SPECIAL REPORT NO. 6

# Digital Computer Programs for Spectral Analysis of Time Series

by

EVERETT J. FEE

Center for Great Lakes Studies
The University of Wisconsin-Milwaukee
Milwaukee, Wisconsin 53201 USA

March, 1969

CONTENTS

# DIGITAL COMPUTER PROGRAMS FOR SPECTRAL ANALYSIS OF TIME SERIES

## ABSTRACT

The computer programs presented here were developed for analysis of extensive time series of surface water level fluctuations in Lake Michigan and Lake Superior. The main results of this research will be presented elsewhere (Mortimer and Fee, 1969, abstract reproduced in Appendix C). The input to the program consists of an arbitrary number of real data points in the time domain from either one or two equally spaced digitized time series (for example, from either one or two water level recording stations). If data from only one time series are presented for analysis, the program yields an estimate of the corresponding energy (variance) spectrum. If two data sets representing two simultaneously recorded time series are presented to the program, the output provides estimates of (i) an energy spectrum for each series, and (ii) information on the relationships between the two series, in the form of co- and quadrature spectra and spectra of coherence and phase.

In comparison with procedures used until recently, considerable saving in computer time has been achieved by making use of the fast Fourier transform. The programs are presented in both FORTRAN and

ALGOL versions and, while they were written for the specific application already mentioned, their logic is general. They can, therefore, find wide application in the analysis of time series, irrespective of the source and length (ALGOL version).

# INTRODUCTION

In the last twenty years, the techniques of spectrum and cross spectrum analysis have completely dominated the methodology of time series analysis. The historical development of the various approaches to spectrum analysis is treated by Tukey (1967) and by Bingham, _et al_. (1967). Blackman and Tukey (1957) and Jenkins, _et al_. (1968) provide standard references on the subject. The main advantages of these methods are that they digest large bodies of data (in Mortimer and Fee, 1969, for example, there were usually more than 10,000 points in each of over 40 time series) into a few graphical spectral presentations, the main features of which can be readily interpreted and which disclose persistent periodicities in the data, if present.

As used here, spectral analysis is a statistical procedure which determines the frequency distribution of the energy (variance) of a time series over a defined frequency range. Subject to the statistical considerations outlined below, peaks in particular frequency bands of the energy spectrum indicate higher contents of energy in those bands than in neighboring ones. Cross spectral analysis of two simultaneously recorded time series provides information on the relationships between them, in the form of estimates of coherences and phase differences, as functions of frequency. Spectral and cross spectral analyses, combined, can provide much more information about persistent periodicities in a pair of simultaneously recorded time series than can either analysis procedure alone. For example, in Figure 1 (Mortimer and Fee, 1969), the phase and coherence diagrams permit unambiguous interpretation of the physical meaning of peaks in the spectra of water level fluctuations at two stations.

Many computer programs for spectral and cross spectral analysis of time series exist in published and unpublished form. Most of them employ the techniques reviewed by Blackman and Tukey (1957), and they become very expensive to operate when large numbers of data points are involved. The new programs presented here use a different set of algorithms (Cooley, *et al.*, 1965) to obtain essentially identical answers, a conclusion which has been carefully tested. The new techniques are much faster, however, and they make it economically feasible to extend the use of spectrum analysis as an experimental tool. For example, the analysis of a pair of time series each with $N = 4,392$ points, which by the old method took 15 to 20 minutes of computer time on an IBM 7090, takes about six or seven seconds on the Univac 1108 using the new method presented here. The savings become even greater with longer time series, since the number of operations increases as $N^2$ with the old method and only as $[N \log_2 N]$ with the new.

The spectrum displays the (estimated) frequency distribution of energy over a defined frequency range (usually 0 to $1/(2\Delta t)$, see below) not as a continuous distribution, but as a "histogram" made up of estimates for successive small sub-divisions of the frequency range. The number of such sub-divisions, usually several hundred, determines the frequency resolution (i.e., the bandwidth of each spectral estimate) and the statistical reliability of the answers obtained. This number is usually referred to as the number of "lags", further defined in Blackman and Tukey (1957).

Given a time series in the form of discrete readings at a constant time interval, $\Delta t$, then $2\Delta t$ is the oscillation of lowest period, i.e., $f_n = 1/(2\Delta t)$ is the oscillation of highest frequency, for which the energy contribution can be estimated. The parameter $f_n$ (discussed below) is called the "Nyquist frequency." Therefore, if we choose to estimate the spectrum at $\underline{m}$ "lags", we break up the interval from 0 to $f_n$ into $\underline{m}$ frequency bands of equal width. Resolution, therefore, increases with increasing number of lags. On the other hand, the approximate number of degrees of freedom is given by the expression $2\underline{N}/\underline{m}$ where $\underline{N}$ is the number of points in the time series. Thus, increasing $\underline{m}$ decreases the statistical reliability of the answers (i.e., the confidence limits widen as $\underline{m}$ increases). There is no simple rule to guide the choice of number of lags for a given number of data points, since a spectrum with more or less resolution may be required, in order to detect particular features of interest. Most authors agree, however, that $\underline{m}$ should not be more than ten or fifteen per cent of $\underline{N}$.

The choice of $f_n$ can greatly influence the spectral estimates. As with the number of lags, selection of this number must be a balance between two opposing considerations. On the one hand, a large $\Delta t$ simplifies the procedure and reduces the cost in computer time. On the other hand, the spectrum becomes contaminated by "aliasing" (defined by Blackman and Tukey, 1957) if there is appreciable energy in the time series corresponding to frequencies greater than $f_n$. Although this energy lies outside the spectral range, 0 to $f_n$, it unavoidably generates spurious (aliased) additions and peaks in that range which, in severe cases, can completely mask the true spectrum. There are two ways of avoiding aliasing. The first is to choose $\Delta t$ so small that energy at frequencies greater than $f_n$ is negligible. However, this requires prior knowledge of the properties of the spectrum at its high frequency end. As this is usually not available, it is commonly necessary to sample the time series in much greater detail than needed to estimate the spectrum in the region of interest. For example, it would be unnecessarily costly to sample and analyze at 30 second intervals if the interest is in periods greater than one hour. Another undesirable effect of this procedure is that increase of $f_n$ increases the bandwidth of each spectral estimate, making it more difficult to resolve peaks at lower frequencies. The other, and usually more practical, solution is to smooth the record before sampling, to remove energy at frequencies greater than $f_n$. This can be accomplished in many ways, for example by designing the recording apparatus so that it is insensitive to higher frequencies, or by sampling at higher frequencies than desired and then averaging. Rarely, in digital time series analysis, can the possibility of aliasing be ignored.

The main uses of the cospectrum and quadrature spectrum are in the estimation of the coherence and phase relationships between two series, say, series 1 and series 2. Coherence is variously defined (Tukey, 1967), and the definition used here is:

$$\text{coherence}_k = \frac{(\text{cospectrum}_k)^2 + (\text{quadspectrum}_k)^2}{(\text{spectrum}_{1_k}) \times (\text{spectrum}_{2_k})}$$

where k refers to the kth lag. This generates a (coherence) number ranging from 0 to 1, which is analogous to the correlation coefficient of classical statistics. That is, a value near 1 indicates that series 1 and 2 correlate closely, in terms of variance contribution, at a particular frequency interval, while a value near 0 indicates little correlation between the two series at that frequency interval. The phase spectrum represents the angular lead of the series 2 over series 1, at each frequency interval.

## RESULTS

The programs are presented in Appendix A (ALGOL) and in Appendix B (FORTRAN). The following steps describe the operation of the programs.

(1) Read in a title card to be used in labeling the output.

(2) Read in the parameters N (number of data points in the time series), DELTAT (time interval between observations), NUMSER (the number of time series to be analyzed) and LAGS (the number of spectral estimates to be made). In the FORTRAN version (Appendix B), N and LAGS must be checked to be sure that they are within the permissible dimensions of the arrays of the program; no such checks are necessary in the ALGOL program, since arrays can be dynamically dimensioned.

(3) Read in the time series.

(4) Remove the trend from the time series by least squares fitting of a straight line to the data and subtraction to obtain residuals.

(5) Append zeros to the end of the data to extend N to an exact power of 2.

(6) Fourier transform the time series to obtain the complex Fourier coefficients, $a_{1k} + ib_{1k}$ where $i = \sqrt{-1}$, $k = 0, 1, \ldots, N$.

(7) If there is a second time series involved, repeat steps (3) through (6) to obtain $a_{2k} + ib_{2k}$ .

(8) Square and sum both the real and imaginary Fourier coefficients over the number of bands desired and normalize by the number of

terms squared

$$\sum_{k=j}^{\ell} (a_k^2 + b_k^2) / (\ell - j + 1)$$

to obtain the raw spectral estimates for both series (1 and 2).

(9) The raw cross spectrum is obtained by taking the sum of the cross products of the real parts of the transformed series plus the cross products of the complex parts

$$\sum_{k=j}^{\ell} (a_{1_k} \times b_{2_k} + b_{1_k} \times b_{2_k}) / (\ell - j + 1).$$

(10) The raw quadrature spectrum is the sum of the products of the real part of series 1 and the imaginary part of series 2, minus the product of the real part of series 2 and the imaginary part of series 1

$$\sum_{k=j}^{\ell} (a_{1_k} \times b_{2_k} - a_{2_k} \times b_{1_k}) / (\ell - j + 1).$$

(11) Smooth the raw estimates by hanning (defined in Blackman and Tukey, 1957).

(12) An estimate of the phase difference between the two series, for a particular frequency band, is given by the arctangent ($\tan^{-1}$) of the ratio of the smoothed value of the quadrature spectrum to that of the co-spectrum for that frequency band.

(13) The coherence at a particular frequency band is the sum of the squares of the cospectrum and the quadrature spectrum, divided by

the product of the two spectra.

(14) Output the results.

The form of the output is illustrated by an example (from Mortimer and Fee, 1969) in Figure 1. Nearly six months of water level records (U. S. Lake Survey analog traces) at two stations were digitized, taking half-hourly means to yield 8545 data points for each station.

No "prewhitening" (defined in Blackman and Tukey, 1957) or other type of data smoothing, other than trend and mean removal, has been incorporated into the program. Such treatment was omitted to keep the basic program simple, so that it can be used in a variety of applications. Data from different sources may require particular kinds of treatment; and it is best left to the investigator, familiar with the data, to decide which treatment is needed.

It should be noted that the program will not work if missing data produce gaps in the series. In practice, however, data gaps will occasionally occur, if mechanical devices are used to record the data. Where the data gap is of short duration (a qualitative term, of course), linear interpolation may be an adequate solution, and this is the technique employed in Mortimer and Fee (1969). Longer gaps may require analysis of the data in two or more sections.

Techniques for setting confidence limits on the estimated energy and phase spectra are given by Munk, *et al.* (1959). Methods of setting confidence limits on the coherence spectra are discussed by Jenkins, *et al.* (1968, p. 379).

## DISCUSSION

The FORTRAN version (Appendix B) has been kept very general, so that it can be compiled on almost any FORTRAN system, including FORTRAN II if the input/output statements are changed. Where possible trouble may arise with different compilers, this has been pointed out in the program. The comments preceding the program explain the type and format of the input data. Subroutine FOURT, which is used to obtain the Fourier transform of the data, is a slightly modified version of a subroutine by Brenner (1967). This program has been tested on both Univac 1108 and Burroughs B5500 computers.

In contrast, the ALGOL version is highly modified to take advantage of the (at present) unique software capabilities of the Burroughs B5500 Extended ALGOL compiler. Because of the extent of the modifications required to make this program compatible with the reference language ALGOL 60, these changes will be considered in detail here rather than by the use of internal comments within the program.

The B5500 ALGOL limits the size of any single dimension of an array to 1023 words; however, there is no limit to the number of dimensions an array may have. Thus, one may use consecutive rows of a multidimensioned array to simulate a singly dimensioned array that alone would far exceed the core memory of the machine. The operating system (Master Control Program) automatically makes the proper row available when it is referenced in the program. This method of handling arrays is referred to as "virtual core." To make use of this concept effectively, it is necessary to use partial word indexing to designate the particular array row that is being used. This is done in the following manner: with 9 binary digits (bits), it is possible to reference numbers from 0 to $(2^9 - 1)$, i.e., 0 to 511. When the number exceeds this size, the next higher bit of the word is turned on and the lower 9 order bits start recounting at 0. For example, the bit configuration for the number 515 would be (using 48 bit word in which bit 47 is the least significant):

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| 0 | | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | ← bit index |
| | | Subscript 1 (row index) | | | | Subscript 2 (column index) | | | | | | | | | |

Therefore, by using the last 9 bits of an integer index as the last subscript and the next 9 high order bits as the first subscript, it is possible to use a double dimensioned array to simulate a single dimensioned array while still using a single index for subscripting. For example, in the program A[K1.[30:9],K1.[39:9]] has been used in place of A [K1]. If K1 were less than 512, this notation would designate an element of the 0th row. An index between 512 and 1023 would cause the 1st row to be referenced, and so on. With this technique, it is possible to use arrays having as many as 512 x 512 = 256,144 elements. If larger arrays are needed, another dimension could easily be added by using the next 9 high order bits of an index as the 3rd subscript.

This same technique can be used with the Univac 1108 Extended ALGOL compiler. To save keypunching, this has been implemented by the use of DEFINE declarations within the program. The first one appears as the first declaration of the program. In order to make this compatible with other systems lacking this capability, these DEFINE declarations should be removed and, at each place where the defined variables occur in the program, they should be changed to a simple subscript, e.g., A[KOP] should be changed to A[K0]. None of the other logic of the program will be affected by these changes. The declarations for the affected arrays must also be changed to reflect the lower numbers of subscripts.

In B5500 ALGOL it is also necessary to specify the lower bounds of subscripts in procedure headings. This can be changed to standard ALGOL 60 by merely removing these lower bounds, e.g., in the procedure heading of REVFFT2 change "ARRAY A, B[0,0];" to "ARRAY A,B;".

The last major deviation is that B5500 ALGOL requires that all labels be declared before they are used. ALGOL 60 does not allow this, and all LABEL declarations should be removed to make the program compatible.

Input/output statements and declarations will not be discussed since they are nonstandardized in ALGOL 60. To resolve these and any other questions about the program, see the Burroughs B5500 Extended

ALGOL Reference Manual (1966). Procedures REVFFT2, REORDER, REALTRAN, and REALTRANSFORM, which are used to find the Fourier transforms, are modified from Singleton (1968).

Of the two programs, the ALGOL version is faster. It is set up to operate efficiently on a system with "virtual core", doing as much computation as possible on one array row before accessing other rows (Singleton, 1967). On such a system (e.g., the B5500) the FORTRAN program spends an excessive amount of time overlaying storage. Also, the ALGOL version is specifically designed for real data, thus using only half the storage space required by the FORTRAN version, which assumes complex data even though the programs are intended for real data only.

ACKNOWLEDGEMENTS

Special thanks go to Mr. G. Haller, president of Shared Computer Systems, Inc., Chicago, for access to the B5500 computer under his management. His help made the development of the program possible. Without it, the large quantities of available data could not have been analyzed so rapidly. Miss Marian Pierce helped with keypunching the programs and document preparation. Mr. Ratko Ristić prepared the figure. Dr. C. H. Mortimer reviewed the manuscript and made invaluable comments and suggestions. Mr. P. C. Volkman not only provided valuable council when the ALGOL program was being developed but also reviewed portions of the manuscript. Further comments or corrections by users would be welcomed.

## REFERENCES


Bingham, C., M.D. Godfrey, and J. W. Tukey. 1967. Modern techniques of power spectrum estimation. IEEE, Trans. on Audio and Electroacoustics, AU-15: 56-66.

Blackman, R. B. and J. W. Tukey. 1957. The Measurement of Power Spectra from the Point of View of Communication Engineering. Dover Pub., New York. 190 pp.

Brenner, N.M. 1967. Three FORTRAN programs that perform the Cooley-Tukey Fourier transform. Tech. Note, 1967-2, Lincoln Laboratory, Mass. Inst. Tech., Lexington, Mass. 29 pp.

Burroughs Corp. 1966. Burroughs B5500 Information Processing Systems. Extended ALGOL Reference Manual. Detroit, Mich.

Cooley, J.W., and J.W. Tukey. 1965. An algorithm for the machine calculation of Fourier series. Math. Comput., 19: 297-301.

Jenkins, G.M. and D. W. Watts. 1968. Spectral Analysis and Its Applications. Holden-Day, San Francisco. 525 pp.

Mortimer, C.H. and E. J. Fee. 1969. Spectra, coherence and phase relationships of low-frequency water level fluctuations at shore stations on Lake Michigan-Huron and on Lake Superior. To be presented at the 12th Conference on Great Lakes Research, Ann Arbor, Michigan, May 5-7, 1969 (abstract presented in App. C).

Munk, W. H., F.E. Snodgrass, and M.J. Tucker. 1959. Spectra of low-frequency ocean waves. Bull. Scripps Inst. Oceanogr., 7: 283-362.

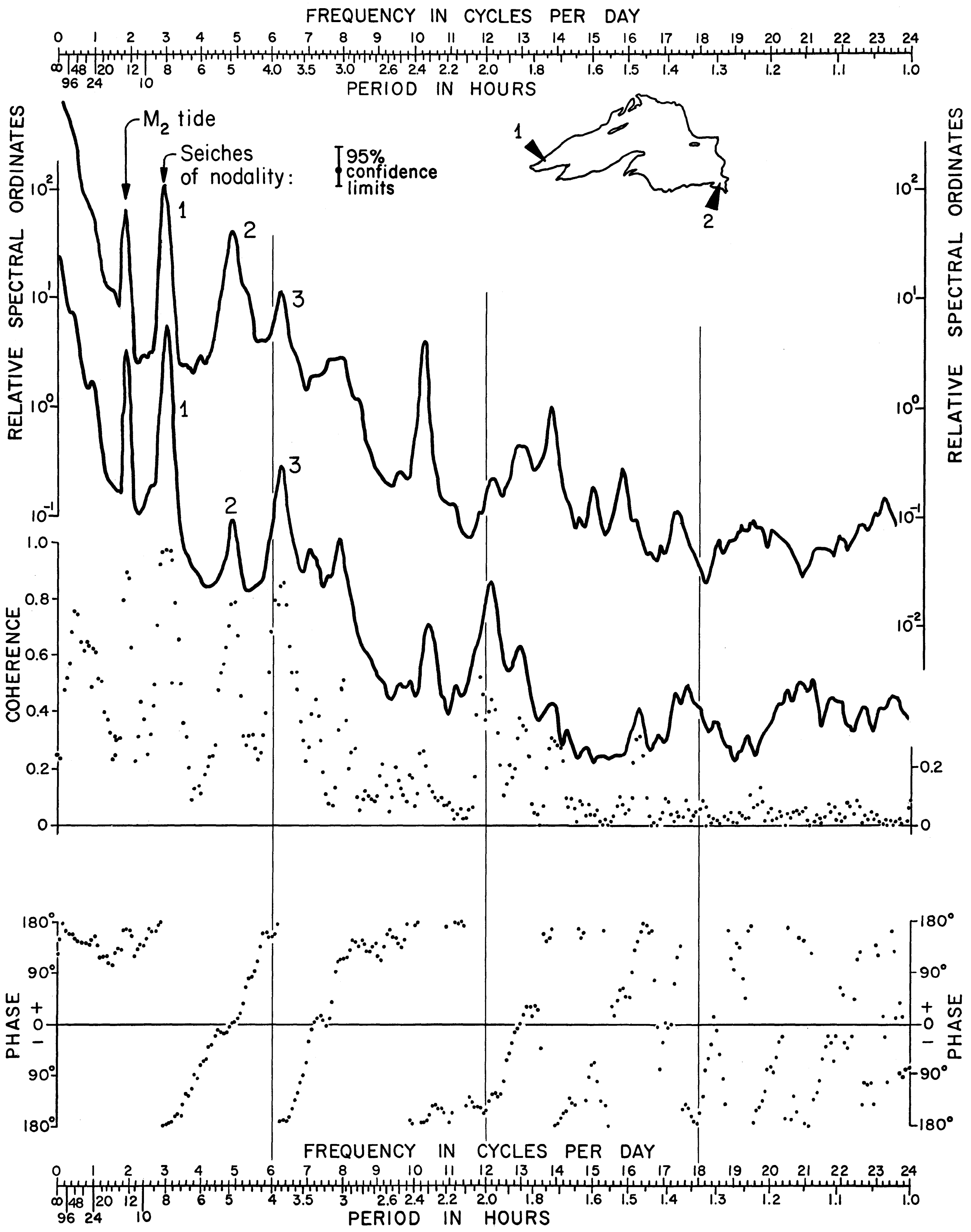
Singleton, R. C. 1967. On computing the fast Fourier transform. Comm. Assoc. Computing Mach., 10: 647-654.

_______. 1968. Algorithm 338. ALGOL Procedures for the fast Fourier transform. Comm. Assoc. Computing Mach., 11: 773-776.

Tukey, J.W. 1967. An introduction to the calculations of numerical spectrum analysis. pp. 25-46, in Harris, B. [ed.], Advanced Seminar on Spectral Analysis of Time Series. John Wiley and Sons, New York. 319 pp.

FIGURE 1

Energy (variance) spectra of water level fluctuations (from a data input of 8545 30-minute means) at two shore stations on Lake Superior: Series 1 (upper spectrum), Two Harbors, Mich; Series 2, Point Iroquois, Mich. The coherence and phase relationships between the stations (defined in the text) are also illustrated. The spectral peaks, identified in the figure, represent the lunar tide and the uninodal, binodal, and trinodal free longitudinal oscillations (seiches) of the basin. Note that the phase diagram confirms that the two stations are out of phase for the lunar tide and the uni- and trinodal seiches (i.e., of odd nodality), while they are in phase for the binodal seiche (even nodality). Further features of this and similar figures are discussed by Mortimer and Fee (1969).


FREQUENCY IN CYCLES PER DAY
PERIOD IN HOURS
RELATIVE SPECTRAL ORDINATES
M2 tide
Seiches of nodality:
95% confidence limits
COHERENCE
PHASE
FREQUENCY IN CYCLES PER DAY
PERIOD IN HOURS

# APPENDIX A

```
BEGIN
  COMMENT  BURROUGHS EXTENDED ALGOL PROGRAM FOR FINDING THE SPECTRA,
             COSPECTRA, QUADSPECTRA, COHERENCE, AND PHASE OF TWO TIME
             SERIES OR A SINGLE SPECTRUM OF ONE SERIES. THIS PROGRAM
             USES THE FAST FOURIER TRANSFORM.  THE INPUT AND OUTPUT
             FILES USED ARE:
         FILE NAME
         ---------
            CRA: CONTAINS TITLE CARD AND INPUT PARAMETERS.
              CARD #1. TITLE CARD CONTAINING UP TO 80 CHARACTERS.
              CARD #2. INPUT PARAMETERS IN FREE FORMAT. THE DATA MUST BE
                         IN THE FOLLOWING ORDER:
                       1) THE NUMBER OF DATA POINTS IN THE TIME SERIES
                          (INTEGER).
                       2) THE TIME INTERVAL BETWEEN SUCCESSIVE DATA
                          POINTS (REAL NUMBER). THE UNITS OF THIS
                          CONSTANT DETERMINE THE UNITS OF THE OUTPUT.
                       3) THE NUMBER OF DATA SETS, MUST BE 1 OR 2
                          (INTEGER).
                       4) THE NUMBER OF LAGS AT WHICH ESTIMATES ARE TO
                          BE MADE (INTEGER).
           FILE1: THE FIRST TIME SERIES, IN FREE FORMAT (REAL NUMBERS).
           FILE2: THE SECOND TIME SERIES.
            CPA: THE CARD PUNCH.
            LPA: THE LINE PRINTER;

   DEFINE KOP=KO.[30:9],KO.[39:9]#,   K1P=K1.[30:9],K1.[39:9]#,
          K2P=K2.[30:9],K2.[39:9]#,   K3P=K3.[30:9],K3.[39:9]#,
          KKP=KK.[30:9],KK.[39:9]#,   KP=K.[30:9],K.[39:9]#,
          NKP=NK.[30:9],NK.[39:9]#,   JP=J.[30:9],J.[39:9]#,
          NP=N.[30:9],N.[39:9]#,   QP=Q.[30:9],Q.[39:9]#;
PROCEDURE REVFFT2 (A,B, N, M, KS); VALUE N, M, KS;
  INTEGER N, M, KS; ARRAY A, B[O,O];
BEGIN
  INTEGER KO,K1,K2,K3,K4,SPAN, NN, J, JJ, K, KB, NT, KN, MK;
  REAL RAD, C1,C2,C3,S1,S2,S3,CK,SK,SQ;
  REAL AO,A1,A2,A3,BO,B1,B2,B3,RE,IM;
  INTEGER ARRAY C[O:M];
  LABEL L,L2,L3,L4,L5,L6;
  SQ := 0.707106781187;
  SK := 0.382683432366;
  CK := 0.923879532511;
  C[O] := KS; KN := O; K4 := 4×KS; MK := M-4;
  FOR K:= 1 STEP 1 UNTIL M DO C[K] := KS := KS + KS;
  RAD := 3.14159265359 / (C[O] X KS);
L : KB := KN + K4; KN := KN + KS;
  IF M = 1 THEN GO TO L5;
  K := JJ := O; J := MK; NT := 3;
  C1 := 1.O; S1 := O;
L2:  SPAN := C[K];
  IF JJ ≠ O THEN
```

```
  BEGIN
    C2 := JJ X SPAN X RAD; C1 := CØS (C2); S1 := SIN (C2);
L3: C2 := C1 ↑ 2 - S1 ↑ 2; S2 := 2.0 X C1 X S1;
    C3 := C2 X C1 - S2 X S1; S3 := C2 X S1 + S2 X C1
  END ELSE S1 := 0;
  K3 := KB - SPAN;
L4: K2 := K3 - SPAN; K1 := K2 - SPAN; K0 := K1 - SPAN;
   A0 := A[K0P]; B0 := B[K0P];
    A1 := A[K1P]; B1 := B[K1P];
    A2 := A[K2P]; B2 := B[K2P];
    A3 := A[K3P]; B3 := B[K3P];
    A[K0P] := A0 + A1 + A2 + A3; B[K0P] := B0 + B1 + B2 + B3;
    IF S1 = 0 THEN
    BEGIN
      A[K1P] := A0 - A1 - B2 + B3; B[K1P] := B0 - B1 + A2 - A3;
      A[K2P] := A0 + A1 - A2 - A3; B[K2P] := B0 + B1 - B2 - B3;
      A[K3P] := A0 - A1 + B2 - B3; B[K3P] := B0 - B1 - A2 + A3
    END
    ELSE
    BEGIN
      RE := A0 - A1 - B2 + B3; IM := B0 - B1 + A2 - A3;
      A[K1P] := RE X C1 - IM X S1; B[K1P] := RE X S1 + IM X C1;
      RE := A0 + A1 - A2 - A3; IM := B0 + B1 - B2 - B3;
      A[K2P] := RE X C2 - IM X S2; B[K2P] := RE X S2 + IM X C2;
      RE := A0 - A1 + B2 - B3; IM := B0 - B1 - A2 + A3;
      A[K3P] := RE X C3 - IM X S3; B[K3P] := RE X S3 + IM X C3
  END;
  K3 := K3 + 1; IF K3 < KB THEN GØ TØ L4;
  NT := NT -1;
  IF NT ≥ 0 THEN
  BEGIN
    C2 := C1;
    IF NT = 1 THEN
      BEGIN C1 := C1 X CK + S1 X SK; S1 := S1 X CK - C2 X SK END
    ELSE BEGIN C1 := (C1 - S1) X SQ; S1 := (C2+S1) X SQ END;
      KB := KB + K4; IF KB ≤ KN THEN GØ TØ L3 ELSE GØ TØ L5
  END;
  IF NT = -1 THEN BEGIN K:= 2; GØ TØ L2 END;
  IF C[J] ≤ JJ THEN
  BEGIN
    JJ:=JJ-C[J]; J := J-1;
    IF C[J] ≤ JJ THEN
      BEGIN JJ := JJ - C[J]; J := J - 1; K := K + 2 END
    ELSE BEGIN JJ := C[J] + JJ; J := MK END
  END
  ELSE BEGIN JJ := C[J] + JJ; J := MK END;
  IF J < MK THEN GØ TØ L2; K := 0; NT := 3;
  KB := KB + K4; IF KB ≤ KN THEN GØ TØ L2;
L5: K := (M ÷ 2) X 2;
  IF K ≠ M THEN
  BEGIN
```

```
    K2 := KN; KO := J := KN -C[K];
L6: K2 := K2 - 1; KO := KO - 1;
    AO := A[K2P]; BO := B[K2P];
    A[K2P] := A[KOP] - AO; A[KOP] := A[KOP] + AO;
    B[K2P] := B[KOP] - BO; B[KOP] := B[KOP] + BO;
    IF K2 > J THEN GØ TØ L6
  END;
  IF KN < N THEN GØ TØ L
END REVFFT2;
PRØCEDURE REØRDER (A, B, N, M, KS, REEL);
  VALUE N, M, KS, REEL; INTEGER N, M, KS;
  BØØLEAN REEL; ARRAY A, B[O,O];
BEGIN INTEGER I, J, JJ, K, KK, KB, K2, KU, LIM, P, Q;
  REAL T;
  INTEGER ARRAY C, LST [O:M];
  LABEL L,L2,L3,L4;
  C[M] := KS;
  FØR K := M STEP -1 UNTIL 1 DØ C[K-1]:= C[K] ÷ 2;
  P := J := M-1; I := KB := O;
  IF REEL THEN
  BEGIN
    KU := N - 2;
    FØR K := O STEP 2 UNTIL KU DØ
      BEGIN Q := K+1; T := A[QP]; A[QP] := B[KP]; B[KP] := T END
  END ELSE M := M - 1;
  LIM := (M + 2) ÷ 2; IF P ≤ O THEN GØ TØ L4;
L: KU := K2 := C[J] + KB; JJ := C[M-J]; KK := KB + JJ;
L2: K := KK + JJ;
L3: T := A[KKP]; A[KKP] := A[K2P]; A[K2P] := T;
  T := B[KKP]; B[KKP] := B[K2P]; B[K2P] := T;
  KK := KK + 1; K2 := K2 + 1;
  IF KK < K THEN GØ TØ L3;
  KK := KK + JJ; K2 := K2 + JJ;
  IF KK < KU THEN GØ TØ L2;
  IF J > LIM THEN
  BEGIN
    J := J-1; I := I+1;
    LST[I] := J; GØ TØ L
  END;
  KB := K2;
  IF I > O THEN
    BEGIN J := LST[I]; I := I-1; GØ TØ L END;
  IF KB < N THEN BEGIN J:= P; GØ TØ L END;
L4:
END REØRDER;
PRØCEDURE REALTRAN (A, B, N, EVALUATE);
  VALUE N, EVALUATE; INTEGER N;
  BØØLEAN EVALUATE; ARRAY A, B[O,O];
BEGIN INTEGER K, NK, NH;
  REAL AA, AB, BA, BB, RE, IM, CK, SK, DC, DS, R;
  NH := N ÷ 2; R := 3.14159265359 / N;
```

```
  DS := SIN (R); R := -(2XSIN(0.5XR))↑2;
  DC := -0.5 x R; CK := 1.0; SK := 0;
  IF EVALUATE THEN
    BEGIN CK := -1.0; DC := -DC END
  ELSE BEGIN A[NP] := A[0,0]; B[NP] := B[0,0] END;
  FØR K := 0 STEP 1 UNTIL NH DØ
  BEGIN
    NK := N - K;
    AA := A[KP] + A[NKP]; AB := A[KP] - A[NKP];
    BA := B[KP] + B[NKP]; BB := B[KP] - B[NKP];
    RE := CK X BA + SK X AB; IM := SK X BA - CK X AB;
    B[NKP] := IM - BB; B[KP] := IM + BB;
    A[NKP] := AA - RE; A[KP] := AA + RE;
    DC := R x CK + DC; CK := CK + DC;
    DS := R X SK + DS; SK := SK + DS
  END
END REALTRAN;
PRØCEDURE REALTRANSFØRM (A,B,M);
  VALUE M;  INTEGER M;  ARRAY A,B[0,0];
BEGIN INTEGER N,J; REAL P;
  N := 2↑M;
  REØRDER (A,B,N,M,N,TRUE);
  REVFFT2(A,B,N,M,1); P:= 0.5;
  FØR J:= N-1 STEP -1 UNTIL 0 DØ
     BEGIN A[JP] := P XA[JP]; B[JP]:= P x B [JP] END;
  REALTRAN (A,B,N,FALSE)
END REALTRANSFØRM;
    FILE FILE1 DISK SERIAL(3,10,30);
    FILE FILE2 DISK SERIAL(3,10,30);
    FILE LPA 18(2,17);
    FILE CPA 0(2,10);
    FILE CRA(2,10);
  INTEGER I,J,K,L,LAGS,NTWØ,N,NUMSER,N1,N2,M;
  REAL FACTØR,DELTAT,F,BW,P;
  DEFINE I1=I.[30:9],I.[39:9]#,J1=J.[30:9],J.[39:9]#,
         K1=K.[30:9],K.[39:9]#,L1=L.[30:9],L.[39:9]#;
   PRØCEDURE HANN(A); ARRAY A[0,0];
    BEGIN
      REAL T1,T2;
      A[0,0]:=0.5X((T1:=A[0,0])+A[0,1]);
      K := LAGS-1;
      FØR I:=1 STEP 1 UNTIL K DØ
        BEGIN
          J:=I+1;   T2:=A[I1];
          A[I1]:=0.5XT2 + 0.25X(A[J1] + T1);
          T1:=T2;
        END;
      A[J1]:=0.5X(T1+A[J1]);
   END HANNING;
   PRØCEDURE INPUT(A,B,INFILE,FIRSTIME);
    ARRAY A,B[0,0];   FILE INFILE;  BØØLEAN FIRSTIME;
```

```
BEGIN
   REAL SUMY,SUMXY,F1,F2,SLØPE,INTER;
   IF FIRSTIME THEN
     BEGIN
       M:=-1; FØR NTWØ:=1,2XNTWØ WHILE NTWØ < N DØ M:=M+1;
       N2:=NTWØ ÷ 2;  NTWØ:=N2-1;  N1:=N-NTWØ-2;
     END;
   READ(INFILE,/,FØR I:=O STEP 1 UNTIL NTWØ DØ A[I1],
                 FØR I:=O STEP 1 UNTIL N1 DØ B[I1]);
   SUMY := SUMXY := 0.0;
   FØR I:=O STEP 1 UNTIL NTWØ DØ
     BEGIN SUMY := SUMY+A[I1]; SUMXY := SUMXY+A[I1]X(I+1); END;
   FØR I:=O STEP 1 UNTIL N1 DØ
     BEGIN SUMY := SUMY+B[I1];  SUMXY := SUMXY+B[I1]X(N2+I+1); END;
   SUMXY := DELTATXSUMXY; F1:=0.5XDELTATX(NX(N+1));  F2:=SUMY/N;
   SLØPE:=(SUMXY-F1XF2)/(F1X(DELTATX(2XN+1)/3-SUMY:=F1/N));
   INTER:=F2-SLØPEXSUMY;   FACTØR:=SLØPEXDELTAT;
   FØR I:=O STEP 1 UNTIL NTWØ DØ A[I1]:=A[I1]-INTER-FACTØRX(I+1);
   FØR I:=O STEP 1 UNTIL N1 DØ B[I1]:=B[I1]-INTER-FACTØRX(I+N2+1);
   FØR I:=N1+1 STEP 1 UNTIL NTWØ DØ B[I1]:=0.0;
   REALTRANSFØRM(A,B,M);
END INPUT;
PRØCEDURE SPECT1(A,B); ARRAY A,B[O,O];
  BEGIN
    FØRMAT HEAD(X45,"LAG",X7,"FREQ",X7,"PERIØD",X2,"LØG10(SPECT)"/),
           LINE(X43,I5,3(X4,F8.4)),   CARD(10F8.4);
    ARRAY WØRK[O:(LAGS-1) ÷ 512,0:511];
    K:=-1; P := 0.0;
    FØR L:=O STEP 1 UNTIL LAGS DØ
      BEGIN
        WØRK[L1]:=0.0;    J:=K+1;
        IF J=O THEN BEGIN P:=BW/2.0; K:=ENTIER(P); END
                ELSE K:=IF(P:=P+BW) > N2 THEN N2 ELSE ENTIER(P+0.5);
        FØR I:=J STEP 1 UNTIL K DØ
                 WØRK[L1] := WØRK[L1] + A[I1]XA[I1] + B[I1]XB[I1];
        WØRK[L1]:=WØRK[L1]/(K-J+1);
      END;
      HANN(WØRK);
      WRITE(LPA[DBL],HEAD);
      FACTØR:=0.5/(LAGSXDELTAT);
      FØR I:=O STEP 1 UNTIL LAGS DØ WRITE(LPA,LINE,I,F:=FACTØRXI,
         IF I>O THEN 1.O/F ELSE 999.9999,WØRK[I1]:=0.43429448191X
                                                      LN(WØRK[I1]));
      WRITE(CPA,CARD,FØR I:=O STEP 1 UNTIL LAGS DØ WØRK[I1]);
END SPECT1;
PRØCEDURE SPECT2(A,B);   ARRAY A,B[O,O];
  BEGIN
    ARRAY A2,B2[O:(N-1) ÷ 512,0:511], SPEC1,SPEC2,QUASP,CØSP[O:
          (LAGS-1) ÷ 512,0:511];
    FØRMAT HEAD(X2,"LAG",X3,"FREQ",X3,"PERIØD",X6,"LØG(SPEC1)",X2,
          "LØG(SPEC2)",X2,"CØHER ↑2",X6,"PHASE",X10,"CØ-SPEC",X9,
```

```
           "QUA-SPEC",/), LINE(I5,2(F8.4,X2),3(X3,F9.5),X3,F10.5,
           2(X3,E15.9)), CARD(7E11.4);
      REAL PHASE,CØH,T1,T2,T3;
      INPUT(A2,B2,FILE2,FALSE);  FACTØR:=0.5/(LAGSXDELTAT);
      T3:=0.4342944819l;  K:=-1;  P := 0;
      FØR L:=0 STEP 1 UNTIL LAGS DØ
        BEGIN
          CØSP[L1]:=QUASP[L1]:=SPEC1[L1]:=SPEC2[L1]:=0.0;
          J:=K+1;
          IF J=0 THEN BEGIN P:=BW/2.0; K:=ENTIER(P); END
                  ELSE K:=IF(P:=P+BW) > N2 THEN N2 ELSE ENTIER(P+0.5);
          FØR I:=J STEP 1 UNTIL K DØ
            BEGIN
              CØSP[L1]:=CØSP[L1]+A[I1]XA2[I1]+B[I1]XB2[I1];
              QUASP[L1]:=QUASP[L1]-A[I1]XB2[I1]+A2[I1]XB[I1];
              SPEC1[L1]:=SPEC1[L1]+A[I1]XA[I1]+B[I1]XB[I1];
              SPEC2[L1]:=SPEC2[L1]+A2[I1]XA2[I1]+B2[I1]XB2[I1];
             END;
          CØSP[L1]:=CØSP[L1]X(F:=1.0/(K-J+1));  SPEC1[L1]:=SPEC1[L1]XF;
          QUASP[L1]:=QUASP[L1]XF;   SPEC2[L1]:=SPEC2[L1]XF;
        END;
        HANN(CØSP);   HANN(QUASP);   HANN(SPEC1);   HANN(SPEC2);
        WRITE(LPA[DBL],HEAD);
        FØR I:=0 STEP 1 UNTIL LAGS DØ
          BEGIN
            WRITE(LPA,LINE,I, F:=IXFACTØR,
              P:=IF I=0 THEN 999.9999 ELSE 1.0/F,
              T1:=T3XLN(SPEC1[I1]),  T2:=T3XLN(SPEC2[I1]),
              CØH:=(CØSP[I1]XCØSP[I1]+QUASP[I1]XQUASP[I1])/(SPEC1[I1] X
                                                         SPEC2[I1]),
              PHASE:=(ARCTAN(QUASP[I1]/CØSP[I1]) + (IF CØSP[I1] ≥ 0 THEN
                  0 ELSE SIGN(QUASP[I1])X3.1415926536))X57.2957795131,
              CØSP[I1],QUASP[I1]);
            WRITE(CPA,CARD,F,T1,T2,CØH,PHASE,CØSP[I1],QUASP[I1]);
          END;
   END SPECT2;
    ALPHA ARRAY TITLE[0:13];
    FØRMAT HEAD(X15,"TIME SERIES SPECTRUM ANALYSIS PRØGRAM. EVERETT J. F
EE. CENTER FØR GREAT LAKES STUDIES, UWM. 1968."//X32,13A6,A2/X51,"NUMBER
 ØF DATA PØINTS = ",I5/X55,"NUMBER ØF LAGS = ",I4/X36,"TIME INTERVAL BET
WEEN DATA PØINTS = ",F10.5," TIME UNITS."//), ALFA(13A6,A2);
    READ(CRA,ALFA,FØR I:=0 STEP 1 UNTIL 13 DØ TITLE[I]);
    READ(CRA,/,N,DELTAT,NUMSER,LAGS);
      BEGIN
        ARRAY A,B[0:(N-1) ÷ 512, 0:511];
        INPUT(A,B,FILE1,TRUE);
        BW := N2 / LAGS;
        WRITE(LPA,HEAD,FØR I:=0 STEP 1 UNTIL 13 DØ TITLE[I],N,LAGS,
                                                           DELTAT);
        WRITE(CPA,ALFA,FØR I:=0 STEP 1 UNTIL 13 DØ TITLE[I]);
        IF NUMSER=1 THEN SPECT1(A,B) ELSE SPECT2(A,B);
      END;
END.
```

# APPENDIX B

```
C ***********************************************************************
C ** POWER SPECTRUM, COHERENCE, AND PHASE PROGRAM WRITTEN IN FORTRAN IV
C     AND UTILIZING THE COOLEY-TUKEY FAST FOURIER TRANSFORM ALGORITHM.
C ** THE LOGICAL UNIT NUMBERS REFER TO THE FOLLOWING PHYSICAL UNITS..
C       1 IS THE CARD PUNCH
C       5 IS THE CARD READER
C       6 IS THE LINE PRINTER
C ** UNLESS THE USER CHANGES THE INPUT/OUTPUT STATEMENTS, THE INPUT DECK
C    MUST BE OF THE FOLLOWING FORM..
C     CARD  1  ALPHANUMERIC TITLE CARD. ALL 80 COLUMNS MAY BE USED.
C     CARD  2
C       COLUMNS
C         1-5   THE NUMBER OF DATA POINTS IN THE TIME SERIES (INTEGER,
C                RIGHT JUSTIFIED). MUST BE < OR = 8192 UNLESS THE USER
C                CHANGES DIMENSION STATEMENTS IN MAIN PROGRAM AND SPEC2.
C         6-10  TIME INTERVAL BETWEEN OBSERVATIONS (REAL NUMBER). FOR
C                EXAMPLE, 0.5 WOULD BE USED FOR 1/2 HOUR, 1/2 MIN., ETC.
C                THE UNITS OF THIS QUANTITY DETERMINE THE UNITS OF THE
C                OUTPUT.
C        11-14  BLANK
C           15  THE NUMBER OF TIME SERIES BEIGN ANALYZED. IF 1 THEN ONLY
C                ONE SPECTRUM WILL BE PRODUCED. IF 2 THEN BOTH SPECTRA,
C                COHERENCE, PHASE, COSPECTRUM, AND QUADSPECTRUM WILL BE
C                OUTPUT. THE RESULTS ARE BOTH PRINTED AND PUNCHED.
C        16-20  THE NUMBER OF LAGS THAT ARE TO BE ESTIMATED IN EACH
C                SPECTRUM, ETC. THE PROGRAM WILL TAKE UP TO 900 AS NOW
C                WRITTEN. THIS CAN BE CHANGED BY ADJUSTING THE DIMENSION
C                STATEMENTS IN SPEC1 AND SPEC2.
C     CARD  3   A FORMAT CARD, DESCRIBING WHERE THE DATA ARE TO BE FOUND
C                ON A LOGICAL RECORD. IF THIS IS ILLEGAL ON YOUR SYSTEM
C                THEN SUBROUTINE INPUT MUST BE COMPILED WITH A FIXED
C                FORMAT DESCRIBING THE DATA.
C     CARD  4 AND FOLLOWING ARE THE DATA. IF THERE IS A SECOND DATA SET,
C                IT MUST BE PRECEDED BY A FORMAT CARD ALSO.
C ***********************************************************************
C
      DIMENSION DATA(2,8192),TITLE(80)
      READ(5,1) TITLE
    1 FORMAT(80A1)
      READ(5,2) N,DELTAT,NUMSER,LAGS
    2 FORMAT(I5,F5.2,I5,I5)
      WRITE(6,3) TITLE,N,LAGS,DELTAT
    3 FORMAT('1',15X,'TIME SERIES SPECTRUM ANALYSIS PROGRAM. EVERETT J.
     1FEE. CENTER FOR GREAT LAKES STUDIES, UWM. 1968.'///32X,80A1/51X
     2,'NUMBER OF DATA POINTS = ',I4/55X,'NUMBER OF LAGS = ',I4,/36X,'TI
     2ME INTERVAL BETWEEN DATA POINTS = ',F10.5,' TIME UNITS.'//)
      IF(N-8192) 6,6,9
    6 IF(LAGS-900) 8,8,9
    8 CALL INPUT(DATA,N,DELTAT,NWORK,5)
      BW = FLOAT(NWORK)/(2.0*FLOAT(LAGS))
      IF(NUMSER-2) 4,5,11
    4 CALL SPEC1(DATA,NWORK/2,DELTAT,LAGS,BW)
      STOP
    5 CALL SPEC2(DATA,N,DELTAT,LAGS,BW)
      STOP
    9 WRITE(6,10)
   10 FORMAT('0TOO MANY DATA POINTS OR LAGS. JOB TERMINATED')
      STOP
   11 WRITE(6,12)
   12 FORMAT('0NUMBER OF SERIES MUST BE 1 OR 2. JOB TERMINATED.')
```

```
      STOP
      END


      SUBROUTINE INPUT(DATA,N,DELTAT,NWORK,IFILE)
      DIMENSION DATA(2,8192),NN(1),FORM(6),WORK(1)
      READ(IFILE,1) FORM
    1 FORMAT(6A6)
C     SOME MACHINES MAY NOT ALLOW MANY AS 6 ALPHANUMERIC CHARACTERS PER
C     WORD OF STORAGE. CHECK YOUR LOCAL INSTALLATION ON THIS POINT.
      READ(IFILE,FORM) (DATA(1,K),K=1,N)
      SUMY=0.0
      SUMXY = 0.0
      DO 2 I=1,N
      SUMY = SUMY + DATA(1,I)
    2 SUMXY = SUMXY + DATA(1,I)*FLOAT(I)
      SUMXY = SUMXY*DELTAT
      F1 = 0.5*DELTAT*FLOAT(N*(N+1))
      F2 = SUMY/FLOAT(N)
      SLOPE = (SUMXY-F1*F2)/(F1*(DELTAT*FLOAT(2*N+1)/3.0-F1/FLOAT(N)))
      AINTER = F2-SLOPE*F1/FLOAT(N)
      FACTOR = SLOPE*DELTAT
      DO 3 I=1,N
    3 DATA(1,I) = DATA(1,I)-AINTER-FACTOR*FLOAT(I)
      NWORK = 2
    4 IF(NWORK-N) 9,7,5
    9 NWORK = 2*NWORK
      GO TO 4
    5 N1 = N+1
      DO 6 I=N1,NWORK
    6 DATA(1,I) = 0.0
    7 DO 8 I=1,NWORK
    8 DATA(2,I) = 0.0
      NN(1) = NWORK
      CALL FOURT(DATA,NN,1,-1,0,WORK)
      RETURN
      END


      SUBROUTINE HANN(A,LAGS)
      DIMENSION A(1)
      T1=A(1)
      A(1) = (0.5*(T1+A(2)))
      K=LAGS-1
      DO 1 I=2,K
      T2 = A(I)
      A(I) = 0.5*T2 + 0.25*(T1+A(I+1))
    1 T1 = T2
      A(LAGS) = (0.5*(T1+A(LAGS)))
      RETURN
      END


      SUBROUTINE SPEC1(DATA,N,DELTAT,NDIM,BW)
      DIMENSION DATA(2,8192),WORK(820)
      FACTOR = 0.5/(FLOAT(NDIM)*DELTAT)
      LAGS=NDIM+1
      WRITE(6,1)
    1 FORMAT('0',45X,'LAG',7X,'FREQ',7X,'PERIOD',2X,'LOG10(SPECTRUM)'/)
      K=0
      P = 0.0
```

```
      N=N+1
      DO 6 I=1,LAGS
      WORK(I) = 0.0
      J=K+1
      IF(J-1) 13,13,2
   13 P = BW/2.0+1
      K = IFIX(P)
      GO TO 4
    2 P = P+BW
      IF(P-N) 14,14,3
   14 K = IFIX(P+0.5)
      GO TO 4
    3 K=N
    4 DO 5 L=J,K
    5 WORK(I) = WORK(I)+DATA(1,L)*DATA(1,L)+DATA(2,L)*DATA(2,L)
    6 WORK(I) = WORK(I)/FLOAT(K-J+1)
      CALL HANN(WORK,LAGS)
      DO 7 I=1,LAGS
    7 WORK(I) = ALOG10(WORK(I))
C     ALOLG10(X) MAY NOT BE A LIBRARY FUNCTION ON YOUR MACHINE. IF NOT,
C     USE 2.3025805*ALOG(X).
      DO 10 I=1,LAGS
      FREQ = FLOAT(I-1)*FACTOR
      IF(I-1) 15,15,8
   15 PERIOD=999.9999
      GO TO 9
    8 PERIOD = 1.0/FREQ
    9 J=I-1
   10 WRITE(6,11) J,FREQ,PERIOD,WORK(I)
   11 FORMAT(43X,I5,3(4X,F8.4))
      WRITE(1,12) (WORK(I),I=1,LAGS)
   12 FORMAT(10F8.4)
      RETURN
      END


      SUBROUTINE SPEC2(DATA1,N,DELTAT,NDIM,BW)
      DIMENSION COSP(900),QUASP(900),SPECT1(900),SPECT2(900)
      DIMENSION DATA1(2,8192),DATA2(2,8192)
      CALL INPUT(DATA2,N,DELTAT,NWORK,5)
      FACTOR = 0.5/(FLOAT(NDIM)*DELTAT)
      N=NWORK/2+1
      LAGS=NDIM+1
      K=0
      P = 0.0
      DO 5 I=1,LAGS
      COSP(I) = 0.0
      QUASP(I) = 0.0
      SPECT1(I) = 0.0
      SPECT2(I) = 0.0
      J=K+1
      IF(J-1) 1,14,1
   14 P = BW/2.0+1
      K = IFIX(P)
      GO TO 3
    1 P = P+BW
      IF(P-N) 13,13,2
   13 K = IFIX(P+0.5)
      GO TO 3
    2 K = N
    3 DO 4 L=J,K
```

```
      COSP(I) = COSP(I)+DATA1(1,L)*DATA2(1,L)+DATA1(2,L)*DATA2(2,L)
      QUASP(I)=QUASP(I)+DATA1(1,L)*DATA2(2,L)-DATA1(2,L)*DATA2(1,L)
      SPECT1(I) = SPECT1(I)+DATA1(1,L)*DATA1(1,L)+DATA1(2,L)*DATA1(2,L)
    4 SPECT2(I) = SPECT2(I)+DATA2(1,L)*DATA2(1,L)+DATA2(2,L)*DATA2(2,L)
      FACT=1.0/FLOAT(K-J+1)
      COSP(I)=COSP(I)*FACT
      QUASP(I) = QUASP(I)*FACT
      SPECT1(I) = SPECT1(I)*FACT
    5 SPECT2(I) = SPECT2(I)*FACT
      CALL HANN(SPECT1,LAGS)
      CALL HANN(SPECT2,LAGS)
      CALL HANN(QUASP,LAGS)
      CALL HANN(COSP,LAGS)
      WRITE(6,6)
    6 FORMAT('0  LAG',3X,'FREQ',3X,'PERIOD',6X,'LOG(SPEC1)',2X,'LOG(SPEC
     12)',2X,'COHER**2',6X,'PHASE',10X,'CO-SPEC',9X,'QUA-SPEC'/)
      DO 10 I=1,LAGS
      COHER = (COSP(I)*COSP(I)+QUASP(I)*QUASP(I))/(SPECT1(I)*SPECT2(I))
C     IF ATAN2(X,Y) IS NOT A LIBRARY FUNCTION, USE THE FOLLOWING
C     ALGORITHM TO OBTAIN IT.. IF Y < 0 THEN ATAN(X/Y)+SIGN(X)*PI ELSE
C     ATAN(X/Y).
      PHASE = ATAN2(QUASP(I),COSP(I))*57.29578
      SPECT1(I) = ALOG10(SPECT1(I))
      SPECT2(I) = ALOG10(SPECT2(I))
      FREQ = FLOAT(I-1)*FACTOR
      J=I-1
      IF(I-1) 12,12,7
   12 PERIOD = 999.9999
      GO TO 8
    7 PERIOD = 1.0/FREQ
    8 WRITE(6,9) J,FREQ,PERIOD,SPECT1(I),SPECT2(I),COHER,PHASE,COSP(I),
     1     QUASP(I)
    9 FORMAT(I5,2(F8.4,2X),3(3X,F9.5),3X,F10.5,2(3X,E14.9))
   10 WRITE(1,11) FREQ,SPECT1(I),SPECT2(I),COHER,PHASE,COSP(I),QUASP(I)
   11 FORMAT(7E11.5)
      RETURN
      END


      SUBROUTINE FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)
      DIMENSION DATA(1),NN(1),IFACT(32),WORK(1)
      RTHLF = 0.7071067812
      TWOPI=6.283185307
      IF(NDIM-1)920,1,1
1     NTOT=2
      DO 2 IDIM=1,NDIM
      IF(NN(IDIM))920,920,2
2     NTOT=NTOT*NN(IDIM)
      ISAVE = NTOT
      NP1=2
      DO 910 IDIM=1,NDIM
      N=NN(IDIM)
      NP2=NP1*N
      IF(N-1)920,900,5
5     M=N
      NTWO=NP1
      IF=1
      IDIV=2
10    IQUOT=M/IDIV
      IREM=M-IDIV*IQUOT
      IF(IQUOT-IDIV)50,11,11
```

```
11      IF(IREM)20,12,20
12      NTWO=NTWO+NTWO
        IFACT(IF) = IDIV
        IF = IF+1
        M=IQUOT
        GO TO 10
20      IDIV=3
        INON2 = IF
30      IQUOT=M/IDIV
        IREM=M-IDIV*IQUOT
        IF(IQUOT-IDIV)60,31,31
31      IF(IREM)40,32,40
32      IFACT(IF)=IDIV
        IF=IF+1
        M=IQUOT
        GO TO 30
40      IDIV=IDIV+2
        GO TO 30
50      INON2 = IF
        IF(IREM) 60,51,60
51      NTWO=NTWO+NTWO
        GO TO 70
60      IFACT(IF)=M
70      ICASE = 1
        IFMIN = 1
        I1RNG = NP1
        IF(IDIM-4) 71,100,100
71      IF(IFORM)72,72,100
72      IF(IDIM-1) 73,74,73
73      ICASE = 2
        I1RNG = NP0*(1+NPREV/2)
        GO TO 100
74      IF(NTWO-NP1) 75,75,76
75      ICASE = 3
        GO TO 100
76      ICASE = 4
        IFMIN=2
        NTWO=NTWO/2
        N=N/2
        NP2=NP2/2
        NTOT=NTOT/2
        I = 3
        DO 80 J=2,NTOT
        DATA(J)=DATA(I)
80      I=I+2
100     IF(NTWO-NP2)200,110,110
110     NP2HF=NP2/2
        J=1
        DO 150 I2=1,NP2,NP1
        IF(J-I2)120,130,130
120     I1MAX=I2+NP1-2
        DO 125 I1=I2,I1MAX,2
        DO 125 I3=I1,NTOT,NP2
        J3=J+I3-I2
        TEMPR=DATA(I3)
        TEMPI=DATA(I3+1)
        DATA(I3)=DATA(J3)
        DATA(I3+1)=DATA(J3+1)
        DATA(J3)=TEMPR
125     DATA(J3+1)=TEMPI
130     M=NP2HF
```

```
140   IF(J-M)150,150,145
145   J=J-M
      M=M/2
      IF(M-NP1)150,140,140
150   J=J+M
      GO TO 300
200   NWORK = 2*N
      DO 270 I1=1,NP1,2
      DO 270 I3=I1,NTOT,NP2
      J = I3
      DO 260 I=1,NWORK,2
      IF(ICASE-3)210,220,210
210   WORK(I) = DATA(J)
      WORK(I+1)=DATA(J+1)
      GO TO 230
220   WORK(I) = DATA(J)
      WORK(I+1) = 0.0
230   IFP2 = NP2
      IF = IFMIN
240   IFP1 = IFP2/IFACT(IF)
      J = J+IFP1
      IF(J-I3-IFP2) 260,250,250
250   J = J-IFP2
      IFP2 = IFP1
      IF = IF+1
      IF(IFP2-NP1) 260,260,240
260   CONTINUE
      I2MAX = I3+NP2-NP1
      I = 1
      DO 270 I2=I3,I2MAX,NP1
      DATA(I2) = WORK(I)
      DATA(I2+1) = WORK(I+1)
270   I = I+2
300   IF(NTWO-NP1) 600,600,305
305   NP1TW = NP1+NP1
      IPAR=NTWO/NP1
310   IF(IPAR-2)350,330,320
320   IPAR=IPAR/4
      GO TO 310
330   DO 340 I1=1,I1RNG,2
      DO 340 K1=I1,NTOT,NP1TW
      K2 = K1+NP1
      TEMPR=DATA(K2)
      TEMPI=DATA(K2+1)
      DATA(K2)=DATA(K1)-TEMPR
      DATA(K2+1)=DATA(K1+1)-TEMPI
      DATA(K1)=DATA(K1)+TEMPR
340   DATA(K1+1)=DATA(K1+1)+TEMPI
350   MMAX=NP1
360   IF(MMAX-NTWO/2)370,600,600
370   IF(NP1TW-MMAX/2) 372,375,375
372   LMAX = MMAX/2
      GO TO 377
375   LMAX = NP1TW
377   DO 570 L=NP1,LMAX,NP1TW
      M = L
      IF(MMAX-NP1) 420,420,380
380   THETA=-TWOPI*FLOAT(L)/FLOAT(4*MMAX)
      IF(ISIGN)400,390,390
390   THETA=-THETA
400   WR=COS(THETA)
```

```
      WI=SIN(THETA)
410   W2R=WR*WR-WI*WI
      W2I=2.*WR*WI
      W3R=W2R*WR-W2I*WI
      W3I=W2R*WI+W2I*WR
420   DO 530 I1=1,I1RNG,2
      KMIN = I1+IPAR*M
      IF(MMAX-NP1)430,430,440
430   KMIN=I1
440   KDIF=IPAR*MMAX
450   KSTEP=4*KDIF
      IF(KSTEP-NTWO) 460,460,530
460   DO 520 K1=KMIN,NTOT,KSTEP
      K2=K1+KDIF
      K3=K2+KDIF
      K4=K3+KDIF
      IF(MMAX-NP1) 470,470,480
470   U1R=DATA(K1)+DATA(K2)
      U1I=DATA(K1+1)+DATA(K2+1)
      U2R=DATA(K3)+DATA(K4)
      U2I=DATA(K3+1)+DATA(K4+1)
      U3R=DATA(K1)-DATA(K2)
      U3I=DATA(K1+1)-DATA(K2+1)
      IF(ISIGN) 471,472,472
471   U4R=DATA(K3+1)-DATA(K4+1)
      U4I=DATA(K4)-DATA(K3)
      GO TO 510
472   U4R=DATA(K4+1)-DATA(K3+1)
      U4I=DATA(K3)-DATA(K4)
      GO TO 510
480   T2R=W2R*DATA(K2)-W2I*DATA(K2+1)
      T2I=W2R*DATA(K2+1)+W2I*DATA(K2)
      T3R=WR*DATA(K3)-WI*DATA(K3+1)
      T3I=WR*DATA(K3+1)+WI*DATA(K3)
      T4R=W3R*DATA(K4)-W3I*DATA(K4+1)
      T4I=W3R*DATA(K4+1)+W3I*DATA(K4)
      U1R=DATA(K1)+T2R
      U1I=DATA(K1+1)+T2I
      U2R=T3R+T4R
      U2I=T3I+T4I
      U3R=DATA(K1)-T2R
      U3I=DATA(K1+1)-T2I
      IF(ISIGN)490,500,500
490   U4R=T3I-T4I
      U4I=T4R-T3R
      GO TO 510
500   U4R=T4I-T3I
      U4I=T3R-T4R
510   DATA(K1)=U1R+U2R
      DATA(K1+1)=U1I+U2I
      DATA(K2)=U3R+U4R
      DATA(K2+1)=U3I+U4I
      DATA(K3)=U1R-U2R
      DATA(K3+1)=U1I-U2I
      DATA(K4)=U3R-U4R
520   DATA(K4+1)=U3I-U4I
      KMIN=4*(KMIN-I1)+I1
      KDIF=KSTEP
      IF(KDIF-NP2HF) 450,530,530
530   CONTINUE
      M = M+LMAX
```

```
      IF(M-MMAX) 540,540,570
540   IF(ISIGN) 550,560,560
550   TEMPR=WR
      WR = (WR+WI)*RTHLF
      WI = (WI-TEMPR)*RTHLF
      GO TO 410
560   TEMPR = WR
      WR = (WR-WI)*RTHLF
      WI = (TEMPR+WI)*RTHLF
      GO TO 410
570   CONTINUE
      IPAR=3-IPAR
      MMAX=MMAX+MMAX
      GO TO 360
600   IF(NTWO-NP2)605,700,700
605   IFP1 = NTWO
      IF = INON2
      NP1HF=NP1/2
610   IFP2 = IFACT(IF)*IFP1
      J1MIN = NP1+1
      IF(J1MIN-IFP1) 615,615,640
615   DO 635 J1=J1MIN,IFP1,NP1
      THETA=-TWOPI*FLOAT(J1-1)/FLOAT(IFP2)
      IF(ISIGN)625,620,620
620   THETA=-THETA
625   WSTPR = COS(THETA)
      WSTPI=SIN(THETA)
      WR=WSTPR
      WI=WSTPI
      J2MIN = J1+IFP1
      J2MAX = J1+IFP2-IFP1
      DO 635 J2=J2MIN,J2MAX,IFP1
      I1MAX=J2+I1RNG-2
      DO 630 I1=J2,I1MAX,2
      DO 630 J3=I1,NTOT,IFP2
      TEMPR=DATA(J3)
      DATA(J3)=DATA(J3)*WR-DATA(J3+1)*WI
630   DATA(J3+1)=TEMPR*WI+DATA(J3+1)*WR
      TEMPR=WR
      WR=WR*WSTPR-WI*WSTPI
635   WI=TEMPR*WSTPI+WI*WSTPR
640   THETA=-TWOPI/FLOAT(IFACT(IF))
      IF(ISIGN)650,645,645
645   THETA=-THETA
650   WSTPR = COS(THETA)
      WSTPI=SIN(THETA)
      J2RNG = IFP1*(1+IFACT(IF)/2)
      DO 695 I1=1,I1RNG,2
      DO 695 I3=1,NTOT,NP2
      J2MAX = I3+J2RNG-IFP1
      DO 690 J2=I3,J2MAX,IFP1
      J1MAX = J2+IFP1-NP1
      DO 680 J1=J2,J1MAX,NP1
      J3MAX=J1+NP2-IFP2
      DO 680 J3=J1,J3MAX,IFP2
      JMIN = J3-J2+I3
      JMAX = JMIN+IFP2-IFP1
      I = 1+(J3-I3)/NP1HF
      IF(J2-I3) 655,655,665
655   SUMR=0.
      SUMI=0.
```

```
      DO 660 J=JMIN,JMAX,IFP1
      SUMR=SUMR+DATA(J)
660   SUMI=SUMI+DATA(J+1)
      WORK(I)=SUMR
      WORK(I+1)=SUMI
      GO TO 680
665   ICONJ = 1+(IFP2-2*J2+I3+J3)/NP1HF
      J=JMAX
      SUMR=DATA(J)
      SUMI=DATA(J+1)
      OLDSR=0.
      OLDSI=0.
      J=J-IFP1
670   TEMPR=SUMR
      TEMPI=SUMI
      SUMR=TWOWR*SUMR-OLDSR+DATA(J)
      SUMI=TWOWR*SUMI-OLDSI+DATA(J+1)
      OLDSR=TEMPR
      OLDSI=TEMPI
      J = J-IFP1
      IF(J-JMIN) 675,675,670
675   TEMPR=WR*SUMR-OLDSR+DATA(J)
      TEMPI=WI*SUMI
      WORK(I)=TEMPR-TEMPI
      WORK(ICONJ)=TEMPR+TEMPI
      TEMPR=WR*SUMI-OLDSI+DATA(J+1)
      TEMPI=WI*SUMR
      WORK(I+1)=TEMPR+TEMPI
      WORK(ICONJ+1)=TEMPR-TEMPI
680   CONTINUE
      IF(J2-I3) 685,685,686
685   WR=WSTPR
      WI=WSTPI
      GO TO 690
686   TEMPR=WR
      WR=WR*WSTPR-WI*WSTPI
      WI=TEMPR*WSTPI+WI*WSTPR
690   TWOWR=WR+WR
      I = 1
      I2MAX = I3+NP2-NP1
      DO 695 I2=I3,I2MAX,NP1
      DATA(I2)=WORK(I)
      DATA(I2+1)=WORK(I+1)
695   I = I+2
      IF = IF+1
      IFP1 = IFP2
      IF(IFP1-NP2) 610,700,700
700   GO TO (900,800,900,701),ICASE
701   NHALF=N
      N=N+N
      THETA=-TWOPI/FLOAT(N)
      IF(ISIGN)703,702,702
702   THETA=-THETA
703   WSTPR = COS(THETA)
      WSTPI=SIN(THETA)
      WR=WSTPR
      WI=WSTPI
      IMIN=3
      JMIN=2*NHALF-1
      GO TO 725
710   J=JMIN
```

```
      DO 720 I=IMIN,NTOT,NP2
      SUMR=(DATA(I)+DATA(J))/2.
      SUMI=(DATA(I+1)+DATA(J+1))/2.
      DIFR=(DATA(I)-DATA(J))/2.
      DIFI=(DATA(I+1)-DATA(J+1))/2.
      TEMPR=WR*SUMI+WI*DIFR
      TEMPI=WI*SUMI-WR*DIFR
      DATA(I)=SUMR+TEMPR
      DATA(I+1)=DIFI+TEMPI
      DATA(J)=SUMR-TEMPR
      DATA(J+1)=-DIFI+TEMPI
720   J=J+NP2
      IMIN=IMIN+2
      JMIN=JMIN-2
      TEMPR=WR
      WR=WR*WSTPR-WI*WSTPI
      WI=TEMPR*WSTPI+WI*WSTPR
725   IF(IMIN-JMIN)710,730,740
730   IF(ISIGN)731,740,740
731   DO 735 I=IMIN,NTOT,NP2
735   DATA(I+1)=-DATA(I+1)
740   NP2=NP2+NP2
      NTOT=NTOT+NTOT
      J=NTOT+1
      IMAX=NTOT/2+1
745   IMIN=IMAX-2*NHALF
      I=IMIN
      GO TO 755
750   DATA(J)=DATA(I)
      DATA(J+1)=-DATA(I+1)
755   I=I+2
      J=J-2
      IF(I-IMAX)750,760,760
760   DATA(J)=DATA(IMIN)-DATA(IMIN+1)
      DATA(J+1)=0.
      IF(I-J)770,780,780
765   DATA(J)=DATA(I)
      DATA(J+1)=DATA(I+1)
770   I=I-2
      J=J-2
      IF(I-IMIN)775,775,765
775   DATA(J)=DATA(IMIN)+DATA(IMIN+1)
      DATA(J+1)=0.
      IMAX=IMIN
      GO TO 745
780   DATA(1)=DATA(1)+DATA(2)
      DATA(2)=0.
      GO TO 900
800   IF(I1RNG-NP1)805,900,900
805   DO 860 I3=1,NTOT,NP2
      I2MAX=I3+NP2-NP1
      DO 860 I2=I3,I2MAX,NP1
      IMIN=I2+I1RNG
      IMAX=I2+NP1-2
      JMAX=2*I3+NP1-IMIN
      IF(I2-I3)820,820,810
810   JMAX=JMAX+NP2
820   IF(IDIM-2)850,850,830
830   J=JMAX+NP0
      DO 840 I=IMIN,IMAX,2
      DATA(I)=DATA(J)
```

```
      DATA(I+1)=-DATA(J+1)
840   J=J-2
850   J=JMAX
      DO 860 I=IMIN,IMAX,NP0
      DATA(I)=DATA(J)
      DATA(I+1)=-DATA(J+1)
860   J=J-NP0
900   NP0=NP1
      NP1=NP2
910   NPREV=N
920   IF(ISIGN) 950,950,930
930   FACT= 2.0/FLOAT(ISAVE)
      DO 940 I=1,ISAVE
940   DATA(I) = DATA(I)*FACT
950   RETURN
      END
```

## APPENDIX C

Mortimer, C. H. and E. J. Fee. ABSTRACT of: Spectra, coherence, and phase relationships of low-frequency water level fluctuations at shore stations on Lake Michigan-Huron and on Lake Superior. (to be presented at the 12th Conference on Great Lakes Research, Ann Arbor, Michigan, May 5-7, 1969).

As a first step, analysis was performed (by C.H.M.) on 18 months of hourly water level readings (by the U.S. Army Corps of Engineers, Lake Survey; June 1962 through November 1963), at eight shore stations on Lake Michigan-Huron, by a Tukey procedure (IBM Share Program 574) similar to that used by Munk, Snodgrass and Tucker (1959, "Spectra of low-frequency ocean waves," Bull. Scripps Inst. Oceanogr., 7: 283-362). For any given station pair, the analysis provided not only a spectrum of periodicities present at each station over a frequency range of 0 to 12 cycles/day divided into 300 frequency intervals of width 0.04 cycles/day, but also information — for each frequency interval — of coherence and phase differences between the two stations (10 station pairs were analyzed).

The second step was the development (by E.J.F.) of a new power spectrum program using the fast Fourier transform and its application to six months of half-hourly averages of water level (digitized from U.S. Lake Survey records, 5 October 1966 through 31 March 1967) at five U.S. stations on Lake Superior and hourly readings covering the same interval at three Canadian stations, as well as to five additional station pairs on Lake Michigan-Huron. We have confirmed that the new program, to be published in FORTRAN and ALGOL versions in the Center for Great Lakes Studies Special Report Series (No. 6), provides the same information (spectra, coherence and phase) as IBM 574, but cuts machine time very substantially and with no restrictions on the number of data points or lags.

Coherence and phase differences between stations provide powerful tools for unambiguous identification of the origins of prominent spectral peaks as forced and free modes of (longitudinal and transverse) oscillation.

## REQUEST FOR LITERATURE EXCHANGE

This report is being sent to you in the hope that literature exchange between you or your organization and this Center will be initiated and maintained. Reports or reprints which you may wish to designate for the Center Library, or inquiries concerning publications and exchanges, should be addressed to:

The Librarian
Center for Great Lakes Studies
University of Wisconsin-Milwaukee
Milwaukee, Wisconsin 53201 USA

Publications designated for individuals within the Center should be addressed to those persons by name.

## SPECIAL REPORT SERIES

Special Reports are issued from time to time, usually to provide accounts of work in progress, or completed, which are too detailed for acceptance by professional journals. This method of publication is, for example, convenient when it is desired to record and deposit collections of data in identifiable and recoverable form, or to fulfill the terms of contracts, or to make the details of a particular technique available to other potential users. This mode of reporting does not preclude later publication--usually abbreviated and selected--in professional journals; indeed, authors are urged to regard such publication as a necessary final step in the completion of a piece of research.

The following Special Reports have been issued so far:

Mortimer, C.H. 1968. Internal waves and associated currents observed in Lake Michigan during the summer of 1963. Special Report No. 1; 24 p., 120 figs.

Schenker, E. 1968. Effects of containerization on Great Lakes ports. Special Report No. 2; 45 p., 18 tables, 3 appendices.

Abstracts, Eleventh Conference on Great Lakes Research, April 1968. Special Report No. 3; 83 p., 91 abstracts

Fee, E. J. 1968. Digital computer programs for the Defant method of seiche analysis. Special Report No. 4; 27 p., 4 appendices, 3 figs.

Schenker, E. 1968. Future general cargo traffic and terminal requirements at the Port of Milwaukee. 13 p.

Fee, E. J. 1969. Digital computer programs for spectral analysis of time series. Special Report No. 6. 16 p., 3 appendices, 1 fig.