC(LX,X,Y)
DIMENSION X(LX),Y(LX)
COMPLEX X,Y
DO 1 I=1,LX
1 Y(I)=X(I)
RETURN
END

DIMENSION R(N,N,LF),A(N,N,LF),AP(N,N,LF),B(N,N,LF),VP(N,N,LF)
& R(N,N,LF),VA(N,N),VB(N,N),DA(N,N),DB(N,N),CA(N,N),CB(N,N)

CALL RZERO(N,N,LF,A)
CALL RZERO(N,N,LF,B)
DO 2 I=1,N
DO 1 J=1,N
VA(I,J)=R(I,J,1)
VR(I,J)=R(I,J,1)
A(I,I,1)=1.
P(I,I,1)=1.
CALL ESCALD(N,VA,CB,1)
M=0
IF(LF.EQ.1)RETURN
DO 8 L=2,LF
CALL RZERO(N,N,DA)
DO 5 I=1,N
DO 4 LI=1,L
LD=L-LI+1
DO 3 K=1,N
DA(I,J)=DA(I,J)-A(I,K,LI)*R(K,J,LD)
4 CONTINUE
DO 5 3=1,0
VP(J,I)=VP(J,I)-CA(I,K)*VA(K,J)
5 CONTINUE
CALL SRMEO(N,CA,VP,DA)
CALL SRMEO(N,CR,VP,DB)
IF((DV-DETVA)/DETVA-0.05)100,100,200
IF(M.GE.15)RETURN
DV=DETVA
M=M+1
CALL MOVE(N,N,L,A,AP)
CALL MOVE(N,N,L,B,BP)
DO 7 J=1,N
DO 6 K=1,N
LD=L-LI+1
DO 5 I=1,N
A(I,J,1)=A(I,J,1)+CA(I,K)*BP(K,J,LD)
B(I,J,1)=B(I,J,1)+CR(I,K)*AP(K,J,LD)
5 CONTINUE
7 CONTINUE
RETURN
END
SUBROUTINE NLOGN(N,X,SIGN,LX)

NMAX=LARGEST VALUE OF N TO BE PROCESSED

NNDUMMAY DIMENSION M(NMAX)

FOR EXAMPLE, IF NMAX=100, THEN

DIMENSION M(100)

DIMENSION X(2**N)

DIMENSION X(LX)

COMPLEX X, M(I), HOLD, Q

DO 1 I=1,N

M(I)=2**(N-I)

DO 4 L=1,N

NBLOCK=2**(L-1)

LBLOCK=LX/NBLOCK

LBHALF=LBLOCK/2

K=0

DO 4 IRLOCK=1,NBLOCK

DO 4 I=1,LBLOCK

DO 4 J=1,LHALF

X(J)=X(J)+M(I)*Q

CONTINUE

DO 4 I=2,N

IF(K.LT.M(I)) GO TO 4

K=K-M(I)

K=K+M(I)

IF(SIGN.LT.0.0) RETURN

DO d r=1,LX

X(I)=X(I)/FLX

RETURN

END
NORMAG

100 SUBROUTINE NORMAG(LX,X)
110 DIMENSION X(LX)
125 R=0.0
130 DO 10 I=1,LX
140 10 B=MAX1(ABS(X(I)),B)
145 IF(R.EQ.0.0)RETUN
150 DO 20 I=1,LX
160 20 X(I)=X(I)/B
170 RETURN
180 END
SUBROUTINE OLDPL0(G2,F1,M)

DIMENSION G2(M),F1(M)

DIMENSION F(64),G(64)

CHARACTER PLANK(63)/" */
CHARACTER STAR/"**"/

ROUND(X)=X+.5
DO 15 I=1,M
10 G(I)=G2(I)
15 F(I)=F1(I)

FG=(RIG(G,M)-SMALL(G,M))/42.
IF(FG.LE.1.0E-07)GO TO 998

DO 1 K=1,M
10 G(K)=G(K)/FG
14 F(K)=2.0*F(K)
18 CALL GFSORT(G,F,M)
19 DO 19 K=1,5
20 WRITE(0,23)
21 FORMAT(1H1)
22 G1=G(1)*FG
23 WRITE(0,100)G1
24 GLAST=G(1)
25 DO 2 J=1,M
26 LGDIF=ROUND(GLAST-G(J))
27 IF(LGDIF)6,8,6
28 DO 6 L=1,LGDIF
29 GG=(GLAST-FLOAT(L))*FG
30 WRITE(0,100)GG
31 FORMAT(1H100 FORMAT(E9.2," "))
32 LFDIF=ROUND(F(J)+1.)
33 WRITE(0,101)(BLANK(K),K=1,LFDIF),STAR
34 FORMAT(1H",10X,64A1)
35 GLAST=G(J)
36 WRITE(0,200)
37 FORMAT(1H10X,32(12))
38 WRITE(0,210)
39 FORMAT(1H10X,0 1 2 3 4 5 6 7 8 9 11 13 15 17 19 21,)
40 FORMAT(1H23 25 27 29 31)
41 RETURN
42 WRITE(0,104)G(1)
43 FORMAT(ALL VALUES ARE EQUAL TO,E9.2)
44 RETURN
500 END
100 *10
110 FL2IND,0
120 FL3IND,0
130 *200
140 STA
150 CLZE
160 CLA
170 TAD MTICKS
180 CLAB
190 CLA
200 TAD K5500
210 CLBE
220 CLA
230 STA
240 DCA FL2IND
250 STA
260 DCA FL3IND
270 DCA TALLY
280 LOOP, CLSK
290 JMP .-1
295 CLSA
300 JMS SAMP
310 ISZ TALLY
320 JMP LOOP
330 HLT
340 SAMP, 0
350 CLA
360 TAD K2000
370 JMS ADCONV
380 6221 / CDF2
390 DCA I FL2IND
400 TAD K2001
410 JMS ADCONV
420 6231 / CDF3
430 DCA I FL3IND
440 JMP I SAMP
450 ADCONV, 0
460 ADSC
470 ADCV
480 AUSF
490 JMP .-1
500 ADRB
510 JMP I ADCONV
520 K2001, 2001
530 K2000, 2000
540 TALLY, 0
550 MTICKS, -1415
560 K5500, 5500
570 *300
580 DISP, LAS
PDP  (continued)

590 SMA
600 JMP .+3
610 6231 /CDF3
620 JMP .+2
630 6221 /CDF2
640 CLA
650 DCA TALLY
652 TAD K2000
654 0552
656 CLA
658 0552
660 TAD I TALLY
670 0551
680 CLA
690 ISZ TALLY
700 JMP .-4
710 JMP DISP
720 $
SUBROUTINE PLOT(C2,F1,M)
DIMENSION G2(M),F1(M)
DIMENSION F(64),G(64)
CHARACIER BLANK(63) '/63**''/
CHARACIER STAR/''**''/
ROUND(X)=X+.5
DO 15 I=1,M
G(I)=G2(I)
F(I)=F1(I)
15 IF(F(I).LE.1.0E-07)G0 TO 998
DO 1 K=1,M
G(K)=G(K)/FG
1 FG(I)=2.0*F(K)
CALL GFSORT(G,F,M)
DO 19 K=1,5
23 FORMAT(IH)
G1=G1*FG
WRITE(0,100)G1
GLAST=G1
LGDIF=ROUND(GLAST-G(J))
IF(LGDIF.8,8,6
6 DO 8 L=1,LGDIF
8 GG=(GLAST-FLOAT(L))*FG
WRITE(0,100)GG
100 FORMAT(IH,E9.2,"")
8 LDIF=ROUND(F(J)+1.)
WRITE(0,101) (PLANK(K),K=1,LFDIF),STAR
101 FORMAT(IH*,10X,64AI)
2 GLAST=J(J)
WRITE(0,200)
200 FORMAT(IOX,32('I'))
WRITE(0,210)
210 FORMAT(IOX,0123456789111315171921')
400 RETURN
998 WRITE(0,104)G(1)
104 FORMAT('ALL VALUES ARE EQUAL TO',E9.2)
RETURN
END
PLOTR

100 SUBROUTINE PLOTR(Y1,X,M)
110 DIMENSION Y1(M),X(M),Y(128)
120 CHARACTER LINE(63)/63*' ', BLAN/' ', STAR/ '*'/, DOT'/ '.'/
129 DO 99 I=1,M
130 99 Y(I)=Y1(I)
131 SMALL=Y(I)
140 BIG=SMALL
150 DO 40 I=2,M
160 VALUE=Y(I)
170 IF(VALUE-BIG)20,20,10
180 10 BIG=VALUE
190 10 TO 40
200 20 IF(VALUE-SMALL)30,40,40
210 30 SMALL=VALUE
220 40 CONTINUE
230 IF(ABS(BIG)-ABS(SMALL))50,60,60
240 50 SCALE=ABS(SMALL)/31.
250 60 TO 70
260 60 SCALE=ABS(BIG)/31.
270 70 IF(ABS(SMALL))100,80,100
280 80 WRITE(0,90)BIG
290 90 FORMAT(' NO GRAPH CAN BE DRAWN SINCE ALL VALUES ARE ', E15.7)
300 RETURN
310 100 WRITE(0,110)BIG,SMALL,SCALE
320 110 FORMAT(' LARGEST VALUE IS ', E15.7, '/ ', 'SMALLEST VALUE IS ', E15.7, '
', 'MULTIPLY EACH ORDIINATE READING BY THE SCALE FACTOR ', E15.7)
340 DO 19 K=1,5
350 19 WRITE(0,23)
360 23 FORMAT(1H )
370 DO 130 I=1,M
380 130 Y(I) = Y(I)/SCALE
390 WRITE(0,210)
400 210 FORMAT(9X,-30,-27,-24,-21,-18,-15,-12,-9,-6,-3,0,3,6,9,12',
410 &13,18,21,24,27,30')
420 WRITE(0,120)
430 120 FORMAT(11X,21('1'))
440 DO 131 I=1,63
450 131 LINE(I)=BLAN
460 DO 1000 I=1,M
470 VALUE=Y(I)
480 INDEX=32.+VALUE
481 INDEX=4AXO(INDEX,1)
482 INDEX=MINO(INDEX,63)
490 LINE(32)=DOT
500 LINE(INDEX)=STAR
510 WRITE(0,180)X(I),LINE
520 LINE(INDEX)=BLAN
530 180 FORMAT(F7.2,3X,63A1)
540 1000 CONTINUE
550 RETURN

80
PLOTIR  (continued)
560 END
SUBROUTINE REMAV(LY, Y)

DIMENSION Y(LY)

S = 0.0

DO 10 I = 1, LY

10   S = S + Y(I)

AV = S / FLOAT(LY)

DO 20 I = 1, LY

20   Y(I) = Y(I) - AV

RETURN

END
SUBROUTINE RZERO(LX,X)
110 DIMENSION X(LX)
120 DO 1 I=1,LX
130 X(I)=0.0
140 RETURN
150 END

RZERO
SIMEQ

1400 SUBROUTINE SIMEQ(M,N,A,R,C)
1410 * NMAX=LARGEST VALUE OF N TO BE PROCESSED
1420 * NONDUMMY DIMENSION S(NMAX,NMAX)
1430 * FOR EXAMPLE, IF NMAX=4 THEN
1440 DIMENSION S(4,4)
1450 DIMENSION A(M,N),R(N,N),C(M,N)
1460 CALL MOVE(M,N,R,S)
1470 CALL ESCAL(N,S,R)
1480 DO 1 I=1,M
1490 DO 1 J=1,N
1500 A(I,J)=0.0
1510 1 DO 1 K=1,N
1520 1 A(I,J)=A(I,J)+C(I,K)*B(K,J)
1530 CALL MOVE(M,N,S,R)
1540 RETURN
1550 END
SUBROUTINE SIMEOQ(M,N,A,B,C,D)
* NMAX=LARGEST VALUE OF N TO BE PROCESSED
* NONDUMMY DIMENSION A(NMAX,NMAX)
* FOR EXAMPLE, IF NMAX=4, THEN
DIMENSION S(4,4)
DIMENSION A(M,N),B(N,N),C(M,N)
call move(n*n,r,s)
call escaIdd(n,s,r,d)
do 1 i=1,n
do 1 j=1,n
a(i,j)=0.0
do 1 k=1,n
a(i,j)=a(i,j)+c(i,j)*b(k,j)
call move(n*n,s,r)
return
end

SIMEOQ

85
SMALL

100 FUNCTION SMALL(A,M)
110 DIMENSION A(M)
120 S=A(1)
130 DO 1 K=2,M
140 IF(A(K)-S)2,1,1
150 2 S=A(K)
160 1 CONTINUE
165 SMALL=S
170 RETURN
180 END
TRANS

100 SUBROUTINE TRANS(P,IDIMP,Y,N,NV,NB,LOG2NS)
105 DIMENSION Y(1024,NV)
110 DIMENSION P(IDIMP)
120 M=LOG2NS
130 DO 1 I=1,NV-1,2
140 CALL FAST(Y(1,I),Y(1,I+1),M,N)
150 NT=N/2
160 DO 6 I=1,NV,2
170 DO 8 J=2,NT
180 K=N-J+2
190 T1=(Y(J,1+I)-Y(K,1+I))
200 T2=(Y(K,1)-Y(J,1))
210 Y(J,1)=Y(J,1)+Y(K,1)
220 Y(J,1+I)=Y(J,1+I)+Y(K,1+I)
230 Y(K,1)=T1
240 Y(K,1+I)=T2
250 DO 17 I=1,NV
260 Y(I,1)=2.0*Y(I,1)
270 NT=NT+1
280 NE=N/(4*NB)
290 IIMIN=NE+1
300 IIMAX=NT-NE
310 IISTEP=NE+NE
320 L=1
330 DO 2 I=1,NV
340 DO 2 J=1,NV
350 P(L)=Y(I,J)*Y(1,J)
360 P(L+1)=0.0
370 DO 3 K=2,IIMIN
380 M=N-K+2
390 P(L)=P(L)+2.0*(Y(K,1)*Y(K,J)+Y(M,1)*Y(M,J))
400 L=L+2
410 DO 4 I=IIMIN,IIMAX,IISTEP
420 P(L)=0.0
430 P(L+1)=0.0
440 KMAX=I+IISTEP
450 DO 5 K=I,KMAX
460 M=N-K+2
470 P(L)=P(L)+Y(K,1)*Y(K,J)+Y(M,1)*Y(M,J)
480 P(L+1)=P(L+1)+Y(K,1)*Y(M,1)*Y(M,J)-Y(M,1)*Y(K,J)
490 4 L=L+2
500 P(L)=Y(NT+1,1)*Y(NT+1,J)
510 P(L+1)=0.0
520 KMIN=NT+1-NE
530 DO 6 K=KMIN,NT
540 M=N-K+2
550 P(L)=P(L)+2.0*(Y(K,1)*Y(K,J)+Y(M,1)*Y(M,J))
560 P(L+1)=P(L+1)+Y(K,1)*Y(M,1)*Y(M,J)
570 RETURN
580 END
UNPACK

100 LET I=0
110 DIM A(4096),S(13)
120 FILE #1: "PUB"
130 FOR J = 1 TO LOF(#1)
140 READ #1: S$
150 CHANGE S$ TO S
160 FOR K= 0 TO 3
170 LET I=I+1
180 LET A(I)=16*S(3*K+1)+INT(S(3*K+2)/16)
190 LET I=I+1
200 LET A(I)=256*MOD(S(3*K+2),16)+S(3*K+3)
210 NEXT K
220 NEXT J
230 SCRATCH #1
240 FOR J= 1 TO I
250 WRITE #1: A(J)
260 NEXT J
270 END
PARZ

100 SUBROUTINE PARZ(M,W)
110 DIMENSION W(M)
120 ** M IS AN EVEN INTEGER
130 DO 1 J=1,M/2+1
140 W(J)=1.-6.*(FLOAT(J-1)/FLOAT(M))**2+6.*(FLOAT(J-1)/FLOAT(M))**3
150 DO 2 J=M/2+2,M
160 W(J)=2.*(1.-FLOAT(J-1)/FLOAT(M))**3
170 RETURN
180 END
ZERO

100 SUBROUTINE ZERO(LX,X)
110 DIMENSION X(LX)
120 COMPLEX X
130 IF(LX.LE.0)RETURN
140 DO 1 I=1,LX
150 1 X(I)=(0,0,0,0)
160 RETURN
170 END
LPSFLTI

100 DIMENSION WT(2095), NTS(49), A1(2095), BI(2047)
110 LIBRARY "NLOGN", "CROSS", "GENwTS", "FILTER"
120 OPENFILE 1, "PUB", "NUMERIC"
130 READ(1) (A1(J), J=1, 2095)
140 CALL GENwTS(2, 24, NTS, 49)
150 CALL FILTER(2095, 49, NTS, WT, A1, 2047, BI, 12)
170 OPENFILE 2, "NPUB", "NUMERIC"
180 WRITE(2) BI
250 STOP
260 END
LPSFLT2

100 DIMENSION N(I(2095),NTS(49),A2(2095),B2(2047))
110 LIBRARY "NLOGN", "CROSS", "GENWTS", "FILTER"
140 CALL GENWTS(2,24,NTS,49)
190 OPENFILE 3,"CH2", "NUMERIC"
200 READ(3) (A2(J), J=1,2095)
210 CALL FILTER(2095,49,NWS,NT,A2,2047,B2,12)
230 OPENFILE 4,"NCH2", "NUMERIC"
240 WRITE(4)B2
250 STOP
260 END
100 DIMENSION Y(320), R1(300), R(32), S(33), F(33), SH(33), TLAG(300)
110 LIBRARY "RESPECT", "REHAV", "LOGIC", "CROSS", "APARZ", "SPECT"
120 & "SPECT", "PLOT", "SORT", "S1", "SMALL", "PLOT"
130 N=32
140 LY=320
150 OPENFILE 3,"REGDAT","NUMERIC"
160 READ(3)Y
170 CALL REHAV(LY,Y)
180 CALL CROSS(320, Y, Y, /0, 11, 9)
190 CALL APARZ(4, Y)
200 H=1./64.
210 M=1+1
220 N=0
230 CALL SPECTRUM(Y, X, X, X, X, M)
240 DO 1 J=1, 33
250 H(J)=H(J-1)
260 CALL PLOT(S, F, "")
270 WRITE(0,100) (F(J), G(J), J=1, 33)
280 WRITE(0,100) (H0, F5, 2.4%, E9.27)
290 100 H=H/J
300 2 CALL PLOT(1, FLAG, 50)
310 WRITE(0,200) (FLAG(J), R1(J), J=1, 10)
320 200 WRITE(100, F7, 14%, E9.27)
330 PRINT, Y
340 STOP
350 END
SUBROUTINE SPECT(R, N, M, NL1, G, *M1)
DIMENSION C(N1), R(N1), G(N1)
COMPLEX C(100)
* I = 'I', R(1)=I, 2*M = 2*K
* DIMENSION C(N1) WHERE I = NUMBER OF DIMENSION MORE THAN
* TWICE "A" REUN LAG "
C(1) = COMPLEX(R(1), 0.)
DO 1 J = 2, M
1 C(J) = C*PLX(2., *R(J)*R(J), 1.)
DO 2 J = 'I' + 1, M + 1
2 C(J) = C(J - 1)
CALL RLDJ(C, C(-1), *M1)
DO 3 J = 1, M
3 C(J) = 2.*HL*REAL(C(J))
RETURN
END
DO 10 J=2, N
   A = .54*SPEC1(J) + .46*SPEC1(J)
   B = .54*SPEC1(J) + .46*SPEC1(J)
   SJ = SPEC1(J)
   SK = SPEC1(J+1)
   10 SPEC1(J) = .54*SJ + .23*(SI+SK)
   SPEC1(1) = A
   SPEC1(N) = B
END
SUBROUTINE RSEXTH(", N, RT, G, M1)

DIMENSION RT(N), G(M1)

COMPLEX C(100)

* M1=1+1; 2*M=2**N

* DIMENSION C(M), WHERE M IS ANY DIMENSION >= M1

* THE MAXIMUM LAG M.

C(1)=CUMPLEX(RT(1),0.)

DO 1 J=2, M

1 C(J)=CUMPLEX(2.*RT(J),0.)

C(M1)=CUMPLEX(RT(M1),0.)

DO 2 J=1+2, M+1

2 C(J)=(0.,0.)

CALL NLOGKIC(C,-1., M+1)

DO 3 J=1, M

3 C(J)=2.*IBREAL(C(J))

RETURN

END
TRUSPECL WAS USED TO COMPUTE THE THEORETICAL SPECTRA OF SIMULATED TIME-SERIES DATA.

SINCE THE DATA WAS OBTAINED BY PASSING WHITE NOISE THROUGH DIGITAL FILTERS, AND THE WHITE NOISE PROCESS WAS OBTAINED BY USING A FAULTY RANDOM NUMBER GENERATOR, THE ESTIMATED COHRENCE SPECTRA AND ESTIMATED PARTIAL COHRENCE SPECTRA DIFFERED SOMewhat FROM THE CORRESPONDING THEORETICAL QUANTITIES.
CONTINUE
DO 7 J=1,NB1
7
DO 7 I=1,NB1
7
S(I,J,J)=S(I,J,J)
510 2  J=1,NV-1
500  K=J+1,NV
520  I=1,NB1
510  CSS=CCAB(S(I,J,J))*CCAB(S(I,K,K))
510  IF(CSS=1.0E-07) 17,18,18
180  S(I,J,J)=0.0,0.0,0.0
200  GO TO 201
510  6 CONTINUE
WHAT(0.300)CH(J),CH(K)
210  300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
220  
98
DO 319 I=1,NB1
319 SP(I)=REAL(S(I,K,J))
319 CONTINUE
CALL OLDPLO(SP,F,NB)
WRITE(0,102)
102 FORMAT(5(IH,4X))
DO 512 L=1,NV
418 IF((J-L)*(K-L))418,500,418
418 DO 512 J=1,NB1
512 SP(I)=12(I,J,K,L)
512 CONTINUE
WRITE(0,301) CH(J),CH(K),CH(L)
301 FORMAT(' PARTIAL COHERENCE BETWEEN CHANNELS ',A1,' AND ',A1)
81,/* AFTER THE INFLUENCE OF CHANNEL ',A1,' HAS BEEN REMOVED*)
CALL OLDPLO(SP,F,NB)
WRITE(0,102)
500 CONTINUE
DO 417 J=1,NV
417 WRITE(0,317)CH(J)
317 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
IX=2*IV1*(1-(J-1)*(J-2))/2-1
DO 416 J=1,NB1
416 SP(I)=REAL(S(I,J,J))
CALL OLDPLO(SP,F,NB)
417 WRITE(0,102)
STOP
END
10) DIMENSION X(1000,2),Z(1000)
11) LIBRARY "WNOISE", "FORTLIB***:SAVEL"
12) DATA LZ/1000/
13) DO 1 J=1,2
14) CALL WNOISE(LZ,Z,8.0)
15) DO I=1,1000
16) 1 X(I,J)=Z(I)
17) CALL FSAVEL("WNSDAT","",1)
18) OPENFILE 2,"WNSDAT","NUMERIC"
19) WRITE(2) ((X(I,J),I=1,1000),J=1,2)
20) STOP
21) END
* N={NUMBER OF CHANNELS}
  * N={NUMBER OF FREQUENCY BANDS(A POWER OF 2)}
  * JSCANS IS ALSO A POWER OF 2
  * SR=SAMPLING RATE=1/H
  * X=INPUT SERIES ARRAY
  * /E=ARRAY FOR STORING CROSS SPECTRA
  * DIMENSION X(1024),E(193),S(132),C(132),F(33)
  * DIMENSION S1(33,2,2)
  * CHARACTER CH(2)="1","2"
  * EQUIVALENCE (S,C)
  * EQUIVALENCE (S,1)
  * DATA NS,N,V,NH,JSCANS,SR,P1/1000,2,32,1024,64,0,1,14169265/
  * LIBRARY "FAST","TRANS","MOVE","MOMAG","COHERE","PLOT","GFSORT"
  * & "KIN","SMALL"
  * OPENFILE 2,"FNSDAT","NUMERIC"  
  * READ(2)(X(J,J),J=1,NS),I=1,NV)
  * * DETRIM THE SERIES BY SUBTRACTING FROM EACH SERIES ITS
  * * LEAST SQUARES LINEAR REGRESSION LINE
  * FNS=NS
  * TFAR=0.5*(FNS+1.0)
  * TSUMSO=(FNS*FNS+FNS+1.0)/(FNS+FNS+1.0))/6.0
  * DO 76 12=1,NV
  * SUM=0.0
  * CRSPRO=0.0
  * DO 32 11=1,NS
  * SUM=SUM+X(II,12)
  * DO 34 77 CRSPRO=CRSPRO+FLOAT(II)*X(II,12)
  * EMEAN=SUM/NS
  * RETA=(CRSPRO-FNS*TFAR*EMEAN)/(TSUMSO-FNS*TFAR*TFAR)
  * DO 78 11=1,NS
  * FREG=EMEAN+RETA*(FLOAT(II)-TFAR)
  * X(II,12)=X(II,12)-FREG
  * 76 CONTINUE
  * * WINDOW EACH SERIES WITH A COSINE TAPER
  * 420 IR=NS/10
  * 430 R=IR
  * 440 DO 79 11=1,IR
  * 450 FI1=11
  * 450 F11=F11-0.5
  * 450 M2=0.5*(1.0-COS(PI*FINT/10))
  * 450 NS=11-F11
  * 450 DO 7C 12=1,NV
  * 470 X(II,12)=WINDOW*X(II,12)
  * 500 DO 60 13=13,12
  * 510 79 CONTINUE
  * 520 LOG2NS=0
  * 530 NSCANS=1
  * 540 54 IF(NS,LE,NSSCAN)GO TO 55
  * 550 LOG2NS=LOG2NS+1
  * 560 NSCANS=NSCANS+NSCANS
  * 570 GO TO 54
  * 590 55 IF(NS,EQ,NSSCAN)GO TO 74
  * 590 * ZERO OUT THE LAST PART OF EACH SERIES IF THE NUMBER OF
  * 590 * SCAN IS NOT A POWER OF 2
  * 600 11=BEGIN=NS+1
  * 620 DO 75 11=1-BEGIN,NSCANS
  * 630 DO 75 12=1,NV
  *
640 /5 X(11,12)=0.0
650 /6 CONTINUE
650 /7 IF(FMOD(JV,2))/0.52/.70
670 * 1: IF THE NUMBER OF SERIES IS ODD, FILL A DUPLICATE SERIES WITH ZEROS
680 70 NVI=NV+1
690 DO 83 I=1,NSCAN
700 83 X(I,NVI)=0.0
710 GO TO 85
720 82 NVI=NV
730 85 CONTINUE
740 IFMPS=NVI*(NVI+1)*(NB+1)
750 CALL TRANS(P,1DIMP,X,NSCAN,NVI,NB,1.0,2*N)
760 * CROSS SPECTRUM ESTIMATES ARE IN ARRAY P
770 * THE CROSS SPECTRUM ESTIMATES IN ARRAY P ARE SCALING BY
780 * MULTIPLYING BY C
790 ANDPWR=FNS-1.25*P
800 SN=NSCAN
810 F=SNB=NB
820 F2=SNB/NSB+FB
830 CI=0.25/(SR*(FD+1.0)*MNDWR)
840 IROWSP=NBS+NB+2
850 ICOLOP=0.5*(NV1+(NVI+1))/2
860 ISIZEP=IROWSP*ICOLSP
870 DO 95 I=1,ISIZEP
880 95 P(I)=CI*P(I)
890 NBS=NB+1
900 DO 100 J=1,NVI
910 DO 100 K=J,NVI
920 DO 100 I=1,NVI
940 IF(J-K)99,1000,1000
960 1000 CONTINUE
970 CALL COHERE(NBS,1,NVI,S,C)
980 DO / J=1,NVI
990 7 F(J)=J-1
1000 WRITE(0,300)CH(J),CH(K)
1010 300 FORMAT(\' COHERENCE FOR CHANNELS \',A1,\' AND \',A1)
1020 CALL PLOT(C(1+NBS*(J-1)+NBS*NVI*(K-1)),F,NB)
1030 100 FORMAT('H ',F5.0,4X,E9.2F)
1040 100 WRITE(0,102)
1050 102 FORMAT(5(1H,1))
1060 500 CONTINUE
1070 DO 105 J=1,NVI
1080 WRITE(0,102)
1090 WRITE(0,107)CH(J)
1010 107 FORM (\' AUTOSPECTRA FOR CHANNEL \',A1)
1100 CALL PLOT(C(1+NBS*(J-1)+NBS*NVI*(J-1)),F,NB)
1110 105 CONTINUE
1120 STOP
1130 END
100 * DIMENSION X(NS*LS),R1(LR*NS*NS),N(M),F(M),S(M*NS*NS)
110 * DIMENSION C(M*NS*NS),TLAG(2*LR-1),Z(2*LR-1),TLAGA(LR)
120 * NS=NUMBER OF CHANNELS
130 * LS=NUMBER OF DATA POINTS FROM EACH CHANNEL
140 * M=MAXIMUM TIME LAG
150 * LR=MAXIMUM DESIRED TIME LAG + LE.
160 * L=SMALLEST INTEGER SUCH THAT |LX|<2**L
170 * M=MAXIMUM LAG, M=2**(N-1)
180 * N IS DEFINED SO THAT 2*M=2**N
190 * H=LENGTH OF SAMPLING INTERVAL
200 * LN=NS=LS
210 * LRNSNS=LR*NS*NS
220 * MNSNS=M*NS*NS
230 DIMENSION X(1000,2),R1(50,2,2),N(32),F(33),S(142),C(33,2,2),TLAG(99)
240 R,Z(99),TLAGA(50)
250 CHARACTER CH(2)="1", "2"/
260 DATA NS,LS,M,LR,N,NS,NS,IN/2,1000,32,33,50,10,6,2000,200,2/
270 LIBRARY "REMAV", "NLOGN", "CROSS", "MPARM", "MACOR", "COQUAD", "MOVE"
280 "NORM", "COHERE", "OLDPLO", "GFSORT", "BIG", "SMALL", "PLOTR"
290 H=1./64.
300 OPENFILE 2,"NNSDAT", "NUMERIC"
310 READ(2)X
315 CALL MPARM(M,N)
320 DO 1 J=1,NS
330 CALL REMAV(LX,X(I,J))
340 CALL MACOR(NS,LS,M,LR,N,NS,LRNSNS,L)
350 CALL COQUAD(H,NS,M,N,W,R1,S,M,LR)
360 CALL COHERE(M1,NS,S,C)
370 10 H=I,J=M1
380 DO 9 J=J+1,NS
390 WRITE(0,300)CH(J),CH(K)
400 WRITE(0,500)K=J+1,NS
410 WRITE(0,300)CH(J),CH(K)
420 300 FORMAT(10,1H, "COHERENCE FOR CHANNELS",A1," AND",A1)
430 WRITE(0,102)C(I,J,K),F,M
440 102 FORMAT(1H,2,F6.2,4X,E9.2,6X)/
450 WRITE(0,102)
460 102 FORMAT(1H,1/))
470 500 CONTINUE
480 700 CONTINUE
490 DO 105 J=1,NS
500 WRITE(0,107)CH(J)
510 107 FORMAT(1H, "AUTO SPECTRA FOR CHANNEL",A1)
520 WRITE(0,102)C(I,J,J),F,M
530 105 WRITE(0,102)
540 DO 700 J=1,NS
550 700 CONTINUE
560 700 CONTINUE
570 WRITE(0,901)CH(I),CH(J)
590 DO 108 I=1,LR
600 TLAG(L)=L-LR
610 108 Z(L)=R1(LR=L+1,J,I)
620 DO 109 I=1,LR+LR-1
630 TLAG(L)=L-LR
640 109 Z(L)=R1(L-LR+1,I,J)
650 CALL PLOTR(Z,TLAG,LR+LR-1)
660 WRITE(0,102)
670 100 CONTINUE
680 100 CONTINUE
690 100 CONTINUE
700 100 CONTINUE
710 100 CONTINUE
720 100 CONTINUE
730 100 CONTINUE
740 100 CONTINUE
750 100 CONTINUE
760 100 CONTINUE
770 100 CONTINUE
780 100 CONTINUE
790 100 CONTINUE
800 100 CONTINUE
810 100 CONTINUE
820 100 CONTINUE
830 100 CONTINUE
840 100 CONTINUE
850 100 CONTINUE
860 100 CONTINUE
870 100 CONTINUE
880 100 CONTINUE
890 100 CONTINUE
900 100 CONTINUE
910 100 CONTINUE
690 701 TLAGA(K) = K - 1
700  DO 801 J = 1, NS
710  WRITE(0,501)CH(J)
720 501 FORMAT(' AUTO CORRELATION FOR CHANNEL ', A1)
730  CALL PLOTT(R1(I,J,J), TLAGA, LR)
740  801 WRITE(0,102)
750  PRINT,X
760  STOP
770  END
90 LIBRARY "FILTER", "SPKFLT"
100 LIBRARY "SIMDAT", "GENFLT", "ORMFLT", "ANOISE", "FORTLIB***:FSAVFL"
101 ", "NLOGN": "CROSS"
110 * LZ, GE, LX+2*L, L=50
120 * LX, GE, LX+2*L, N=50
130 DIMENSION Z(1000), X(1000,4), XF(6), Y(690)
140 & A(101, 0), N(1000), ZK(1000,5), C(4)
145 & A(4,5), 0(4,5), H(4,5)
150 DATA H, L, Z, LX, LY, NWT, LNZ/*0.015625, 1000, 900, 800, 101, 10/
160 DATA C(1), C(2), C(3), C(4)/0.4, 0.1, 0.1, 0.1,
170 DATA XF(4, 5)/0.015625, 1.0, 2.0, 3.0, 4.0,
180 DATA YF(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)/0.015625, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0,
190 DATA H(4, 5)/0.015625, 1.0, 2.0, 3.0,
200 DATA H(4, 5)/0.015625, 1.0, 2.0, 3.0,
210 DATA A(1, 1), B(1, 1), N(1, 1)/2.0, 1.0,
220 DATA A(1, 2), B(1, 2), N(1, 2)/4.0, 6.0,
230 DATA A(1, 3), B(1, 3), N(1, 3)/3.0, 10.0,
240 DATA A(1, 4), B(1, 4), N(1, 4)/4.0, 14.0,
250 DATA A(1, 5), B(1, 5), N(1, 5)/4.0, 29.0,
260 DATA A(2, 1), B(2, 1), N(2, 1)/10.0, 10.0,
270 DATA A(2, 2), B(2, 2), N(2, 2)/7.0, 20.0,
280 DATA A(3, 1), B(3, 1), N(3, 1)/3.0, 3.0,
290 DATA A(3, 2), B(3, 2), N(3, 2)/2.0, 25.0,
300 DATA A(4, 1), B(4, 1), N(4, 1)/25.0, 12.0,
310 DATA A(4, 2), B(4, 2), N(4, 2)/0.0, 1.0,
320 DO 1 K=1, 5
330 CALL SIMDAT(H, A(1, K), D(1, K), N(1, K), LZ, Z, LX, XZ(1, K), NWT, AN, NT, LNZ)
340 CALL ANOISE(LZ, Z, C(1))
350 DO 2 J=1, LX
360 X(J, 1)=Z(J)
370 DO 2 K=1, 5
380 X(J, 1)=X(J, 1)+XZ(J, K)
390 CALL ORMLFT(LY, Y, NWT, AN, LX, X(1, 1), H, XF, YF, 0, 9, 0, NT, LNZ)
400 DO 3 N=2, 4
410 DO 4 K=1, 2
420 CALL SIMDAT(H, A(N, K), D(N, K), N(N, K), LZ, Z, LY, XZ(1, K), NWT, AN, NT, LNZ)
430 CALL ANOISE(LZ, Z, C(N))
440 DO 3 J=1, LY
450 X(J, N)=0.
460 DO 5 K=1, 2
470 X(J, N)=X(J, N)+XZ(J, K)
480 DO 3 X(J, N)=X(J, N)+Z(J)+Y(J)
490 CALL FSAVFL("FCHDAT", ",I")
500 OPENFILE: 2,"FCHDAT","NUMERIC"
510 WRITE(2)(X(I, J), I=1, LY, J=1, 4)
520 STOP
530 END
10) SUBROUTINE GMFIL(LY, Y, NWT, AW, LX, X, H, XF, YF, LX, LN, LN2, O, WT, LN2N)
11) DIMENSION Y(LY), X(LX), NWT(NWT), N1(LX), XF(LX), YF(LX), O(LX)
12) * LX, GE, LY+2*NW
13) * LN2, GE, LW+1
14) NW=(NWT-1)/2
15) NWT=NWT+1
16) CALL GENFL(T,Y,AW(NWT),NWT,XF,YF,LX,O,L(2)
17) DO 1 J=1,NW
18) 1 N=AW(NWT-J)=NW+J
19) CALL FILTER(LX, NWT, AW, WT, X, LY, Y, LN2N)
20) RETURN
21) END
SUBROUTINE SPKFLT(H, A1, D, N, N1, A)

DIMENSION A(N1)
P=6.2831853
C=A1/(19.73925*H*I)
140 IF(D,EQ.0.0)GO TO 200
150 A(1)=(P*I*H)**2
160 C1=P*I*(D-I)
170 C2=P*I*H
180 C3=P*I*(D+H)
190 DO 1 J=2, N1
200 K=J-1
210 A(J)=-(1./K**2)*((COS(C1*K)-2.*COS(C2*K)+COS(C3*K))
220 1 CONTINUE
230 DO 2 J=1, N1
240 2 A(J)=C*A(J)
250 RETURN
260 200 C1=P*I*H
270 A(1)=(0.5)*C1**2
280 DO 3 J=2, N1
290 K=J-1
300 3 A(J)=(-1.,0)*((1./K**2)*((COS(C1*K)-1.,0)
310 DO 4 J=1, N1
320 4 A(J)=C*A(J)
330 RETURN
340 END
SUBROUTINE SIMDAT(H, A1, D, N, LZ, Z, LX, X, NWT, AN, WT, LN2)
DIMENSION Z(LZ), X(LX), AN(NWT), WT(LZ)
* LZ .GE. LX + 2 * NW = LX + NWT - 1
NN = (NWT - 1) / 2
NW = NW + 1
CALL SKFLI(H, A1, D, N, NW, AN(NW))
DO 1 J = 1, NW
1 AW(NW - J) = AW(NW) + J
CALL WNOISE(LZ, Z, 1.0)
CALL FILTER(LZ, NW, AN, WT, Z, LX, X, LN2)
RETURN
END
APPENDIX B

PROGRAM OUTPUTS
COHERENCE FOR CHANNELS 1 AND 2

EEG COHERENCE ESTIMATE OBTAINED BY USING PROGRAM MULSPECT
SAMPLING RATE: 64 PER SECOND, DEGREES OF FREEDOM: 37
AUTOSPECTRA FOR CHANNEL 1

EEG AUTO SPECTRUM ESTIMATE OBTAINED BY USING PROGRAM MULSPECT
37 DEGREES OF FREEDOM
AUTOSPECTRA FOR CHANNEL 2

EEG AUTO SPECTRUM ESTIMATE OBTAINED BY USING PROGRAM MULSPECT
37 DEGREES OF FREEDOM
CROSS CORRELATION BETWEEN CHANNELS 1 AND 2
LARGEST VALUE IS \(1.401775 \times 10^7\)
SMALLEST VALUE IS \(-1.426841 \times 10^7\)
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR \(0.4602714 \times 10^5\)

-30-27-24-21-18-15-12 -9 -6 -3 0 3 6 9 12 15 18 21 24 27 30

-49.00
-43.00
-47.00
-49.00
-45.00
-44.00
-43.00
-42.00
-41.00
-40.00
-39.00
-38.00
-37.00
-36.00
-35.00
-34.00
-33.00
-32.00
-31.00
-30.00
-29.00
-28.00
-27.00
-26.00
-25.00
-24.00
-23.00
-22.00
-21.00
-20.00
-19.00
-18.00
-17.00
-16.00
-15.00
-14.00
EEG CROSS CORRELATION FUNCTION OBTAINED BY USING PROGRAM MULSPECT
AUTO CORRELATION FOR CHANNEL 1
LARGEST VALUE IS  .2638448E+07
SMALLEST VALUE IS  -.1674095E+07
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR  .8511122E+05
EEG AUTO CORRELATION FUNCTION OBTAINED BY USING PROGRAM MULSPECT
AUTO CORRELATION FOR CHANNEL 2
LARGEST VALUE IS \(2.506558 \times 10^7\)
SMALLEST VALUE IS \(-1.1673505 \times 10^7\)
MULTIPLY EACH ORDinate READING BY THE SCALE FACTOR \(8.085670 \times 10^5\)

EEG AUTOCORRELATION FUNCTION OBTAINED BY USING PROGRAM MULSPECT
COHERENCE FOR CHANNELS 1 AND 2

The next 22 graphs are estimates of coherences and partial coherence spectra of 4 channel time series, obtained by passing white noise through digital filters. The estimates were obtained by using program SPECTRAN, with 25 degrees of freedom.
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 2
AFTER THE INFLUENCE OF CHANNEL 3 HAS BEEN REMOVED
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 2
AFTER THE INFLUENCE OF CHANNEL 4 HAS BEEN REMOVED.
COHERENCE FOR CHANNELS 1 AND 3
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 3
AFTER THE INFLUENCE OF CHANNEL 2 HAS BEEN REMOVED
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 3
AFTER THE INFLUENCE OF CHANNEL 4 HAS BEEN REMOVED
COHERENCE FOR CHANNELS 1 AND 4

![Graph showing coherence for channels 1 and 4.](image-url)
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 4
AFTER THE INFLUENCE OF CHANNEL 2 HAS BEEN REMOVED
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 4
AFTER THE INFLUENCE OF CHANNEL 3 HAS BEEN REMOVED
COHERENCE FOR CHANNELS 2 AND 3
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 3
AFTER THE INFLUENCE OF CHANNEL 1 HAS BEEN REMOVED
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 3
AFTER THE INFLUENCE OF CHANNEL 4 HAS BEEN REMOVED
COHERENCE FOR CHANNELS 2 AND 4

1.0E+01
.9E+00
.8E+00
.7E+00
.6E+00
.5E+00
.4E+00
.3E+00
.2E+00
.1E+00
.0E+00

.50E+00
.45E+00
.4E+00
.3E+00
.2E+00
.1E+00

.19E+00
.17E+00
.15E+00
.12E+00
.1E+00
.72E-01
.63E-01
.25E-01
.89E-02

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 4
AFTER THE INFLUENCE OF CHANNEL 1 HAS BEEN REMOVED
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 4
AFTER THE INFLUENCE OF CHANNEL 3 HAS BEEN REMOVED
COHERENCE FOR CHANNELS 3 AND 4

\[ \text{.10E+01} \]
\[ \text{.9E+00} \]
\[ \text{.2E+00} \]
\[ \text{.9E+00} \]
\[ \text{.7E+00} \]
\[ \text{.8E+00} \]
\[ \text{.2E+00} \]
\[ \text{.7E+00} \]
\[ \text{.7E+00} \]
\[ \text{.7E+00} \]
\[ \text{.4E+00} \]
\[ \text{.7E+00} \]
\[ \text{.5E+00} \]
\[ \text{.5E+00} \]
\[ \text{.4E+00} \]
\[ \text{.3E+00} \]
\[ \text{.5E+00} \]
\[ \text{.4E+00} \]
\[ \text{.3E+00} \]
\[ \text{.3E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
\[ \text{.2E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
\[ \text{.1E+00} \]
PARTIAL COHERENCE BETWEEN CHANNELS 3 AND 4
AFTER THE INFLUENCE OF CHANNEL 1 HAS BEEN REMOVED
PARTIAL COHERENCE BETWEEN CHANNELS 3 AND 4
AFTER THE INFLUENCE OF CHANNEL 2 HAS BEEN REMOVED
ALL VALUES ARE EQUAL TO .00E-21
AUTOSPECTRA FOR CHANNEL 2

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
AUTOSPECTRA FOR CHANNEL 3

0 1 2 3 4 5 6 7 8 9 11 13 15 17 19 21 23 25 27 29 31

-4.0E-03
-2.7E-02
-0.69E-02
0.12E-01
0.15E-01
0.17E-01
0.19E-01
0.21E-01
0.23E-01
0.25E-01
0.27E-01
0.29E-01
0.30E-01
0.32E-01
0.33E-01
0.35E-01
0.37E-01
0.38E-01
0.39E-01
0.41E-01
0.43E-01
0.45E-01
0.47E-01
0.49E-01
0.51E-01
0.53E-01
0.55E-01
0.57E-01
0.59E-01
0.61E-01
0.63E-01
0.65E-01
0.67E-01
0.69E-01
0.71E-01
0.73E-01
0.75E-01
0.77E-01
0.79E-01
0.81E-01
0.83E-01
0.85E-01
0.87E-01
0.89E-01
0.91E-01
0.93E-01
0.95E-01
0.97E-01
1.00E+00
1.02E+00
1.04E+00
1.06E+00
1.08E+00
1.10E+00
1.12E+00
1.14E+00
1.16E+00
1.18E+00
1.20E+00
1.22E+00
1.24E+00
1.26E+00
1.28E+00
1.30E+00
1.32E+00
1.34E+00
1.36E+00
1.38E+00
1.40E+00
1.42E+00
1.44E+00
1.46E+00
1.48E+00
1.50E+00
1.52E+00
1.54E+00
1.56E+00
1.58E+00
1.60E+00
1.62E+00
1.64E+00
1.66E+00
1.68E+00
1.70E+00
1.72E+00
1.74E+00
1.76E+00
1.78E+00
1.80E+00
1.82E+00
1.84E+00
1.86E+00
1.88E+00
1.90E+00
1.92E+00
1.94E+00
1.96E+00
1.98E+00
2.00E+00
COHERENCE FOR CHANNELS 1 AND 2

EEG COHERENCE ESTIMATE OBTAINED BY USING PROGRAM SPECTCLTK
10 DEGREES OF FREEDOM
AUTOSPECTRA FOR CHANNEL 2
EEG auto spectra obtained by using the program SPCTCLKK.
10 degrees of freedom
<table>
<thead>
<tr>
<th></th>
<th>Prediction Errors for Autoregressive Schemes of Order 1 to 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.359876E+07</td>
</tr>
<tr>
<td>2</td>
<td>2.179372E+07</td>
</tr>
<tr>
<td>3</td>
<td>2.038137E+07</td>
</tr>
<tr>
<td>4</td>
<td>2.033934E+07</td>
</tr>
<tr>
<td>5</td>
<td>1.992783E+07</td>
</tr>
<tr>
<td>6</td>
<td>1.936475E+07</td>
</tr>
<tr>
<td>7</td>
<td>1.913139E+07</td>
</tr>
<tr>
<td>8</td>
<td>1.920902E+07</td>
</tr>
<tr>
<td>9</td>
<td>1.920908E+07</td>
</tr>
<tr>
<td>10</td>
<td>1.928289E+07</td>
</tr>
<tr>
<td>11</td>
<td>1.911853E+07</td>
</tr>
<tr>
<td>12</td>
<td>1.910493E+07</td>
</tr>
<tr>
<td>13</td>
<td>1.893031E+07</td>
</tr>
<tr>
<td>14</td>
<td>1.892531E+07</td>
</tr>
<tr>
<td>15</td>
<td>1.889173E+07</td>
</tr>
</tbody>
</table>

.24E+06
.23E+06
.23E+06
.22E+06
.22E+06
.21E+06
.21E+06
.20E+06
.20E+06
.19E+06
.19E+06
.19E+06
.18E+06
.17E+06
.17E+06
.16E+06
.16E+06
.15E+06
.14E+06
.14E+06
.13E+06
.13E+06
.12E+06
.12E+06
.11E+06
.11E+06
.10E+06
.97E+05
.92E+05
COHERENCE FOR CHANNELS 1 AND 2

EEG COHHERENCE COMPUTED BY USING PROGRAM MAUTOREG
AUTOSPECTRA FOR CHANNEL 1

EEG AUTOSPECTRUM ESTIMATE OBTAINED BY USING THE PROGRAM
MAUTOREG
AUTO SPECTRA FOR CHANNEL 2

EEG AUTO SPECTRUM ESTIMATE OBTAINED BY USING THE PROGRAM MAUTOREG
COHESION FOR CHANNELS 1 AND 2

COHESION SPECTRUM OF TWO WHITE NOISE PROCESSES OBTAINED
BY USING FORTRAN DTSS RANDOM NUMBER GENERATOR.
CHANNELS ARE HIGHLY COHESIVE, SHOWING POOR PER-
FORMANCE OF RANDOM NUMBER GENERATOR.
ESTIMATES WERE COMPUTED BY USING PROGRAM WNOISPEC.
31 DEGREES OF FREEDOM.
AUTOSPECTRA FOR CHANNEL 1

SPECTRUM OF WHITE NOISE GENERATED BY USING FORTRAN DTSS RANDOM NUMBER GENERATOR. ESTIMATES WERE OBTAINED BY USING PROGRAM WNOISPEC. 31 DEGREES OF FREEDOM WERE USED. STATISTICAL ANALYSIS OF SPECTRUM SHOWS POOR PERFORMANCE OF RANDOM NUMBER GENERATOR.
SPECTRUM FOR WHITE NOISE PROCESS OBTAINED BY USING A RANDOM NUMBER GENERATOR. PROGRAM WNOISPEC WAS USED, WITH 31 DEGREES OF FREEDOM. TEST SHOWS POOR PERFORMANCE OF GENERATOR.
COHERENCE FOR CHANNELS 1 AND 2

COHERENCE FUNCTION FOR TWO WHITE NOISE PROCESSES OBTAINED BY USING DTSS FORTRAN RANDOM NUMBER GENERATOR. PROGRAM WHOITEST WAS USED WITH 116 DEGREES OF FREEDOM. TEST SHOWS POOR PERFORMANCE OF RANDOM NUMBER GENERATOR.
AUTOSPECTRA FOR CHANNEL 1

AUTOSPECTRUM OF WHITE NOISE PROCESS OBTAINED BY USING DTSS FORTRAN RANDOM NUMBER GENERATOR. PROGRAM VNOITEST, WITH 116 DEGREES OF FREEDOM WAS USED. TEST SHOWS POOR PERFORMANCE OF RAND. NUMBER GENERATOR.
AUTOSPECTRUM FOR A WHITE NOISE PROCESS OBTAINED BY USING DTSS FORTRAN RANDOM NUMBER GENERATOR. PROGRAM WN0TEST, WITH 116 DEGREES OF FREEDOM WAS USED. TEST SHOWS POOR PERFORMANCE OF RAND. NUMBER GENERATOR.
AUTO CORRELATION FOR CHANNEL 1
LARGEST VALUE IS 6.915590E+02
SMALLEST VALUE IS -5.173850E+01
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .2230835101

AUTO CORRELATION FOR WHITE NOISE PROCESS
AUTO CORRELATION FOR CHANNEL 2
LARGEST VALUE IS .6397735E+02
SMALLEST VALUE IS -.8320822E+01
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .2063785E+01
AUTO CORRELATION FOR WHITE NOISE PROCESS.
Cross correlation for two white noise processes obtained by using random number generator. Generator performs poorly.
**UNCLASSIFIED**

**DOCUMENT CONTROL DATA - R & D**

<table>
<thead>
<tr>
<th>1. ORIGINATING ACTIVITY (Corporate author)</th>
<th>2a. REPORT SECURITY CLASSIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.S. Naval Academy, Annapolis.</td>
<td>UNCLASSIFIED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. REPORT TITLE</th>
<th>2b. GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematical-statistical and digital computer analysis of time series data.</td>
<td>UNCLASSIFIED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. DESCRIPTIVE NOTES (Type of report and inclusive dates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research report.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. AUTHOR(S) (First name, middle initial, last name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steven J. Raher.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. REPORT DATE</th>
<th>7a. TOTAL NO. OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>17 May 1976</td>
<td>152</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8a. CONTRACT OR GRANT NO.</th>
<th>9a. ORIGINATOR'S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U.S. Naval Academy, Annapolis - Trident Scholar project report, no. 81 (1976)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. DISTRIBUTION STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>This document has been approved for public release; its distribution is UNLIMITED.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. SPONSORING MILITARY ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.S. Naval Academy, Annapolis.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. ABSTRACT</th>
</tr>
</thead>
</table>

The purpose of this study is to develop a series of computer programs for use in analyzing time series data (waves) - such as EEG readings.

Programs were used to produce reliable estimates of correlations, spectra, cross spectra, and partial coherences of multi-channel random processes. The software package was written to be easily adaptable to different sampling rates, amounts of data, and numbers of channels. Provisions for digital pre-filtering of data, detrending, and smoothing (using a number of lag windows) were also included. Techniques for estimation of spectra by fitting single- and multi-channel autoregressive schemes to sampled data were also applied and found to yield results consistent with the other methods.

All programs were written in FORTRAN and run on the USNA/DTSS computer system. All data and charts included in the paper.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>ROLE</td>
<td>ROLE</td>
</tr>
<tr>
<td></td>
<td>WT</td>
<td>WT</td>
<td>WT</td>
</tr>
</tbody>
</table>

Security Classification