COMPUTER PROGRAMS FOR SPECTRAL AND TIME SERIES ANALYSES FOR RANDOM WAVES

by

DANIEL T. COX, NOBUHISA KOBAYASHI

and ANDOJO WURJANTO

Sponsored by
U.S. Army Coastal Engineering Research Center

RESEARCH REPORT NO. CACR-91-06
JULY, 1991

CENTER FOR APPLIED COASTAL RESEARCH
DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF DELAWARE
NEWARK, DELAWARE
19716

Abstract

Fourteen subroutines are presented herein for standard spectral and time series analyses since such subroutines may not be easily accessible to a user of the numerical model RBREAK reported previously. These subroutines have been used to specify numerically generated incident random waves as input to the numerical model RBREAK as well as to analyze and interpret the computed time series associated with random waves on impermeable coastal structures and beaches. These subroutines have also been used to conduct irregular wave tests in a wave flume for the evaluation and calibration of the numerical model RBREAK.

The mathematical background, computer program and example for each of the fourteen subroutines are presented in a user-friendly manner. The function of each subroutine is explained concisely to allow the selection of an appropriate subroutine for a specific spectral or time series analysis. The combined effective use of the numerical model RBREAK and appropriate subroutines from those included in this report is essential for predicting and interpreting random wave motions on coastal structures and beaches.

Acknowledgement

This is the third report resulting from research sponsored by the Coastal Engineering Research Center, U.S. Army Engineer Waterways Experiment Station under contract No. DACW 39-90-K-0006-P001. The authors would like to thank D.L. Ward, J.P. Ahrens and D.D. Davidson for their enthusiastic support of our continued efforts to improve the capabilities of numerical models for the design of coastal structures.

Contents

List of Figures

List of Tables

Part I: Introduction

Background

Wurjanto and Kobayashi (1991) presented a numerical model called RBREAK for random waves on impermeable coastal structures and beaches. Their report summarized the previous work related to RBREAK and described the detailed computational aspects of RBREAK. This report presents the subroutines used for the spectral and time series analyses for random waves used in the previous work by Kobayashi, Cox and Wurjanto (1990, 1991), Kobayashi, Wurjanto and Cox (1990a, 1990b), and Kobayashi and Wurjanto (1991).

The subroutines presented in this report are normally required to conduct irregular wave tests in a wave flume and analyze the measured time series as well as to specify numerically-generated incident random waves as input to RBREAK and analyze the computed time series. The methods of the spectral and time series analyzes used in these subroutines are standard and are explained in books for spectral methods (*e.g.* Bendat and Piersol, 1986) and for random waves (*e.g.* Goda, 1985). Consequently, similar subroutines may already be available to users of RBREAK. Nevertheless, well-documented, user-friendly subroutines for the spectral and time series analyses may not be accessible easily.

Fourteen subroutines are presented in this report. These subroutines are listed in Appendix A. The magnetic disk accompanying this report and containing the subroutines is explained in Appendix B. Parts II through XIV of this report describing the routines are each divided into three sections. The first section is the mathematical background for each subroutine and is explained to the degree that a user will be able to comprehend the content of each subroutine. The algebraic manipulations required to derive most of the equations used in this report are omitted herein. The derivation of these equations may not be straight forward but is presented in available books for spectral methods (*e.g.* Bendat and Piersol, 1986) and for random waves (*e.g.* Goda, 1985). In the second section, the computer program for each subroutine is included to show the usage of the subroutine by a main program or another subroutine. The input and output associated with each subroutine are explained thoroughly so that a user may be able to apply the subroutine without knowing every detail of the subroutine. In the third section, an example is presented for each subroutine so that a user can become familiar with the subroutine. The physical interpretations of the analyzed results were given in the previous works related to RBREAK.

Summary of Subroutines

The name and function of each of the fourteen subroutines is summarized concisely as follows:

- TMASPC: computes the TMA spectrum for wind waves in finite water depth as a function of frequency
- SPCPAR: computes standard spectral parameters for a specified spectrum
- TIMEPH: generates a time series for a specified spectrum using a random phase scheme where the generated time series depends on the seed value used to initialize the random number generator

- TIMEDC: generates the time series determined uniquely for specified Fourier components
- TIMPAR: computes standard parameters and ranked wave statistics based on a zero-upcrossing method for a specified time series
- SPCTRA: computes the unsmoothed and smoothed power density spectrum for a specified time series
- IRSORT: computes the incident and reflected wave time series from measured free surface oscillations at three locations in front of a reflective structure or beach
- COHPHS: computes the smoothed coherence squared and phase between two specified time series
- DISTNR: computes the probability distribution of the free surface elevation in comparison to the normal distribution as well as the exceedance probability of individual wave heights in comparison to the Rayleigh distribution
- USRSPC: accommodates the generation of time series from a power density spectrum whose shape is known but can not be expressed by a formula
- PRORBR: produces an input wave train for the numerical model RBREAK from the time series generated by either the TIMEPH or TIMEDC subroutine
- FFTIMSL: computes the complex Fourier coefficients for a specified time series using the IMSL subroutine FFT2D for a fast Fourier transform (FFT) as well as the time series for specified complex Fourier coefficients using the IMSL subroutine FFT2B for an inverse FFT
- RDMGEN: generates an array of pseudo-random numbers distributed uniformly between zero and one using the IMSL subroutines RNSET and RNUN
- WAVNUM: computes the wave number based on the linear wave dispersion relation for specified frequency and water depth where use is made of the gravitational acceleration $g = 9.81ms^{-2}$. It is noted that subroutines should have dimensions based on the SI units because of the use of the WAVNUM subroutine. Alternatively, a user may wish to modify this subroutine to use other units

The above subroutines are written for the spectral and time series analyses for the free surface oscillation. However, some of these subroutines can also be applied to other time-varying quantities such as the shoreline oscillation and velocities associated with random waves. The IMSL subroutines FFT2D, FFT2B, RNSET, and RNUN are used in this report since they are computationally efficient and widely used in the U.S.A. These subroutines can be replaced by other equivalent subroutines such as those included in the book of Press *et al.* (1986).

Part II: Subroutine TMASPC

Mathematical Background

A self-similar spectral shape given by Bouws *et al.* (1985) is used to define the sea state. The TMA spectral form (for TEXEL, MARSEN, and ARSLOE data sets) is an extension of the JONSWAP shape to finite water depth. The JONSWAP spectrum for wind waves in deep water is given by

$$S_J = S_P(f)\Phi_{PM}(f/f_p)\Phi_J(f, f_p, \gamma, \sigma) \tag{1}$$

where

$$S_P(f) = \alpha g^2(2\pi)^{-4}f^{-5} \tag{2}$$

$$\Phi_{PM}(f/f_p) = \exp\left[(-5/4)(f/f_p)^{-4}\right] \tag{3}$$

$$\Phi_J(f, f_p, \gamma, \sigma) = \exp\left\{\ln(\gamma)\exp\left[-(f - f_p)^2/2\sigma^2 {f_p}^2)\right]\right\} \tag{4}$$

$$\sigma = \begin{cases} \sigma_a & f_p \geq f \\ \sigma_b & f_p < f \end{cases} \tag{5}$$

where S_P is the Phillips formula for equilibrium range, α is a variable coefficient, g is the gravitational acceleration, f is the frequency, Φ_{PM} is the Pierson-Moskowitz shape function, f_p is the spectral peak frequency, Φ_J is the JONSWAP shape function, and σ and γ are variable coefficients.

In finite water depth, Bouws *et al.* (1985) assumed the validity of the JONSWAP spectrum expressed in terms of the wave number, k, and included the transformation factor, $\Phi_K(\omega_H)$, given explicitly by

$$\Phi_K(\omega_H) = \tanh^2(kh)\left[1 + \frac{2kh}{\sinh 2kh}\right]^{-1} \tag{6}$$

where h is the water depth, and kh can be found for given ω_H specified by

$$\omega_H = 2\pi f\left(\frac{h}{g}\right)^{1/2} \tag{7}$$

using the linear dispersion relation

$$(2\pi f)^2 = gk\tanh kh \tag{8}$$

Adopting the transformation factor initially introduced for shallow water by Kitaigorodskii *et al.* (1975), the TMA spectral shape is given by

$$S_{TMA} = S_J\Phi_K(\omega_H) \tag{9}$$

Computer Program

The TMASPC subroutine was written to yield S_{TMA} as a function of f for given spectral parameters for specifying the incident wave spectrum in finite water depth as well as for the generation of irregular waves in a flume. The subroutine is called by a main program or another subroutine

```
CALL TMASPC (NP, DT, FP, DH, IP, HR, AP, SP)
```

where the arguments are defined as

- IN:
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval, Δt (s)
  - FP = peak frequency of target spectrum, f_p (s^{-1})
  - DH = water depth, h (m)
  - IP = option to specify either root-mean-square wave height or spectral constant
- IN/OUT:
  - HR = root-mean-square wave height, H_{rms} (m)
  - AP = spectral constant, α
- OUT:
  - SP(NP/2+1) = TMA spectral array, S_{TMA} $(m^2 s)$
- EXTERNAL ROUTINES:
  - WAVNUM to return the wave number based on the linear dispersion relation

where the SI units of length, L, is in meters (m), and time, t, is in seconds (s).

Since the subroutines in this report were written for the case of analyzing discrete time series from the numerical model and from measurements in a flume, two arguments are contained in nearly all the subroutines. They are NP (N), the number of data point in the time series, and DT (Δt), the sampling interval of the time series. The sampling interval determines the largest frequency, that is, the Nyquist frequency, f_{Nyq}, where

$$f_{\mathrm{Nyq}} = \frac{1}{2\Delta t} \tag{10}$$

The Nyquist frequency is related in this and other subroutines to the length NH of the arrays in the frequency domain, where NH=NP/2+1. The frequency resolution DF (Δf) is related in these subroutines by DF=1/TM or

$$\Delta f = \frac{1}{t_{\mathrm{max}}} \tag{11}$$

where TM (t_{max}) is the duration of the time series and is given by TM=NP*DT or

$$t_{\mathrm{max}} = N\Delta t \tag{12}$$

The N-th element in the spectral array, SP(N), corresponds to the (N-1)*DF-th frequency where N=1,2,...,NH.

Specific to the TMASPC subroutine are the arguments FP, DH, IP, HR and AP. The user must specify the peak frequency FP of the TMA spectrum. The variable coefficients SIGA (σ_a), SIGB (σ_b) and GAMMA (γ) of the JONSWAP shape function are written in the subroutine with the standard values of SIGA=0.07, SIGB=0.09, and GAMMA=3.3.

The water depth DH (h) must be specified in meters since the WAVNUM subroutine was written with the gravitational constant g in SI units. The user may specify either the root-mean-square wave height HR ($H_{\rm rms}$) in which case IP=1, or the spectral constant AP (α) in which case IP=2. If $H_{\rm rms}$ is specified, then α is computed from the zeroth moment of the TMA spectrum with $\alpha = 1$

$$\alpha = \frac{H_{\rm rms}^2}{8m_0} \tag{13}$$

In general, the zeroth moment for the TMA spectrum is given by

$$m_0 = \int_0^\infty S_{TMA}(f)df \tag{14}$$

If α is specified, then $H_{\rm rms}$ is returned and computed by

$$H_{\rm rms} = \sqrt{8m_0} \tag{15}$$

The subroutine will return the TMA spectral array of length NP/2+1.

Example

The TMASPC subroutine can be used to generate irregular waves in a flume (*e.g.* Cox, 1989). In this example, TMASPC is called by a main program

```
      PARAMETER (NP=8192,IP=1)
      PARAMETER (DT=0.04,FP=0.6,DH=0.4,HR=0.06)
      DIMENSION SP(4097)
C Call TMASPC subroutine
      CALL TMASPC(NP,DT,FP,DH,IP,HR,AP,SP)
C Make a graph
      . . .
```

and returns the array SP and coefficient AP. The TMA spectral form is shown in Figure 1 with AP=0.0078. Although the Nyquist frequency is $12.5s^{-1}$ for this example, only the range $0.0 < f < 3.5s^{-1}$ is shown.

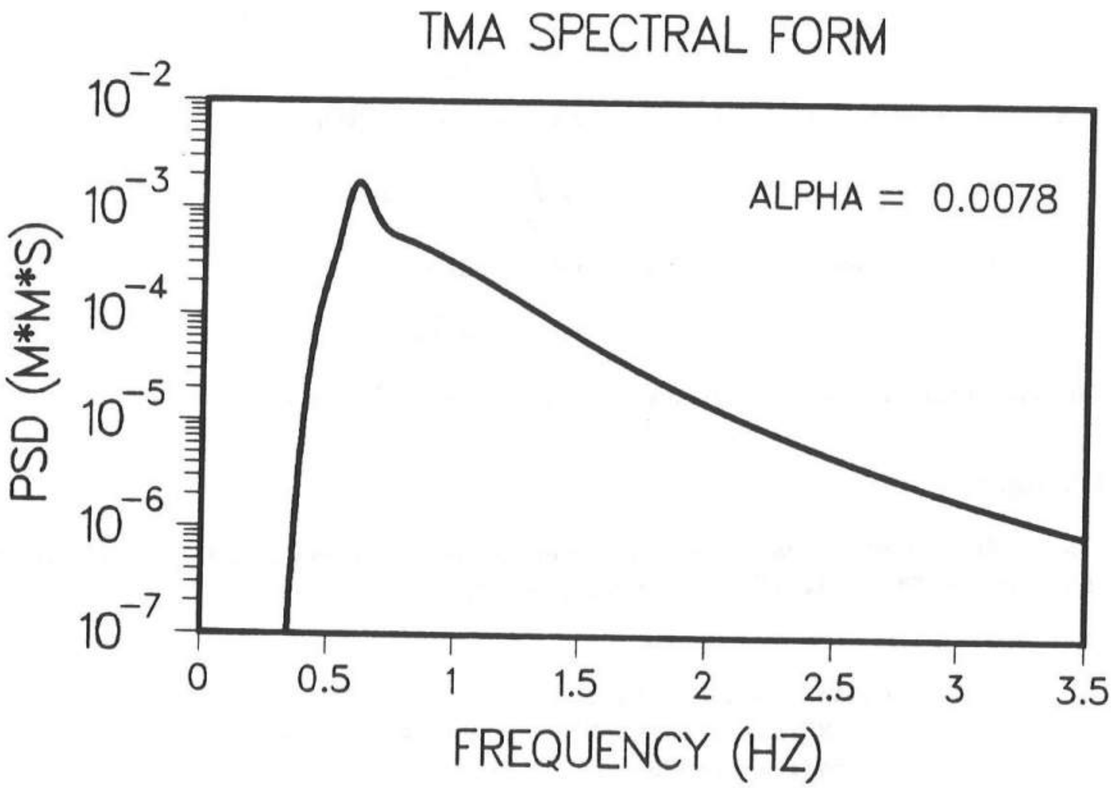

Figure 1: TMA Spectrum for Laboratory Experiment Returned by TMASPC.

Part III: Subroutine SPCPAR

Mathematical Background

The characteristics of the wave spectrum, $S(f)$, are typically described by various spectral parameters. Cartwright and Longuet-Higgins (1956) first defined the spectral width parameter, ϵ, as

$$\epsilon = \left[1 - \frac{m_2^2}{m_0 m_4}\right]^{1/2} \quad \text{for } 0 < \epsilon < 1 \tag{16}$$

with the n-th spectral moment defined as

$$m_n = \int_0^\infty f^n S(f) df \tag{17}$$

For narrow spectra, ϵ is near zero; and for broad spectra, ϵ is near unity. However, accurate computation of the fourth spectral moment for the high frequency part of the spectrum is difficult since $S(f) \propto f^{-5}$ in deep water. Instead, Longuet-Higgins (1957) defined a second spectral width parameter, ν, as

$$\nu = \left[\frac{m_0 m_2}{m_1^2} - 1\right]^{1/2} \tag{18}$$

On the other hand in 1970, Goda introduced the peakedness parameter, Q_p, as

$$Q_p = \frac{2}{m_0^2} \int_0^\infty f \left[S(f)\right]^2 df \tag{19}$$

where the value of Q_p is near 2 for wind waves (Goda, 1985).

The standard deviation of the free surface oscillation, $\eta_{\rm rms}$, is related to the zeroth moment as

$$\eta_{\rm rms} = \sqrt{m_0} = \left(\overline{\eta^2}\right)^{1/2} \tag{20}$$

where $\overline{\eta^2}$ is the mean of the square of the free surface oscillation with zero mean. Assuming the Rayleigh distribution of wave heights, the spectral estimate of the root-mean-square wave height, $H_{\rm rms}$, and the spectral estimate the significant wave height, $H_{\rm mo}$, are given by

$$H_{\rm rms} = \sqrt{8 m_0} \tag{21}$$

and

$$H_{\rm mo} = 4.004\sqrt{m_0} \tag{22}$$

In addition to the wave height parameters, the wave period parameters can be computed. A spectral estimate of the mean period of the zero-upcrossing waves, T_{02}, is given by

$$T_{02} = \sqrt{\frac{m_0}{m_2}} \tag{23}$$

A second spectral estimate of the mean period, T_{01}, is given by

$$T_{01} = \frac{m_0}{m_1} \tag{24}$$

It is noted that T_{02} and T_{01} may not be the same as the mean wave period calculated using a zero-crossing method with the time series of the free surface oscillation.

Computer Program

The SPCPAR subroutine, written to calculate the parameters for a given spectrum, is called by a main program or another subroutine

```
CALL SPCPAR (SP, NP, DT, EP, VU, QP, ER, HR, HM, T1, T2)
```

where the arguments are defined as

- IN:
  - SP(NP/2+1) = spectral array, $S(f)$ (L^2s)
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval, Δt (s)
- OUT:
  - EP = spectral width parameter, ϵ
  - VU = spectral width parameter, ν
  - QP = spectral peakedness parameter, Q_p
  - ER = standard deviation of the free surface oscillation, η_{rms} (L)
  - HR = spectral estimate of the root-mean-square wave height, H_{rms} (L)
  - HM = spectral estimate of the significant wave height, H_{mo} (L)
  - T1 = spectral estimate of the mean period, T_{01} (s)
  - T2 = spectral estimate of the mean period, T_{02} (s)
- EXTERNAL ROUTINES:
  - none

where L refers to the unit of length. In the subroutine the spectral moments are computed by trapezoidal approximation.

Example

The same TMA spectral form shown in Figure 1 is used in this example, where SPCPAR is called by a main program

```
      PARAMETER (NP=8192,IP=1)
      PARAMETER (DT=0.04,FP=0.6,DH=0.4,HR=0.06)
      DIMENSION SP(4097)
C Call TMASPC subroutine
      CALL TMASPC(NP,DT,FP,DH,IP,HR,AP,SP)
C Call SPCPAR subroutine
      CALL SPCPAR(SP,NP,DT,EP,VU,QP,ER,HRMS,HM,T1,T2)
C Make a table
      . . .
```

and the values of the spectral parameters are returned and are given in Table 1. It is noted that in the example program the $H_{\rm rms}$ argument is written HRMS to differentiate it from the parameter HR used as input to the TMASPC subroutine. As expected, the $H_{\rm rms}$ value returned by the SPCPAR subroutine agrees with the value specified to the TMASPC subroutine.

Table 1: Spectral Parameters Returned by SPCPAR.

Argument	Value	Units
ϵ	0.861	
ν	0.456	
Q_p	2.233	
$\eta_{\rm rms}$	2.121	cm
$H_{\rm rms}$	6.000	cm
$H_{\rm mo}$	8.492	cm
T_{01}	1.188	s
T_{02}	1.080	s

Part IV: Subroutine TIMEPH and TIMEDC

Mathematical Background

Two common methods for generating a time series for a given spectrum are discussed by Elgar *et al.* (1985), one of which is summarized here. This method is called the random phase scheme where the free surface oscillation is composed of the superposition of sinusoidal waves with random phase angles and with amplitudes based on the spectrum, $S(f_n)$. The free surface, $\eta(t)$, as a function of time is expressed by

$$\eta(t) = \sum_{n=1}^{N/2} C_n \cos(2\pi f_n t + \phi_n) \quad \text{for} \quad 0 \leq t < t_{\max} \tag{25}$$

with

$$C_n = [2S(f_n)\Delta f]^{1/2} \tag{26}$$

and

$$\Delta f = \frac{1}{t_{\max}} = \frac{1}{N\Delta t} \tag{27}$$

where N is the even number of data points in the time series, Δt is the sampling interval, $t_{\max}$ is the duration of the time series, C_n are the real Fourier amplitudes, Δf is the frequency bandwidth, $f_n = n\Delta f$ is the frequency, and ϕ_n are the random phase angles uniformly distributed in $[0, 2\pi]$.

This method represents a Gaussian sea only as the number of harmonics, $N/2$, approaches infinity (Tucker *et al.*, 1984). For finite N and Δf = constant, the free surface profile repeats after $t = t_{\max}$. However, the duration of $0 \leq t < t_{\max}$ is of interest, and the use of Δf = constant is normally sufficient for small Δf.

A second routine is presented in this section to return the time series for determined Fourier coefficients and is based on the following equation

$$\eta(t) = \sum_{n=1}^{N/2} \Big[a_n \cos(2\pi f_n t) + b_n \sin(2\pi f_n t)\Big] \qquad 0 \leq t < t_{\max} \tag{28}$$

where a_n and b_n are the Fourier coefficients. Comparing Equations 25 and 28, a_n and b_n can be expressed

$$a_n = C_n \cos\phi_n \tag{29}$$

and

$$b_n = -C_n \sin\phi_n \tag{30}$$

This subroutine is essentially an inverse FFT routine but was written as a separate routine for clarity. This subroutine is used by the IRSORT subroutine to return incident and reflected wave trains.

Computer Program

The TIMEPH subroutine, written for calculating a time series for a given spectrum, is called by a main program or another subroutine

```
CALL TIMEPH (SP, NP, DT, IS, TS)
```

where the arguments are defined as

- IN:
  - `SP(NP/2+1)` = power density spectrum, $S(f_n)$ $(L^2 s)$
  - `NP` = even number of data points in the time series, N
  - `DT` = time step or sampling interval, Δt (s)
  - `IS` = seed value to initialize the random number generator
- OUT:
  - `TS(NP)` = time series, $\eta(t)$ (L)
- EXTERNAL ROUTINES:
  - `FFTIMSL` to inverse Fourier transform the coefficients and return the time series
  - `RDMGEN` to return an array of random numbers uniformly distributed between zero and one

Only these two subroutines, `FFTIMSL` and `RDMGEN`, contain calls to the IMSL library of subroutines and can be easily substituted by standard FFT routines and random number generators for computers without this library (Press *et. al.*, 1986). However, the IMSL version of the FFT is computationally efficient and is used in these routines.

The second routine, `TIMEDC`, which is essentially an inverse Fourier transform of known Fourier coefficients to return the time series, is called by a main program

```
CALL TIMEDC (A, B, NP, TS)
```

where the arguments are defined as

- IN:
  - `A(NP/2+1)` = real part of the complex Fourier coefficients, a_n (L)
  - `B(NP/2+1)` = imaginary part of the complex Fourier coefficients, b_n (L)
  - `NP` = even number of data points in the time series
- OUT:
  - `TS(NP)` = time series, $\eta(t)$ (L)
- EXTERNAL ROUTINES:
  - `FFTIMSL` to inverse Fourier transform the coefficients and return the time series

The relation between the complex Fourier coefficients, c_n, used in the **FFTIMSL** subroutine and the real Fourier coefficients, a_n and b_n, is given by

$$c_n = 0 \qquad \text{for } n = 0 \tag{31}$$

$$c_n = \frac{a_n - ib_n}{2} \qquad \text{for } n = 1, 2, \ldots, N/2 - 1 \tag{32}$$

$$c_n = a_n \qquad \text{for } n = N/2 \tag{33}$$

$$c_n = \frac{a_{N-n} + ib_{N-n}}{2} \qquad \text{for } n = N/2 + 1, \ldots, N - 1 \tag{34}$$

where i^2=−1 and N=$(n+1)$ is used in the computer program listed in Appendix A.

Example

The example in this section illustrates the use of the **TIMEPH** subroutine by way of the TMA spectrum generated in the previous two examples. The **TIMEPH** subroutine is called by a main program

```
          PARAMETER (NP=8192,IP=1,IS=123457)
          PARAMETER (DT=0.04,FP=0.6,DH=0.4,HR=0.06)
          DIMENSION SP(4097),TS(8192)
     C To provide IMSL workspace
          COMMON /WORKSP/ RWKSP
          REAL RWKSP(65592)
          CALL IWKIN(65592)
     C Call TMASPC subroutine
          CALL TMASPC(NP,DT,FP,DH,IP,HR,AP,SP)
     C Call TIMEPH subroutine
          CALL TIMEPH(SP,NP,DT,IS,TS)
     C Make a graph
          . . .
```

and the time series is returned. It is noted that in order to run the FFT routines provided in the IMSL library, it is necessary to provide adequate workspace in the main (calling) program.

Figure 2 shows the time series of the TMA spectrum. Although the maximum duration is $t_{\max} = 327.68s$, only the range $0.0 < t < 160.0s$ is shown in this figure. An example of the TIMEDC subroutine is not provided here but is discussed in conjunction with the IRSORT subroutine in Part VII.

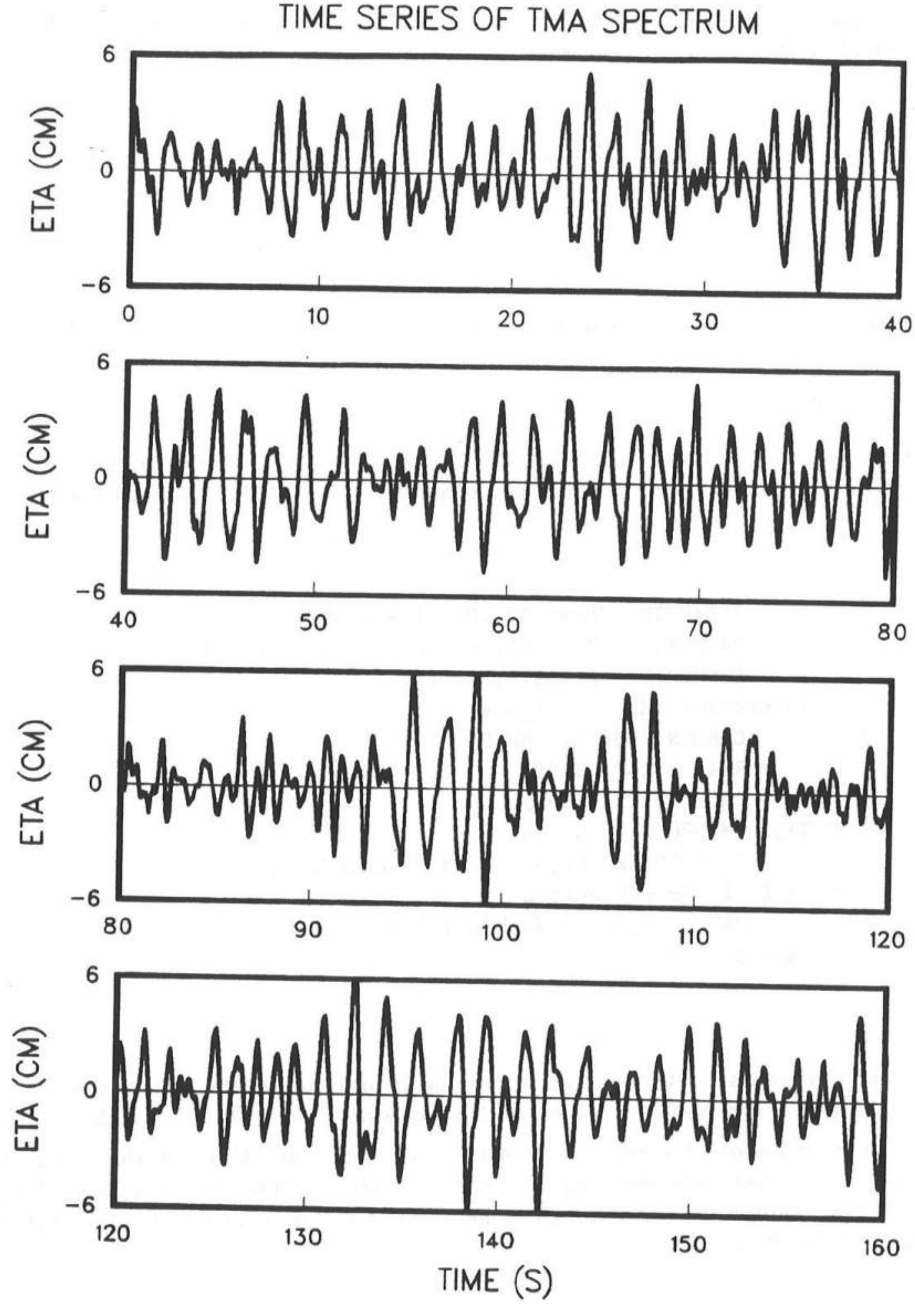

Figure 2: Time Series of TMA Spectrum for Laboratory Experiment Returned by TIMEPH.

Part V: Subroutine TIMPAR

Mathematical Background

There are two general categories for random wave analyses: one is the spectral method, and the other is the zero-crossing method. These two methods are completely different in approach although gross statistics such as the root-mean-square wave height tend to agree for time series of long duration. Individual waves can be identified using either successive zero-upcrossing points or successive zero-downcrossing points where a "zero-crossing" is the location where the time series crosses the zero of the abscissa. The zero-upcrossing method is adopted in this report.

Beginning the zero-upcrossing analysis, the mean water level due to wave setup or setdown, $\overline{\eta}$, is computed by using the arithmetic mean of the time series, η_i, measured from still water level,

$$\overline{\eta(t)} = \frac{1}{N}\sum_{i=1}^{N}\eta_i \tag{35}$$

where N is the number of data points. Next, the adjusted free surface oscillation is computed by subtracting the setup; and the adjusted free surface, $(\eta_i - \overline{\eta})$, is denoted hereafter as η_i for brevity. The root-mean-square of the free surface oscillation, η_{rms}, is found by

$$\eta_{\mathrm{rms}}^2 = \frac{1}{N}\sum_{i=1}^{N}\eta_i^2 \tag{36}$$

Additional parameters are calculated using the zero-upcrossing method in which the point of zero-upcrossing is located using the linear interpolation between two points satisfying the conditions

$$\eta_i \cdot \eta_{i+1} < 0 \text{ and } \eta_{i+1} > 0 \tag{37}$$

where η_i is the i-th data point of the free surface elevation. An individual wave is defined using two adjacent zero-upcrossing points. The wave period of an individual wave, T_i, is the duration between the two adjacent zero-upcrossing points. The largest value of the data points in an individual wave can be found by comparing the values of η_i included in each wave. A parabolic curve is fitted to the three discrete data points about the largest value, η_i, to improve the estimate of the maximum elevation, η_{max}, which is given by

$$\eta_{\mathrm{max}} = C - \frac{B^2}{4A} \tag{38}$$

where

$$A = \frac{1}{2}(\eta_{i-1} - 2\eta_i + \eta_{i+1}) \tag{39}$$

$$B = \frac{1}{2}(\eta_{i+1} - \eta_{i-1}) \tag{40}$$

$$C = \eta_i \tag{41}$$

The minimum elevation of the individual wave, η_{min}, can be found in a similar manner. The corresponding wave height of an individual wave, H_i, is given by

$$H_i = \eta_{\mathrm{max}} - \eta_{\mathrm{min}} \tag{42}$$

The average wave height, $\overline{H}$, and the average wave period, $\overline{T}$, can be found by arithmetic mean

$$\overline{H} = \frac{1}{N_0}\sum_{i=1}^{N_0} H_i \tag{43}$$

and

$$\overline{T} = \frac{1}{N_0}\sum_{i=1}^{N_0} T_i \tag{44}$$

where N_0 is the number of individual waves with wave period and wave height denoted by T_i and H_i, respectively. The root-mean-square wave height, $H_{\rm rms}$, is defined as

$$H_{\rm rms} = \left[\frac{1}{N_0}\sum_{i=1}^{N_0} H_i^2\right]^{1/2} \tag{45}$$

To determine the significant wave height, H_s, and the significant wave period, T_s, the individual waves are ranked in descending order of H_i. The significant wave height, H_s, is defined as the arithmetic mean of the one-third highest waves

$$H_s = \frac{3}{N_0}\sum_{r=1}^{N_0/3} H_r \tag{46}$$

where H_r is the wave height of r-th rank. Similarly, the significant wave period, T_s, is given by

$$T_s = \frac{3}{N_0}\sum_{r=1}^{N_0/3} T_r \tag{47}$$

where T_r is the period corresponding to the r-th ranked wave. Additionally, the average height and period of the one-tenth highest waves, H_{10} and T_{10}, respectively, are given by

$$H_{10} = \frac{10}{N_0}\sum_{r=1}^{N_0/10} H_r \tag{48}$$

and

$$T_{10} = \frac{10}{N_0}\sum_{r=1}^{N_0/10} T_r \tag{49}$$

Lastly, the run length of wave groups are computed. The run length is equal to the number of waves in a sequence for which the wave heights are larger than a specified wave height (Goda, 1985) which is taken to be the significant wave height, H_s, in this routine.

Computer Program

The TIMPAR subroutine, written to compute the statistical parameters of given time series, is called by a main program or another subroutine

```
CALL TIMPAR (TS, NP, DT, SD, ER, NZ, HB, TB, HV, HS, T3,
             HT, TT, HRK, TRK, LRN, NK)
```

where the arguments are defined as

- IN:
  - TS(NP) = time series, $\eta(t)$ (L)
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval, Δt (s)
- OUT:
  - SD = mean of time series (setup or setdown), $\overline{\eta(t)}$ (L)
  - ER = root-mean-square of the free surface elevation, η_{rms} (L)
  - NZ = number of zero-upcrossing waves, N_0
  - HB = mean wave height, $\overline{H}$ (L)
  - TB = mean wave period, $\overline{T}$ (s)
  - HV = root-mean-square wave height, H_{rms} (L)
  - HS = significant wave height, *i.e.*, the average of the one-third highest waves, H_s (L)
  - T3 = significant wave period, *i.e.*, the average period of the one-third highest waves, T_s (s)
  - HT = average height of the one-tenth highest waves, H_{10} (L)
  - TT = average period of the one-tenth highest waves, T_{10} (s)
  - HRK(NZ) = array of wave heights ranked with HRK(1) being the highest, H_r (L)
  - TRK(NZ) = array of wave periods corresponding to the ranked wave heights with TRK(1) being the period corresponding to the highest wave, T_r (s)
  - LRN(NK) = run length of wave heights exceeding HS
  - NK = number of runs
- EXTERNAL ROUTINES:
  - none

In the TIMPAR subroutine, the time series array is copied, ATS(I)=TS(I), and an additional point is added, ATS(NP+1)=TS(1), to make the time series periodic. The mean is removed from the copied time series, ATS(I), to leave the original unchanged.

It is noted that the number of zero-upcrossing waves is NZ (N_0) and that the last wave period T(NZ) ($T_{i=N_0}$) is computed by T(NZ)=(TM-TMZERO(NZ))+TMZERO(1) where TM ($t_{\max}$) is the duration of the time series, TMZERO(NZ) is the last zero-upcrossing point near the end of the time series and TMZERO(1) is the first zero-upcrossing point near the beginning of the time series.

Since the wave heights are generally of greater importance than the wave periods to the design engineer, the waves are ranked by height and not by period in this subroutine although this condition is easy to change in the subroutine. The wave period array returned contains elements corresponding to the wave periods of the ranked waves, and the highest waves do not generally have the longest periods (Goda, 1985).

Example

The example in this section uses the total time series generated by the TIMEPH subroutine, part of which is shown in Figure 2. The TIMEPH subroutine is called by a main program

```
      PARAMETER (NP=8192,IP=1,IS=123457)
      PARAMETER (DT=0.04,FP=0.6,DH=0.4,HR=0.06)
      DIMENSION SP(4097), TS(8192), HRK(1000), TRK(1000)
C To provide IMSL workspace
      COMMON /WORKSP/ RWKSP
      REAL RWKSP(65592)
      CALL IWKIN(65592)
C Call TMASPC subroutine
      CALL TMASPC(NP,DT,FP,DH,IP,HR,AP,SP)
C Call TIMEPH subroutine
      CALL TIMEPH(SP,NP,DT,IS,TS)
C Call TIMPAR subroutine
      CALL TIMPAR(TS,NP,DT,DS,ER,NZ,HB,TB,HV,HS,T3,HT,TT,
     &   HRK,TRK,LRN,NK)
C Make a table
      . . .
```

and the subroutine returns the values of the time series parameters given in Table 2. For this example, $H_{\text{rms}} = 5.666cm$ which is roughly equivalent to $H_{\text{rms}} = 6.0cm$ specified to the TMASPC subroutine to generate the TMA spectrum. The value of H_{rms} returned by TIMPAR should approach the specified H_{rms} of the target spectrum as the length of the time series increases. The mean wave period, $\overline{T} = 1.050s$, is roughly equivalent to the mean period based on the spectral moments of the SPCPAR subroutine, $T_{02} = 1.080s$, as shown in Table 1. A partial ranking of the waves is given in Table 3, and the run lengths are given in Table 4.

Table 2: Time Series Parameters Returned by TIMPAR.

Argument	Value	Units
$\overline{\eta(t)}$	-2.57×10^{-4}	cm
$\eta_{\rm rms}$	2.121	cm
N_0	312	
$\overline{H}$	4.946	cm
$\overline{T}$	1.050	s
$H_{\rm rms}$	5.666	cm
H_s	8.116	cm
T_s	1.367	s
H_{10}	10.345	cm
T_{10}	1.394	s

Table 3: Wave Height Rankings with Corresponding Wave Periods Returned by TIMPAR.

Rank, r	H_r (cm)	T_r (s)
1	14.165	1.370
2	13.359	1.195
3	12.841	1.163
4	12.671	1.647
5	12.285	1.472
306	0.450	0.369
307	0.364	0.199
308	0.263	0.201
309	0.236	0.142
310	0.104	0.203
311	0.049	0.060

Table 4: Run Lengths Returned by **TIMPAR**.

I	LRN(I)	I	LRN(I)
1	1	14	1
2	3	15	1
3	1	16	2
4	2	17	4
5	1	18	1
6	1	19	1
7	1	20	4
8	1	21	1
9	3	22	1
10	2	23	1
11	1	24	2
12	1	25	1
13	2	26	1

Part VI: Subroutine SPCTRA

Mathematical Background

The power density spectrum, $S(f_n)$, is found by Fourier transform of the time series, $\eta(t)$, expressed in the form of Equation 28 and is given by

$$S(f_n) = 0 \qquad \text{for } n = 0 \tag{50}$$

$$S(f_n) = \frac{1}{2\Delta f}(a_n^2 + b_n^2) \qquad \text{for } n = 1, \ldots, \frac{N}{2} - 1 \tag{51}$$

$$S(f_n) = \frac{1}{2\Delta f} a_n^2 \qquad \text{for } n = N/2 \tag{52}$$

where a_n and b_n are the Fourier coefficients at each frequency, $f_n = n\Delta f$. It was noted in Part II that the frequency resolution, Δf, is related to the duration of the time series, $t_{\max}$, by

$$\Delta f = \frac{1}{t_{\max}} = \frac{1}{N\Delta t} \tag{53}$$

The largest frequency, that is, the Nyquist frequency, f_{Nyq}, depends only on the sampling interval, Δt, and is given by

$$f_{\mathrm{Nyq}} = \frac{1}{2\Delta t} = \frac{N\Delta f}{2} \tag{54}$$

In practice, the spectrum obtained from the Fourier transform may have a high statistical resolution but may have a low statistical reliability, particularly for time series with much noise. Hence, it may be advantageous to use a smoothing procedure to increase the reliability at the expense of the spectral resolution. The simplest method is a rectangular filter with the smoothed spectrum, $\hat{S}(f_k)$, given by

$$\hat{S}(f_k) = \frac{1}{m} \sum_{j=(k-1)m+1}^{km} S(f_j) \tag{55}$$

with

$$f_k = \left[\frac{1}{2} + (k - \frac{1}{2})m\right]\Delta f \tag{56}$$

where m indicates the number of unsmoothed spectral values used for the averaging. The corresponding degree of freedom is $2m$ (*e.g.* Goda, 1985).

It is noted that the spectrum is computed without regard to aliasing or spectral leakage. These two points should be considered when the sampling interval, Δt, and the number of samples, N, are determined for an experiment. The choice of the sampling interval alone determines the Nyquist frequency. Energy contained in the record beyond the Nyquist frequency is folded back in to the frequency band below the Nyquist; therefore, the sampling interval should be chosen so that there is negligible energy above the Nyquist frequency. The frequency resolution, Δf, determined by the duration of the record should be chosen so that there is negligible energy below Δf.

Computer Program

The SPCTRA subroutine, written to compute the power density spectrum for a given time series, is called by a main program

```
CALL SPCTRA (TS, NP, DT, NB, SP, FS, SM)
```

where the arguments are defined as

- IN:
  - TS(NP) = time series to be transformed, $\eta(t)$ (L)
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval, Δt (s)
  - NB = number of band-averaged data points for smoothing, m
- OUT:
  - SP(NP/2+1) = unsmoothed power density spectrum, $S(f_n)$ $(L^2 s)$
  - FS(NP/2/NB) = frequency array of smoothed spectrum, f_k (s^{-1})
  - SM(NP/2/NB) = smoothed spectrum, $\hat{S}(f_k)$ $(L^2 s)$
- EXTERNAL ROUTINES:
  - FFTIMSL to return the complex Fourier coefficients

Example

The example for this section starts with the time series data file CM06G1. The time series for CM06G1 is the measured free surface oscillation in a flume at the toe of a 1:20 smooth, impermeable slope with a water depth of $h = 0.47m$, and with a peak frequency of the target spectrum of $f_p = 0.6s^{-1}$. The length of this file is $N = 8192$ points and the sampling interval is $\Delta t = 0.04s$. The free surface displacement, $\eta(t)$, is in units of centimeters. The SPCTRA subroutine is called by a main program

```
      PARAMETER (NB=16)
      DIMENSION SP(4097), TS(16384), FS(256), SM(256)
C To provide IMSL workspace
      COMMON /WORKSP/ RWKSP
      REAL RWKSP(65592)
      CALL IWKIN(65592)
C Read in time series data file CM06G1
      OPEN(UNIT=11,FILE='CM06G1')
      READ(11,1) NP, DT
1     FORMAT(I10,F10.4)
      READ(11,*) (TS(I), I=1,NP)
      CLOSE(11)
C Remove mean from time series
      CALL TAKEMN(TS,NP,SD)
C Call SPCTRA subroutine
      CALL SPCTRA(TS,NP,DT,NB,SP,FS,SM)
```

```
C Make a graph
        . . .
C Subroutine to remove the mean from the time series
      SUBROUTINE TAKEMN(TS,NP,SD)
      REAL TS(NP)
      SUM=0.0
      DO 1 I = 1, NP
         SUM = SUM + TS(I)
1     CONTINUE
      SD = SUM/FLOAT(NP)
      DO 2 I = 1, NP
         TS(I) = TS(I) - SD
2     CONTINUE
      RETURN
      END
```

and the subroutine returns for the given time series the unsmoothed power spectral density as well as the smoothed power spectral density and corresponding frequency array. The unsmoothed spectrum has a frequency resolution of $\Delta f = 0.00305 s^{-1}$ and Nyquist frequency of $f_{\mathrm{Nyq}} = 12.5 s^{-1}$. Figure 3 shows the unsmoothed spectrum in the range $0.0 < f < 3.5 s^{-1}$ for the time series CM06G1. For the smoothed spectrum, the frequency resolution is $\Delta f = 0.0488 s^{-1}$ and the Nyquist frequency is essentially unchanged. Figure 4 shows the smoothed power spectrum for the data file CM06G1 with 32 degrees of freedom ($m = 16$).

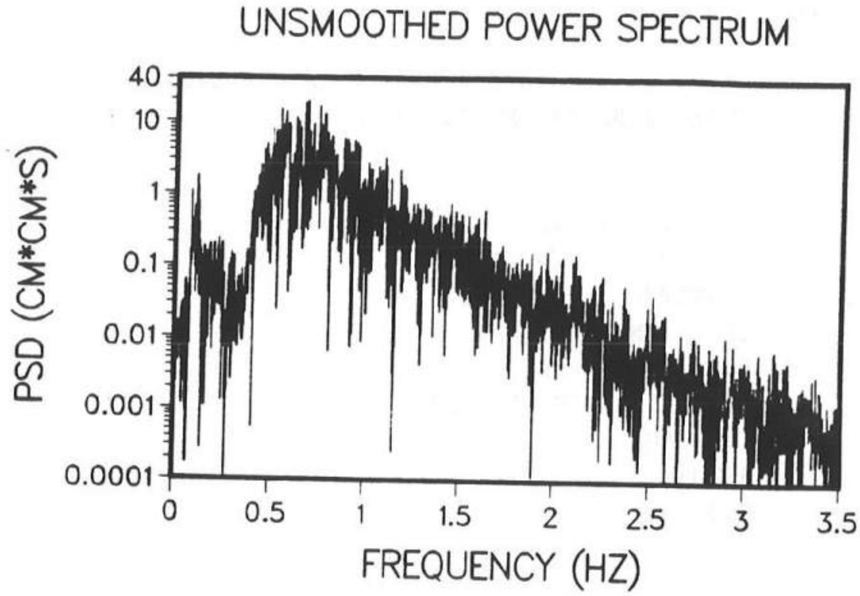

Figure 3: Unsmoothed Power Spectrum for Time Series CM06G1 Returned by SPCTRA.

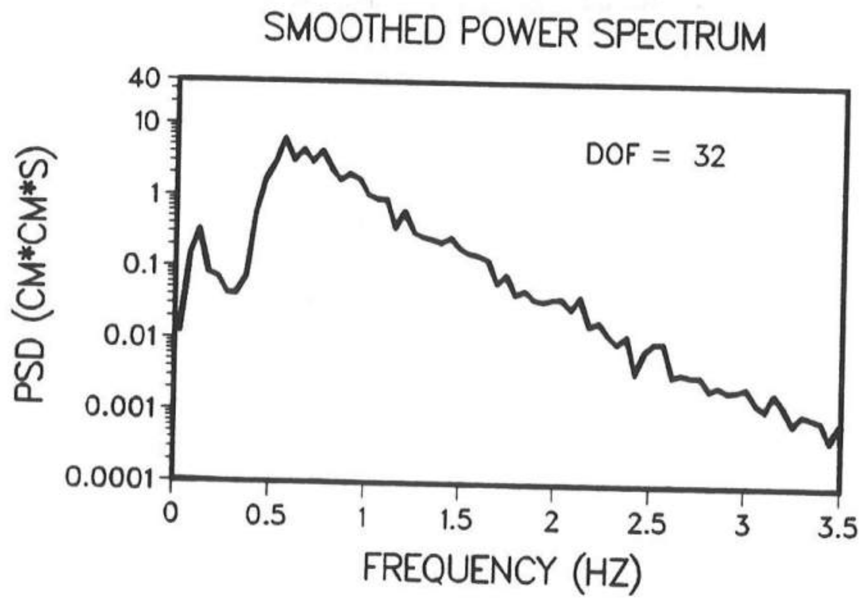

Figure 4: Smoothed Power Spectrum with 32 Degrees of Freedom for Time Series CM06G1 Returned by SPCTRA.

Part VII: Subroutine IRSORT

Mathematical Background

Since laboratory experiments are often affected by multireflection from the slope and wavemaker, it is important to know the reflective properties of the slope for any hydraulic test in a flume. In the case of irregular waves, it is desirable to estimate the reflection as a function of frequency and to be able to separate incident and reflected waves. A method to separate incident and reflected waves with an array of wave gages is presented here following Thornton and Calhoun (1972), Goda and Suzuki (1976), and Seelig (1980).

The mean water level is first removed from the time series, and the free surface displacement, η^i, is given by

$$\eta^i(t) = \eta^i(t)' - \overline{\eta^i(t)'} \quad \text{for } 0 \le t \le t_{\max} \tag{57}$$

where i is the gage number, $\eta^i(t)'$ is the time series before removal, and $\overline{\eta^i(t)'}$ is the mean water level. Assuming a horizontal seabed seaward of the slope in the region $x \le 0$ where the horizontal coordinate x is taken to be positive landward, the incident and reflected time series are assumed to be expressed as

$$\eta_i(x,t) = \sum_{n=1}^{N/2} \Big[(a_i)_n \cos(k_n x - \omega_n t) + (b_i)_n \sin(k_n x - \omega_n t) \Big] \quad \text{for } x \le 0 \tag{58}$$

and

$$\eta_r(x,t) = \sum_{n=1}^{N/2} \Big[(a_r)_n \cos(k_n x + \omega_n t) + (b_r)_n \sin(k_n x + \omega_n t) \Big] \quad \text{for } x \le 0 \tag{59}$$

in which $(a_i)_n$, $(b_i)_n$, $(a_r)_n$, and $(b_r)_n$ with $n = 1, 2, \ldots, N/2$ are the unknown coefficients for the wave trains of length N, where $\eta_i(x,t)$ and $\eta_r(x,t)$ are the incident and reflected wave trains, respectively. The total free surface variation seaward of the toe of the slope is given by

$$\eta(x,t) = \eta_i(x,t) + \eta_r(x,t) \quad \text{for } x \le 0 \tag{60}$$

which can be written in expanded form as

$$\begin{aligned} \eta(x,t) = \sum_{n=1}^{N/2} & \left\{ \Big[(a_i)_n + (a_r)_n \Big] \cos(k_n x) + \Big[(b_i)_n + (b_r)_n \Big] \sin(k_n x) \right\} \cos \omega_n t \\ & + \left\{ \Big[(b_r)_n - (b_i)_n \Big] \cos(k_n x) + \Big[(a_i)_n - (a_r)_n \Big] \sin(k_n x) \right\} \sin \omega_n t \end{aligned} \tag{61}$$

On the other hand, the free surface oscillations are known at each gage $x = x_i$, where i is the gage number, and is expressed as

$$\eta(x_i,t) = \sum_{n=1}^{N/2} \Big[a_n^i \cos(\omega_n t) + b_n^i \sin(\omega_n t) \Big] \quad \text{for } 0 \le t \le t_{\max} \tag{62}$$

where a_n^i and b_n^i are the Fourier coefficients computed for the known time series. Comparing Equations 61 and 62, the following equations must be satisfied

$$a_n^i = \Big[(a_i)_n + (a_r)_n \Big] \cos k_n x_i + \Big[(b_i)_n + (b_r)_n \Big] \sin k_n x_i \tag{63}$$

and

$$b_n^i = \Big[(b_r)_n - (b_i)_n\Big] \cos k_n x_i + \Big[(a_i)_n - (a_r)_n\Big] \sin k_n x_i \tag{64}$$

where $n = 1, 2, \ldots, N/2$ indicates each harmonic and i indicates the location of the first gage of the pair. The location of the second gage, j, with $x_i > x_j$ and $j > i$ gives

$$a_n^j = \Big[(a_i)_n + (a_r)_n\Big] \cos k_n x_j + \Big[(b_i)_n + (b_r)_n\Big] \sin k_n x_j \tag{65}$$

and

$$b_n^j = \Big[(b_r)_n - (b_i)_n\Big] \cos k_n x_j + \Big[(a_i)_n - (a_r)_n\Big] \sin k_n x_j \tag{66}$$

Using these four equations, the unknown coefficients, $(a_i)_n$, $(b_i)_n$, $(a_r)_n$, and $(b_r)_n$, are solved in terms of the known Fourier coefficients, a_n^i, b_n^i, a_n^j, and b_n^j, and the gage positions, x_i and x_j.

The unknown coefficients are given by

$$(a_i)_n = \frac{1}{2 \sin k_n(x_i - x_j)} \Big[-a_n^i \sin k_n x_j + a_n^j \sin k_n x_i + b_n^i \cos k_n x_j - b_n^j \cos k_n x_i\Big] \tag{67}$$

$$(b_i)_n = \frac{1}{2 \sin k_n(x_i - x_j)} \Big[+a_n^i \cos k_n x_j - a_n^j \cos k_n x_i + b_n^i \sin k_n x_j - b_n^j \sin k_n x_i\Big] \tag{68}$$

$$(a_r)_n = \frac{1}{2 \sin k_n(x_i - x_j)} \Big[-a_n^i \sin k_n x_j + a_n^j \sin k_n x_i - b_n^i \cos k_n x_j + b_n^j \cos k_n x_i\Big] \tag{69}$$

$$(b_r)_n = \frac{1}{2 \sin k_n(x_i - x_j)} \Big[+a_n^i \cos k_n x_j - a_n^j \cos k_n x_i - b_n^i \sin k_n x_j + b_n^j \sin k_n x_i\Big] \tag{70}$$

where k_n is the wave number calculated at each frequency using the linear dispersion relation

$$(2\pi f_n)^2 = g k_n \tanh k_n h \tag{71}$$

where h is the water depth. An inverse transform of the Fourier coefficients gives the incident time series, $\eta_i(x,t)$, and reflected time series, $\eta_r(x,t)$, at $x = 0$, the position of the first gage seaward of the toe of the slope.

A limitation of this method is the singularity of $1/\sin k_n(x_i - x_j)$. Goda (1985) and Goda and Suzuki (1976) recommend that the effective frequency range of resolution should be limited to

$$\frac{\pi}{10} \leq k_n(x_i - x_j) \leq \frac{9\pi}{10} \tag{72}$$

For an array of three gages, there are three gage pairs and, therefore, three estimates. These estimates are averaged in the case that two or three estimates are within the cutoff criteria. Estimates outside the cutoff range are not used. With proper choice of gage spacing, a wide frequency band can be resolved; however, the lowest resolvable frequency is limited by the largest gage spacing. Frequencies below this limit are not resolvable. Additionally, in the higher frequencies, there will be frequency bands where all three estimates are outside the cutoff criteria, and no estimate is possible. Appropriate gage spacing for given water depth and peak wave period should be chosen such that most of the energy is contained in the effective frequency range of resolution.

Computer Program

The IRSORT subroutine, written to separate incident and reflected waves for a three gage array, is called by a main program or another subroutine

```
CALL IRSORT (TS, ND, NW, NP, DT, XG, DH, FMN, FMX, TI, TR)
```

where the arguments are defined as

- IN:
  - TS(ND,NW) = free surface oscillations at NW wave gages, $\eta^i(t)$ (L)
  - ND = dimension of TS in calling program equal to NP or greater
  - NW = width of TS in calling program (equal to number of gages)
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval, Δt (s)
  - XG(NW) = location of each gage with the x-axis positive shoreward and gage number decreasing shoreward, x (m)
  - DH = water depth, h (m)
- OUT:
  - FMN = minimum resolvable frequency based on largest gage spacing, $f_{\min}$ (s^{-1})
  - FMX = maximum resolvable frequency based on smallest gage spacing, $f_{\max}$ (s^{-1})
  - TI(NP) = incident time series at $x = 0$, $\eta_i(t)$ (L)
  - TR(NP) = reflected time series at $x = 0$, $\eta_r(t)$ (L)
- EXTERNAL ROUTINES:
  - FFTIMSL to return the Fourier coefficients for the time series at each gage location
  - WAVNUM to return the wave number based on the linear dispersion relation at each frequency
  - TIMEDC to return the time series for known Fourier coefficients

The minimum and maximum resolvable frequencies are limited by the maximum and minimum gage spacings, respectively, following the restriction suggested by Goda (1985) given by

$$\frac{\pi}{10} \leq k\Delta x \leq \frac{9\pi}{10} \qquad (73)$$

where Δx is the gage spacing. The minimum resolvable frequency, $f_{\min}$, is given by the linear dispersion relation

$$(2\pi f_{\min})^2 = gk_{\min} \tanh(k_{\min} h) \tag{74}$$

where h is the water depth and $k_{\min}$ is the minimum wave number given by

$$k_{\min} = \frac{\pi}{10\Delta x_{\max}} \tag{75}$$

where $\Delta x_{\max}$ is the maximum gage spacing. Similarly, the maximum resolvable frequency, $f_{\max}$, is limited by the minimum gage spacing, $\Delta x_{\min}$, and is found by

$$(2\pi f_{\max})^2 = gk_{\max} \tanh(k_{\max} h) \tag{76}$$

where

$$k_{\max} = \frac{9\pi}{10\Delta x_{\min}} \tag{77}$$

Additionally, it is noted that this program can be used for a two gage array provided that the arrays in the main program are dimensioned correctly. Also, although it is necessary that the gage positions and the water depth be specified in meters, the free surface elevation can be specified with arbitrary units of length.

Example

The example for this section starts with three time series data files: CM06G1, CM06G2, and CM06G3. These time series data files are the measured total free surface oscillations for a three gage array where CM06G1 is the first gage at $x = 0m$ (see example, Part VI), CM06G2 is the second gage at $x = -1.4m$, and CM06G3 is the third gage at $x = -2.0m$. After reading in the data files, the main program calls the IRSORT subroutine. Within the IRSORT subroutine, three other subroutines are called. The FFTIMSL subroutine returns the Fourier coefficients for the time series at each gage location. The WAVNUM subroutine returns the wave number based on the linear dispersion relation at each frequency. It should be noted that this subroutine is written in SI units and that the water depth, DH, and gage locations, XG, should be specified in meters (m). The TIMEDC subroutine returns the incident and reflected wave time series for the computed Fourier coefficients. After IRSORT returns the incident and reflected wave time series, the main routine in this example calls the SPCTRA subroutine twice to return the incident and reflected wave spectra. The main program is as follows

```
      PARAMETER (NB=8,NW=3,ND=16384)
      PARAMETER (DH=0.47)
      DIMENSION SMI(256), SMR(256), REFL(256)
      DIMENSION SPI(4097), SPR(4097), FSI(256), FSR(256)
      DIMENSION TS(16384,3), TI(16384), TR(16384), XG(3)
      CHARACTER*6 FLNM(3)
C To provide IMSL workspace
      COMMON /WORKSP/ RWKSP
      REAL RWKSP(65592)
```

```
      CALL IWKIN(65592)
C Wave gage spacing (in meters) and data file names
      DATA XG / 0.0, -1.4, -2.0/
      DATA FLNM / 'CM06G1', 'CM06G2', 'CM06G3'/
C Read in three files
      DO 10 J = 1, NW
         OPEN(UNIT=11,FILE=FLNM(J))
         READ(11,1) NP, DT
1        FORMAT(I10,F10.4)
         READ(11,*) (TS(I,J), I=1,NP)
         CLOSE(11)
C Remove mean from each time series
         CALL TAKEMN(TS(1,J),NP,SD)
10    CONTINUE
C Call IRSORT subroutine
      CALL IRSORT(TS,ND,NW,NP,DT,XG,DH,FMN,FMX,TI,TR)
C To compare incident and reflected spectra
      CALL TAKEMN(TI,NP,SD)
      CALL TAKEMN(TR,NP,SD)
      CALL SPCTRA(TI,NP,DT,NB,SPI,FSI,SMI)
      CALL SPCTRA(TR,NP,DT,NB,SPR,FSR,SMR)
C Estimate reflection as a function of frequency
      DO 20 I = 1, NP/2/NB
         IF (FSI(I).GE.FMN.AND.FSI(I).LE.FMX) THEN
            REFL(I) = SQRT(SMR(I)/SMI(I))
         ELSE
            REFL(I) = 0.0
         ENDIF
20    CONTINUE
C Make graphs
      . . .
```

and the output is given in Figure 5 and Figure 6. The minimum resolvable frequency returned from the subroutine is $f_{\text{min}} = 0.0549s^{-1}$, and the maximum resolvable frequency is $f_{\text{max}} = 1.068s^{-1}$. Figure 5 shows the smoothed incident spectrum (solid line) and reflected spectrum (dashed line) in the range $0.0 < f < 1.2s^{-1}$ for the incident and reflected wave time series at $x = 0m$. The spectra are smoothed with $m = 8$, and, correspondingly, there are 16 degrees of freedom. Figure 6 shows the reflection coefficient calculated as a function of frequency, where the reflection coefficient at the n-th harmonic, r_n, is estimated as

$$r_n = \sqrt{\frac{(S_r)_n}{(S_i)_n}} \tag{78}$$

where $(S_r)_n$ and $(S_i)_n$ are the smoothed reflected and incident spectral estimates at the n-th harmonic, respectively. This example is intended to show the case of the resolvable frequency range which was not ideal for the specified incident wave spectrum.

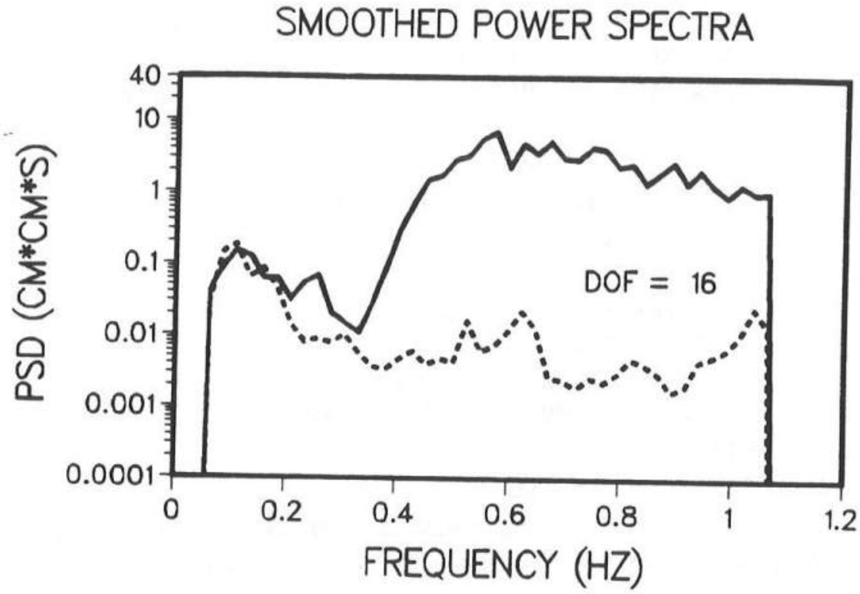

Figure 5: Smoothed Incident and Reflected Spectra at x=0m with 16 Degrees of Freedom Returned by IRSORT.

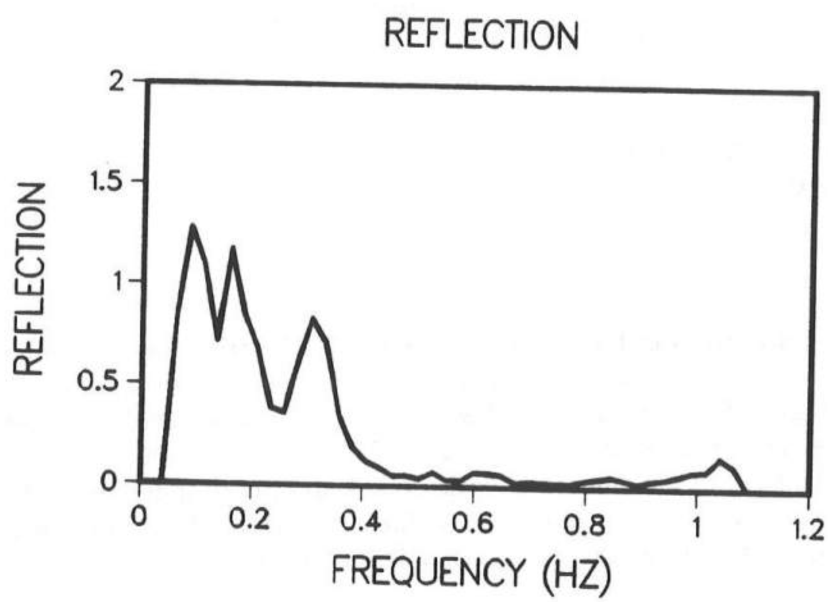

Figure 6: Reflection Coefficient as a Function of Frequency with 16 Degrees of Freedom.

Part VIII: Subroutine COHPHS

Mathematical Background

The two-sided cross-spectrum $S_{12}(f)$ in the frequency range $-\infty < f < \infty$ is related to the cross correlation function $C_{12}(\tau)$ expressed in terms of the time lag τ between two time series $\eta_1(t)$ and $\eta_2(t)$ and is given by

$$S_{12}(f) = \int_{-\infty}^{\infty} C_{12}(\tau) e^{-i2\pi f\tau} d\tau \tag{79}$$

This complex function is normally separated into the real and imaginary parts (*e.g.* Bendat and Piersol, 1986) and can be written

$$S_{12}(f) = K_{12}(f) + iQ_{12}(f) \tag{80}$$

where $K_{12}(f)$ is the (real) co-spectrum and $Q_{12}(f)$ is the (real) quadrature spectrum.

Alternatively, the two-sided cross-spectrum, $S_{12}(f)$, can be expressed in terms of the coherence squared, $\gamma^2(f)$, and the phase, $\theta(f)$, defined as

$$\gamma^2(f) = \frac{|S_{12}(f)|^2}{S_{11}(f)S_{22}(f)} \quad \text{for } 0 \le \gamma^2 \le 1 \tag{81}$$

and

$$\theta(f) = \tan^{-1}\left[\frac{Q_{12}(f)}{K_{12}(f)}\right] \quad \text{for } -\pi \le \theta \le \pi \tag{82}$$

where $S_{11}(f)$ and $S_{22}(f)$ are the two-sided auto-spectra for η_1 and η_2, respectively.

The coherence squared and phase are computed using the complex Fourier coefficients c_n^1 and c_n^2 at the frequency f_n computed from the time series $\eta_1(t)$ and $\eta_2(t)$, respectively. The coherence squared, $\gamma^2(f_n)$, and the phase, $\theta(f_n)$, at each frequency, f_n, are computed using the following equations

$$\gamma^2(f_n) = \frac{|(c_n^1)^* \, c_n^2|^2}{|c_n^1|^2 \, |c_n^2|^2} \tag{83}$$

and

$$\theta(f_n) = \tan^{-1}\left\{\frac{Im[(c_n^1)^* \, c_n^2]}{Re[(c_n^1)^* \, c_n^2]}\right\} \tag{84}$$

where $(c_n^1)^*$ is the complex conjugate of c_n^1 and Re and Im indicate the real and imaginary parts of $(c_n^1)^* \, c_n^2$, respectively. The smoothed coherence squared and phase are computed in a manner similar to the smoothed power density spectrum explained in Part VI where the smoothed values of $|c_n^1|^2$, $|c_n^2|^2$, and $(c_n^1)^* c_n^2$ are used in Equations 83 and 84 and are denoted by $\hat{\gamma}^2(f_k)$ and $\hat{\theta}(f_k)$.

Computer Program

The COHPHS subroutine, written to calculate the coherence squared and phase between two time series, is called by a main program or another subroutine

```
CALL COHPHS (TS1, TS2, NP, DT, NB, FS, CH, PH)
```

where the arguments are defined as

- IN:
  - TS1(NP) = first time series, $\eta_1(t)$ (L)
  - TS2(NP) = second time series, $\eta_2(t)$ (L)
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval, Δt (s)
  - NB = number of band-averaged data points for smoothing, m
- OUT:
  - FS(NP/2/NB) = smoothed frequency array, f_k (s^{-1})
  - CH(NP/2/NB) = smoothed coherence squared between TS1 and TS2, $\hat{\gamma}^2(f_k)$
  - PH(NP/2/NB) = smoothed phase between TS1 and TS2, $\hat{\theta}(f_k)$ (degrees)
- EXTERNAL ROUTINES:
  - FFTIMSL to compute the complex Fourier coefficients

Example

In this example, the coherence squared, γ^2, and phase, θ, is estimated between the two time series CM06G1 and CM06G3. These two data files are read into the main program and the mean is removed from both records. The main routine calls the COHPHS subroutine

```
      PARAMETER (NB=16)
      DIMENSION FS(1024), CH(1024), PH(1024)
      DIMENSION TS1(16384), TS2(16384)
C To provide IMSL workspace
      COMMON /WORKSP/ RWKSP
      REAL RWKSP(65592)
      CALL IWKIN(65592)
C Read in time series at x=0.0m (toe) and x=-2.0m (seaward of toe)
      OPEN(UNIT=11,FILE='CM06G1')
      READ(11,1) NP, DT
1     FORMAT(I10,F10.4)
      READ(11,*) (TS1(I), I=1,NP)
      CLOSE(11)
      OPEN(UNIT=11,FILE='CM06G3')
      READ(11,1) NP, DT
      READ(11,*) (TS2(I), I=1,NP)
      CLOSE(11)
C Remove mean from time series
      CALL TAKEMN(TS1,NP,SD)
      CALL TAKEMN(TS2,NP,SD)
```

```
C Call COHPHS subroutine
      CALL COHPHS(TS1,TS2,NP,DT,NB,FS,CH,PH)
C Make a graph
      . . .
```

and the subroutine returns smoothed coherence squared and phase arrays and an array of the corresponding frequency. The output is shown in Figure 7 and Figure 8 with 32 degrees of freedom ($m = 16$).

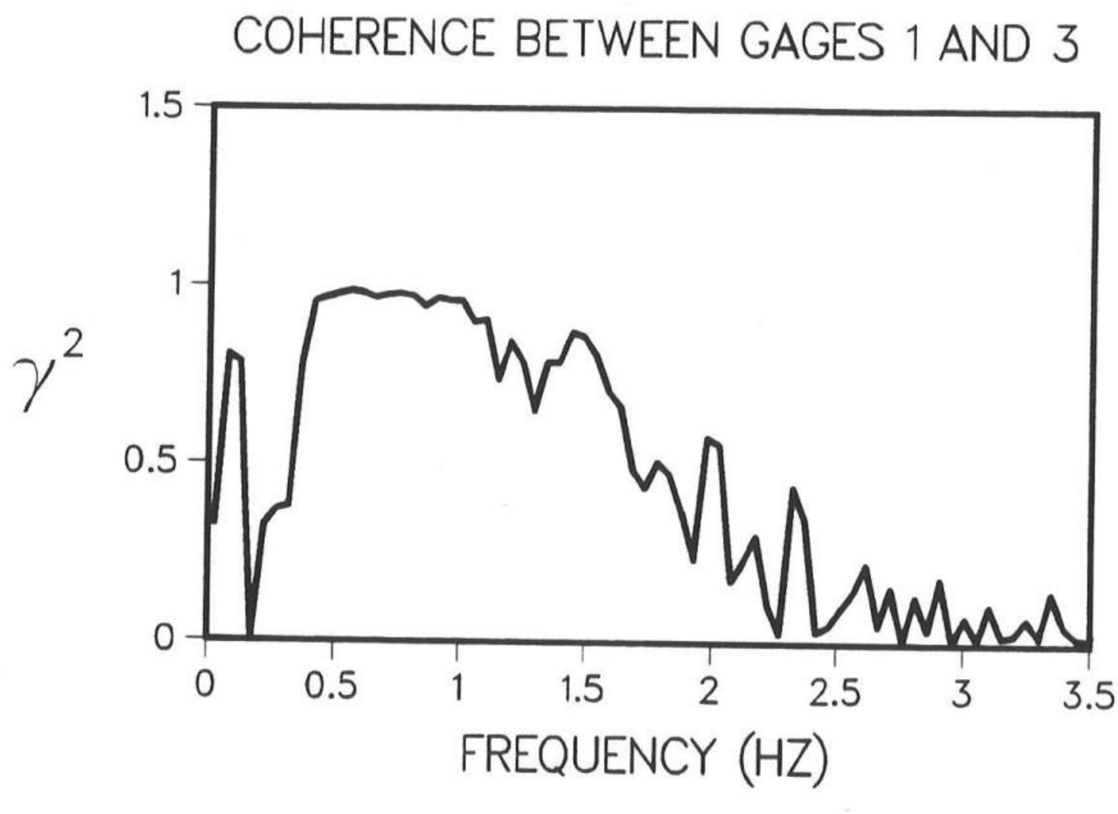

Figure 7: Coherence Squared for Two Time Series CM06G1 and CM06G3 Returned by COHPHS.

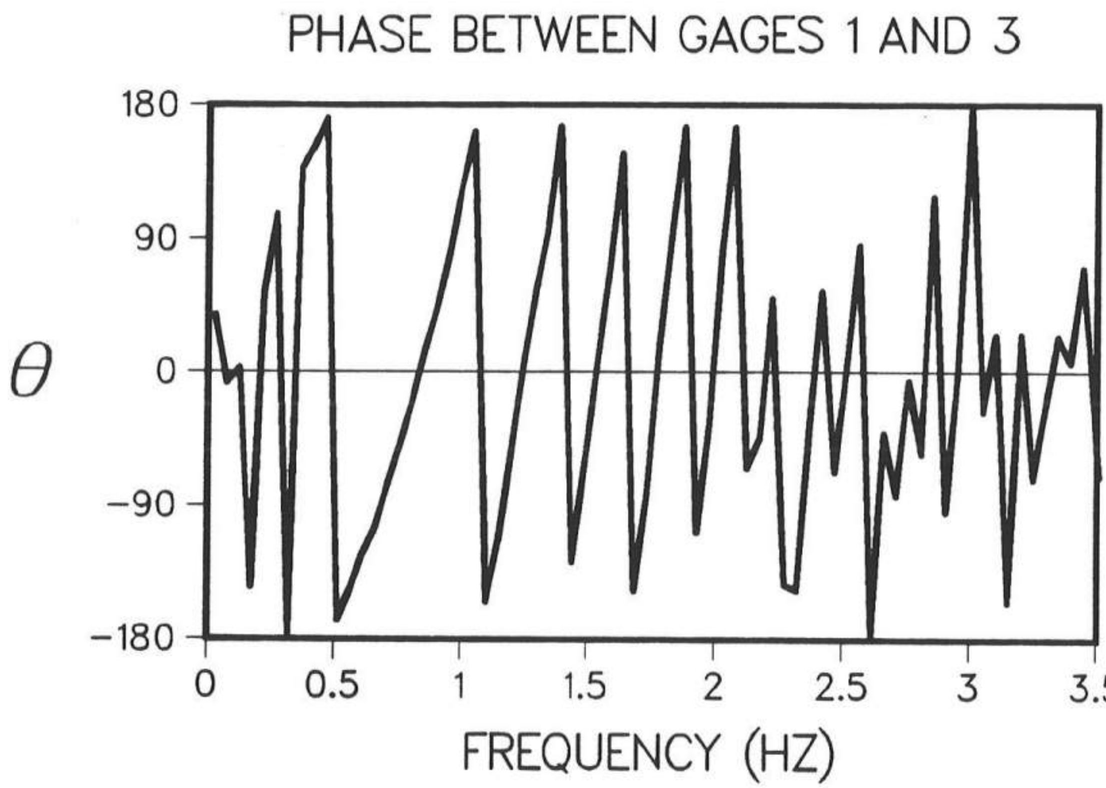

Figure 8: Phase for Two Time Series CM06G1 and CM06G3 Returned by COHPHS.

Part IX: Subroutine DISTNR

Mathematical Background

As in Part V, for a given time series, $\eta(t)$, the mean is computed by

$$\overline{\eta(t)} = \frac{1}{N}\sum_{i=1}^{N}\eta_i \tag{85}$$

The adjusted free surface is found by subtracting the mean, $(\eta_i - \overline{\eta})$, and denoted as η_i in Equations 86 and 87. The variance, Var, is given by

$$Var = \eta_{\text{rms}}^2 = \frac{1}{N}\sum_{i=1}^{N}\eta_i^2 \tag{86}$$

and the skewness, Skw, is given by

$$Skw = \frac{1}{N\eta_{\text{rms}}^3}\sum_{i=1}^{N}\eta_i^3 \tag{87}$$

The probability density function, PDF, for $x = \eta_i$ with non-zero mean can be compared with the normal distribution, $g(x)$, given by

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)^2\right] \tag{88}$$

where μ is the mean $\overline{\eta(t)}$, and σ is the standard deviation with $\sigma = \sqrt{Var}$. The PDF of the free surface is essentially a histogram of the free surface with the range from η_{min} to η_{max} divided into bins of width δx.

On the other hand, using the method of zero-upcrossing, the exceedance probability, P_E, corresponding to the height of the p-th ranked wave, H_p, is estimated by

$$P_E = \frac{p}{N_0 + 1} \tag{89}$$

where N_0 is the number of zero-upcrossing waves. If the probability distribution of the wave heights follows the Rayleigh distribution, then the exceedance probability associated with H_p is given by

$$P'_E = \exp\left[-2\left(\frac{H_p}{H_s}\right)^2\right] \tag{90}$$

where H_s is the significant wave height defined in Part V.

Computer Program

The DISTNR subroutine, written to compute the free surface distribution and exceedance probability, is called by a main program or another subroutine

```
CALL DISTNR (TS, NP, DT, XMN, XMX, DX, SD, VAR, SKW, F, XP,
             XN, NDX, HS, G, PE, PR, NZ)
```

where the arguments are defined as

- IN:
  - TS(NP) = time series, $\eta(t)$ (L)
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval, Δt (s)
  - XMN = minimum value for free surface displacement, $\eta_{\min}$ (L)
  - XMX = maximum value for free surface displacement, $\eta_{\max}$ (L)
  - DX = increment for estimating the probability density function for $x = \eta_i$, δx (L)
- OUT:
  - SD = mean of free surface, $\overline{\eta(t)}$ (L)
  - VAR = variance, Var (L^2)
  - SKW = skewness, Skw
  - F(NDX) = free surface array (*i.e.* bins), x (L)
  - XP(NDX) = probability density function of free surface, PDF
  - XN(NDX) = normal distribution, $g(x)$
  - NDX = length of array for PDF and normal distribution
  - HS = significant wave height from one-third highest waves (L), H_s
  - G(NZ) = values of HP/HS
  - PE(NZ) = exceedance probability, P_E
  - PR(NZ) = exccedance probability following Rayleigh distribution, P'_E
  - NZ = number of zero-upcrossing waves, N_0
- EXTERNAL ROUTINES:
  - TIMPAR to return the wave height rankings

Example

This example illustrates how the DISTNR subroutine can be used to compare the free surface distribution to the normal distribution and to compare the wave height distribution to the Rayleigh distribution. The time series for this example is CM06G1. The DISTNR subroutine is called by a main program

```
      PARAMETER (XMN=-6.0,XMX=6.0,DX=0.1)
      DIMENSION TS(16384), G(1000), PE(1000), PR(1000)
      DIMENSION F(1000), XP(1000), XN(1000)
C Read in time series at x=0.0m (toe)
      OPEN(UNIT=11,FILE='CM06G1')
      READ(11,1) NP, DT
```

```
1     FORMAT(I10,F10.4)
      READ(11,*) (TS(I), I=1,NP)
      CLOSE(11)
C Call DISTNR subroutine
      CALL DISTNR(TS,NP,DT,XMN,XMX,DX,SD,VAR,SKW,F,XP,XN,
     &    NDX,HS,G,PE,PR,NZ)
C Make graphs
      . . .
```

and the output of the subroutine is shown in Figure 9 and Figure 10.

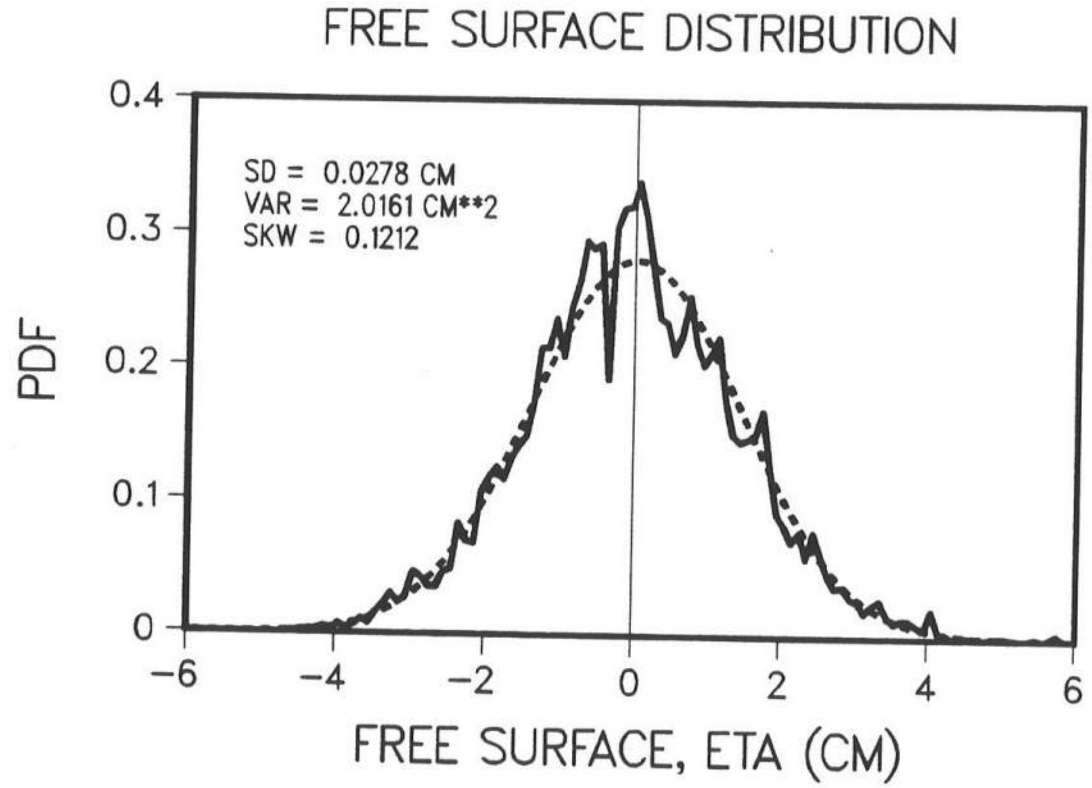

Figure 9: Probability Distribution Function of Free Surface Elevation for Time Series CM06G1 Compared with Normal Distribution Returned by DISTNR.

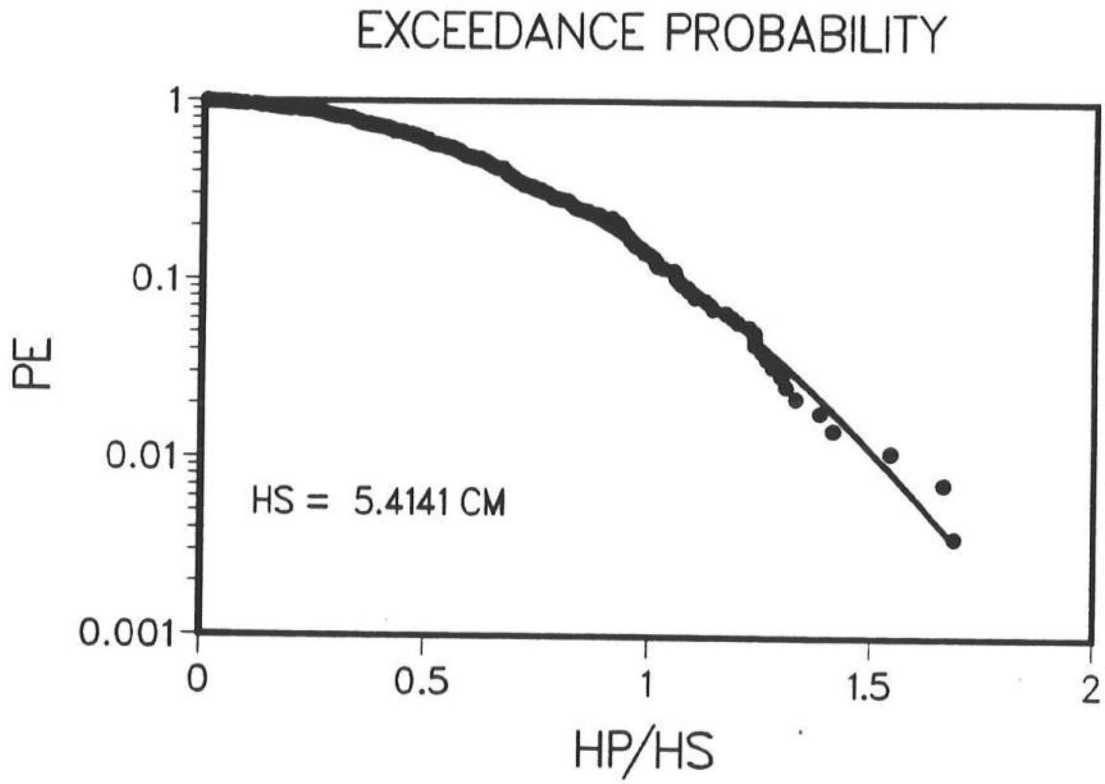

Figure 10: Wave Height Exceedance Probability for Time Series CM06G1 Compared with Rayleigh Distribution Returned by DISTNR.

Part X: Subroutine USRSPC

Mathematical Background

Time series of surface elevation can be generated from any power density spectrum of known shape. Subroutine USRSPC is created to accommodate the generation of time series from a power density spectrum whose shape is known but can not be expressed by a formula.

In using the USRSPC subroutine, a user needs to divide the given spectrum into a number of linear segments of known geometry as depicted in Figure 11 where the end points of the linear segments are referred to as the *raw points*. The largest frequency of the raw points is indicated by f_M. The coordinates of the raw points are specified as input to the USRSPC subroutine.

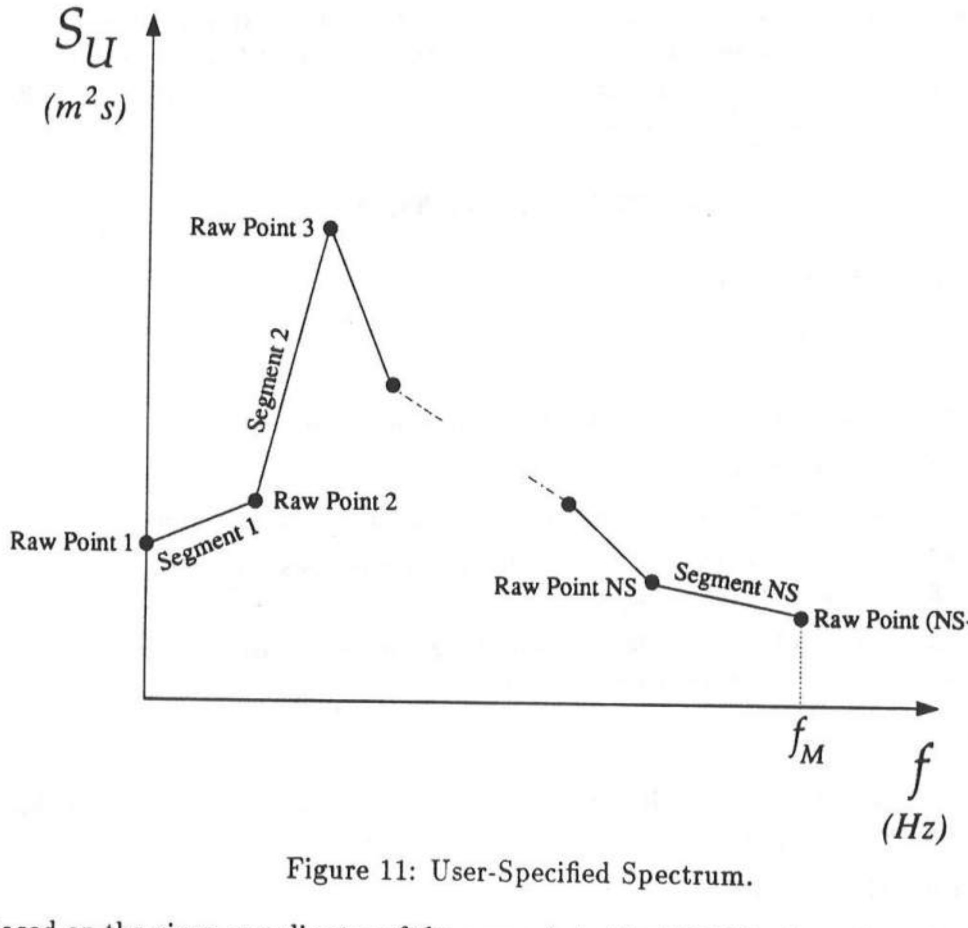

Figure 11: User-Specified Spectrum.

Based on the given coordinates of the raw points, the USRSPC subroutine calculates the *fine points*, *i.e.*, the values of spectral density at equally spaced discrete frequencies. The frequency resolution of the fine points is Δf given by

$$\Delta f = \frac{1}{N \Delta t} \tag{91}$$

where N and Δt are the even number of points and the sampling interval, respectively, of the requested time series. The fine points cover the range of $0 \leq f \leq f_{\mathrm{Nyq}}$ where the Nyquist frequency f_{Nyq} is determined by the sampling interval Δt and is given by

$$f_{\mathrm{Nyq}} = \frac{1}{2\Delta t} \tag{92}$$

For $f \leq f_{\mathrm{M}}$, the ordinates of the fine points, S_U, are obtained from a linear interpolation of the raw points. For $f > f_{\mathrm{M}}$, S_U is simply taken as zero. In addition, it is imposed that $S_U(f = 0) = 0$.

The fine points calculated by the USRSPC subroutine can be exported to the TIMEPH subroutine to yield a corresponding time series.

Computer Program

The USRSPC subroutine produces an array of the values of S_U at equally spaced discrete frequencies where the conditions $S_U(f = 0) = 0$ and $S_U(f > f_{\mathrm{M}}) = 0$ are imposed. The frequency resolution of the output spectrum is Δf given by Equation 91. The subroutine USRSPC is called by a main program or another subroutine

```
CALL USRSPC (NP, DT, NS, FR, SR, SP)
```

where the arguments are defined as

- IN:
  - NP = even number of data points in the time series, N
  - DT = sampling interval, Δt (s)
  - NS = number of linear segments specifying the given spectrum
  - FR = array of length (NS+1) containing the abscissas f (s^{-1}) of the raw points where FR(1)=0 and FR(NS+1)=f_{M}
  - SR = array of length (NS+1) containing the ordinates S_U $(L^2 s)$ of the raw points where SR(1) does *not* have to be zero
- OUT:
  - SP = array of length (NP/2+1) containing the ordinates S_U $(L^2 s)$ of the fine points where SP(1)=0 corresponding to $f = 0$ is imposed
- EXTERNAL ROUTINES:
  - none

Example

Kobayashi and Wurjanto (1991) simulated the Santa Barbara, California, "Feb 4, 1980" field data reported by Elgar and Guza (1985a, 1985b). They specified an incident wave spectrum based on the reported wave spectrum at the water depth $1.7m$, and generated a wave train based on the former spectrum.

The input of their data to the USRSPC subroutine consisted of 34 raw points corresponding to NS=33 as presented in Table 5. Other important parameters included the largest frequency of the raw points, $f_M = 0.4s^{-1}$; the sampling interval, DT = $\Delta t = 0.5s$; the Nyquist frequency, $f_{Nyq} = 1.0s^{-1}$; the peak period of the given spectrum, $t_p = 13.5s$; the length of the requested time series, $590t_p$; and the number of data points in the time series, NP = $590t_p/\Delta t = 15930$.

The USRSPC subroutine is called by a main program and the spectrum is returned.

```
      PARAMETER (NMAX=20000,NS=33)
      REAL TS(NMAX)
      REAL FN(NMAX/2+1), SP(NMAX/2+1)
      REAL FR(NS+1), SR(NS+1)
C Coordinates of the raw points
      DATA FR /.000000, .014052, .025293, .035597, .044965,
     2         .054801, .064169, .074941, .082904, .088525,
     3         .106792, .110539, .121780, .131148, .142389,
     4         .152693, .171429, .189227, .200468, .220141,
     5         .232319, .249180, .259953, .270726, .290398,
     6         .318501, .327400, .338173, .348478, .358782,
     7         .367213, .378454, .387354, .400000/
      DATA SR /.127742, .175202, .066442, .085046, .051275,
     2         .113203, .895258,1.091278, .483540, .355365,
     3         .123616, .086938, .062502, .108330, .157460,
     4         .150682, .072908, .049068, .043001, .059812,
     5         .055992, .041150, .038522, .034509, .033758,
     6         .038522, .033024, .031602, .028311, .031602,
     7         .030242, .034509, .033024, .034509/
C IMSL requirements
      COMMON /WORKSP/ RWKSP
      REAL RWKSP(127496)
      CALL IWKIN(127496)
C Parameters NP=number of data points, DT=sampling interval
      NP = 15930
      DT = 0.5
C Call USRSPC to obtain the fine points
      CALL USRSPC(NP,DT,NS,FR,SR,SP)
C Call TIMEPH to generate a corresponding time series
C with a seed value IS = 517644
      IS = 517644
      CALL TIMEPH(SP,NP,DT,IS,TS)
C Make a graph
      . . .
```

Figure 12 shows the spectrum for the range of $0 \le f \le f_M = 0.4s^{-1}$. It is imposed that

$S_U(f=0)=0$ and $S_U(f>f_M)=0$. Figure 13 shows the first $160t_p$ of the time series, which was generated by the TIMEPH subroutine with a seed value IS=517644, in a normalized form where the abscissa t denotes the time normalized by the peak period t_p=13.5s, and the ordinate η_i denotes the surface elevation normalized by a reference wave height of 0.9m.

Table 5: Raw Points Specifying the User-Specified Spectrum in Kobayashi and Wurjanto (1991).

I	FR(I) s^{-1}	SR(I) m^2s	I	FR(I) s^{-1}	SR(I) m^2s
1	0.000000	0.127742	18	0.189227	0.049068
2	0.014052	0.175202	19	0.200468	0.043001
3	0.025293	0.066442	20	0.220141	0.059812
4	0.035597	0.085046	21	0.232319	0.055992
5	0.044965	0.051275	22	0.249180	0.041150
6	0.054801	0.113203	23	0.259953	0.038522
7	0.064169	0.895258	24	0.270726	0.034509
8	0.074941	1.091278	25	0.290398	0.033758
9	0.082904	0.483540	26	0.318501	0.038522
10	0.088525	0.355365	27	0.327400	0.033024
11	0.106792	0.123616	28	0.338173	0.031602
12	0.110539	0.086938	29	0.348478	0.028311
13	0.121780	0.062502	30	0.358782	0.031602
14	0.131148	0.108330	31	0.367213	0.030242
15	0.142389	0.157460	32	0.378454	0.034509
16	0.152693	0.150682	33	0.387354	0.033024
17	0.171429	0.072908	34	0.400000	0.034509

I is the array element number
FR(I) is the abscissa of the raw point, s^{-1}
SR(I) is the ordinate of the raw point, m^2s

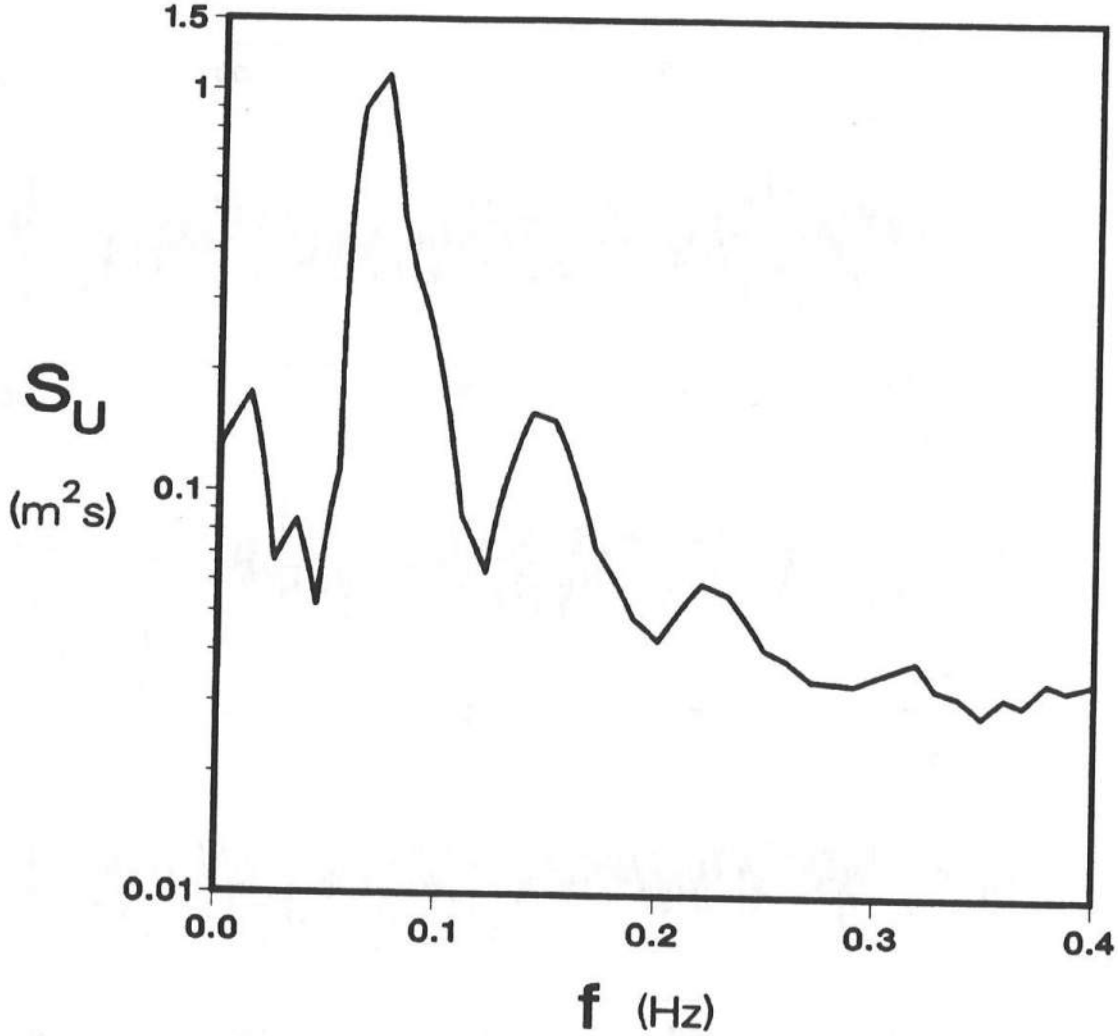

Figure 12: Spectrum for Field Data Returned by USRSPC.

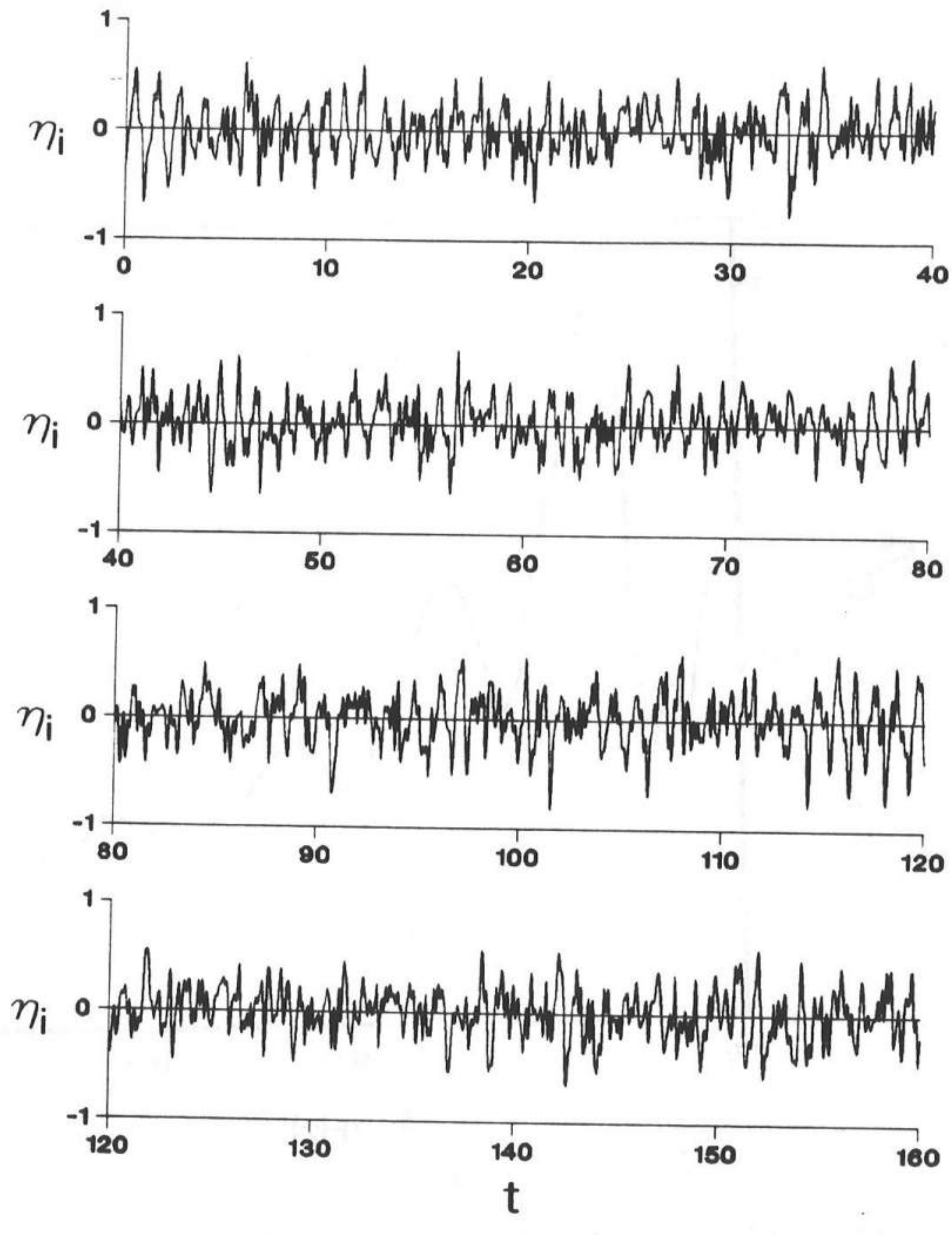

Figure 13: Normalized Time Series of Spectrum for Field Data Returned by TIMEPH.

Part XI: Subroutine PRORBR

Mathematical Background

For the case of irregular waves, the numerical model RBREAK (Wurjanto and Kobayashi, 1991) requires that a *normalized* time series of the free surface elevation at the seaward boundary, referred to as the *input wave train*, be specified in certain FORTRAN format. In addition, the input wave train needs to begin with a small value to provide a smooth transition from the initial condition of no wave action to a condition of full wave action as was done in the previous work by

- Kobayashi, Cox and Wurjanto (1990) where the input wave trains were obtained from a laboratory experiment, and
- Kobayashi, Wurjanto and Cox (1990a, 1990b) and Kobayashi and Wurjanto (1991) where the input wave trains were simulated using the TIMEPH subroutine.

It is noted that in the latter work, the input wave trains began with a sufficiently small negative value immediately following a zero-downcrossing point. Based on their experience, it is recommended that a *simulated* input wave train for the numerical model RBREAK begin in this way.

The TIMEPH and TIMEDC subroutines presented in Part IV generate *dimensional* time series, referred to as the *original* time series, which can not be used directly as input to the numerical model RBREAK. The subroutine PRORBR prepares an input wave train that satisfies the above requirements, based on the time series generated by either the TIMEPH or TIMEDC subroutine. First, the PRORBR subroutine shifts the original time series, which is periodic, such that the shifted time series has the required characteristics. The normalization of the shifted time series will then yield the desired input wave train for the numerical model RBREAK.

The relation between the original time series η'_j with j=1,2,...,N and the input wave train η_i with i=1,2,...,(N+1) with N being the even number of points in the original time series, is given by

$$\eta_i = \begin{cases} \left[\eta'_{i+j_0-1}\right] / H' & \text{for} \quad i \leq (N - j_0 + 1) \\ \left[\eta'_{i-(N-j_0+1)}\right] / H' & \text{for} \quad i > (N - j_0 + 1) \end{cases} \tag{93}$$

where H' is the reference wave height used for the normalization of the free surface elevation, which can be

1. the significant wave height of the original time series, H_s (Part V),
2. the spectral estimate of the significant wave height, H_{mo} (Part III), or
3. a user-specified value,

and j_0 marks the data point next to the first zero-downcrossing point in the original time series that satisfies the following conditions:

$$\eta'_{j_0-1} > 0 \quad \text{and} \quad \eta'_{j_0} < 0 \quad \text{and} \quad \frac{\left|\eta'_{j_0}\right|}{H'} < \varepsilon \tag{94}$$

with ε being a small positive value. The values of ε=0.001 and 0.005 have been used in the previous work mentioned above. Use of ε=0.001 is made in the PRORBR subroutine listed in Appendix A.

Computer Program

The PRORBR subroutine produces an input wave train for the numerical model RBREAK from the time series generated by either the TIMEPH or TIMEDC subroutine. The input wave train is stored in an output file formatted according to the RBREAK's convention. The subroutine PRORBR is called by a main program or another subroutine

```
CALL PRORBR (TS, NP, DT, FNAME, IP, HW, JO, TJO)
```

where the arguments are defined as

- IN:
  - TS(NP) = original time series, η' (L)
  - NP = even number of data points in the original time series, N
  - DT = sampling interval, Δt (s)
  - FNAME = name of the output file containing the input wave train for the numerical model RBREAK (FNAME is a CHARACTER*10 variable)
  - IP = option to specify the reference wave height used for the normalization of the dimensional time series η' as follows:
    - * H_s = significant wave height based on the time series (L) (IP=1 and HW is returned as H_s)
    - * H_{mo} = spectral estimate of the significant wave height (L) (IP=2 and HW is returned as H_{mo})
    - * A user-specified reference wave height (L) (IP=3 and HW needs to be specified as input to the PRORBR subroutine)
- IN/OUT:
  - HW = reference wave height used for the normalization of the dimensional time series η' (L)
- OUT:
  - JO = index j_0 satisfying the conditions given by Equation 94
  - TJO = dimensional time corresponding to the index $j_0 = (j_0 - 1)\Delta t$ (s)
- EXTERNAL ROUTINES:
  - TIMPAR to compute the significant wave height based on the time series for IP=1
  - SPCTRA to transform the original time series to the corresponding power density spectrum for IP=2
  - SPCPAR to compute the spectral estimate of the significant wave height for IP=2

Example

Part X of this report presents an example on how to generate a time series from a user-specified power density spectrum using the USRSPC subroutine. The example ends by calling the TIMEPH subroutine that generates a dimensional time series from the specified power density spectrum.

The following example is a continuation of the example of Part X. Added in this example is a call to the PRORBR subroutine that produces the corresponding input wave train for the numerical model RBREAK.

```
      PARAMETER (NMAX=20000,NS=33)
      REAL TS(NMAX)
      REAL FN(NMAX/2+1), SP(NMAX/2+1)
      REAL FR(NS+1), SR(NS+1)
C Character variable to provide name for the output file created
C   by the PRORBR subroutine
      CHARACTER*10 FNAME
C Coordinates of the raw points
      DATA FR /.000000, .014052, .025293, .035597, .044965,
     2         .054801, .064169, .074941, .082904, .088525,
     3         .106792, .110539, .121780, .131148, .142389,
     4         .152693, .171429, .189227, .200468, .220141,
     5         .232319, .249180, .259953, .270726, .290398,
     6         .318501, .327400, .338173, .348478, .358782,
     7         .367213, .378454, .387354, .400000/
      DATA SR /.127742, .175202, .066442, .085046, .051275,
     2         .113203, .895258,1.091278, .483540, .355365,
     3         .123616, .086938, .062502, .108330, .157460,
     4         .150682, .072908, .049068, .043001, .059812,
     5         .055992, .041150, .038522, .034509, .033758,
     6         .038522, .033024, .031602, .028311, .031602,
     7         .030242, .034509, .033024, .034509/
C IMSL requirements
      COMMON /WORKSP/ RWKSP
      REAL RWKSP(127496)
      CALL IWKIN(127496)
C Parameters NP=number of data points, DT=sampling interval
      NP = 15930
      DT = 0.5
C Call USRSPC to obtain the fine points
      CALL USRSPC(NP,DT,NS,FR,SR,SP)
C Call TIMEPH to generate a corresponding time series
C with a seed value IS = 517644
      IS = 517644
      CALL TIMEPH(SP,NP,DT,IS,TS)
C
```

```
C The above procedure is identical to the example of Part X,
C   except for the declaration of the character variable FNAME,
C   which is added in this example.
C The TIMEPH subroutine returns a dimensional time series, TS,
C   from which the PRORBR subroutine will produce the
C   corresponding input wave train for RBREAK
C The input wave train is not returned to the calling program,
C   but is stored in an output file, the name of which is
C   specified by the character variable FNAME
C
      FNAME = 'TKWSB2     '
      IP = 3
      HW = 0.9
      CALL PRORBR(TS,NP,DT,FNAME,IP,HW,JO,TJO)
      WRITE (*,*) ' Index JO =',JO
      WRITE (*,*) ' Time corresponding to the index JO, TJO =',TJO
C Make a graph
      . . .
```

It is noted that the option IP=3 with the corresponding reference wave height HW=$0.9m$ is used in this example. This reference wave height was actually meant to be the spectral estimate of the significant wave height: the two wave heights indeed agreed to the third decimal place. This example could have used the option IP=2, which would have resulted in an almost identical input wave train. The reason why the option IP=3 was used in this example was to get the exact value of $0.9m$ for the reference wave height.

The index j_0 was found to be 970, corresponding to the dimensional time t'=$484.5s$ in the original time series, which has been presented in the normalized form in Figure 13 of Part X where the normalized time t=35.889 corresponds to t'=$484.5s$.

Figure 14 shows the first $160t_p$ of the input wave train where the abscissa t denotes the time normalized by the peak period t_p=$13.5s$, and the ordinate η_i denotes the surface elevation normalized by the reference wave height, HW=$0.9m$.

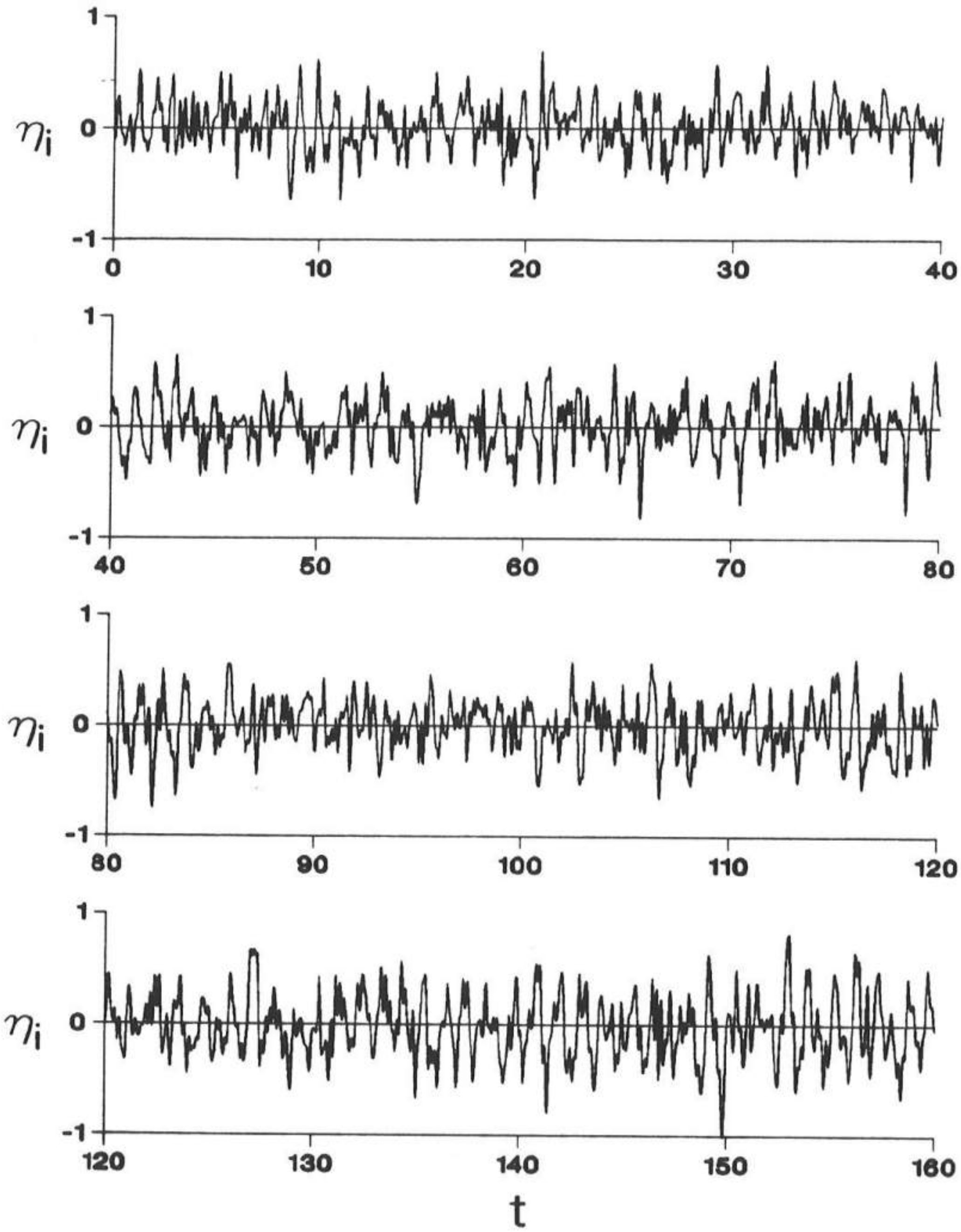

Figure 14: Input Wave Train for RBREAK Created by PRORBR.

Part XII: Subroutine FFTIMSL

Mathematical Background

The subroutine FFT2D in the IMSL library is used to compute the complex Fourier coefficients, c_n, given by Equations 31 to 34 where the real Fourier coefficients a_n and b_n for the time series $\eta(t)$ with zero mean are defined in Equation 28. The discrete time series η_j can be expressed

$$\eta_j = \eta(t_j) \quad \text{for} \quad j = 1, 2, \ldots, N \tag{95}$$

with

$$t_j = (j-1)\Delta t \tag{96}$$

where Δt is the sampling interval and N is the even number of data points. Using this definition, Equation 28 can be shown to yield

$$N c_n = \sum_{j=1}^{N} \eta_j \exp\left[-\frac{2\pi i(j-1)(n-1)}{N}\right] \tag{97}$$

where $i^2 = -1$. Equation 97 is in the form which allows the direct use of the subroutine FFT2D to compute c_n with $n = 1, 2, \ldots, N$. It is noted that c_n in Equations 31 to 34 corresponds to c_{n-1} obtained from Equation 97.

The subroutine FFT2B is the IMSL subroutine to compute the inverse Fourier transform of given coefficients c_n to find η_j with $j = 1, 2, \ldots, N$. To apply the subroutine FFT2B, Equation 28 is rewritten as

$$\eta_j = \sum_{n=1}^{N} c_n \exp\left[\frac{2\pi i(j-1)(n-1)}{N}\right] \quad \text{for} \quad j = 1, 2, \ldots, N \tag{98}$$

where η_j is real and c_n in this equation corresponds to c_n in Equation 97.

Computer Program

The FFTIMSL subroutine is called by a main program or another subroutine

```
CALL FFTIMSL (TS, CN, NP, IO)
```

where the arguments are defined as

- IN/OUT:
  - TS(NP) = time series, $\eta(t)$ (L)
  - CN(NP) = complex Fourier coefficients, c_n (L)
- IN:
  - NP = even number of data points in the time series, N
  - IO = option for an FFT (IO=+1) to return complex coefficients of known time series or for an inverse FFT (IO=-1) to return time series for known complex coefficients
- EXTERNAL ROUTINES:
  - FFT2D, an IMSL routine for FFT
  - FFT2B, an IMSL routine for inverse FFT

Example

The FFTIMSL subroutine is illustrated in this example by considering a periodic saw-tooth wave form given by

$$f(t) = -\frac{t}{\pi} \quad \text{for} \quad -\pi \le t \le \pi \tag{99}$$

with period, $T = 2\pi$. The function, $f(t)$, with zero mean can be written as a finite sum

$$f(t) = \sum_{n=1}^{N/2} \left[a_n \cos(n\omega t) + b_n \sin(n\omega t) \right] \quad -\pi \le t \le \pi \tag{100}$$

where the angular frequency $\omega = 2\pi/T = 1$ and the Fourier coefficients are given by (*e.g.* Bendat and Piersol, 1986)

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(\tau)\cos(n\omega\tau)d\tau \tag{101}$$

and

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(\tau)\sin(n\omega\tau)d\tau \tag{102}$$

It is noted that $f(t)$ obtained from Equation 100 approaches $f(t)$ given by Equation 99 as N approaches infinity.

For this example, by inspection the values of a_n are zero since $f(t)$ is an odd function and $\cos(n\omega t)$ is an even funtion. Solving for b_n gives

$$b_n = \frac{2}{n\pi}(-1)^n \tag{103}$$

To check the FFT routine, the discrete function, f_i, $i = 1, 2, \ldots, N$ where $N = 16$ for this case, is constructed corresponding to $f(t)$ above. Since the FFT is defined for $t \ge 0$, the range of $0 \le t \le 2\pi$ is considered and the discrete time domain is given by

$$t_i = \frac{\Delta t}{2} + (i-1)\Delta t \quad ; \quad \Delta t = \frac{2\pi}{N} \tag{104}$$

It is noted that since N is small for this example, the time domain is shifted slightly by $\Delta t/2$ to reduce the effect of spectral leakage. This effect is negligible for large N and the discrete time domain of Equation 96 is generally adopted. In the main program, the Fourier coefficients b_n are computed and compared with the real and imaginary parts of the complex Fourier coefficients returned from the FFT routine. The main program is written as

```
      PARAMETER (NP=16,NL=4096)
      REAL B(4096), TS(256), TSR(256), TSB(256), T(256)
      COMPLEX CN(256)
C Constants
      PI = 4.*ATAN(1.)
C Fourier coefficients for analytic solution
      DO 3 I = 1, NL, 2
         B(I) = -2./(PI*FLOAT(I))
```

```
          B(I+1) = 2./(PI*FLOAT(I+1))
3       CONTINUE
C Time step
        DT = 2.*PI/FLOAT(NP)
        DO 7 I = 1, NP
C Make time array and time series array from zero to two pi
          T(I) = DT/2. + FLOAT(I-1)*DT
          TS(I) = -T(I)/PI
          IF(T(I).GT.PI) TS(I) = TS(I)+2.
C Reconstruct time series using coefficients of analytic solution
          SUM = 0.0
          DO 5 K = 1, NL
              SUM = SUM + B(K)*SIN(FLOAT(K)*T(I))
5         CONTINUE
          TSB(I) = SUM
7       CONTINUE
C Call FFTIMSL to return complex Fourier coefficients
        CALL FFTIMSL(TS,CN,NP,+1)
C Call FFTIMSL to return time series for comparison
        CALL FFTIMSL(TSR,CN,NP,-1)
C Make a table
        . . .
```

and the output is given in Table 6. Table 6 shows the values of a_n and b_n based on the complex coefficient array, CN, returned by the FFTIMSL subroutine. The last column shows the Fourier coefficients calculated by Equation 103. The values of a_n and b_n will approach the analytical values as N increases from 16.

Table 7 shows the original time series, $f(t_i)$, the reconstructed time series using the inverse FFT, $f(t_i)'$, and the reconstructed time series using a large number of terms and the Fourier coefficients computed by Equation 103, $f(t_i)''$, where

$$f(t_i)'' = \sum_{n=1}^{4096} b_n \sin(n\omega t_i) \tag{105}$$

and $N/2 = 4096$ was chosen as a large number approaching infinity.

Table 6: Fourier Coefficients Returned by FFTIMSL.

n	$(a_n)_{\text{FFT}}$	$(b_n)_{\text{FFT}}$	b_n
0	0.0000	0.0000	0
1	-0.1250	-0.6284	-0.6366
2	0.1250	0.3018	0.3183
3	-0.1250	-0.1871	-0.2122
4	0.1250	0.1250	0.1592
5	-0.1250	-0.0835	-0.1273
6	0.1250	0.0518	0.1061
7	-0.1250	-0.0249	-0.0909
8	0.1250	0.0000	0.0796

n is the harmonic
$(a_n)_{\text{FFT}}$ is the real part, $a_n = 2Re(c_n)$ for $1 \leq n \leq 8$
$(b_n)_{\text{FFT}}$ is the imaginary part, $b_n = -2Im(c_n)$ for $1 \leq n \leq 8$
b_n is the Fourier coefficient from the analytic part

Table 7: Saw-Tooth Time Series Reconstructed with an Inverse FFT by FFTIMSL.

i	time, t_i	$f(t_i)$	$f(t_i)'$	$f(t_i)''$
1	0.1964	-0.0625	-0.0625	-0.0624
2	0.5890	-0.1875	-0.1875	-0.1874
3	0.9817	-0.3125	-0.3125	-0.3123
4	1.3744	-0.4375	-0.4375	-0.4373
5	1.7671	-0.5625	-0.5625	-0.5623
6	2.1598	-0.6875	-0.6875	-0.6872
7	2.5525	-0.8125	-0.8125	-0.8121
8	2.9452	-0.9375	-0.9375	-0.9365
9	3.3397	0.9375	0.9375	0.9366
10	3.7306	0.8125	0.8125	0.8121
11	4.1233	0.6875	0.6875	0.6872
12	4.5160	0.5625	0.5625	0.5623
13	4.9087	0.4375	0.4375	0.4373
14	5.3014	0.3125	0.3125	0.3123
15	5.6941	0.1875	0.1875	0.1874
16	6.0868	0.0625	0.0625	0.0625

t_i is the time level
$f(t_i)$ is the original time series
$f(t_i)'$ is the reconstructed time series using the inverse FFT
$f(t_i)''$ is the reconstructed time series using the analytic Fourier coefficients

Part XIII: Subroutine RDMGEN

Mathematical Background

Random number generators typically require a seed value to begin their algorithm and return an array of random numbers uniformly distributed between zero and one. By specifying the same seed, the routine will generally return the same array of numbers. Therefore, these random number simulators actually generate pseudo-random numbers.

Computer Program

The RDMGEN subroutine is called by the TIMEPH subroutine

```
CALL RDMGEN (NR, IS, AR)
```

where the arguments are defined as

- IN:
  - NR = number of random numbers
  - IS = seed value to initialize random number generator
- OUT:
  - AR(NR) = random numbers distributed uniformly between 0 and 1
- EXTERNAL ROUTINES:
  - RNSET, an IMSL routine for setting the random number generator
  - RNUN, an IMSL routine for generating the random numbers

The purpose of the RNSET subroutine is to initialize a random seed for use in the IMSL random number generators. If the seed value is set to zero, then the random number generator is started by a seed value from the system clock. The purpose of the RNUN subroutine is to generate pseudo-random numbers from a uniform distribution from zero to one, excluding zero and one. All values returned by RNUN are positive and less than one.

Example

This example is to show the probability distribution function for an array of numbers returned from the RDMGEN subroutine. The main program is written as

```
      PARAMETER (NP=8192,IS=123457)
      PARAMETER (DX=0.01)
      DIMENSION TS(8192), X(1000), XP(1000)
C Get array of random numbers
      CALL RDMGEN(NP,IS,TS)
```

```
C Probability density function (a histogram)
      NDX = 1./DX
      DO 10 J = 1,NDX
          KOUNT=0
          X(J) = FLOAT(J-1)*DX+DX/2.0
          DO 5 I = 1,NP
               IF (TS(I).GT.FLOAT(J-1)*DX .AND.
     &              TS(I).LE.FLOAT(J)*DX) THEN
                    KOUNT=KOUNT+1
               ENDIF
 5        CONTINUE
          IF(KOUNT.EQ.0)THEN
               XP(J) = 0.0
          ELSE
               XP(J) = FLOAT(KOUNT)/FLOAT(NP)/DX
          ENDIF
10    CONTINUE
C Make a graph of probability distribution
      . . .
```

and the output is given in Figure 15. In theory, the probability density function should be unity in the range $0 < x < 1$. The agreement is expected to improve as the value of **NP** is increased.

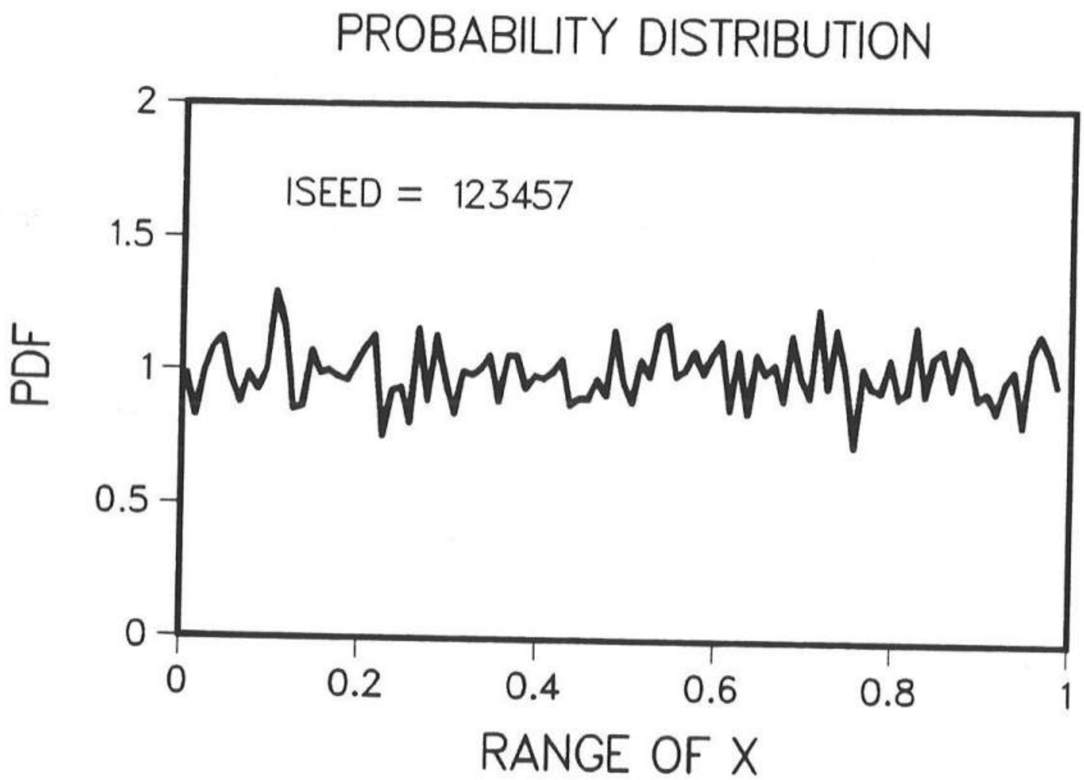

Figure 15: Probability Distribution Function of Random Numbers Returned by RDMGEN.

Part XIV: Subroutine WAVNUM

Mathematical Background

Following linear wave theory (*e.g. Shore Protection Manual*, 1984), the profile of a wave, $\eta(t)$, propagating in the positive x-direction in time, t, can be written

$$\eta(t) = \frac{H}{2}\cos(kx - \omega t) \tag{106}$$

where H is the wave height, T is the wave period related to the angular frequency, ω, and the frequency, f, by

$$\omega = \frac{2\pi}{T} = 2\pi f \tag{107}$$

and k is the wave number related to the wave length, L, by

$$k = \frac{2\pi}{L} \tag{108}$$

For given f, the wave number can be determined from the linear dispersion relation

$$(2\pi f)^2 = gk\tanh(kh) \tag{109}$$

where h is the water depth. For shallow water where $h/L < 1/25$, the wave number can be found from

$$k = \frac{2\pi f}{\sqrt{gh}} \tag{110}$$

and for deep water where $h/L > 1/2$,

$$k = \frac{(2\pi f)^2}{g} \tag{111}$$

Computer Program

The subroutine **WAVNUM**, written to return the wave number based on linear wave theory, is called by a main program or another subroutine

```
CALL WAVNUM (FQ, DH, WN)
```

where the arguments are defined as

- IN:
  - FQ = frequency, f (s^{-1})
  - DH = water depth, h (m)
- OUT:
  - WN = wave number, k (m^{-1})
- EXTERNAL ROUTINES:
  - none

It is noted that the SI units with $g = 9.81ms^{-2}$ are used in this routine.

Example

The example for the **WAVNUM** subroutine computes the wave number, k, for the wave conditions used in earlier examples related to laboratory experiments. The main program calls the **WAVNUM** subroutine by

```
      PARAMETER (NP=16384)
      PARAMETER (DH=0.47,DT=0.04)
C Constants
      PI = 4.*ATAN(1.)
      C1 = 9.81/(2.*PI)
      DF = 1./(FLOAT(NP)*DT)
C Do loop for the frequency range of interest
      DO 10 I = 2, NP/2+1
C Frequency and deep water wave length
         FQ = DF*FLOAT(I-1)
         DLO = DH/(C1*(1./FQ)**2)
C Call WAVNUM subroutine
         CALL WAVNUM(FQ,DH,WN)
C Compute water depth to wave length ratio, DL
         DL = DH/(2.*PI/WN)
10    CONTINUE
C Make a table
      . . .
```

and the subroutine returns the wave number at given frequency. It is noted that N was increased from 8192 of previous examples to 16384 in this example to increase the frequency resolution for ease of comparison of the values returned by **WAVNUM** with those of Table C-1 of the *Shore Protection Manual* (1984). Table 8 compares selected values of h/L computed with the wave number returned by the **WAVNUM** subroutine with the values from Table C-1, $(h/L)_{SPM}$, as a function of h/L_0. It is noted that the values in Table C-1 have four significant digits and that five significant digits have been included for the values computed by the **WAVNUM** subroutine in Table 8.

Table 8: Wave Number for Laboratory Wave Conditions Returned by **WAVNUM**.

I	h/Lo	h/L	$(h/Lo)_{SPM}$	$(h/L)_{SPM}$
276	0.053005	0.097268	0.05300	0.09726
435	0.132016	0.16826	0.1320	0.1682
603	0.254005	0.27136	0.2540	0.2714
683	0.32600	0.33573	0.3260	0.3357
787	0.43301	0.43661	0.4330	0.4366
864	0.52200	0.52345	0.5220	0.5235

I is the counter of the DO loop
h/Lo is the ratio of water depth to deep water wave length with $h = 0.47m$
h/L is the ratio of water depth to wave length from the **WAVNUM** subroutine
$(h/Lo)_{SPM}$ is the ratio of water depth to deep water wave length to four significant digits from Table C-1
$(h/L)_{SPM}$ is the ratio water depth to wave length to four significant digits from Table C-1

Part XV: Conclusions

Fourteen subroutines have been presented herein for standard spectral and time series analyses. These subroutines have been used to specify numerically-generated incident random waves as input to the numerical model RBREAK as well as to analyze and interpret the computed time series by Kobayashi, Wurjanto and Cox (1990a, 1990b) and Kobayashi and Wurjanto (1991). These subroutines have also been used to conduct irregular wave tests in a wave flume to calibrate and evaluate the capabilities and limitations of RBREAK by Kobayashi, Cox and Wurjanto (1990, 1991). Users of this report are recommended to read these papers for the actual applications of the subroutines to the problems associated with random waves on coastal structures and beaches.

The subroutines included in this report are relatively short and may be modified easily by users where necessary. In any case, these subroutines are essential for the analyses of random waves and the interpretations of the measured and computed time series. The standard spectral and time series analysis methods employed in these subroutines may not yield clear interpretations for highly nonlinear random waves on coastal structures and beaches. Consequently, new analysis methods will need to be developed for highly nonlinear random waves.

References

Bendat, J.S. and Piersol, A.G. (1986), *Random Data Analysis and Measurement Procedures*, John Wiley & Sons, Inc., New York.

Bouws, E., Günther, H., Rosenthal, W., and Vincent, C. L. (1985), "Similarity of the Wind Wave Spectrum in Finite Depth Water, 1. Spectral Form," *Journal of Geophysical Research*, 90(C1), 975–986.

Cartwright, D. E. and Longuet-Higgins, M. S. (1956), "The Statistical Distribution of the Maxima of Random Functions," *Proc. Royal Society of London*, A(237) 212–232.

Cox, D.T. (1989), "Irregular Wave Reflection and Runup on Rough, Permeable Slopes," thesis presented to the University of Delaware, at Newark, Delaware, in partial fulfillment of the requirements for the degree of Master of Civil Engineering.

Elgar, S. and Guza, R.T. (1985a), "Shoaling Gravity Waves: Comparisons Between Field Observations, Linear Theory, and a Nonlinear Model," *J. Fluid Mechanics*, 158, 47-70.

Elgar, S. and Guza, R.T. (1985b), "Observation of Bispectra of Shoaling Surface Gravity Waves," *J. Fluid Mechanics*, 161, 425-448.

Elgar, S., Guza, R. T., and Seymour, R. J. (1985), "Wave Group Statistics from Numerical Simulation of a Random Sea," *Applied Ocean Research*, 7(2), 93–96.

Goda, Y. (1985), *Random Seas and Design of Maritime Structures*, University of Tokyo Press, Tokyo.

Goda, Y. and Suzuki, Y. (1976), "Estimation of Incident and Reflected Waves in Random Wave Experiments," *Proc. of 15th International Coastal Engineering Conference*, Hawaii, 828–845.

Kitaigorodskii, S. A., Krasitskii, V. P., and Zaslavskii, M. M. (1975), "On Phillips' Theory of Equilibrium Range in the Spectra of Wind-Generated Water Waves," *Journal of Physical Oceanography*, 5, 410–420.

Kobayashi, N., Cox, D.T. and Wurjanto, A. (1991), "Permeability Effects on Irregular Wave Run-up and Reflection," *J. Coastal Research*, 7(1), 127–136.

Kobayashi, N., Cox, D.T. and Wurjanto, A. (1990), "Irregular Wave Reflection and Run-up on Rough Impermeable Slopes," *J. Waterway, Port, Coastal, and Ocean Engineering* , ASCE, 116(6), 708–726.

Kobayashi, N., and Wurjanto, A. (1991), "Irregular Wave Setup and Run-up on Beaches," Submitted to *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE.

Kobayashi, N., Wurjanto, A. and Cox, D.T. (1990a), "Irregular Waves on Rough Permeable Slopes," *J. Coastal Research*, 7(SI), 167–184.

Kobayashi, N., Wurjanto, A. and Cox, D.T. (1990b), "Rock Slopes under Irregular Wave Attack," *Proc. 22nd Coastal Engineering Conference*, ASCE, 1306–1319.

Longuet-Higgins, M. S. (1957), "The Statistical Analysis of a Random, Moving Surface," *Trans. Royal Society of London*, 249, 321–387.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986), *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, United Kingdom.

Seelig, W.N. (1980), "Two-Dimensional Tests of Wave Transmission and Reflection Characteristics of Laboratory Breakwaters," *Technical Report No. 80-1*, U.S. Army Coastal Engineering Research Center, Fort Belvoir, VA.

Shore Protection Manual (1984), Department of the Army, Waterways Experiment Station, Coastal Engineering Research Center, Vicksburg, MS.

Tucker, M. J., Challenor, P. G., and Carter, D. J. T. (1984), "Numerical Simulation of a Random Sea: A Common Error and its Effect upon Wave Group Statistics," *Applied Ocean Research*, 6(2), 118–122.

Thornton, E.B. and Calhoun, R.J. (1972), "Spectral Resolution of Breakwater Reflected Waves," *J. Waterway, Harbor, Coastal Engineering Division*, ASCE, 98(4), 443–460.

Wurjanto, A. and Kobayashi, N. (1991), "Numerical Model for Random Waves on Impermeable Coastal Structures and Beaches," *Research Report No. CACR-91-05*, Center for Applied Coastal Research, University of Delaware, Newark, Delaware.

Appendix A: Subroutine Listings

TMASPC Subroutine

```
C======================================================================C
C     TMASPC
C
C     COMPUTES TMA POWER DENSITY SPECTRUM FOR WIND WAVES IN
C     FINITE WATER DEPTH
C
C        IN:
C     NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C     DT...........TIME STEP (SAMPLING INTERVAL) (S)
C     FP...........PEAK FREQUENCY OF TMA SPECTRUM (HZ)
C     DH...........WATER DEPTH (M)
C     IP...........OPTION TO SPECIFY EITHER HRMS (IP=1) OR ALPHA (IP=2)
C
C        IN/OUT:
C     HR...........ROOT MEAN SQUARE WAVE HEIGHT (M)
C     AP...........SPECTRAL CONSTANT, ALPHA
C
C        OUT:
C     SP(NP/2+1)...SPECTRAL ARRAY (M*M*S)
C                  WHERE SP(1)=0 CORRESPONDS TO FREQUENCY = 0
C
C        EXTERNAL ROUTINE:
C     WAVNUM.......RETURN WAVE NUMBER
C======================================================================C
      SUBROUTINE TMASPC (NP, DT, FP, DH, IP, HR, AP, SP)
C
      REAL       SP(NP/2+1)
C
      NH = NP/2+1
      TM = NP*DT
      DF = 1.0/TM
C                                COMPUTE SPECTRUM; AP=1. IF IP=1
      IF (IP .EQ. 1) THEN
         AP = 1.0
      ENDIF
      SP(1) = 0.0
      DO 100 I = 2,NH
         FQ = (I-1) * DF
         CALL WAVNUM(FQ,DH,WN)
         AKH=WN*DH
         AKH2=WN*DH*2.
C                                LIMIT FOR SINH(ARG), ARG > 175.366
         IF (AKH2.GT.150) THEN
              PHIK = 1.0
         ELSE
              PHIK = (TANH(AKH))**2 / (1.+AKH2/SINH(AKH2))
         ENDIF
         SP(I) = AP * EJ(FQ,FP) * PHIK
100   CONTINUE
```

```
C                                IF HR SPECIFIED, CALCULATE AP
      CALL INTGRL(SP,NH,DF,ZM)
      IF (IP .EQ. 1) THEN
         AP = HR**2 / (8.0 * ZM)
         DO 200 I = 1,NH
              SP(I) = AP * SP(I)
200      CONTINUE
      ELSE
         HR = SQRT( 8.0 * ZM)
      ENDIF
C
      RETURN
      END
C
C FUNCTION SUBROUTINE TO GET JONSWAP FUNCTION (WITH GAMMA = 3.3)
      REAL FUNCTION EJ (FQ, FP)
C
      TWOPI = 8.0 * ATAN(1.0)
      GRAV = 9.810
      SIGA = 0.07
      SIGB = 0.09
      GAMMA = 3.3
C                                CALCULATE EP
      C1 = GRAV**2 * TWOPI**(-4)
      EP = C1 * FQ**(-5)
C                                CALCULATE PHIPM
      C = 1.25 * (FQ/FP)**(-4)
      IF (C .GT. 150) THEN
         C = 150
      ENDIF
      PHIPM = EXP(-C)
C                                JONSWAP SHAPE FUNCTION
      IF (FQ .LE. FP) THEN
         SIG = SIGA
      ELSE
         SIG = SIGB
      ENDIF
      C2 = (1.0/(2.0*SIG**2)) * (FQ/FP - 1.0)**2
      IF (C2 .GT. 150) THEN
         C2 = 150
      ENDIF
      PHIJ = GAMMA ** (EXP(-C2))
C                                JONSWAP SPECTRUM
      EJ = EP * PHIPM * PHIJ
C
      RETURN
      END
C
C     INTGRL - INTEGRATION ROUTINE BASED ON SIMPSON'S RULE
      SUBROUTINE INTGRL (F, N, DF, AREA)
C
      REAL      F(N)
C
```

```
      SE = F(2)
      SO = F(3)
      DO 100 I = 1, N/2-2
         SE = SE + F(2 + I*2)
         SO = SO + F(3 + I*2)
100   CONTINUE
      AREA = (DF/3.0) * (F(1) + 4.0*SE + 2.0*SO + F(N))
C
      RETURN
      END
```

SPCPAR Subroutine

```
C=======================================================================C
C     SPCPAR
C
C     COMPUTES SPECTRAL PARAMETERS
C
C        IN:
C     SP(NP/2+1)..INPUT SPECTRUM (L*L*S)
C     NP..........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C     DT..........TIME STEP (SAMPLING INTERVAL) (S)
C
C        OUT:
C     EP...........SPECTRAL WIDTH PARAMETER (EQ. 16)
C     VU...........SPECTRAL WIDTH PARAMETER (EQ. 18)
C     QP...........PEAKEDNESS PARAMETER (EQ. 19)
C     ER...........STANDARD DEVIATION OF FREE SURFACE OSCILLATION (L)
C     HR...........ROOT-MEAN-SQUARE WAVE HEIGHT (L)
C     HS...........SIGNIFICANT WAVE HEIGHT (L)
C     T1...........MEAN PERIOD BASED ON FIRST MOMENT (EQ. 24) (S)
C     T2...........MEAN PERIOD BASED ON SECOND MOMENT (EQ. 23) (S)
C=======================================================================C
      SUBROUTINE SPCPAR (SP, NP, DT, EP, VU, QP, ER, HR, HS, T1, T2)
C
      PARAMETER (NDS=16384)
      REAL      SP(NP/2+1), SM(5), FQ(NDS/2+1)
C
      NH = NP/2+1
      TM = NP * DT
      DF = 1.0/TM
C                                  SPECTRAL MOMENTS BY TRAPAZOIDS
      DO 25 I = 1, NH
         FQ(I) = (I-1) * DF
25    CONTINUE
      SUM = 0.5 * SP(1)
      DO 50 I = 2, NH-1
          SUM = SUM + SP(I)
50    CONTINUE
      SM(1) = (SUM + 0.5*SP(NH)) * DF
      DO 200 J = 1,4
         SUM = 0.0
         IF (J .NE. 3) THEN
              SUM = 0.0
              DO 100 I = 2, NH-1
                   SUM = SUM + SP(I) * (FQ(I))**J
100           CONTINUE
         ENDIF
         SM(J+1) = (SUM + 0.5*SP(NH)*(FQ(NH))**J) * DF
200   CONTINUE
      SUM = 0.0
      DO 300 I = 2, NH-1
         SUM = SUM + (SP(I)**2)*(I-1)
300   CONTINUE
```

```
      RESULT = (SUM + 0.5*(SP(NH)**2)*(NH-1)) * DF**2
C                                        SPECTRAL STATISTICS
      EP = SQRT(1 - SM(3)**2 / (SM(1) * SM(5)))
      VU = SQRT((SM(1) * SM(3) / SM(2)**2) - 1)
      QP = (2.0/SM(1)**2) * RESULT
      ER = SQRT(SM(1))
      HR = SQRT(8.0 * SM(1))
      HS = 4.004 * SQRT(SM(1))
      T1 = SM(1)/SM(2)
      T2 = SQRT(SM(1)/SM(3))
C
      RETURN
      END
```

TIMEPH Subroutine

```
C=====================================================================C
C     TIMEPH
C
C     COMPUTES TIME SERIES FOR GIVEN POWER DENSITY SPECTRUM USING
C     A RANDOM PHASE SCHEME
C
C        IN:
C     SP(NP/2+1)..POWER DENSITY SPECTRUM (L*L*S)
C     NP..........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C     DT..........TIME STEP (SAMPLING INTERVAL) (S)
C     IS..........SEED VALUE TO INITIALIZE RANDOM NUMBER GENERATOR
C
C        OUT:
C     TS(NP)......TIME SERIES SOLUTION (L)
C
C        EXTERNAL ROUTINES:
C     RDMGEN......RANDOM NUMBER GENERATOR
C     FFTIMSL.....INVERSE FOURIER TRANSFORM (IMSL) FOR IO= -1
C=====================================================================C
      SUBROUTINE TIMEPH (SP, NP, DT, IS, TS)
C
      PARAMETER (NDS=16384)
      REAL      SP(NP/2+1), TS(NP), PHI(NDS)
      COMPLEX   CN(NDS)
C
      TWOPI = 8.0 * ATAN(1.0)
      NH=NP/2+1
      TM=NP*DT
      DF = 1.0/TM
C                             GENERATE RANDOM NUMBERS FROM 0 TO 1
      CALL RDMGEN(NH-1,IS,PHI)
C                             FILL UP COMPLEX COEFFICIENTS
C                             AVERAGE VALUE
      CN(1) = CMPLX (0.0, 0.0)
C                             FIRST HALF OF ARRAY
      DO 500 I = 2, NH-1
         PHX = TWOPI * PHI(I-1)
         CX = SQRT(2.0*SP(I) * DF)
         CN(I) = 0.5 * CX * CMPLX(COS(PHX),SIN(PHX))
500   CONTINUE
C                             AT NYQUIST FREQUENCY
      PHX = TWOPI * PHI(NH-1)
      CX = SQRT(2.0*SP(NH)* DF)
      CN(NH) = CX * CMPLX(COS(PHX),0.0)
C                             SECOND HALF OF ARRAY
      DO 600 I = NH+1, NP
         NN = NP - I + 2
         PHX = TWOPI * PHI(NN-1)
         CX = SQRT(2.0*SP(NN)* DF)
         CN(I) = 0.5 * CX * CMPLX(COS(PHX),-SIN(PHX))
600   CONTINUE
```

```
C                                INVERSE FOURIER TRANSFORM                TIM00520
      CALL FFTIMSL(TS,CN,NP,-1)                                           TIM00530
C                                                                         TIM00540
      RETURN                                                              TIM00550
      END                                                                 TIM00560
```

TIMEDC **Subroutine**

```
C=====================================================================C
C     TIMEDC
C
C     DETERMINISTIC COEFFICIENT SCHEME TO COMPUTE TIME SERIES FOR
C     GIVEN FOURIER COEFFICIENTS
C
C        IN:
C     A(NP/2+1)....FOURIER COEFFICIENTS FOR COSINE (L)
C     B(NP/2+1)....FOURIER COEFFICIENTS FOR SINE (L)
C     NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C
C        OUT:
C     TS(NP).......TIME SERIES SOLUTION (L)
C
C        EXTERNAL ROUTINE:
C     FFTIMSL......INVERSE FOURIER TRANSFORM (IMSL) FOR IO= -1
C=====================================================================C
      SUBROUTINE TIMEDC (A, B, NP, TS)
C
      PARAMETER (NDS=16384)
      REAL      TS(NP), A(NP/2+1), B(NP/2+1)
      COMPLEX   CN(NDS)
C
      NH = NP/2+1
C                                AVERAGE VALUE SHOULD BE ZERO
      CN(1) = CMPLX(0.0, 0.0)
C                                FILL FIRST PART OF COMPLEX ARRAY
      DO 500 I = 2, NH-1
         CN(I) = 0.5 * CMPLX(A(I),-B(I))
500   CONTINUE
C                                AT NYQUIST FREQUENCY, B(NYQ) IS ZERO
      CN(NH) = CMPLX(A(NH), 0.0)
C                                FILL SECOND PART OF COMPLEX ARRAY
      DO 600 I = NH+1, NP
         CN(I) = 0.5 * CMPLX(A(NP-I+2),B(NP-I+2))
600   CONTINUE
C                                INVERSE TRANSFORM TO RETURN TIME SERIE
      CALL FFTIMSL(TS,CN,NP,-1)
C
      RETURN
      END
```

TIMPAR Subroutine

```
C=======================================================================C
C     TIMPAR
C
C     COMPUTES PARAMETERS FOR GIVEN TIME SERIES BY ZERO-UPCROSS METHOD
C
C        IN:
C     TS(NP).......TIME SERIES (L)
C     NP...........EVEN NUMBER OF DATA POINTS
C     DT...........TIME STEP (SAMPLING INTERVAL) (S)
C
C        OUT:
C     SD...........AVERAGE OF TS (SETUP OR SETDOWN) (L)
C     ER...........ROOT MEAN SQUARE OF FREE SURFACE (L)
C     NZ...........NUMBER OF ZERO-UPCROSSINGS
C     HB...........MEAN WAVE HEIGHT (L)
C     TB...........MEAN WAVE PERIOD (S)
C     HV...........ROOT-MEAN-SQUARE WAVE HEIGHT (L)
C     HS...........SIGNIFICANT WAVE HEIGHT (L) OF 1/3 HIGHEST WAVES
C     T3...........SIGNIFICANT WAVE PERIOD (S) OF 1/3 HIGHEST WAVES
C     HT...........MEAN HEIGHT OF ONE-TENTH HIGHEST WAVES (L)
C     TT...........MEAN PERIOD OF ONE-TENTH HIGHEST WAVES (S)
C     HRK(NZ)......RANKED WAVE HEIGHTS (L) WITH HRK(1) THE HIGHEST
C     TRK(NZ)......WAVE PERIODS (S) FOR RANKED WAVE HEIGHTS
C     LRN(NK)......RUN LENGTH OF WAVE HEIGHTS EXCEEDING HS
C     NK...........NUMBER OF RUNS
C=======================================================================C
      SUBROUTINE TIMPAR (TS, NP, DT, SD, ER, NZ, HB, TB, HV, HS, T3,
     &   HT, TT, HRK, TRK, LRN, NK)
C
      PARAMETER (NDS1=16385,NDZ=1000)
      REAL       ATS(NDS1)
      REAL       TS(NP), H(NDZ),HRK(NDZ),TRK(NDZ)
      REAL       TMZERO(NDZ), T(NDZ), TSMAX(NDZ), TSMIN(NDZ)
      INTEGER    IZERO(NDZ), JMAX(NDZ), JMIN(NDZ), IRK(NDZ), LRN(NDZ)
      LOGICAL    SORTED
C
      TM=NP*DT
C                                ADJUST TIME SERIES TO MAKE
C                                PERIODIC OVER TIME
      DO 10 I = 1,NP
         ATS(I)=TS(I)
10    CONTINUE
      ATS(NP+1)=TS(1)
C                                CORRECT FOR MEAN WATER LEVEL
      SUM = 0.0
      DO 20 I = 1, NP
         SUM = SUM + ATS(I)
20    CONTINUE
      SD = SUM / FLOAT(NP)
      DO 30 I = 1,NP+1
         ATS(I) = ATS(I) - SD
```

```
30    CONTINUE                                                          TIM00520
C                                        COMPUTE ROOT-MEAN-SQUARE OF ETA(T) TIM00530
      SUM= 0.0                                                          TIM00540
      DO 40 I = 1,NP                                                    TIM00550
        SUM= SUM+ (ATS(I)*ATS(I))                                       TIM00560
40    CONTINUE                                                          TIM00570
      ER = SQRT(SUM / FLOAT(NP))                                        TIM00580
C                                        ZERO-UPCROSSING POINTS         TIM00590
      NZ = 0                                                            TIM00600
      DO 200 J = 1,NP                                                   TIM00610
         IF (ATS(J) .EQ. 0.0) THEN                                      TIM00620
              NZ = NZ + 1                                               TIM00630
              IZERO(NZ) = J + 1                                         TIM00640
              TMZERO(NZ) = (J-1) * DT                                   TIM00650
         ELSEIF (ATS(J) .LT. 0.0 .AND. ATS(J+1) .GT. 0.0) THEN          TIM00660
              NZ = NZ + 1                                               TIM00670
              IZERO(NZ) = J + 1                                         TIM00680
              TMZERO(NZ) = (J-1)*DT + (-ATS(J)/(ATS(J+1)                TIM00690
     &             - ATS(J)))*DT                                        TIM00700
         ENDIF                                                          TIM00710
200   CONTINUE                                                          TIM00720
C                                        CALCULATE WAVE PERIOD, T, OF EACH WAVETIM00730
      DO 250 I = 1, NZ-1                                                TIM00740
         T(I) = TMZERO(I+1) - TMZERO(I)                                 TIM00750
250   CONTINUE                                                          TIM00760
      T(NZ) = (TM - TMZERO(NZ)) + TMZERO(1)                             TIM00770
C                                        NEED TO FIND TSMAX, TSMIN      TIM00780
      DO 350 I = 1, NZ                                                  TIM00790
         TSMAX(I) = 0.0                                                 TIM00800
         TSMIN(I) = 0.0                                                 TIM00810
         J1 = IZERO(I)                                                  TIM00820
         IF (I .EQ. NZ) THEN                                            TIM00830
              J2 = NP + IZERO(1) -1                                     TIM00840
         ELSE                                                           TIM00850
              J2 = IZERO(I+1) - 1                                       TIM00860
         ENDIF                                                          TIM00870
         DO 325 J = J1, J2                                              TIM00880
              IF (J .GT. NP) THEN                                       TIM00890
                   JT = J - NP                                          TIM00900
              ELSE                                                      TIM00910
                   JT = J                                               TIM00920
              ENDIF                                                     TIM00930
              IF (ATS(JT) .GT. TSMAX(I)) THEN                           TIM00940
                   TSMAX(I) = ATS(JT)                                   TIM00950
                   JMAX(I) = JT                                         TIM00960
              ENDIF                                                     TIM00970
              IF (ATS(JT) .LT. TSMIN(I)) THEN                           TIM00980
                   TSMIN(I) = ATS(JT)                                   TIM00990
                   JMIN(I) = JT                                         TIM01000
              ENDIF                                                     TIM01010
325     CONTINUE                                                        TIM01020
350   CONTINUE                                                          TIM01030
C                                        IMPROVE ESTIMATES W/ PARABOLIC CURVE  TIM01040
```

```
      DO 400 I = 1, NZ
         J = JMAX(I)
         J1 = J - 1
         IF (J1 .LT. 1) THEN
              J1 = NP
         ENDIF
         TS1 = ATS(J1)
         TS2 = ATS(J)
         TS3 = ATS(J+1)
         TSMAX(I) = TS2 - (TS3 -TS1)**2 / (8.0 * (TS1 - 2*TS2
     &        + TS3))
         J = JMIN(I)
         J1 = J - 1
         IF (J1 .LT. 1) THEN
              J1 = NP
         ENDIF
         TS1 = ATS(J1)
         TS2 = ATS(J)
         TS3 = ATS(J+1)
         TSMIN(I) = TS2 - (TS3 -TS1)**2 / (8.0 * (TS1 - 2*TS2
     &        + TS3))
C                                WAVE HEIGHT, H, OF EACH WAVE
         H(I) = TSMAX(I) - TSMIN(I)
400   CONTINUE
C                                STATISTICS OF INDIVIDUAL WAVE HEIGHTS
      SUM = 0.0
      SUMHB = 0.0
      SUMHV = 0.0
      DO 450 I = 1, NZ
         SUM = SUM + T(I)
         SUMHB = SUMHB + H(I)
         SUMHV = SUMHV + H(I)*H(I)
450   CONTINUE
      TB = SUM / FLOAT(NZ)
      HB = SUMHB / FLOAT(NZ)
      HV = SQRT(SUMHV / FLOAT(NZ))
C SORTING ROUTINE FOR WAVE HEIGHT RANKING.
C                                SET UP HRANK, TRANK, IRANK ARRAYS
      DO 500 I = 1, NZ
         IRK(I) = I
         HRK(I) = H(I)
         TRK(I) = T(I)
500   CONTINUE
      SORTED = .FALSE.
      IPASS = 0
550   IF (.NOT. SORTED) THEN
         IPASS = IPASS + 1
         SORTED = .TRUE.
         DO 600 I = 1, NZ - IPASS
              IF (H(IRK(I)) .LT. H(IRK(I+1))) THEN
                   ITEMP = IRK(I)
                   IRK(I) = IRK(I+1)
                   IRK(I+1) = ITEMP
```

```
                  HTEMP = HRK(I)                                          TIM01580
                  HRK(I) = HRK(I+1)                                       TIM01590
                  HRK(I+1) = HTEMP                                        TIM01600
                  TTEMP = TRK(I)                                          TIM01610
                  TRK(I) = TRK(I+1)                                       TIM01620
                  TRK(I+1) = TTEMP                                        TIM01630
                  SORTED = .FALSE.                                        TIM01640
               ENDIF                                                      TIM01650
600         CONTINUE                                                      TIM01660
            GOTO 550                                                      TIM01670
         ENDIF                                                            TIM01680
C                                REPRESENTATIVE WAVE HEIGHT AND PERIOD    TIM01690
         ITHIRD = NZ/3                                                    TIM01700
         ITENTH = NZ /10                                                  TIM01710
         HS = 0.0                                                         TIM01720
         T3 = 0.0                                                         TIM01730
         DO 650 I = 1, ITHIRD                                             TIM01740
            HS = HS + HRK(I)                                              TIM01750
            T3 = T3 + TRK(I)                                              TIM01760
650      CONTINUE                                                         TIM01770
         HS = HS/FLOAT(ITHIRD)                                            TIM01780
         T3 = T3/FLOAT(ITHIRD)                                            TIM01790
         HT = 0.0                                                         TIM01800
         TT = 0.0                                                         TIM01810
         DO 700 I = 1,ITENTH                                              TIM01820
            HT = HT + HRK(I)                                              TIM01830
            TT = TT + TRK(I)                                              TIM01840
700      CONTINUE                                                         TIM01850
         HT = HT/FLOAT(ITENTH)                                            TIM01860
         TT = TT/FLOAT(ITENTH)                                            TIM01870
C                              RUN LENGTH OF WAVE HEIGHTS EXCEEDING HS    TIM01880
         NK = 0                                                           TIM01890
         NCOUNT = 0                                                       TIM01900
         I = 1                                                            TIM01910
750      IF (I .LE. NZ) THEN                                              TIM01920
            IF (H(I) .GT. HS) THEN                                        TIM01930
               NK = NK + 1                                                TIM01940
               NCOUNT = NCOUNT + 1                                        TIM01950
725            IF (H(I+NCOUNT) .GT. HS) THEN                              TIM01960
                  NCOUNT = NCOUNT + 1                                     TIM01970
                  GOTO 725                                                TIM01980
               ENDIF                                                      TIM01990
               LRN(NK) = NCOUNT                                           TIM02000
               I = I + NCOUNT                                             TIM02010
               NCOUNT = 0                                                 TIM02020
            ENDIF                                                         TIM02030
            I = I + 1                                                     TIM02040
            GOTO 750                                                      TIM02050
         ENDIF                                                            TIM02060
C                                                                         TIM02070
         RETURN                                                           TIM02080
         END                                                              TIM02090
```

SPCTRA Subroutine

```
C=======================================================================
C     SPCTRA
C
C     COMPUTES SMOOTHED AND UNSMOOTHED SPECTRA FOR GIVEN TIME SERIES
C
C        IN:
C     TS(NP).......TIME SERIES TO BE TRANSFORMED (L)
C     NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C     DT...........TIME STEP (SAMPLING INTERVAL) (S)
C     NB...........NUMBER OF DATA POINTS IN EACH BAND FOR SMOOTHING
C                  FOR NB=1, NO SMOOTHING
C
C        OUT:
C     SP(NP/2+1)...UNSMOOTHED SPECTRUM (L*L*S)
C     FS(NP/2/NB)..FREQUENCY OF SMOOTHED SPECTRUM (HZ)
C     SM(NP/2/NB)..SMOOTHED SPECTRUM (L*L*S)
C
C        EXTERNAL ROUTINE:
C     FFTIMSL......RETURNS COMPLEX FOURIER COEFFICIENTS FOR IO=1
C=======================================================================
      SUBROUTINE SPCTRA (TS, NP, DT, NB, SP, FS, SM)
C
      PARAMETER (NDS=16384)
      REAL       TS(NP), SP(NP/2+1), SM(NP/2/NB), FS(NP/2/NB)
      COMPLEX    CN(NDS)
C
      NH = NP/2+1
      TM=NP*DT
      DF = 1.0/TM
C                                     FAST FOURIER TRANSFORM
      CALL FFTIMSL(TS,CN,NP,+1)
C                                     POWER SPECTRAL DENSITY
      SP(1) = 0.0
      DO 400 I = 2,NH-1
         A = 2.*REAL(CN(I))
         B = -2.*AIMAG(CN(I))
         SP(I) = 1.0/(2.0*DF) * (A**2 + B**2)
400   CONTINUE
         A = REAL(CN(NH))
         SP(NH) = 1.0/(2.0*DF) * (A**2)
C                                     BAND AVERAGE SMOOTHING
      IF (NB.GT.1)THEN
         FBS=FLOAT(NB)*DF
         NS=(NH-1)/NB
         FS(1)=DF/2. + FBS/2.
         DO 10 K = 1,NS
              IF(K.GT.1) FS(K)=FS(1) + (K-1)*FBS
              JB=(K-1)*NB + 2
              JE=K*NB + 1
              SUM = 0.0
              DO 5 J = JB,JE
```

```
            SUM=SUM+SP(J)
5         CONTINUE
          SM(K) = SUM/FLOAT(NB)
10      CONTINUE
      ENDIF
C
      RETURN
      END
```

IRSORT Subroutine

```
C=========================================================================
C      IRSORT
C
C      SEPARATES INCIDENT AND REFLECTED WAVE TRAINS USING THREE WAVE
C      GAGES AND METHOD OF GODA+SUZUKI
C
C         IN:
C      TS(ND,NW)....FREE SURFACE OSCILLATIONS (AT NW GAGES) (L)
C      ND...........DIMENSION OF TS IN CALLING PROGRAM
C      NW...........WIDTH OF TIME SERIES ARRAY (EQUAL TO NO. OF GAGES)
C      NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C      DT...........TIME STEP (SAMPLING INTERVAL) (S)
C      XG(NW).......LOCATION OF EACH GAGE WITH X-AXIS POSITIVE SHOREWARD
C                   AND GAGE NUMBER DECREASING SHOREWARD (M)
C      DH...........WATER DEPTH (M)
C
C         OUT:
C      FMN..........MINUMUM RESOLVABLE FREQUENCY (HZ) BASED ON LARGEST
C                   GAGE SPACING
C      FMX..........MAXIMUM RESOLVABLE FREQUENCY (HZ) BASED ON SMALLEST
C                   GAGE SPACING
C      TI(NP).......INCIDENT TIME SERIES (L)
C      TR(NP).......REFLECTED TIME SERIES (L)
C
C         EXTERNAL ROUTINES:
C      FFTIMSL......FAST FOURIER TRANSFORM
C      WAVNUM.......WAVE NUMBER
C      TIMEDC.......RETURN TIME SERIES FOR KNOWN FOURIER COEFFICIENTS
C=========================================================================
      SUBROUTINE IRSORT (TS, ND, NW, NP, DT, XG, DH, FMN, FMX, TI, TR)
C
      PARAMETER (NDS=16384,NWS=3)
      REAL      TS(ND,NW), XG(NW), TI(NP),TR(NP)
      REAL      AI(NDS/2+1),AR(NDS/2+1),BI(NDS/2+1),BR(NDS/2+1)
      REAL      A(NDS/2+1,NWS),B(NDS/2+1,NWS), XI(NWS,NWS), XJ(NWS,NWS)
      COMPLEX   CN(16384)
      LOGICAL   DONE
C
      TM=NP*DT
      DF=1.0/TM
      NH=NP/2+1
      GRAV=9.81
      PI=4.0*ATAN(1.0)
      CMIN=0.1*PI
      CMAX=0.9*PI
C
C                                  SET THE CORRECT GAGE LOCATION SO THAT
C XI(I,J) IS THE LOCATION OF GAGE I SHOREWARD OF GAGE J AND XJ(I,J) IS
C THE LOCATION OF GAGE J SEAWARD OF GAGE I AND XI(I,J)-XJ(I,J) > 0
      DO 10 I=1,NW
        DO 5 J=1,NW
            XI(I,J) = 0.0
```

```
              XJ(I,J) = 0.0
5           CONTINUE
10      CONTINUE
        XI(1,2) = XG(1)
        XJ(1,2) = XG(2)
        IF(NW.GT.2)THEN
        XI(1,3) = XG(1)
        XJ(1,3) = XG(3)
        XI(2,3) = XG(2)
        XJ(2,3) = XG(3)
        ENDIF
C                                       GET FOURIER COEFFICIENTS
        DO 550 I = 1, NW
           CALL FFTIMSL(TS(1,I),CN,NP,+1)
           A(1,I) = 0.0
           B(1,I) = 0.0
           DO 525 K = 2,NH-1
                A(K,I) = 2.*REAL(CN(K))
                B(K,I) = -2.*AIMAG(CN(K))
525        CONTINUE
           A(NH,I) = REAL(CN(NH))
           B(NH,I) = 0.0
550     CONTINUE
C                                       LOOP TO SORT INC. AND REFL. A'S,B'S
        DO 910 L = 2, NH
           FQ = (L-1) * DF
           CALL WAVNUM(FQ,DH,WVNM)
           KOUNT = 0
           AI(L) = 0.0
           AR(L) = 0.0
           BI(L) = 0.0
           BR(L) = 0.0
           DO 880 I = 1, NW
                DO 870 J = 1, NW
                     IF (I .LT. J) THEN
C                                       CRITERION FOR .1PI < KX < .9PI
                          ARGIS = WVNM * (XI(I,J) - XJ(I,J))
                          IF (ARGIS.GE.CMIN.AND.ARGIS.LE.CMAX) THEN
                               KOUNT = KOUNT + 1
                               SI = SIN(WVNM * XI(I,J))
                               SJ = SIN(WVNM * XJ(I,J))
                               CI = COS(WVNM * XI(I,J))
                               CJ = COS(WVNM * XJ(I,J))
                               D1 = 0.5 / SIN(ARGIS)
                               D2 = A(L,I) * SJ
                               D3 = A(L,J) * SI
                               D4 = A(L,I) * CJ
                               D5 = A(L,J) * CI
                               D6 = B(L,I) * SJ
                               D7 = B(L,J) * SI
                               D8 = B(L,I) * CJ
                               D9 = B(L,J) * CI
                               AI(L) = AI(L) + D1*(-D2+D3+D8-D9)
```

```
                         BI(L) = BI(L) + D1*(+D4-D5+D6-D7)
                         AR(L) = AR(L) + D1*(-D2+D3-D8+D9)
                         BR(L) = BR(L) + D1*(+D4-D5-D6+D7)
                     ENDIF
                 ENDIF
870          CONTINUE
880      CONTINUE
         IF (KOUNT .NE. 0) THEN
             AI(L) =       AI(L) / FLOAT(KOUNT)
             BI(L) = -1 * BI(L) / FLOAT(KOUNT)
             AR(L) =       AR(L) / FLOAT(KOUNT)
             BR(L) =       BR(L) / FLOAT(KOUNT)
         ENDIF
910   CONTINUE
C                                   SET ZERO-TH HARMONIC TO ZERO
      AI(1) = 0.0
      BI(1) = 0.0
      AR(1) = 0.0
      BR(1) = 0.0
C                                   RESOLVABLE FREQUENCY RANGE
C                                   FIND MAX AND MIN GAGE PAIR
      XMIN=XG(1)-XG(2)
      XMAX=XMIN
      IF(NW.GT.2)THEN
         XMAX=XG(1)-XG(3)
         IF(XG(2)-XG(3) .LT. XMIN) XMIN=XG(2)-XG(3)
      ENDIF
C                                   FMN IS FROM LARGEST GAGE PAIR
      WNMIN = CMIN/XMAX
      FMN = SQRT(WNMIN*GRAV*TANH(WNMIN*DH)) / (2.*PI)
C                                   FMX IS FROM SMALLEST GAGE PAIR
      WNMAX = CMAX/XMIN
      FMX = SQRT(WNMAX*GRAV*TANH(WNMAX*DH)) / (2.*PI)
C                                   TO GET TIME SERIES AT X=0.0
      CALL TIMEDC(AI,BI,NP,TI)
      CALL TIMEDC(AR,BR,NP,TR)
C
      RETURN
      END
```

COHPHS Subroutine

```
C=====================================================================CCOH00010
C     COHPHS                                                            COH00020
C                                                                       COH00030
C     COMPUTES COHERENCE SQUARED AND PHASE OF CROSS SPECTRUM            COH00040
C     BETWEEN TWO GIVEN TIME SERIES                                     COH00050
C                                                                       COH00060
C        IN:                                                            COH00070
C     TS1(NP)......FIRST TIME SERIES (L)                                COH00080
C     TS2(NP)......SECOND TIME SERIES (L)                               COH00090
C     NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES            COH00100
C     DT...........TIME STEP (SAMPLING INTERVAL) (S)                    COH00110
C     NB...........NUMBER OF DATA POINTS IN BAND WIDTH FOR SMOOTHING    COH00120
C                                                                       COH00130
C        OUT:                                                           COH00140
C     FS(NP/2/NB)..SMOOTHED FREQUENCY ARRAY (HZ)                        COH00150
C     CH(NP/2/NB)..SMOOTHED COHERENCE SQUARED (1,2)                     COH00160
C     PH(NP/2/NB)..SMOOTHED PHASE OF CROSS SPECTRUM (1,2) (DEGREES)     COH00170
C                                                                       COH00180
C        EXTERNAL ROUTINE:                                              COH00190
C     FFTIMSL......FOURIER TRANSFORM (IMSL)                             COH00200
C=====================================================================CCOH00210
      SUBROUTINE COHPHS (TS1, TS2, NP, DT, NB, FS, CH, PH)              COH00220
C                                                                       COH00230
      PARAMETER (NDS=16384)                                             COH00240
      REAL       TS1(NP), TS2(NP), FS(NP/2/NB), CH(NP/2/NB), PH(NP/2/NB) COH00250
      REAL       AS1(NDS/2+1), AS2(NDS/2+1), BU(NDS/2+1)                COH00260
      COMPLEX    XS(NDS/2+1), CN1(NDS), CN2(NDS)                        COH00270
C                                                                       COH00280
      NH = NP/2+1                                                       COH00290
      PI = 4.0*ATAN(1.0)                                                COH00300
C                                   COMPLEX FOURIER COEFFICIENTS        COH00310
      CALL FFTIMSL(TS1,CN1,NP,+1)                                       COH00320
      CALL FFTIMSL(TS2,CN2,NP,+1)                                       COH00330
C                                   AUTO SPECTRA AND CROSS SPECTRUM     COH00340
      DO 10 I = 1,NH                                                    COH00350
         AS1(I) = CN1(I)*CONJG(CN1(I))                                  COH00360
         AS2(I) = CN2(I)*CONJG(CN2(I))                                  COH00370
         XS(I) = CONJG(CN1(I))*CN2(I)                                   COH00380
10    CONTINUE                                                          COH00390
C                                   SMOOTH THE SPECTRA                  COH00400
      CALL SMOOTH(AS1,NP,DT,NB,FS,AS1)                                  COH00410
      CALL SMOOTH(AS2,NP,DT,NB,FS,AS2)                                  COH00420
      CALL SMTHCX(XS,NP,NB,XS)                                          COH00430
C                                   COMPUTE COHERENCE AND PHASE         COH00440
      DO 110 I = 1,NP/2/NB                                              COH00450
         RDUM = CABS(XS(I)*CONJG(XS(I)))                                COH00460
         CH(I) = RDUM/(AS1(I)*AS2(I))                                   COH00470
         XSR = REAL(XS(I))                                              COH00480
         XSI = AIMAG(XS(I))                                             COH00490
         XPH = ATAN2(XSI,XSR)                                           COH00500
         PH(I) = 180.*XPH/PI                                            COH00510
```

```
110   CONTINUE
C
      RETURN
      END
C                                  SMOOTHING SPECTRUM
      SUBROUTINE SMOOTH (SP, NP, DT, NB, FS, SM)
C
      REAL      SP(NP/2+1),FS(NP/2/NB),SM(NP/2/NB)
C
      NH = NP/2+1
      TM=NP*DT
      DF=1./TM
      FBS=FLOAT(NB)*DF
      NS=(NH-1)/NB
      FS(1)=DF/2. + FBS/2.
      DO 10 K = 1,NS
         IF(K.GT.1) FS(K)=FS(1) + (K-1)*FBS
         JB=(K-1)*NB + 2
         JE=K*NB + 1
         SUM = 0.0
         DO 5 J = JB,JE
            SUM=SUM+SP(J)
5        CONTINUE
         SM(K) = SUM/FLOAT(NB)
10    CONTINUE
      RETURN
      END
C                                  SMOOTHING COMPLEX SPECTRUM
      SUBROUTINE SMTHCX (XU, NP, NB, XS)
C
      COMPLEX   XU(NP/2+1), XS(NP/2/NB), COSUM
C
      NH = NP/2+1
      NS=(NH-1)/NB
      DO 10 K = 1,NS
         JB=(K-1)*NB + 2
         JE=K*NB + 1
         COSUM = (0.0,0.0)
         DO 5 J = JB,JE
            COSUM=COSUM+XU(J)
5        CONTINUE
         XS(K) = COSUM/FLOAT(NB)
10    CONTINUE
C
      RETURN
      END
```

DISTNR Subroutine

```
C=======================================================================C
C     DISTNR
C
C     TO CHECK WHETHER TIME SERIES FOLLOWS NORMAL DISTRIBUTION AND
C     WHETHER EXCEEDANCE PROBABILITY OF WAVE HEIGHTS FOLLOWS RAYLEIGH
C     DISTRIBUTION
C
C        IN:
C     TS(NP).......TIME SERIES (L)
C     NP...........NUMBER OF DATA POINTS
C     DT...........TIME STEP (SAMPLING INTERVAL) (S)
C     XMN..........MINIMUM VALUE FOR FREE SURFACE DISPLACEMENT (L)
C     XMX..........MAXIMUM VALUE FOR FREE SURFACE DISPLACEMENT (L)
C     DX...........INCREMENT FOR DISCRETE PROBABILITY DENSITY FUNCTION
C                  (L) FOR RESOLUTION NDX=(XMX-XMN)/DX
C
C        OUT:
C     SD...........MEAN (L) CORRESPONDS TO SET DOWN OR SET UP
C     VAR..........VARIANCE (L*L)
C     SKW..........SKEWNESS
C     F(NDX).......FREE SURFACE ARRAY ("BINS")
C     XP(NDX)......PROBABILITY DENSITY FUNCTION OF TIME SERIES
C     XN(NDX)......NORMAL DISTRIBUTION
C     NDX..........LENGTH OF ARRAYS FOR NORMAL DISTIBUTION
C     HS...........SIGNIFICANT WAVE HEIGHT (L) FOR 1/3 HIGHEST WAVES
C     G(NZ)........HP/HS VALUES
C     PE(NZ).......EXCEEDANCE PROBABILITY PE(N)=N/(NZ+1)
C     PR(NZ).......EXCEEDANCE BASED ON RAYLEIGH DISTRIBUTION
C     NZ...........LENGTH OF ARRAYS FOR RAYLEIGH DISTRIBUTION
C
C        ADDITIONAL ROUTINES:
C     TIMPAR.......TO RETURN WAVE HEIGHT RANKINGS
C=======================================================================C
      SUBROUTINE DISTNR (TS,NP,DT,XMN,XMX,DX,SD,VAR,SKW,F,XP,XN,NDX,
     &   HS,G,PE,PR,NZ)
C
      PARAMETER (NDS1=16385,NDS=1000)
      DIMENSION ATS(NDS1),TS(NP), XP(NDS),XN(NDS),F(NDS)
      DIMENSION PE(NDS),PR(NDS),G(NDS)
      DIMENSION HRK(NDS),TRK(NDS)
      DIMENSION LRN(NDS)
C
      TWOPI=8.0*ATAN(1.0)
C                                 COPY TS AND REMOVE MEAN FROM ATS
      SUM = 0.0
      DO 10 I = 1,NP
         ATS(I) = TS(I)
10    CONTINUE
C                                 CORRECT FOR MEAN WATER LEVEL
      SUM = 0.0
      DO 20 I = 1,NP
```

```
         SUM = SUM + ATS(I)
 20   CONTINUE
      SD = SUM / FLOAT(NP)
      DO 30 I = 1,NP
         ATS(I) = ATS(I) - SD
 30   CONTINUE
C                                 COMPUTE VARIANCE
      SUM= 0.0
      DO 40 I = 1,NP
         SUM= SUM+ (ATS(I)*ATS(I))
 40   CONTINUE
      VAR = SUM / FLOAT(NP)
C                                 COMPUTE SKEWNESS
      SUM= 0.0
      DO 45 I = 1,NP
         SUM= SUM + (ATS(I))**3
 45   CONTINUE
      ERMS=SQRT(VAR)
      SKW= SUM/ FLOAT(NP) /ERMS**3
C                                 PROBABILITY DENSITY FUNCTION
      NDX = (XMX-XMN)/DX
      DO 60 J = 1,NDX
         KOUNT=0
         F(J) = XMN+(J-1)*DX+DX/2.0
         DO 50 I = 1,NP
              IF (TS(I).GT.XMN+(J-1)*DX .AND. TS(I).LE.XMN+J*DX) THEN
              KOUNT=KOUNT+1
              ENDIF
 50      CONTINUE
         IF(KOUNT.EQ.0)THEN
              XP(J) = 0.0
         ELSE
              XP(J) = FLOAT(KOUNT)/FLOAT(NP)/DX
         ENDIF
 60   CONTINUE
C                                 NORMAL DISTRIBUTION
      C1=1.0/SQRT(TWOPI*VAR)
      DO 70 I=1,NDX
         XN(I)=C1*EXP((-(F(I)-SD)**2)/(2.*VAR))
 70   CONTINUE
C                                 CALL TIMPAR TO GET WAVE HEIGHT DIST
C                                 DO NOT WANT NEW "SD" SO USE "XX"
      CALL TIMPAR(ATS,NP,DT,XX,ER,NZ,HB,TB,HV,HS,T3,HT,TT,HRK,TRK,LRN,
     &  NK)
C                                 COMPUTE EXCEEDANCE PROBABILITY
      DO 80 I = 1, NZ
         PE(I)=FLOAT(I)/(FLOAT(NZ+1))
 80   CONTINUE
C                                 COMPUTE RAYLEIGH DISTRIBUTION
      DO 90 I = 1, NZ
         HRAT=HRK(I)/HS
         G(I) = HRAT
         PR(I)=EXP(-2.*HRAT*HRAT)
```

```
 90   CONTINUE                                                          DIS01050
C                                                                       DIS01060
      RETURN                                                            DIS01070
      END                                                               DIS01080
```

USRSPC **Subroutine**

```
C=======================================================================C
C      USRSPC
C
C      COMPUTES SPECTRAL DENSITY AT EQUALLY-SPACED DISCRETE FREQUENCIES
C      FROM USER-SPECIFIED RAW POINTS
C
C         IN:
C      NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C      DT...........TIME STEP (SAMPLING INTERVAL) (S)
C      NS...........NUMBER OF LINEAR SEGMENTS SPECIFYING THE SPECTRUM
C      FR...........ARRAY OF LENGTH (NS+1) CONTAINING THE ABSCISSAS OF
C                   THE RAW POINTS (HZ)
C      SR...........ARRAY OF LENGTH (NS+1) CONTAINING THE ORDINATES OF
C                   THE RAW POINTS (L*L*S)
C
C         OUT:
C      SP(NP/2+1)...SPECTRAL ARRAY (L*L*S)
C                   WHERE SP(1)=0 CORRESONDS TO FREQUENCY=0
C=======================================================================C
      SUBROUTINE USRSPC (NP, DT, NS, FR, SR, SP)
C
      REAL SP(NP/2+1)
      REAL FR(NS+1),SR(NS+1)
C
      NH = NP/2+1
      TM = REAL(NP)*DT
      DF = 1.0/TM
      NI = INT(FR(NS+1)/DF)
C                                  COMPUTE FINE POINTS
      K      = 1
      SLOPE  = (SR(K+1)-SR(K))/(FR(K+1)-FR(K))
      FRIGHT = FR(K+1)
C                                  ENFORCED: SP(1)=0.0
      SP(1) = 0.0
C                                  INTERPOLATION OF RAW POINTS
      DO 120 I = 2,NI+1
        FREQ = REAL(I-1)*DF
        IF (FREQ.GT.FRIGHT) THEN
          K      = K+1
          FRIGHT = FR(K+1)
          SLOPE  = (SR(K+1)-SR(K))/(FR(K+1)-FR(K))
        ENDIF
        SP(I) = SR(K) + (FREQ-FR(K))*SLOPE
  120 CONTINUE
C                                  ASSIGN ZERO TO SP(NI+1) TO SP(NH)
      DO 130 I = NI+2,NH
        SP(I) = 0.0
  130 CONTINUE
C
      RETURN
      END
```

PRORBR **Subroutine**

```
C=======================================================================C
C     PRORBR
C
C     CREATES AN OUTPUT FILE CONTAINING AN INPUT WAVE TRAIN FOR RBREAK
C     BASED ON A GIVEN DIMENSIONAL, PERIODICAL TIME SERIES
C
C        IN:
C     TS(NP).......TIME SERIES (L)
C     NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C     DT...........TIME STEP (SAMPLING INTERVAL) (S)
C     FNAME........NAME OF THE OUTPUT FILE CONTAINING THE INPUT WAVE
C                  TRAIN FOR RBREAK
C     IP...........OPTION TO SPECIFY THE REFERENCE WAVE HEIGHT USED
C                  FOR THE NORMALIZATION OF THE DIMENSIONAL TIME
C                  SERIES TS AS FOLLOWS:
C                  . SIGNIFICANT WAVE HEIGHT BASED ON THE TIME SERIES
C                    (IP=1 and HW is returned as the significant wave
C                    height)
C                  . SPECTRAL ESTIMATE OF THE SIGNIFICANT WAVE HEIGHT
C                    (IP=2 and HW is returned as the spectral estimate
C                    of the significant wave height)
C                  . A USER-SPECIFIED REFERENCE WAVE HEIGHT
C                    (IP=3 and HW needs to be specified as input to
C                    the PRORBR subroutine)
C
C        IN/OUT:
C     HW...........REFERENCE WAVE HEIGHT USED FOR THE NORMALIZATION OF
C                  THE DIMENSIONAL TIME SERIES TS (L)
C
C        OUT:
C     JO...........MARKS THE LOCATION IN THE ORIGINAL TIME SERIES OF
C                  THE DATA POINT THAT WILL BE ASSIGNED TO THE FIRST
C                  DATA POINT IN THE INPUT WAVE TRAIN FOR RBREAK
C     TJO..........TIME (S) IN THE ORIGINAL TIME SERIES CORRESPONDING
C                  TO THE INDEX JO
C
C        EXTERNAL ROUTINES:
C     TIMPAR.......COMPUTES THE SIGNIFICANT WAVE HEIGHT BASED ON THE
C                  TIME SERIES FOR IP=1
C     SPCTRA.......TRANSFORMS THE ORIGINAL TIME SERIES TO THE
C                  CORRESPONDING POWER DENSITY SPECTRUM FOR IP=2
C     SPCPAR.......COMPUTES THE SPECTRAL ESTIMATE OF THE SIGNIFICANT
C                  WAVE HEIGHT FOR IP=2
C=======================================================================C
      SUBROUTINE PRORBR (TS, NP, DT, FNAME, IP, HW, JO, TJO)
C
      PARAMETER (NMAX=20001,NDZ=1000,TINY=1.E-06,SMALL=1.E-03)
      CHARACTER*10 FNAME
      REAL     TS(NP),TI(NMAX)
      REAL     SP(NMAX/2+1),FS(NMAX/2),SM(NMAX/2)
      REAL     HRK(NDZ),TRK(NDZ)
```

```
      INTEGER  LRN(NDZ)
C                                 CHECK THE OPTION IP AND
C                                 DETERMINE REFERENCE WAVE HEIGHT
      IF (IP.LT.1.OR.IP.GT.3) THEN
        WRITE (*,2010)
        STOP
      ELSE
        IF (IP.EQ.1) THEN
          CALL TIMPAR(TS,NP,DT,SD,ER,NZ,HB,TB,HV,HS,T3,
     +                HT,TT,HRK,TRK,LRN,NK)
          HW = HS
        ELSEIF (IP.EQ.2) THEN
          NB = 1
          CALL SPCTRA(TS,NP,DT,NB,SP,FS,SM)
          CALL SPCPAR(SP,NP,DT,EP,VU,QP,ER,HR,HS,T1,T2)
          HW = HS
        ELSE
          IF (HW.LT.TINY) THEN
            WRITE (*,2020)
            STOP
          ENDIF
        ENDIF
      ENDIF
C                                 FIND THE LOCATION J0
      J0 = -1
      J  =  1
  900 IF (J0.EQ.-1) THEN
        J = J+1
        IF (TS(J).LT.0.D+00.AND.TS(J-1).GT.0.D+00) THEN
          VALUE = TS(J)/HW
          IF (ABS(VALUE).LT.SMALL) J0=J
        ENDIF
      GOTO 900
      ENDIF
      TJ0 = REAL(J0-1)*DT
C                                 CONSTRUCT AND STORE THE
C                                 INPUT WAVE TRAIN FOR RBREAK
      NP1 = NP+1
      I0  = NP-J0+1
      DO 100 I = 1,NP1
        IF (I.LE.I0) THEN
          TI(I) = TS(I+J0-1)/HW
        ELSE
          TI(I) = TS(I-I0)/HW
        ENDIF
  100 CONTINUE
C
      OPEN (UNIT=90,FILE=FNAME,STATUS='NEW',ACCESS='SEQUENTIAL')
      WRITE (90,9000) NP1
      WRITE (90,8000) (TI(I),I=1,NP1)
C                                 FORMATS
 2010 FORMAT (' Error message from PRORBR: Option out of range.'/
     +        ' Program aborted by PRORBR.')
```

```
 2020 FORMAT (' Error message from PRORBR: Must specify HW for IP=3.'/  PR001050
     +         ' Program aborted by PRORBR.')                          PR001060
 8000 FORMAT (5D15.6)                                                  PR001070
 9000 FORMAT (I8)                                                      PR001080
C                                                                      PR001090
      RETURN                                                           PR001100
      END                                                              PR001110
```

FFTIMSL Subroutine

```
C=====================================================================C
C     FFTIMSL
C
C     FAST FOURIER TRANSFOMR USING IMSL ROUTINES
C
C        IN/OUT:
C     TS(NP).......TIME SERIES (L)
C     CN(NP).......COMPLEX FOURIER COEFFICIENTS (L)
C
C        IN:
C     NP...........NUMBER OF DATA POINTS
C     IO...........+1 THEN FOURIER TRANS OF TS AND CN RETURNED
C                  -1 THEN INVERSE TRANS OF CN AND TS RETURNED
C
C        EXTERNAL ROUTINE:
C     FFT2D........FAST FOURIER TRANSFORM (IMSL)
C     FFT2B........INVERSE FAST FOURIER TRANSFORM (IMSL)
C=====================================================================C
      SUBROUTINE FFTIMSL (TS, CN, NP, IO)
C
      PARAMETER (NDS=16384)
      REAL       TS(NP)
      COMPLEX    CN(NP)
      COMPLEX    CTS(NDS,1), COEF(NDS,1), AFFT(NDS,1)
C                                IF IO IS +1 THEN FFT OF TIME SERIES
      IF (IO .EQ. 1) THEN
C                                CHANGE TO 2-D ARRAY FOR IMSL
      DO 100 I = 1,NP
         CTS(I,1) = CMPLX(TS(I), 0.0)
100   CONTINUE
      NRA = NP
      NCA = 1
      LDA = NDS
      LDCOEF = NDS
      CALL FFT2D (NRA, NCA, CTS, LDA, COEF, LDCOEF)
C
      DO 200 I = 1,NP
         CN(I) = 1.0/FLOAT(NP) * COEF(I,1)
200   CONTINUE
C                                IF IO IS -1 THEN INVERSE FFT OF CN'S
      ELSEIF (IO .EQ. -1) THEN
C                                CHANGE TO 2-D ARRAY FOR IMSL
      DO 300 I = 1,NP
         COEF(I,1) = CN(I)
300   CONTINUE
      NRCOEF = NP
      NCCOEF = 1
      LDCOEF = NDS
      LDA = NDS
      CALL FFT2B(NRCOEF, NCCOEF, COEF, LDCOEF, AFFT, LDA)
C                                TAKE REAL PART FOR TIME SERIES
```

```
      DO 400 I = 1, NP
         TS(I) = REAL(AFFT(I,1))
400   CONTINUE
      ENDIF
C
      RETURN
      END
```

RDMGEN Subroutine

```
C======================================================================C
C     RDMGEN
C
C     RANDOM NUMBER GENERATOR
C
C        IN:
C     NR..........NUMBER OF RANDOM NUMBERS GENERATED
C     IS..........SEED VALUE TO INITIALIZE RANDOM NUMBER GENERATOR
C
C        OUT:
C     AR(NR)......RANDOM NUMBERS DISTRIBUTED UNIFORMLY 0 TO 1.0
C
C        EXTERNAL ROUTINES:
C     RNSET.......SET RANDOM NUMBER GENERATOR (IMSL)
C     RNUN........RETURN RANDOM NUMBER ARRAY (IMSL)
C======================================================================C
      SUBROUTINE RDMGEN (NR, IS, AR)
C
      REAL      AR(NR)
C                                 RANDOM NUMBERS FROM IMSL SUBROUTINES
      CALL RNSET (IS)
      CALL RNUN (NR,AR)
C
      RETURN
      END
```

WAVNUM Subroutine

```
C=======================================================================C
C     WAVNUM
C
C     COMPUTES WAVE NUMBER, K=2PI/L, FOLLOWING LINEAR WAVE THEORY
C     NOTE THAT DIMENSION IS 1/METER
C
C        IN:
C     FQ...........FREQUENCY (HZ)
C     DH...........WATER DAPTH (M)
C
C        OUT:
C     WN...........WAVE NUMBER (1/M)
C=======================================================================C
      SUBROUTINE WAVNUM (FQ,DH, WN)
C
      PARAMETER (GRAV=9.81, TOL=1.0E-5)
      LOGICAL ERROR
C
      TWOPI = 8.0 * ATAN(1.0)
      CORR = 2 * TOL
      ERROR = .FALSE.
C SOLVE FOR KH (X1) USING ITERATIVE METHOD
C                             INITIAL GUESS, X1, BASED ON WHSQ
      WHSQ = (DH/GRAV) * (TWOPI * FQ)**2
      IF (WHSQ .GT. 1.0) THEN
         X1 = WHSQ
C                             DEEP WATER LIMITS
         IF (TANH(X1) .GT. 1.-TOL) GO TO 200
      ELSE
         X1 = SQRT(WHSQ)
C                             SHALLOW WATER LIMITS
         IF (ABS(X1-TANH(X1)) .LT. TOL) GO TO 200
      ENDIF
500      IF (ABS(CORR) .GT. TOL .AND. .NOT. ERROR) THEN
         FUNC = X1 * TANH(X1) - WHSQ
         DFUNC = TANH(X1) + X1/(COSH(X1))**2
C                             CHECK SLOPE TO AVOID DIVISION BY ZERO
         IF (ABS(DFUNC) .LT. TOL) THEN
              ERROR = .TRUE.
              WRITE(6,*) 'ERROR IN SLOPE CHECK IN WAVNUM SUBROUTINE'
              GOTO 100
         ELSE
              CORR = FUNC/DFUNC
              X1 = X1 - CORR
         ENDIF
         GOTO 500
      ENDIF
200   WN = X1 / DH
C
100   RETURN
      END
```

Appendix B: Contents of Accompanying Disk

The accompanying 3.5 inch, double density floppy disk contains the subroutine files and data files referred throughout this report. The extension "`*.doc`" indicates a document file which may be read to further explain the contents of the disk. The extension "`*.for`" indicates a FORTRAN file. The extension "`*.dat`" indicates a data file. These time series data files are of measured laboratory waves. Table 9 summarizes the files.

Table 9: Files on Accompanying Disk.

FILE NAME	DESCRIPTION	REFERENCE
`readme.doc`	document file for further explanation of accompanying disk	Appendix B
`TMASPC.for`	computes the TMA spectrum for wind waves	Part II
`SPCPAR.for`	computes spectral parameters	Part III
`TIMEPH.for`	generates time series following random phase scheme	Part IV
`TIMEDC.for`	generates time series for known Fourier coefficients	Part IV
`TIMPAR.for`	computes time series parameters	Part V
`SPCTRA.for`	computes power density spectrum	Part VI
`IRSORT.for`	computes incident and reflected waves for 3-gage array	Part VII
`COHPHS.for`	computes coherence squared and phase between two time series	Part VIII
`DISTNR.for`	computes PDF and exceedance probability	Part IX
`USRSPC.for`	accommodates generation of time series from known spectrum	Part X
`PRORBR.for`	produces input wave train for RBREAK	Part XI
`FFTIMSL.for`	computes the complex Fourier coefficients (FFT)	Part XII
`RDMGEN.for`	generates an array of pseudo-random numbers	Part XIII
`WAVNUM.for`	computes the wave number based on linear wave theory	Part XIV
`CM06G1.dat`	time series data file at $x = 0.0m$	Part VI, VII, VIII, IX
`CM06G2.dat`	time series data file at $x = -1.4m$	Part VII
`CM06G3.dat`	time series data file at $x = -2.0m$	Part VII, VIII