

WHOI-85-17

**Programs for Computing Properties of  
Coastal-Trapped Waves and Wind-Driven Motions  
Over the Continental Shelf and Slope**

by

Kenneth H. Brink

David C. Chapman

Woods Hole Oceanographic Institution  
Woods Hole, Massachusetts 02543

---

April 1985

**Technical Report**

*Funding was provided by the National Science Foundation under  
grant Number OCE 84-08563.*

*Reproduction in whole or in part is permitted for any purpose of the  
United States Government. This report should be cited as:  
Woods Hole Oceanog. Inst. Tech. Rept. WHOI-85-17.*

*Approved for publication; distribution unlimited.*

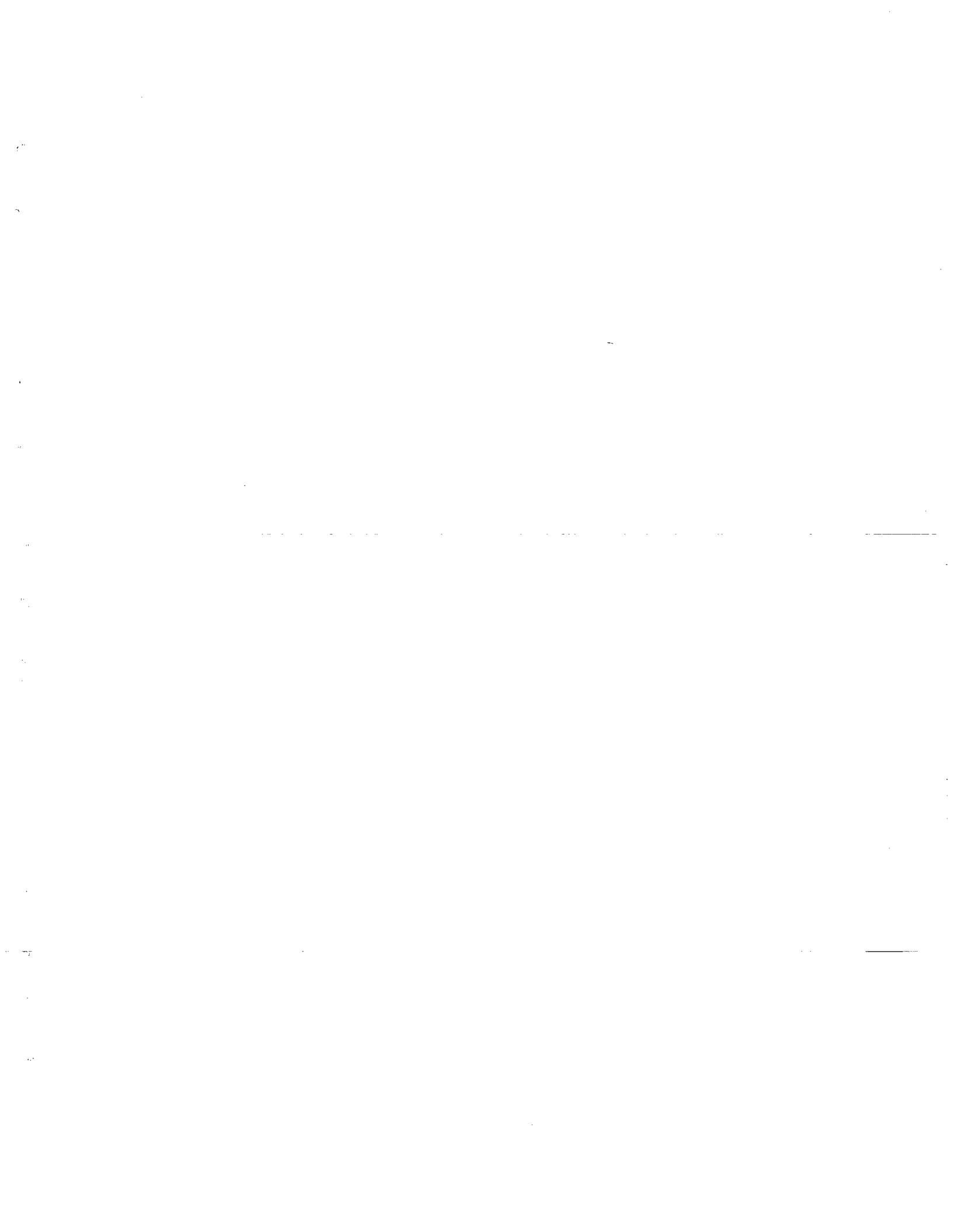
**Approved for Distribution:**

---

*Robert C. Beardsley*

Robert C. Beardsley, Acting Chairman  
Department of Physical Oceanography





## Abstract

Documentation and listings are presented for a sequence of computer programs to be used for problems in continental shelf dynamics. Three of the programs are to be used for computing properties of free and forced coastal-trapped waves. A final program may be used to compute wind-driven fluctuations over the continental shelf and slope.

## Table of Contents

	<u>Page</u>
1. General Introduction	1
2. Barotropic Shelf Waves	5
3. Coastal-Trapped Waves with Stratification and Topography	16
4. Near-Inertial Coastal-Trapped Waves with Stratification and Topography	25
5. Wind-Driven Motions	31
Word of Caution	43
Acknowledgements	44
References	45
Program Listings	
BTCSW	46
BIGLOAD2	60
CROSS	76
BIGDRV2	81

## CHAPTER 1

### GENERAL INTRODUCTION

In a recent sequence of papers (Brink, 1982a,b; Chapman, 1983; Clarke and Brink, 1985) a number of computer programs have been described which compute properties of linear coastal-trapped waves and wind-driven motions over the continental shelf. These programs, since they allow rather arbitrary choices of topography, stratification, etc., may be of fairly general use to the oceanographic community. For this reason, listings and documentation for these algorithms have been assembled here in an accessible form.

Some definitions are common to all of the following routines. Specifically, we use the coordinate system shown in Figure 1, such that the coast (if present) lies at  $x = 0$  and the ocean in the region  $x > 0$ . The alongshelf coordinate is  $y$  and the vertical coordinate is  $z$  (positive upwards), such that  $z = 0$  at the ocean surface. The  $x$ ,  $y$  and  $z$  velocity components are then  $u$ ,  $v$  and  $w$  respectively. Depth-integrated  $u$  and  $v$  velocities are defined as  $U$  and  $V$ , respectively. Pressure and density are given as  $p$  and  $\rho$ , respectively. A few other commonly used variables are  $N^2$ ,  $f$ ,  $g$ ,  $h$ ,  $\omega$  and  $\ell$ , which represent the Brunt-Väisälä frequency squared, the Coriolis parameter, the acceleration due to gravity, the water depth, wave frequency and alongshelf wavenumber.

A few assumptions are common to all programs below. First, only linear problems are considered. Second, the water depth is always assumed to be a function of  $x$  only. Third, the Brunt-Väisälä frequency may vary in  $z$  only, and must be non-zero everywhere. The only exceptions are in computing barotropic continental shelf waves (program BTCSW, Chapter 2) where the problem is linearized and the Brunt-Väisälä frequency is not specified.

The general free-wave programs BTCSW and BIGLOAD2 (coastal-trapped waves with continuous stratification; Chapter 3) search for free-wave solutions using resonance iteration. The general approach is to assume that the dependent variables are sinusoidal in time and the alongshelf direction, e.g.

$$U(x,y,t) = \hat{U}(x) \exp[i(\omega t + \ell y)] ,$$

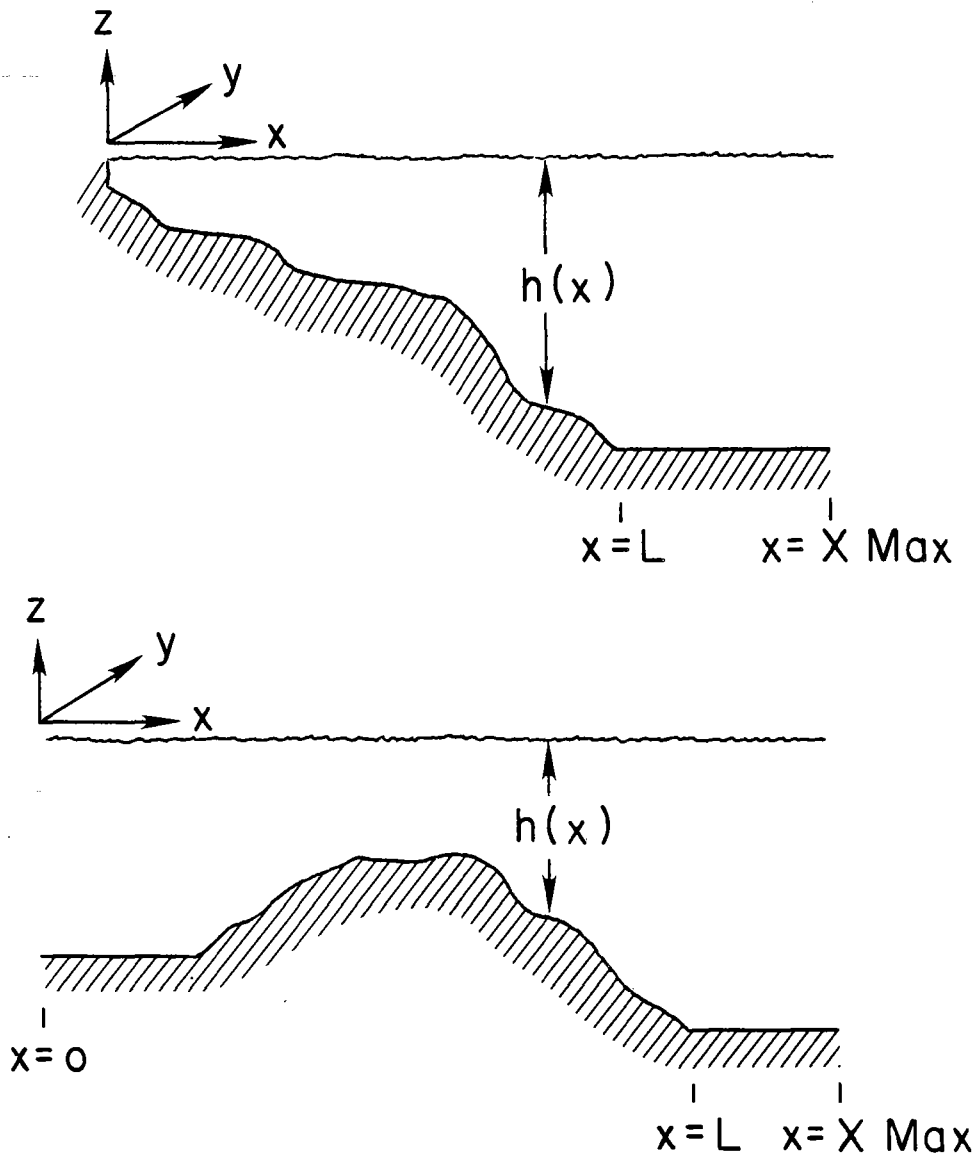


Figure 1: Topography and coordinate system definitions used in all programs: (upper) with a coast, (lower) without a coast.

reducing the problem to a two-dimensional eigenvalue problem in  $(\omega, \ell)$ :

$$\mathcal{L}(\hat{U}(x; \omega, \ell)) = 0 .$$

This is solved for arbitrary forcing and a fixed  $\ell$ . The frequency  $\omega$  is then varied until the free-mode resonance is reached. Resonance is defined as the frequency at which the integrated field variable squared,

$$I_v = \int_0^{\infty} \hat{U}^2 dx$$

or

$$I_p = \int_0^{\infty} \int_{-h}^0 \hat{p}^2 dz dx ,$$

is at a maximum.

A few comments are in order about the workings of the programs. The units internal to all programs are cgs, although input values are often in convenient units (e.g. km for  $x$ ). The input file is always number 5, and the output file number 6. All programs are self-contained except for BIGLOAD2, which requires the use of IMSL subroutine LEQT1B. This subroutine is used to solve the banded matrix equation by L-U decomposition.

The programs described below can be briefly summarized as follows:

- 1) BTCSW: barotropic continental shelf waves (e.g. Buchwald and Adams, 1968) and barotropic bank or trench waves (e.g. Brink, 1983; Mysak, LeBlond and Emery, 1979). Dispersion curves, modal structures, and wind coupling coefficients can be computed for arbitrary topography and mean alongshore flow.
- 2) BIGLOAD2: Coastal-trapped waves in the presence of continuous stratification (e.g. Wang and Mooers, 1976; Huthnance, 1978; Brink, 1982a,b). Dispersion curves (up to  $\omega \cong 0.9f$ ), modal structures and wind coupling coefficients can be computed for arbitrary topography and (horizontally uniform) stratification.
- 3) CROSS: Finding flat-bottom baroclinic modes and where  $\omega = f$  for general coastal-trapped waves (Chapman, 1983). The program allows arbitrary stratification and monotonic bottom topography.

- 4) BIGDRV2: Wind-driven motions over the continental margin (e.g. Clarke and Brink, 1985). The velocity, pressure and density fluctuations driven by a wind stress of the form  $\hat{\tau}(x) \exp[i(\omega t + \lambda y)]$  can be computed for general topography, stratification and bottom friction.

Finally, the user should be aware that programs BTCSW and CROSS require very little CPU time to complete, whereas program BIGLOAD2 uses approximately one minute of CPU time for each point on a dispersion curve and program BIGDRV2 requires approximately one minute of CPU time to complete (both on a VAX 11/780).



CHAPTER 2  
 BAROTROPIC SHELF WAVES  
 Documentation for BTCSW

A. Introduction

This program computes modal structures and dispersion curves for free barotropic shelf waves. It can also compute bottom friction and wind coupling coefficients as in Brink and Allen (1978). Either a free surface or a rigid lid may be used, and a stable mean alongshelf flow can also be included. A variety of boundary conditions are available as options.

B. Formulation

For a linearized, inviscid barotropic ocean, the depth-integrated equations of motion are:

$$\epsilon U_t + \epsilon v_0 U_y - fV = -gh \zeta_x, \quad (2.1a)$$

$$V_t + v_0 V_y + Uv_{0x} + fU = -gh \zeta_y, \quad (2.1b)$$

$$\delta \zeta_t + v_0 \zeta_y + U_x + V_y = 0, \quad (2.1c)$$

where the onshore, and alongshelf directions are  $x$  and  $y$ , respectively, and there are no alongshelf variations in the mean flow  $v_0(x)$  or in the depth  $h$ .  $U$  and  $V$  are the depth-integrated velocities in the  $x$  and  $y$  directions. The free surface elevation is  $\zeta$ , and subscripts  $x$ ,  $y$  and  $t$  represent partial differentiation. The constants  $g$  and  $f$  are the acceleration due to gravity and the Coriolis parameter. The variables  $\epsilon$  and  $\delta$  are defined as follows:

- $\epsilon = 0$  for the long-wave approximation,
- $\epsilon = 1$  for general frequencies and wavenumbers,
- $\delta = 0$  for the rigid-lid approximation,
- and  $\delta = 1$  for a free surface.

With the assumption that  $U$ ,  $V$  and  $\zeta$  vary as  $\exp[i(\omega t + \lambda y)]$ , (2.1) become

$$i\omega'\epsilon U - fV = -gh\zeta_x ,$$

$$i\omega'V + f'U = -i\ell gh\zeta ,$$

$$i\omega'\delta\zeta + U_x + i\ell V = 0 ,$$

where

$$\omega' = \omega + \ell v_0$$

and

$$f' = f + v_{0x} .$$

These can be reduced to either:

$$\begin{aligned} 0 = & U_{xx} \left[ \delta \frac{\omega'^2}{g} - h \ell^2 \right] h \\ & + U_x \left[ h_x \ell^2 - \frac{2\omega' \ell v_{0x} \delta}{g} \right] h \\ & + U \left[ -\frac{\delta^2 \omega'^2}{g^2} (ff' - \epsilon \omega'^2) + \epsilon h^2 \ell^4 \right. \\ & \quad \left. + \frac{\delta}{g} h \ell^2 (ff' - 2\omega'^2 \epsilon + 2f' v_{0x}) \right. \\ & \quad \left. - \frac{h \ell^3 f' h_x}{\omega'} - h \frac{\ell}{\omega'} v_{0xx} \left( \delta \frac{\omega'^2}{g} - h \ell^2 \right) \right] \end{aligned} \quad (2.2)$$

or

$$\begin{aligned} 0 = & \zeta_{xx} [ff' - \epsilon \omega'^2] h \\ & + \zeta_x \left[ (ff' - \omega'^2 \epsilon) h_x - h (f v_{0xx} - 2\omega' \ell v_{0x} \epsilon) \right] \\ & + \zeta \left[ -\frac{\delta}{g} (ff' - \epsilon \omega'^2)^2 + h_x \frac{f \ell}{\omega'} (ff' - \epsilon \omega'^2) \right. \\ & \quad \left. - h \frac{f \ell}{\omega'} (f v_{0xx} - 2\epsilon \omega' \ell v_{0x}) \right. \\ & \quad \left. - h \ell^2 \epsilon (ff' - \omega'^2 \epsilon) \right] . \end{aligned} \quad (2.3)$$

Each of these equations presents a practical difficulty. The  $\zeta$  equation (2.3) possesses a spurious solution. For example, when  $v_0 = 0$  this solution has  $\omega = f$ , and  $\zeta = \zeta_0 \exp(-\ell x)$ . (See Pedlosky, 1979, pp. 79-81 for an explanation.) This spurious mode may, in turn, affect the true solutions. The U equation (2.2) does not possess a spurious mode, but can lead to numerical difficulties for very shallow water (e.g. solving for a laboratory case where  $h < 1$  m everywhere). In general, it is preferable to use the U equation, and to check it against the results of the  $\zeta$  equation. The program allows the choice of the U or  $\zeta$  equation.

### C. Program Input

As explained below, the user provides a bottom topographic profile, a mean flow profile (if desired), and choices for boundary conditions. The program returns modal structures for U and  $\zeta$ , and frequencies for the prescribed wavenumbers. All outputs are in either cgs or arbitrary units, although inputs are in convenient units. Two geometries are possible (Figure 1, p. 2). The first case (Figure 1a) contains a coastal barrier, while the second case (Figure 1b) does not. The second case is useful for bank- or trench-trapped waves. Note that  $\omega > 0$  is assumed, so that waves propagating in the positive y direction (opposite to standard shelf waves in the northern hemisphere) must be found using  $\ell < 0$ .

The following presentation of input parameters describes the user options. A compact list of parameters is given in section 2E. All data are read from file 5.

line 1: IMDM NN

IMDM is the number of cases to be studied. If  $IMDM \neq 1$ , all of the other lines of input must be repeated for each case. This is useful if, for example, several geometries are to be studied in one run.

NN is the number of grid points in the x direction. Presently,  $NN \leq 100$ , but this could be easily changed by the user.

line 2: NITM ISD EPS DEL

These are all parameters used in the search for the resonant frequency. NITM is the maximum number of iterations allowed for finding a resonant frequency (typically 20-40).

ISD directs the frequency search.

For ISD = 0, the program searches for the free-wave frequency closest to the initial guesses.

For ISD = 1, the program searches only towards lower frequencies.

For ISD = -1, the program searches only towards higher frequencies.

EPS is the nominal fractional accuracy desired for  $\omega$ . This is always less than the true error range within which  $\omega$  is known. Typically, EPS = 0.005 (0.5 percent accuracy).

DEL is the fractional step size used for initially searching for  $\omega$ . Typically, DEL = 0.05 (5 percent).

line 3: IUP ILLW

IUP provides the choice of searching with the U or  $\zeta$  equation (see section 2B).

IUP = 0 specifies a search using U.

IUP = 1 specifies a search using  $\zeta$ .

If some other value is given, the program defaults to IUP = 0.

ILLW allows the option of making the long-wave approximation exactly.

ILLW = 0 for long waves ( $\epsilon = 0$  in section 2B).

ILLW = 1 for the general case ( $\epsilon = 1$  in section 2B).

If some other value is given, the program defaults to ILLW = 0.

Also, if ILLW = 0, NCALM (see below) is set to 1, since the waves will be nondispersive.

line 4: IDD1 IDD3 IDD4

These parameters select the boundary conditions.

IDD1 = 0 for a rigid lid ( $\delta = 0$  in section 2B).

IDD1 = 1 for a free surface ( $\delta = 1$  in section 2B).

Other values set the default of IDD1 = 0.

IDD3 = 0; the boundary condition at  $x = XMAX$  is  $U_x = 0$ . This is not the "real" condition, but is used for comparison with the stratified wave program (Chapter 3).

IDD3 = 1; the boundary condition at  $x = XMAX$  is  $U = 0$ . This simulates a channel problem.

IDD3 = 2 sets up the real, exponentially decaying condition at  $x = XMAX$ . This is the preferred condition, but it is only valid if  $h$  and  $v_0$  are constant near  $x = XMAX$ .

If another choice is made for IDD3, the program reverts to IDD3 = 0.

IDD4 = 0, the boundary condition at  $x = 0$  is  $U_x = 0$ . This may be useful for bank or trench waves.

IDD4 = 1 sets  $U = 0$  at  $x = 0$ . This is the desired condition for shelf waves.

IDD4 = 2 uses the exponential decay condition at  $x = 0$ . This is again the preferred condition for bank or trench waves, but it is only valid for  $h$  and  $v_0$  constant at  $x = 0$  (geometry of Figure 1b).

Other values of IDD4 cause the program to revert to IDD4 = 1, the shelf wave case.

---

line 5: NCALM ILW

NCALM is the maximum number of  $(\omega, \ell)$  pairs to be calculated for a given dispersion curve.

ILW provides an option on calculating parameters valid for the long-wave limit. These will only be computed for the first  $(\omega, \ell)$  pair.

If ILW = 0, then no long-wave parameters are computed.

If ILW  $\neq$  0, then the "streamfunction"  $\phi_n(x)$ , wind-coupling coefficient  $b_n$  and bottom drag coefficient  $a_{nn}$  are computed. The definitions follow from Brink and Allen (1978).

For computation and conceptual reasons, these parameters will not be computed if either  $IDD4 \neq 1$  or if  $h(0) = 0$ , even if ILW = 1.

ILLW need not be set to 0.

line 6: RLF DRL

These parameters define the wavenumbers for which  $\omega$  is calculated.

The wavenumbers used in the program will be:

$$\ell = (RLF + (n - 1) DRL) \times 10^{-8} \text{cm}^{-1}$$

when  $n$  represents the number of the  $(\omega, \ell)$  pair on the dispersion curve.  $n$  ranges from 1 to NCALM (see line 5).

For example, if  $RLF = 0.5$  and  $DRL = 1.0$ , then the first wavenumber to be computed is  $\ell = 0.5 \times 10^{-8} \text{ cm}^{-1}$  and the others will be  $(1.5, 2.5, 3.5, \dots) \times 10^{-8} \text{ cm}^{-1}$ .

line 7: IPC

If  $IPC \neq 0$ , then the program prints out modal structures as well as search information for each  $(\omega, \ell)$  pair.

If  $IPC = 0$ , then the modal structure is printed only for the first  $(\omega, \ell)$  pair.

line 8: F XMAX

F is the Coriolis parameter, which is multiplied by  $10^{-5}$  within the program. Thus,  $F = 7.5$  represents  $f = 7.5 \times 10^{-5} \text{ s}^{-1}$ .

XMAX is the distance (in km) from  $x = 0$  to the offshore boundary of the grid (Figure 1). Typically,  $XMAX \sim 2L$ , so that about one half of the domain has a flat bottom.

line 9: NRX

This is the number of  $[x, h(x)]$  pairs to be input to define the bottom topography.

line 10 and following: X H

These are pairs of offshore distance (x) in km and depth (h) in m. These can be arbitrarily spaced, and the information is linearly interpolated to the grid points. The first pair must have  $x = 0$ . For  $x >$  (the last x value read), depth is set to the last h value read.

NRD

This is the number of  $[x, v_0(x)]$  pairs to be input. If  $NRD = 0$ , the program sets  $v_0 = 0$  everywhere.

X V

These are the NRD pairs of offshore distance (x) in km and mean along-shelf velocity ( $v_0$ ) in cm/s. These can be arbitrarily spaced. For  $x <$  (the first x value read), the program sets  $v_0 = 0$ . For  $x >$  (the last x value read), the program sets  $v_0$  equal to the last  $v_0$  value read.

WW(1) WW(2) WW(3)

These are three initial guesses at the free-wave frequency  $\omega$  for the first value of  $\ell (= RLF \times 10^{-8} \text{ cm}^{-1})$ . The program multiplies WW(I) by  $10^{-5}$ , so  $WW(1) = 0.5$  corresponds to  $\omega = 0.5 \times 10^{-5} \text{ s}^{-1}$ .

NW

This is the number of  $x$  (in km) and friction weight function (WF, non-dimensional) pairs to be input. This is useful for  $x$  dependent bottom drag, i.e.

$$E_0^{1/2} = E' WF(x),$$

where  $E_0$  is the Ekman number,  $E'$  is the Ekman number at  $x = 0$ , and  $WF(x)$  a weighting function such that  $WF(0) = 1$ . If  $NW = 0$ , then  $WF(x) = 1$  for all  $x$  as in Brink and Allen (1978). If  $WF$  varies, then

$$a_{nn} = \int_0^L WF(x) (\phi_{nx}(x))^2 dx,$$

where  $\phi_n$  is the streamfunction modal structure.

X WF

These are [ $x$  (in km), WF (non-dimensional)] pairs to be input. The first pair must start at  $x = 0$ . For  $x >$  (last  $x$  value read), WF is set to the last value read.

#### D. General Comments

- i.) The program will work with  $h = 0$  at  $x = 0$  only in the  $U$  equation mode. Thus, if  $h(0) = 0$ , use  $IUP = 0$ . Alternatively,  $h(0)$  can be very small with either  $IUP = 0$  or 1.
- ii.) When the  $\zeta$  equation is being used (e.g.  $IUP = 1$ ), there is a check for small diagonal elements in the finite-difference matrix equation. If a diagonal element is less than  $10^{-36}$ , a message is printed and the solution is omitted.
- iii.) As a check of the  $U_x$  boundary condition against the "real" boundary condition, calculations were run for  $XMAX = 2L$ , no mean flow and  $n = 1, 2$ . The worst error in  $\omega$  was 3.6 percent for  $n = 1$ , and the error decreased for large  $\ell$ . The  $n = 2$  long-wave coefficients ( $a_{22}$  and  $b_2$ ) varied substantially, however. The error in  $b_2$  was about 50 percent.

- iv.) Identifying modes. The Kelvin wave mode will have no zero crossings of  $\zeta$ . The first shelf wave mode will have 1 zero crossing, the second 2, etc. The first shelf wave mode will have no sign changes in  $U$ , although  $U = 0$  at  $x = 0$ . The second mode has one zero crossing, etc.
- v.) When  $v_0 \neq 0$ , the program checks for critical layers, and prints out the number of critical layers in the solution. Further, the program checks to see if the necessary condition for barotropic instability is satisfied. That is, if

$$\left( \frac{f + v_0 x}{h} \right)_x$$

changes sign, a warning is given.





page, beginning at  $x = 0$  and proceeding to  $x = XMAX$ .  $\Delta x$  is given in the header information.

All units in the output are cgs, except for  $U$  and  $\zeta$  which are in arbitrary units.  $\phi(x)$  is normalized as

$$1 = \int_0^L \frac{h_x}{h^2} \phi_n^2 dx ,$$

so that  $\phi$  has units of  $cm^{1/2}$ .

The coefficients  $b_n$  and  $a_{nn}$  for  $\phi_n$  are only strictly valid for  $v_0 = 0$ , and for a rigid lid. Two different  $a_{nn}, b_n$  pairs are given. The first (streamfunction) set is as defined in Brink and Allen (1978). The second analogous set is defined for the long-wave problem in terms of pressure. This is useful if there is a free surface, since the streamfunction is invalid. In this case

$$p = \sum_n F_n(x) Y_n(y,t)$$

where the free-wave modal structures  $F_n(x)$  are orthogonal by

$$\delta_{nm} = (hF_n F_m) \Big|_{x=0} + \int_0^\infty h_x F_n F_m dx ,$$

and  $Y_n$  obeys

$$b_n' \tau_0^y = Y_{ny} - \frac{1}{c_n} Y_n t - r_0 \sum_m a_{nm}' Y_m .$$

The program prints out  $b_n', a_{nn}'$  and the pressure normalized as above. The bottom stress is taken to have the form

$$\tau_B^y = \rho r_0 WF(x)v ,$$

where  $WF$  is as above,  $r_0$  is a bottom resistance coefficient in  $cm s^{-1}$ , and  $\rho$  the fluid density.

G. An Example

Input File:

```
1      100
20     0      0.001    0.05
0      1
0      2      1
1      1
1.0    1.0
0
10.0   400.
3
0.     10.
100.   150.
200.   4000.
3
0.     0.
50.    100.
100.   0.
0.5    0.52    0.54
0
```

The result is, after 14 iterations,  $\omega = 0.6867 \times 10^{-5} \text{s}^{-1}$ ,  $a_{11} = 0.19865 \times 10^{-7} \text{cm}^{-1}$  and  $b_1 = 0.1428 \times 10^{-1} \text{cm}^{-1/2}$ . This is the  $n = 1$  mode.

CHAPTER 3  
 COASTAL-TRAPPED WAVES WITH STRATIFICATION AND TOPOGRAPHY  
 Documentation for BIGLOAD2

A. Introduction

This program calculates free-wave dispersion curves ( $\omega, \ell$  pairs) by resonance iteration, given input parameters including arbitrary bottom topography and stratification. Options include the choice of a free-surface or a rigid-lid boundary condition, and the inclusion of the component of planetary  $\beta$  perpendicular to the coast.

Note that this program uses an external (IMSL) subroutine in the solution procedure.

B. Formulation

The problem is formulated in the geometry of Figure 1a. Note that the depth at the coast  $h(0)$  is non-zero, although it can be arbitrarily small.

The governing equations are

$$\epsilon u_t - fv = -\frac{1}{\rho_0} p_x$$

$$v_t + fu = -\frac{1}{\rho_0} p_y$$

$$0 = -p_z - g\rho$$

$$u_x + v_y + w_z = 0$$

and

$$\rho_t + w\rho_{0z} = 0 .$$

} (3.1)

The variables  $u, v$  and  $w$  are the velocity components in the  $x, y$  and  $z$  directions, respectively. The Coriolis parameter is  $f$ , the acceleration due to gravity is  $g$ , and the pressure is  $p$ . Density is defined by

$$\hat{\rho}(x,y,z,t) = \rho_0(z) + \rho(x,y,z,t) .$$

The Boussinesq approximation is made throughout. Finally, subscripts  $x, y, z$  and  $t$  represent partial differentiation. The quantity  $\epsilon$  is set to either 0 (long-wave approximation) or 1 (general frequency and wavenumber).

All variables are taken to vary as  $\exp[i(\omega t + \ell y)]$ , so that equations (3.1) reduce to:

$$0 = p_{xx} + \frac{2f\beta}{(f^2 - \epsilon\omega^2)} p_x - p[-\epsilon \ell^2 + \frac{\ell\beta}{\omega} - \frac{2f^2\beta\ell}{\omega(f^2 - \epsilon\omega^2)}] + (f^2 - \epsilon\omega^2) \left( \frac{p_z}{N^2} \right)_z$$

subject to

$$p_z + \delta \frac{N^2}{g} p = 0 \quad \text{at } z = 0$$

$$w + h_x u = 0 \quad \text{at } z = -h(x)$$

$$u = 0 \quad \text{at } x = 0$$

and

$$u_x = 0 \quad \text{at } x = XMAX.$$

where  $N$  is the Brunt-Väisälä frequency. The fourth boundary condition (Brink, 1982b) replaces the more desirable

$$p \text{ bounded as } x \rightarrow \infty ,$$

which is not very practical on a finite difference grid. The parameter  $\delta$  is either 0 (rigid-lid surface) or 1 (free surface) at the user's discretion. Note that only the component of  $\beta$  perpendicular to the coast has been included, so that  $f = f_0 - \beta x$ . This means that if the land is north of the ocean, then  $\beta > 0$ , while if the land is south of the ocean, then  $\beta < 0$ . The component of  $\beta$  parallel to the coast is not included because of the considerable complications involved.

The problem is solved by using the coordinate transformation

$$\theta = \frac{z}{h(x)} .$$

This maps the domain into a rectangle, where the problem is solved on a fixed 17 (vertical) by 25 (horizontal) point grid. Thus, vertical resolution is far better close to shore, in shallow water.

### C. Program Input

The user must supply stratification, topography, the Coriolis parameter, and other information. The program then, after converging to a free wave solution, prints out frequency, wavenumber and the modal structure. All program outputs are either in arbitrary or cgs units.

The contents of the input file (file 5) are as follows.

line 1: EPS EST DD1.

EPS is the nominal fractional accuracy desired for the free-wave frequency, i.e.  $\Delta\omega/\omega$ . The program stops searching when its next frequency estimate agrees with the previous best estimate to this accuracy. Typically, EPS = 0.005.

EST is the fractional initial search increment for  $\omega$ . Typically, EST = 0.05.

DD1 determines whether there is a rigid lid (DD1 = 0.) or a free surface (DD1 = 1.0). This corresponds to the  $\delta$  in section 3B.

line 2: ICCM NCALM NITM ISD

ICCM is the number of dispersion curves to be calculated. If ICCM  $\neq$  1, all of the remaining lines of input must be repeated for each dispersion curve.

NCALM is the number of  $(\omega, \ell)$  pairs to be calculated along each dispersion curve.

NITM is the maximum number of iterations allowed for finding a single frequency on the dispersion curve. If NITM is exceeded, the program terminates.

ISD determines the direction of search for frequency.

If ISD = 0, the program searches for the free-wave frequency closest to the initial guesses.

If ISD = 1, the program searches only toward frequencies lower than the initial estimates.

If ISD = -1, the program searches only towards higher frequencies.

line 3: F XMAX

F is the Coriolis parameter, which is multiplied by  $10^{-5} \text{s}^{-1}$  within the program. For example,  $F = 5$ , represents  $f_0 = 5 \times 10^{-5} \text{s}^{-1}$ .

XMAX is the offshore extent of the grid in km. Typically,  $XMAX \sim 2L$  (see Figure 1a).

line 4: BETA ILWW

BETA is the component of planetary  $\beta$  perpendicular to the coast (see section 3B) entered in units of  $\text{s}^{-1} \text{cm}^{-1}$ .

ILWW is  $\epsilon$  of section 3B.

If  $ILWW = 0$ , the long-wave limit is taken exactly.

If  $ILWW = 1$ , the program runs for general frequency and wavenumber.

If  $ILWW$  is not equal to 0 or 1, the program terminates.

line 5: NCAL, WH(1)

For a new dispersion curve,  $NCAL = 1$  and  $WH(1)$  is any number.

When resuming an older curve which has been partially completed,  $NCAL = 2$  and  $WH(1)$  is the frequency of the last  $\ell$  of the previous run.

This must correspond to RLF (see line 7). This information will allow better estimates at succeeding frequencies. Note that  $WH(1)$  is multiplied by  $10^{-6} \text{s}^{-1}$ , so that  $WH(1) = 0.5$  corresponds to  $\omega = 0.5 \times 10^{-6} \text{s}^{-1}$ .

line 6: IDIAG

If  $IDIAG \neq 0$ , then the  $v$ ,  $u$  and  $\rho$  fields (as well as  $p$ ) will be output for the first  $(\omega, \ell)$  pair on the dispersion curve.

If  $IDIAG = 0$ , then only  $v$  (and of course  $p$ ) will be output.

Regardless of  $IDIAG$ , only  $p$  will be output for points after the first  $(\omega, \ell)$  pair.

line 7: RLF DRL

These parameters determine the wavenumbers for which  $\omega$  is computed. Specifically,

$$\ell = (RLF + (n-1) DRL) \times 10^{-7} \text{cm}^{-1},$$

for  $n = 1, 2, 3, \dots, NCALM$ .

line 8: WW(1) WW(2) WW(3)

These are three initial estimates of the free-wave frequency for the starting wavenumber. The program multiplies these values by  $10^{-6} \text{s}^{-1}$ , so a value of 0.5 corresponds to  $\omega = 0.5 \times 10^{-6} \text{s}^{-1}$ .

line 9: NRX

This is the number of  $[x, h(x)]$  pairs to be input.  $NRX \geq 1$  is required.

line 10 and following: X H

These are values of offshore distance (x) in km and water depth (h) in m. There must be NRX pairs, and the first pair must have  $x = 0$ . The spacing in x is arbitrary, and the program fills out the topography by linear interpolation. For values of x greater than the last value read, the program assigns the last depth read.

NR DZR ALPH

These are parameters used for reading the profile of  $N^2$  (the Brunt-Väisälä frequency squared).

NR is the number of  $N^2$  values to be read.

DZR is the vertical spacing of  $N^2$  values in m.

ALPH describes the exponential tail on the  $N^2$  profile. Often  $N^2$  is not available from surface to bottom. In this case, an exponential extrapolation is used:

$$N^2 = N_0^2 \exp(\zeta_0 - \zeta)/ALPH)$$

where

$N_0^2$  is the last  $N^2$  value read,

$\zeta_0$  is the depth of the last  $N^2$  value read, and

$\zeta$  is the depth of the point, i.e.  $\zeta = -z$ .

ALPH is then the exponential length scale of  $N^2$  decay, in km.

CMLT

This is a conversion factor by which the input  $N^2$  are multiplied in order to get units of  $(\text{rad/s})^2$ . Specifically,

$$N^2(\text{rad}^2/\text{s}^2) = \text{CMLT} \times N^2(\text{user units})$$

following lines:  $N^2$

These are the values of  $N^2$  in user units, one per line. There must be NR regularly spaced values. The first  $N^2$  value should be at  $z = 0$ , and  $N^2$  should never equal zero.

NRR

This is the number of  $[x, r(x)]$  pairs to be input, where  $r(x)$  is a bottom resistance coefficient in  $\text{cm s}^{-1}$  defined by



$$\frac{1}{\rho_0} \tau_B = r(x) \underline{v}(x, -h) .$$

This information is used in subroutine LGWH for computing the bottom drag coefficient.  $NRR \geq 1$  is required.

X R

These are the NRR pairs of offshore distance (x) in km and bottom resistance coefficient (r) in cm/s. The first x value read must be zero. The x spacing is arbitrary, and is filled out by linear interpolation. For values of x greater than the last value read, the last value of R will be used.

#### D. General Comments

i.) Identifying modes. Generally, the barotropic Kelvin wave ( $n = 0$ ) will have no zero crossings in pressure. The first coastal-trapped wave ( $n = 1$ ) will have one zero crossing, etc. Occasionally, isolated small pockets of reversed sign in p will exist, representing numerical error. These extraneous zero crossings are usually obvious when the modal structure is plotted.

ii.) The program does not generally work well when the shelf-slope width is small relative to the first internal Rossby radius of the deep-ocean. For such a case, the user should experiment to see if  $\omega$  is stable with respect to small changes in XMAX.

iii.) Since the governing equation is formulated in terms of pressure, a spurious mode exists for  $\beta = 0$  and  $\omega = f$ . It has

$$p = p_0 e^{-\ell x} ,$$

with  $p_z = 0$ . (See Pedlosky, 1979, pp. 79-81 for more detail.) This mode makes the program's performance suspect near  $\omega = f$ .

iv.) For  $\omega > f$ , the inertia-gravity wave continuum is quantized by the offshore boundary condition, and the results are useless. The program will stop after three iterations if  $\omega > f$  is sought.

- v.) The program has trouble finding the barotropic Kelvin wave.
- vi.) The program uses an external (IMSL) subroutine to solve the matrix equation.

E. Input Summary

EPS	EST	DD1	
ICCM	NCALM	NITM	ISD
F	XMAX		
BETA	ILWW		
NCAL	WH(1)		
IDIAG			
RLF	DRL		
WW(1)	WW(2)	WW(3)	
NRX			
X	H		
			NRX times
NR	DZR	ALPH	
CMLT			
N <sup>2</sup>			
			NR times
NRR			
X	R		
			NRR times

F. Program Output

The program first lists the boundary conditions chosen, and a few parameters, such as  $f$  and  $\beta$ .

Next,  $N^2$  at  $x = XMAX$  is listed at grid point locations, starting at the bottom of the water column. (The first point is at  $z = -h$ , and the last at  $z = 0$ ). The  $\Delta z$  can be found in the pressure listing.

Then, information about the frequency search is listed. For each iteration,  $\omega$ ,  $\ell$  and  $c = \omega/\ell$  are listed, along with

$$RI = \int p^2 dz dx ,$$

a measure of resonance, and IER, an IMSL error code. A message announces convergence.

The  $v$  (alongshelf velocity) field is listed, beginning at  $x = 0$ . Total depth ( $h$ ) and depth increment ( $DZ$ ) are given for each  $x$ . Then  $v$  is listed, beginning at  $z = -h$ . The  $v$  field is computed after  $p$  has been normalized so that

$$1 = \int_{-h}^0 p^2 dz \Big|_{x=0} + \int_0^{\infty} h_x p^2 dx \Big|_{z=-h} .$$

The pressure field is also listed, and (optionally)  $u$  and  $\rho$ . All units are consistent so that if  $p$  were in  $\text{dyne/cm}^2$ , then  $v$  would be in  $\text{cm/s}$ .

After the first  $(\omega, \ell)$  point on a dispersion curve, only  $p$  will be listed, and in this case it is not normalized.

Immediately after the  $v$  printout,  $a_{nn}$  and  $b_n$  are listed. (See Brink, 1982a.) This is an improved version due to Clarke and Van Gorder (1985). Finally, at various points in the output, the contributions of  $u$  and  $v$  to wave kinetic energy, and of  $\rho$  and free-surface height to wave potential energy are given. These can be used to compute the diagnostic

$$R = \frac{\text{kinetic energy}}{\text{potential energy}} .$$

This quantity approaches 1 for a baroclinic Kelvin-like wave, and becomes large ( $> 10$ ) for a barotropic shelf wave.

G. An Example

Input file:

```

0.005      0.05      0.
1          1          20      0
10.0      200.
0.         0
1          0.
0
0.1        0.5
3.0        3.1      3.2
2
0.         10.
100.       4000.
2          5000.    5.
1.0 E-06
1.375
1.375
1
0.0        0.05

```

This represents a uniform  $N^2$  and a uniformly sloping shelf. After seven iterations,  $\omega/\ell = 315.05$  cm/s for the  $n = 1$  mode. The coupling coefficients are

$$b_n = -0.36614 \times 10^{-2} \text{ cm}^{-1/2}$$

$$a_{nn} = 0.18207 \times 10^{-8} \text{ cm}^{-1} .$$

Note that the sign of  $b_n$  can be positive or negative, depending on the sign of  $p$ .

This result can be compared to that obtained by Huthnance (1978) of  $\omega/\ell = 310$  cm/s. Note that he had  $h(0) = 0$ .

CHAPTER 4  
NEAR-INERTIAL COASTAL-TRAPPED WAVES WITH STRATIFICATION AND TOPOGRAPHY  
Documentation for CROSS

A. Introduction

This program finds the wavenumbers (if any) at which the dispersion curves for free coastal-trapped waves approach  $\omega = f$ . Also determined is the lowest-order pressure field at  $\omega = f$ . Input parameters include arbitrary bottom topography and vertical stratification. Options include the choice of a free-surface or a rigid-lid boundary condition. The program is designed to be compatible with BIGLOAD2 (Chapter 3).

This program can also be used to find the vertical structures and phase speeds of flat-bottom baroclinic Kelvin waves for arbitrary vertical stratification.

B. Formulation

The solution procedure is based on the near-inertial analysis of Huthnance (1978, Section 6c, see also Chapman, 1983). For a coastal-trapped wave with frequency  $\omega$  slightly less than  $f$ , i.e.  $\omega = f(1-\gamma)$  where  $\gamma \ll 1$ , then the pressure may be assumed to take the form

$$p = [p_0(z) + \gamma p_1(x, z)] e^{-\ell x}$$

where  $x$  is positive offshore,  $z$  positive upwards and  $\ell$  the alongshelf wavenumber. The topography is shown in Figure 1a. It can be shown that with these assumptions, the lowest order pressure field obeys

$$\frac{d}{dz} \left( \frac{f^2}{N^2(z)} e^{-2\ell \bar{X}(z)} \frac{dp_0}{dz} \right) + \ell^2 e^{-2\ell \bar{X}(z)} p_0 = 0 \quad (4.1)$$

where  $N^2(z)$  is the squared Brunt-Väisälä frequency, and  $\bar{X}(z)$  is the inverse topography defined by  $z = -h(\bar{X})$  where  $h$  is the depth. Note that the topography must be monotonic to be inverted uniquely. The program checks for this. Boundary conditions are

$$\frac{dp_0}{dz} = 0 \quad \text{at } z = -H \quad (4.2a)$$

$$\frac{dp_0}{dz} + \frac{\delta N^2(0)}{g} p_0 = 0 \quad \text{at } z = 0 \quad (4.2b)$$

where  $H$  is the maximum depth at  $L < x < XMAX$ ,  $g$  gravitational acceleration, and  $\delta = 1$  for a free surface or  $\delta = 0$  for a rigid lid. Thus, for known  $f$ , topography and stratification, equations (4.1, 4.2) can be solved to find the wavenumber(s)  $\lambda$  at which  $\omega = f$ .

Solutions are found by a shooting technique in which (4.1) is represented in finite difference form and (4.2b) is assumed satisfied. Then  $\lambda$  is varied until the integration of (4.1) from  $z = 0$  to  $z = -H$  results in a pressure distribution which satisfies (4.2a). The wavenumber  $\lambda$  is found to a relative fractional accuracy of  $10^{-7}$ .

### C. Program Input

The user must supply such information as stratification, topography, the Coriolis parameter, etc. All program outputs are either in arbitrary or cgs units.

The contents of the input file (file 5) are as follows. They are designed to be similar to the contents of the input file used with BIGLOAD2.

line 1: NV DD1

NV is the number of grid points (in the vertical) to be used in the solution. First, the topography is computed exactly as in BIGLOAD2 to obtain 25 depths. Then the topography between the coast and the flat bottom ( $0 < x < L$ ) is filled with NV points by linear interpolation. The maximum NV is 101.

DD1 determines whether there is a rigid lid (DD1 = 0.0) or a free surface (DD1 = 1.0). This corresponds to  $\delta$  in (4.2b).

line 2: F XMAX

F is the Coriolis parameter, which is multiplied by  $10^{-5} \text{ s}^{-1}$  within the program. Thus,  $F = 5.0$  corresponds to  $f = 5. \times 10^{-5} \text{ s}^{-1}$ .

XMAX is the offshore extent of the BIGLOAD2 grid in km. It is used here only to insure that the original 25 depths (before interpolation) are computed as in BIGLOAD2.

line 3: NRX

This is the number of  $[x, h(x)]$  pairs to be input. ( $NRX \geq 1$ ).

line 4 and following: X H

These are values of offshore distance (x) in km and water depths (h) in m. There must be NRX pairs, and the first pair must have  $x = 0$ . The spacing in x is arbitrary, and the program fills out the topography by linear interpolation. For values of x greater than the last value read, the program assigns the last depth read.

NR DZR ALPH

These are parameters used for reading the profile of  $N^2$  (the Brunt-Väisälä frequency squared).

NR is the number of  $N^2$  values to be read.

DZR is the vertical spacing of  $N^2$  values in m.

ALPH describes the exponential tail on the  $N^2$  profile. Often  $N^2$  is not available from surface to bottom. In this case, an exponential extrapolation is used:

$$N^2 = N_0^2 \exp((\zeta_0 - \zeta)/ALPH)$$

where

$N_0^2$  is the last  $N^2$  value read,

$\zeta_0$  is the depth of the last  $N^2$  value read, and

$\zeta$  is the depth of the point, i.e.  $\zeta = -z$ .

ALPH is then the exponential length scale of  $N^2$  decay, in km.

CMLT

This is a conversion factor by which the input  $N^2$  are multiplied in order to get units of  $(\text{rad/s})^2$ . Specifically,

$$N^2(\text{rad}^2/\text{s}^2) = \text{CMLT} \times N^2(\text{user units})$$

following lines:  $N^2$

These are the values of  $N^2$  in user units. There must be NR regularly spaced values. The first  $N^2$  value should be at  $z = 0$ , and  $N^2$  should never equal zero.

NRS

This is the number of wavenumber searches to be made. For each search, an ( $\ell_{\text{minimum}}$ ,  $\ell_{\text{maximum}}$ ,  $\Delta\ell$ ) set is read. This allows several searches for the same mode or searches for several modes.

following lines: X5 X6 X7

X5 is the minimum  $\ell$  to start the search

X6 is the maximum  $\ell$  to end the search

X7 is the  $\Delta\ell$  used to locate the solution.

The program multiplies these values by  $10^{-7}$  to obtain  $\text{cm}^{-1}$ . Thus, (X5, X6, X7) = (2., 3., .1) corresponds to  $\ell_{\text{min}} = 2. \times 10^{-7} \text{cm}^{-1}$ ,  $\ell_{\text{max}} = 3. \times 10^{-7} \text{cm}^{-1}$ ,  $\Delta\ell = 0.1 \times 10^{-7} \text{cm}^{-1}$ . The program then locates a root by shooting using  $\ell = \ell_{\text{min}} + n\Delta\ell$  ( $n = 0, 1, 2, \dots$ ) up to  $\ell = \ell_{\text{max}}$ . If a root is located, Newton's method is used to home in on it. If no root is found between  $\ell_{\text{min}}$  and  $\ell_{\text{max}}$ , then the search moves to the next choice for X5, X6, X7. There must be NRS sets of X5, X6, X7.

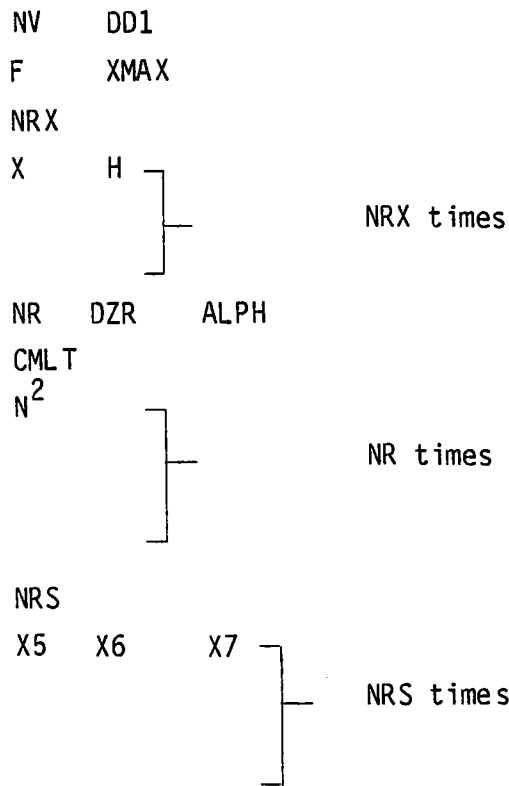
#### D. General Comments

- i.) Identifying modes. The barotropic or Kelvin wave ( $n = 0$ ) will have no zero crossings in pressure. It will exist only if a free surface is used ( $DD1 = 1.0$ ). The first coastal-trapped wave ( $n = 1$ ) will have one zero crossing, the second ( $n = 2$ ) will have two, etc.
- ii.) The program is probably best used when BIGLOAD2 predicts a dispersion curve which reaches  $\omega \approx .9f$ , above which BIGLOAD2 has problems. The BIGLOAD2 dispersion curve can be used to estimate the search parameters for CROSS. The same topography and stratification should be used in both. It may be dangerous to use CROSS if the dispersion curves are unknown, because they may never reach  $\omega = f$  and CROSS will only report that no root was found in the specific interval (i.e. CROSS cannot tell whether or not a dispersion curve ever reaches  $\omega = f$ , only whether or not it reaches  $\omega = f$  in the specified interval).



iii.) This program can find flat-bottom baroclinic Kelvin wave modal structures and phase speeds by using one depth and the desired stratification (NRX = 1, [x, h] = [0., H]). Since Kelvin waves are nondispersive, the phase speed at  $f$  is the same as at any other frequency.

E. Input Summary



F. Program Output

The program first lists the surface boundary condition chosen, and the parameters used ( $f$ , XMAX). Then  $\Delta z$  is listed followed by the inverse topography at  $z = -(n\Delta z)$  where  $n = 0, 1, 2, \dots, NV-1$ . Next  $N^2$  is listed also at  $z = -(n\Delta z)$ .

For each search, if a solution is found, the wavenumber  $\ell$  and phase speed ( $f/\ell$ ) are listed followed by the pressure structure ( $p_0 e^{-\ell x}$ )

normalized by  $(\int_{-H}^0 p_0^2 dz)^{1/2}$ . The pressure is listed at  $z = -(n + 1/2) \Delta z$  where  $n = 1, 2, 3, \dots, NV-1$ . That is, the pressures are given midway between the topography grid points.

G. An Example

51		0.	
10.		200.	
2			
0.		1.	
100.		4000.	
2	5000		5.
1.0 E-6			
56.25			
56.25			
2			
2.	3.	.1	
4.	5.	.1	

This represents a uniformly sloping shelf with uniform stratification.

The analytical solution of (4.1, 4.2) is  $\ell = \frac{n\pi}{L} [(\frac{NH}{fL})^2 - 1]^{-1/2}$  (Huthnance, 1978, p. 83) from which  $\ell_1 = 1.11 \times 10^{-7} \text{cm}^{-1}$ ,  $\ell_2 = 2.22 \times 10^{-7} \text{cm}^{-1}$ . The values predicted by CROSS are  $\ell_1 = 1.11 \times 10^{-7} \text{cm}^{-1}$ ,  $\ell_2 = 2.22 \times 10^{-7} \text{cm}^{-1}$ .

CHAPTER 5  
WIND-DRIVEN MOTIONS  
Documentation for BIGDRV2

A. Introduction

This program computes the velocity, pressure and density response of stratified shelf and slope waters to a time and space harmonic wind stress. Options include using

- a) rigid lid or free surface,
- b) "long wave" or general parameters,
- c) alongshelf or cross-shelf winds.

The cross-shelf distributions of bottom resistance coefficient and of wind stress are at the user's discretion.

B. Formulation

The interior region (away from surface and bottom boundary layers) is described by the linear, inviscid equations:

$$\epsilon u_t - fv = \frac{-1}{\rho_0} p_x \quad (5.1a)$$

$$v_t + fu = \frac{-1}{\rho_0} p_y \quad (5.1b)$$

$$0 = -p_z - g\rho \quad (5.1c)$$

$$u_x + v_y + w_z = 0 \quad (5.1d)$$

$$0 = \rho_t + w\rho_{0z} \quad (5.1e)$$

The variables  $u$ ,  $v$  and  $w$  are the velocity components in the  $x$ ,  $y$  and  $z$  directions, respectively. The Coriolis parameter is  $f$ , the acceleration due to gravity is  $g$ , and the pressure is  $p$ . Density is defined by

$$\hat{\rho}(x,y,z,t) = \rho_0(z) + \rho(x,y,z,t) .$$

The Boussinesq approximation is made throughout. Finally, subscripts  $x, y, z$  and  $t$  represent partial differentiation. The quantity  $\epsilon$  is set to either 0 (long-wave approximation) or 1 (general frequency and wavenumber). Equations (5.1) can be reduced to a single field equation for pressure,

$$0 = p_{xxt} + \epsilon p_{yyt} + (f^2 + \epsilon \frac{\partial^2}{\partial t^2}) (\frac{p_z}{N^2})_{zt}, \quad (5.2)$$

where  $N^2$  is the Brunt-Väisälä frequency squared.

The problem is solved by assuming wind stress in the form of

$$\tau_0^y = T^y(x) \exp[i(\omega t + \ell y)],$$

or

$$\tau_0^x = T^x(x) \exp[i(\omega t + \ell y)],$$

and all of the variables ( $u, v, \rho, p$ ) are assumed to have a similar  $y$  and  $t$  dependence. Given these assumptions, (5.2) reduces to

$$0 = p_{xx} - \ell^2 \epsilon p + (f^2 - \epsilon \omega^2) (\frac{p_z}{N^2})_z. \quad (5.3)$$

The boundary conditions are

$$0 = w + h_x u + (f^2 - \omega^2)^{-1} [-(f r v_B + i \omega r \epsilon u_B)_x + \omega \ell \epsilon r v_B + i \ell f \epsilon r u_B] \quad (5.4a)$$

at  $z = -h(x)$ ,

$$0 = -\rho_0 w + i \omega \delta g^{-1} p + (f^2 - \epsilon \omega^2)^{-1} [(i \omega \epsilon T^x + f T^y)_x + \epsilon \ell (-i f T^x - \omega T^y)] \quad (5.4b)$$

at  $z = 0$ ,

$$0 = u_x \quad \text{at } x = XMAX, \quad (5.4c)$$

and

$$0 = -i(\ell f_p + \omega p_x)h + f(T^y - \rho_0 r v_B) + i \omega \epsilon (T^x - \rho_0 r u_B) \quad (5.4d)$$

at  $x = 0$ .

The variables  $u_B$  and  $v_B$  are the interior velocities evaluated at the bottom:

$$u_B = -i(f^2 - \epsilon\omega^2)^{-1}(\ell fp + \omega p_x) \Big|_{z = -h} \quad (5.5a)$$

and

$$v_B = (f^2 - \epsilon\omega^2)^{-1}(fp_x + \epsilon\omega \ell p) \Big|_{z = -h} . \quad (5.5b)$$

The parameter  $\delta$  is either 0 (rigid-lid surface) or 1 (free surface) at the user's discretion. Implicit in (5.4a,b) is the assumption that the surface and bottom frictional boundary layers are infinitesimally thin. The offshore boundary condition, (5.4c) has been shown to be reasonably accurate for free coastal-trapped waves (Brink, 1982b), and is applied here as well.

The coastal boundary condition (5.4d) has been justified by Clarke and Brink (1985). It states that the net onshore transport (interior plus Ekman) sums to zero, with the further assumption that  $u_z \cong 0$  at  $x = 0$ . In practice, this appears to be reasonable. The work of Mitchum and Clarke (1985) suggests that the "coast" be placed such that

$$h(0) = \frac{6r(0)}{f} , \quad (5.6)$$

where  $r(x)$  is defined by

$$\underline{T}_B = \rho_0 r v_B .$$

The general problem defined by (5.3) and (5.4) reduces to that of Clarke and Brink (1985) when

$$\delta = 0$$

$$\epsilon = 0$$

$$T^x = 0$$

$$T_x^y = 0 .$$

Note that the cross-shelf component of wind stress only enters when the long-wave assumption is not made. Our sensitivity studies suggest that the cross-shelf wind stress is rarely an effective driving agency except near resonance with a coastal-trapped wave.

### C. Program Input

The user provides an  $N^2$  profile, bottom topography information, choices of assumptions (e.g. rigid lid), the bottom resistance coefficient, wind stress profiles,  $f$ ,  $\omega$  and  $\ell$ . The program returns  $v$ ,  $u$ ,  $\rho$  and  $p$  in the form of amplitude and phase, as well as diagnostic information.

A full explanation of input is given here, and a compact listing in section 5E. All data are read from file 5.

#### line 1: ICCM

This is the number of  $(\omega, \ell)$  pairs for which the program will run. All other parameters stay the same for each run.

#### line 2: F XMAX

$F$  is the Coriolis parameter, multiplied by  $10^{-5} \text{ s}^{-1}$  within the program.

For example,  $F = 5.$  represents  $f = 5. \times 10^{-5} \text{ s}^{-1}$ .

XMAX is the offshore extent of the grid in km. Typically, XMAX should be about twice the shelf-slope width.

#### line 3: ILW IRL IXY

ILW determines whether the long-wave assumption is made. It is  $\epsilon$  in section 5B.

If ILW = 1, general frequency and wavenumber.

If ILW = 0, long-wave limit.

If ILW is neither 0 nor 1, the program defaults to ILW = 0.

IRL determines whether the rigid-lid assumption is made. It is  $\delta$  in section 5B.

If IRL = 1, free surface.

If IRL = 0, rigid lid.

If IRL is neither 0 nor 1, the program defaults to IRL = 0.

IXY determines which wind stress component is used.

If IXY = 1, cross-shelf ( $T^X$ ) winds.

If IXY = 0, alongshelf ( $T^Y$ ) winds.

If IXY is neither 0 nor 1, the program defaults to IXY = 0.

If the user sets IXY = 1 and ILW = 0, the program automatically stops and prints out an error message.

line 4: NRX

This is the number of  $[x, h(x)]$  pairs to be input.  $50 \geq NRX \geq 1$  is required.

line 5 and following: X H

These are the values of offshore distance ( $x$ ) in km and water depth ( $h$ ) in m. There must be NRX pairs, and the first pair must have  $x = 0$ . The spacing in  $x$  is arbitrary, and the program fills out the topography by linear interpolation. For values of  $x$  greater than the last value read, the program assigns the last depth read.

NR DZR ALPH

These are parameters used for reading the profile of  $N^2$  (the Brunt-Väisälä frequency squared).

NR is the vertical spacing of  $N^2$  values to be read.

DZR is the vertical spacing of  $N^2$  values in m.

ALPH describes the exponential tail of the  $N^2$  profile. Often  $N^2$  is not available from surface to bottom. In this case, an exponential extrapolation is used:

$$N^2 = N_0^2 \exp((\zeta_0 - \zeta)/ALPH)$$

where

$N_0^2$  is the last  $N^2$  value read,

$\zeta_0$  is the depth of the last  $N^2$  value read, and

$\zeta$  is the depth of the point, i.e.  $\zeta = -z$ .

ALPH is then the exponential length scale of  $N^2$  decay, in km.

CMLT

This is a conversion factor by which the input  $N^2$  are multiplied in order to get units of  $(\text{rad/s})^2$ . Specifically,

$$N^2(\text{rad}^2/\text{s}^2) = \text{CMLT} \times N^2(\text{user units}).$$

following lines:  $N^2$

These are the values of  $N^2$  in user units, one per line. There must be NR regularly spaced values. The first  $N^2$  value should be at  $z = 0$ , and  $N^2$  should never equal zero.

NF

This is the number of  $[x, r(x)]$  pairs to be read.  $NF \geq 1$  is required. The format for reading the bottom resistance coefficient  $r$  is exactly like that for depth  $h$ .

following lines: X R

These are values of offshore distance ( $x$ ) in km and of the resistance coefficient ( $r$ ) in cm/s. There must be NF pairs, and the first pair must have  $x = 0$ . The spacing in  $x$  is arbitrary, and the program fills out  $r$  by linear interpolation. For values of  $x$  greater than the last value read, the program assigns the last  $r$  value read.

NW

This is the number of values of  $T(x)$  to be read. Whether it is  $T^x$  or  $T^y$  depends upon the choice of IXY in line 3. If  $NW = 0$ ,  $T = 1$  dyne/cm<sup>2</sup> for all  $x$ .

following lines: X T

These are values of offshore distance ( $x$ ) in km and wind stress amplitude in dyne/cm<sup>2</sup>. If  $NW = 0$ , these lines should not be inserted. The first pair must be for  $x = 0$ . The  $x$  spacing is arbitrary, and the program fills out the wind stress by linear interpolation. For values of  $x$  greater than the last value read, the program assigns the last wind stress read.

following lines: W RL

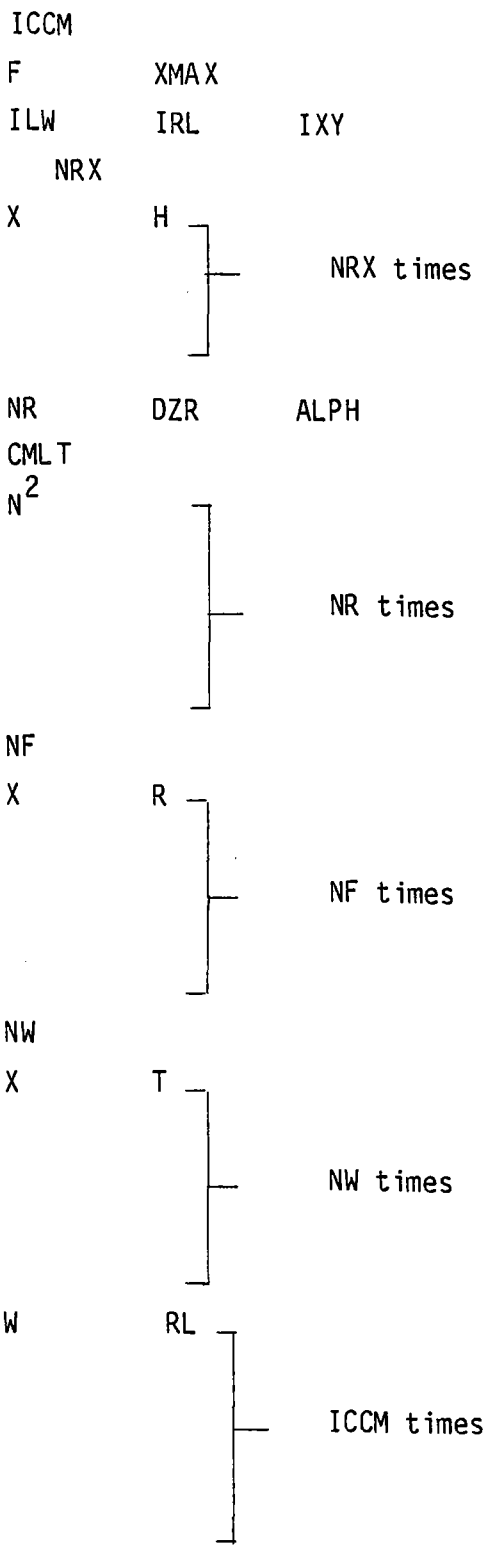
These are the frequency, wavenumber ( $\omega, \ell$ ) pairs for which the program runs. There should be ICCM lines. Units are  $s^{-1}$  and  $cm^{-1}$  respectively. The program includes no internal multiplications for these parameters.



D. General Comments

- i.) Using  $r = 0$  results in a divide by zero. Thus, inviscid problems should not be attempted.
- ii.) Using  $\lambda = 0$  causes no problem until the program is about to print the last values of pressure. An error message will result, but there is nothing wrong with the program's output, which is virtually complete.
- iii.) No external subroutines are required.
- iv.) The program uses the same 25 x 17 stretched grid as in BIGLOAD2.

E. Input Summary



F. Program Output

The program first lists  $f$  and  $XMAX$  in  $s^{-1}$  and  $cm$  respectively. The assumptions chosen on line 3 of input are then stated.

Input functions are then listed:

- i.)  $N^2$  ( $rad/s$ )<sup>2</sup> at  $x = XMAX$ , beginning at  $z = -h$  up to  $z = 0$  in increments of  $\Delta z$  at  $x = XMAX$  (i.e.  $h(XMAX)/16$ ).
- ii.)  $r(x)$  ( $cm/s$ ), beginning at  $x = 0$  out to  $x = XMAX$  in increments of  $\Delta x$  (i.e.  $XMAX/24$ ).
- iii.)  $T(x)$  ( $dyne/cm^2$ ) in the same format as  $r(x)$ .

Following this, the program prints out  $\omega$  ( $s^{-1}$ ) and  $\ell$  ( $cm^{-1}$ ), and the results for this particular input pair. All field variables ( $v$ ,  $u$ ,  $\rho$ ,  $p$ ) are listed as amplitude and phase at each grid point, beginning at the bottom for each  $x$ . For each  $x$ , water depth  $h$  ( $cm$ ) and  $\Delta x$  ( $cm$ ) are also given. The phase is negative for wind leading the response. The field variables are:

- iv.)  $v$  ( $cm/s$ ), followed by the  $v$  contribution to kinetic energy per unit length of coast ( $erg/cm$ ), and the alongshelf bottom stress beginning at  $x = 0$  ( $dyne/cm^2$ ).
- v.)  $u$  ( $cm/s$ ), followed by the  $u$  contribution to kinetic energy per unit length of coast ( $erg/cm$ ).
- vi.)  $\rho$  ( $\sigma_t$  units), followed by the  $\rho$  contribution to fluctuating potential energy per unit length of coast ( $erg/cm$ ). This is followed immediately by the free-surface height contribution to fluctuating potential energy. The free-surface contribution is set to zero if a rigid lid is imposed. At this point, the total (kinetic plus potential) fluctuating energy per unit length of coast ( $erg/cm$ ) is given, along with the ratio of kinetic to potential energy  $R$  (Brink, 1982b). For  $R \gtrsim 10$ , the response is generally highly barotropic, and for  $R \lesssim 2$ , it can be regarded as very baroclinic.
- vii.)  $p$  ( $dyne/cm^2$ ).

G. An Example

Input File:

```
1
10.0  200.
1      1      0
2
0.     30.
100.  4000.
2     5000.    5.
1.0 E-06
1.375
1.375
1
0.0    0.05
0
1.0 E-05      2.0 E-08
```

The resulting output has the following energy components:

$v \gg 2.94 \times 10^{14}$  erg/cm

$u \gg 0.08 \times 10^{14}$  erg/cm

$\rho \gg 0.31 \times 10^{14}$  erg/cm

$p \gg 0.45 \times 10^{12}$  erg/cm

and  $R = 9.6$ .

The alongshelf velocity (Figure 2) is uniform in depth at  $x = 0$  at 34.2 cm/s and a phase of  $-9^\circ$ . The  $v$  maximum is at the surface at  $x = 8.33$  km ( $85, -48^\circ$ ).

The cross-shelf velocity is depth-independent at  $x = 0$  ( $2.5, -21^\circ$ ), and has a maximum at the surface at  $x = 8.33$  km ( $6.6, -136^\circ$ ).

The maximum in density is at the bottom at  $x = 8.33$  km with  $\rho = 0.032 \sigma_t$  and a phase of  $-39^\circ$ . Density goes nearly to zero at the surface, and its phase is consequently unreliable there. When a rigid lid is used and  $NW = 0$ , density fluctuations are zero at the free surface in the long-wave limit.

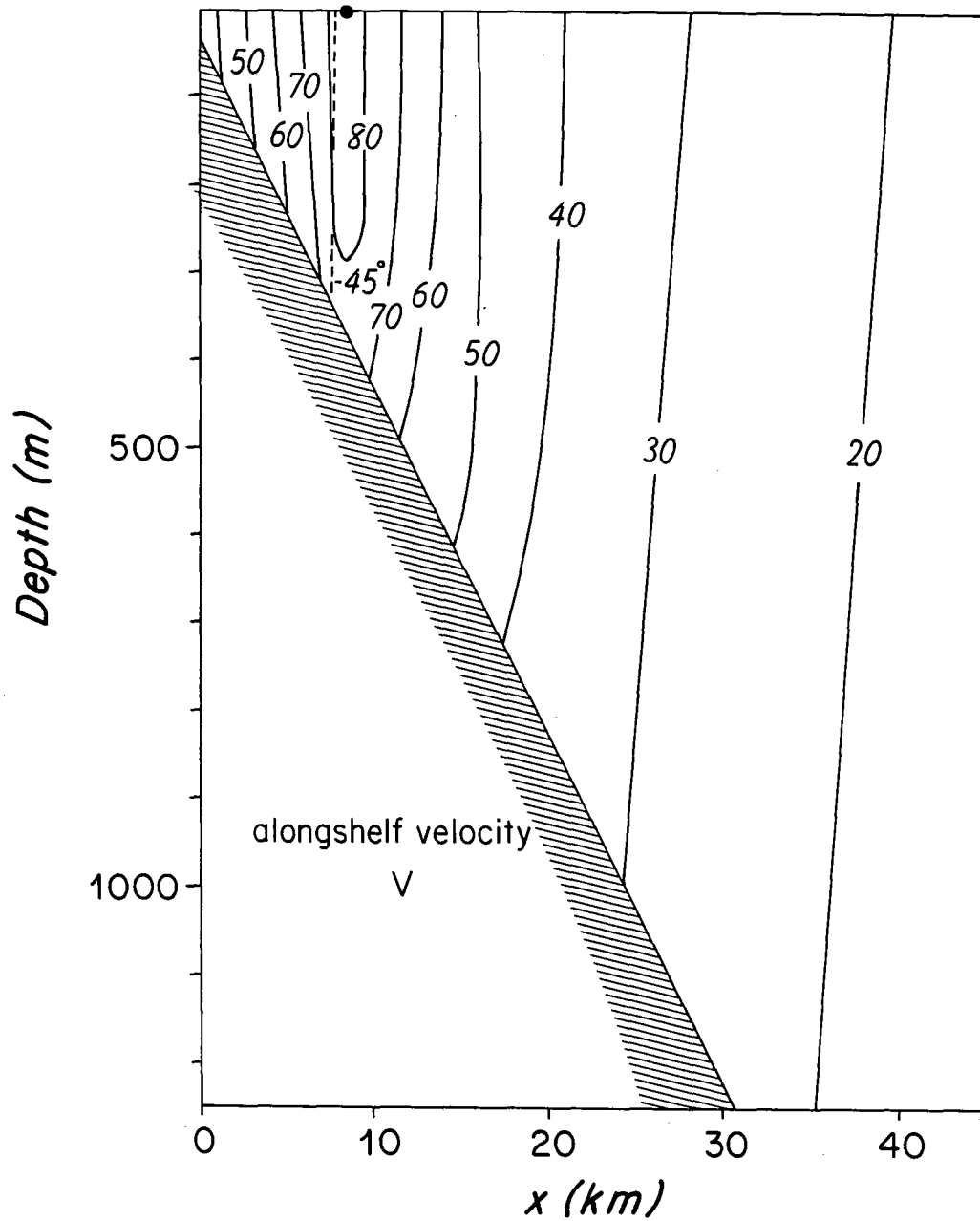


Figure 2: Alongshelf velocity for the example in section 5G. Amplitude (cm/s) is shown in solid lines and phase is shown by dashed contours. Only the upper 1250 m is shown.

The pressure is depth-independent and at a maximum at  $x = 0$  ( $0.190 \times 10^5$  dyne/cm<sup>2</sup>, phase =  $131^\circ$ ). To get sea level, divide by  $g = 981$  cm/s<sup>2</sup> to obtain 19 cm.

All fields become weaker far offshore and at great depth. The  $v$  and  $p$  fields have a roughly  $180^\circ$  phase change far offshore. The strength and structure of response vary radically near  $(\omega, \ell)$  resonances with free coastal-trapped waves.

### Word of Caution

We have performed what we feel are extensive tests with all of the programs contained herein. However, we cannot guarantee that the programs will give sensible results in all situations. That is, it may be possible to find parameter combinations for which a program will complete the run, but the computed results will not make physical sense. Therefore, we cannot be responsible for the ways in which the programs are applied. On the other hand, if actual programming bugs or inconsistencies appear which are not mentioned in this document, please contact us with the details.

### Acknowledgements

We thank the many people who helped us with these programs, both in setting up the original efforts, and by their comments as users.

This work was supported by National Science Foundation (NSF) grant OCE84-08563.



## REFERENCES

- Brink, K. H., 1982a. The effect of bottom friction on low-frequency coastal trapped waves. Journal of Physical Oceanography, 12, 127-133.
- Brink, K. H., 1982b. A comparison of long coastal trapped wave theory with observations off Peru. Journal of Physical Oceanography, 12, 897-913.
- Brink, K. H., 1983. Low-frequency free wave and wind-driven motions over a submarine bank. Journal of Physical Oceanography, 13, 103-116.
- Brink, K. H. and J. S. Allen, 1978. On the effect of bottom friction on barotropic motion over the continental shelf. Journal of Physical Oceanography, 8, 919-922.
- Buchwald, V. T. and J. K. Adams, 1968. The propagation of continental shelf waves. Proceedings of the Royal Society of London, A305, 235-250.
- Chapman, D. C., 1983. On the influence of stratification and continental shelf and slope topography on the dispersion of subinertial coastally trapped waves. Journal of Physical Oceanography, 13, 1641-1652.
- Clarke, A. J. and K. H. Brink, 1985. The response of stratified, frictional shelf and slope waters to fluctuating large-scale low-frequency wind forcing. Journal of Physical Oceanography, in press.
- Clarke, A. J. and S. Van Gorder, 1985. A method for estimating wind driven shelf and slope water flow. Submitted to Journal of Physical Oceanography.
- Huthnance, J. M., 1978. On coastal trapped waves: analysis and numerical calculation by inverse iteration. Journal of Physical Oceanography, 8, 74-92.
- Mitchum, G. T. and A. J. Clarke, 1985. A coastal boundary condition for large-scale, low-frequency flow. Submitted to Journal of Physical Oceanography.
- Mysak, L. A., P. M. LeBlond and W. J. Emery, 1979. Trench waves. Journal of Physical Oceanography, 9, 1001-1013.
- Pedlosky, J., 1979. Geophysical Fluid Dynamics. Springer-Verlag, New York, 624 pp.
- Wang, D-P. and C. N. K. Mooers, 1976. Coastal trapped waves in a continuously stratified ocean. Journal of Physical Oceanography, 6, 853-863.

Program BTCSW  
Listing

```

10      PROGRAM BTBW
20      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
30      DIMENSION BH(400),WW(3),RX(3),RLH(400),WH(400),CH(400)
40      DIMENSION WF(400)
50      C
60      C          MAIN PROGRAM CALLS ALL OF THE PIECES
70      C
80      C          NN LESS THAN OR EQUAL 100
90      IMD = 1
100     READ(5,*) IMDM,NN
110     C          DD1 = 1.  FREE SURFACE
120     C          DD1 = 0.  RIGID LID
130     C          IDD3 = 1  U = 0 AT X = XMAX
140     C          IDD3 = 0  UX = 0 AT X = XMAX
150     C          IDD3 = 2 REAL B.C. AT X = XMAX FOR VX,HX = 0. THERE
160     C          IDD4 =1   U = 0 AT X = 0
170     C          IDD4 = 0  UX = 0 AT X = 0
180     C          IDD4 = 2 DECAY B.C. AT X = 0 FOR VX,HX = 0 THERE
190     C          IPC = 0  REDUCED PRINT OUT
200     C          IUP = 0 SEARCH IN U
210     C          IUP = 1 SEARCH IN P
220     C          ILLW = 0 STRICTLY LONG WAVE
230     C          ILLW = 1 GENERAL FREQUENCY AND WAVENUMBER
240     C          REVISED 11/9/84
250     2      READ(5,*) NITM,ISD,EPS,DEL
260     READ(5,*) IUP,ILLW
270     READ(5,*) IDD1,IDD3,IDD4
280     DD1 = FLOAT(IDD1)
290     NCAL = 1
300     READ(5,*) NCALM,ILW
310     READ(5,*) RLF,DRL
320     READ(5,*) IPC
330     DRL = DRL*1.0E-08
340     RLF = RLF*1.0E-08
350     READ(5,*) F,XMAX
360     XMAX = XMAX*1.0E+05
370     F = F*1.0E-05
380     DX = XMAX/FLOAT(NN-1)
390     WRITE(6,907) F,XMAX,DX
400     IF (IUP.NE.0) GO TO 315
410     312   WRITE(6,924)
420     IUP = 0
430     GO TO 320
440     315   IF (IUP.NE.1) GO TO 312
450     WRITE(6,925)
460     320   IF (IDD4.NE.0) GO TO 305
470     WRITE(6,921)
480     GO TO 310
490     305   IF (IDD4.EQ.2) GO TO 306
500     IDD4 = 1
510     WRITE(6,922)
520     GO TO 310
530     306   WRITE(6,928)
540     310   IF (IDD1.EQ.1) GO TO 3
550     DD1 = 0.
560     WRITE(6,917)
570     GO TO 4
580     3     WRITE(6,918)
590     4     IF (IDD3.EQ.1) GO TO 6
600     IF (IDD3.EQ.2) GO TO 7

```

```

610      IDD3 = 0
620      WRITE(6,919)
630      GO TO 8
640      6      WRITE(6,920)
650      GO TO 8
660      7      WRITE(6,923)
670      8      IF (ILLW.EQ.1) GO TO 11
680      ILLW = 0
690      NCALM = 1
700      WRITE(6,926)
710      GO TO 12
720      11     WRITE(6,927)
730      12     REPS = FLOAT(ILLW)
740      CALL DEP(NN,XMAX)
750      CALL VRD(NN,XMAX,F)
760      DO 10 I = 1,NCALM
770      10     RLH(I) = DRL*FLOAT(I-1) + RLF
780      RL = RLH(NCAL)
790      READ(5,*) (WW(J),J=1,3)
800      DO 5 J = 1,3
810      5      WW(J) = WW(J)*1.0E-05
820      1      RB = 0.
830      WRITE(6,908)
840      DO 18 J = 1,3
850      W = WW(J)
860      IF (IUP.EQ.1) GO TO 330
870      CALL MUTS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
880      GO TO 9
890      330    CALL MATS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
900      9      CP = W/RL
910      CALL CRLC(W,RL,NN,ICR)
920      WRITE(6,905) W,RL,CP,RI,ICR
930      RX(J) = RI
940      IF (RI.LT.RB) GO TO 18
950      RB = RI
960      ICRH = ICR
970      WB = W
980      DO 15 I = 1,NN
990      15     BH(I) = B(I)
1000     18     CONTINUE
1010     NIT = 3
1020     IGP = 0
1030     20     CALL NGW(WW,RX,WB,WN,ISUC,EPS,DEL,IN,ISD,IGP)
1040     IF (ISUC.EQ.1) GO TO 100
1050     IF (IUP.EQ.1) GO TO 340
1060     CALL MUTS(NN,XMAX,DD1,F,WN,RL,IDD3,RI,IDD4,REPS)
1070     GO TO 345
1080     340    CALL MATS(NN,XMAX,DD1,F,WN,RL,IDD3,RI,IDD4,REPS)
1090     345    IF (IN.NE.0) GO TO 29
1100     IF (RI.GT.RB) GO TO 21
1110     IF (WN.LT.WW(2)) GO TO 23
1120     22     IN = 3
1130     GO TO 29
1140     23     IN = 1
1150     GO TO 29
1160     21     IF (WN.GT.WW(2)) GO TO 23
1170     GO TO 22
1180     29     RX(IN) = RI
1190     WW(IN) = WN
1200     CP = WN/RL

```

```

1210      CALL CRLC(WN,RL,NN,ICR)
1220      WRITE(6,905) WN,RL,CP,RI,ICR
1230      IF ( RI.LT.RB) GO TO 30
1240      WB = WN
1250      ICRH = ICR
1260      RB = RI
1270      DO 25 I = 1,NN
1280 25    BH(I) = B(I)
1290 30    NIT = NIT + 1
1300      IF ( NIT.LT.NITM) GO TO 20
1310      WRITE(6,902) NIT
1320      GO TO 140
1330 100   DO 110 I = 1,NN
1340 110   BH(I) = BH(I)/RB
1350      CC = WB/RL
1360      IF ( ICRH.EQ.0) GO TO 115
1370      WRITE(6,911) ICRH
1380 115   WRITE(6,903) WB,RL,CC,EPS,DX,NIT
1390      IF (NCAL.EQ.1) GO TO 120
1400      IF (IPC.EQ.0) GO TO 150
1410 120   IF (IUP.EQ.1) GO TO 350
1420      WRITE(6,913)
1430      GO TO 360
1440 350   WRITE(6,912)
1450 360   WRITE(6,904) (BH(I),I=1,NN)
1460      IF (IUP.EQ.0) GO TO 135
1470      CALL MUTS(NN,XMAX,DD1,F,WB,RL,IDD3,RI,IDD4,REPS)
1480      DO 143 I = 1,NN
1490 143   B(I) = B(I)/RI
1500      WRITE(6,913)
1510      WRITE(6,904) (B(I),I = 1,NN)
1520 135   IF (RH(1).EQ.0.0) GO TO 150
1530      CALL MATS(NN,XMAX,DD1,F,WB,RL,IDD3,RI,IDD4,REPS)
1540      DO 130 I = 1,NN
1550 130   B(I) = B(I)/RI
1560      IF (IUP.EQ.1) GO TO 145
1570      WRITE(6,912)
1580      WRITE(6,904) (B(I),I=1,NN)
1590 145   IF (ILW.EQ.0) GO TO 150
1600      IF (NCAL.NE.1) GO TO 150
1610      IF ( RH(1).EQ.0.) GO TO 150
1620      IF (IDD4.NE.1) GO TO 150
1630      WRITE(6,908)
1640      CALL LGWV(NN,XMAX,F,WF,WB,RL)
1650 150   WH(NCAL) = WB
1660      CH(NCAL) = CC
1670      NCAL = NCAL + 1
1680      IF ( NCAL.GT.NCALM) GO TO 250
1690      RL = RLH(NCAL)
1700      IF (NCAL.GE.3) GO TO 200
1710      WW(2) = CC*RL
1720      GO TO 205
1730 200   I1 = NCAL - 2
1740      I2 = NCAL - 1
1750      CG = (WH(I2) - WH(I1))/(RLH(I2) - RLH(I1))
1760      WW(2) = WH(I2) + CG*(RLH(NCAL)-RLH(I2))
1770 205   WW(1) = WW(2)*(1. - DEL)
1780      WW(3) = WW(2)*(1. +DEL)
1790      GO TO 1
1800 250   WRITE(6,909)

```

```

1810      DO 260 I = 1,NCALM
1820 260  WRITE(6,910) WH(I),RLH(I),CH(I)
1830      IMD = IMD +1
1840      IF (IMD.LE.IMDM) GO TO 2
1850 902  FORMAT(//° USED UP°,I3,° ITERATIONS°//)
1860 903  FORMAT(//° CONVERGED: W,L,C,EPS,DX,NIT =°,5E15.5,I5)
1870 904  FORMAT(10E13.5)
1880 905  FORMAT(//° W,RL,CP,RI,ICR =°,4E15.5,I10)
1890 907  FORMAT(//° F,XMAX,DX =°,3E15.5//)
1900 908  FORMAT(///)
1910 909  FORMAT(///°          W          L          C°//)
1920 910  FORMAT(3E15.5)
1930 911  FORMAT(//° SOLUTION HAS °,I3,° CRITICAL LAYERS°//)
1940 912  FORMAT(//° ZETA°//)
1950 913  FORMAT(//° U°//)
1960 917  FORMAT(//° RIGID LID°//)
1970 918  FORMAT(//° FREE SURFACE°//)
1980 919  FORMAT(° UX = 0 AT X = XMAX°//)
1990 920  FORMAT(° U = 0 AT X = XMAX°//)
2000 921  FORMAT(° UX = 0 AT X = 0°//)
2010 922  FORMAT(° U = 0 AT X = 0°//)
2020 923  FORMAT(° REAL B.C.AT X = XMAX°)
2030 924  FORMAT(//° SEARCH IN U°)
2040 925  FORMAT(//° SEARCH IN P°)
2050 926  FORMAT(° LONG WAVE EXACTLY°)
2060 927  FORMAT(° GENERAL FREQUENCY AND WAVENUMBER°)
2070 928  FORMAT(° DECAY CONDITION AT X = 0°)
2080 140  STOP
2090      END
2100
2110      SUBROUTINE NGW(WW,RX,WB,WN,ISUC,EPS,DEL,IN,ISD,IGP)
2120      DIMENSION WW(3),RX(3)
2130      C
2140      C          SUBROUTINE TO GUESS THE NEXT W
2150      C
2160 5      IC = 0
2170      DO 10 I = 1,2
2180      I1 = I +1
2190      AA = ABS(WW(I1))
2200      BB = ABS(WW(I))
2210      IF (AA.GT.BB) GO TO 10
2220      IC = 1
2230      RI = RX(I)
2240      WX = WW(I)
2250      RX(I) = RX(I1)
2260      WW(I) = WW(I1)
2270      WW(I1) = WX
2280      RX(I1) = RI
2290 10     CONTINUE
2300      IF (IC.NE.0) GO TO 5
2310      ISUC = 0
2320      IF (RX(3).GT.RX(2)) GO TO 150
2330      IF (RX(1).GT.RX(2)) GO TO 160
2340      IL = 1
2350      IH = 3
2360      IF (RX(3).GT.RX(1)) GO TO 9
2370      IL = 3
2380      IH = 1
2390 9      WB1 = (WW(1) +WW(2))/2.
2400      WB2 = (WW(2) + WW(3))/2.

```

```

2410      RW1 = (RX(2) - RX(1))/(WW(2) - WW(1))
2420      RW2 = (RX(3) - RX(2))/(WW(3) - WW(2))
2430      A = (RW2 - RW1)/(WB2 - WB1)
2440      B = RW1 - A*WB1
2450      WN = -B/A
2460      EP = (WN - WB)/WB
2470      EP = ABS(EP)
2480      IF ( EP.LT.EPS) GO TO 100
2490      IGP = IGP + 1
2500      IF (WN.LT.WW(1)) GO TO 15
2510      IF (WN.LT.WW(3)) GO TO 20
2520      15      WN = (WW(2) + WW(IH))/2.
2530      20      IN = 0
2540      GO TO 130
2550      100     ISUC = 1
2560      IF (IGP.NE.0) GO TO 130
2570      IGP = IGP + 1
2580      ISUC = 0
2590      WN = (WW(2) + WW(IL))/2.
2600      GO TO 20
2610      150     IF (RX(1).GT.RX(2)) GO TO 180
2620      155     IF (ISD.EQ.1) GO TO 165
2630      WN = WW(3)*(1. + DEL)
2640      WW(1) = WN
2650      RX(1) = 0.
2660      IN = 1
2670      GO TO 130
2680      160     IF (RX(3).GT.RX(2)) GO TO 180
2690      165     IF (ISD.EQ.-1) GO TO 155
2700      WN = WW(1)*(1. - DEL)
2710      WW(3) = WN
2720      RX(3) = 0.
2730      IN = 3
2740      GO TO 130
2750      180     IF (RX(3).GT.RX(1)) GO TO 155
2760      GO TO 165
2770      130     RETURN
2780      END
2790
2800      SUBROUTINE CALC(NN,RX)
2810      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
2820      DOUBLE PRECISION RI,DSQRT
2830      C
2840      C          SUBROUTINE TO CALCULATE THE INTEGRAL OF RESPONSE SQUARED
2850      C
2860      RI = (B(1)**2 + B(NN)**2)/2.
2870      NX = NN - 1
2880      DO 5 I = 2,NX
2890      5      RI = RI + B(I)**2
2900      RI = DSQRT(RI)
2910      RX = RI
2920      RETURN
2930      END
2940
2950      SUBROUTINE DEP(NN,XMAX)
2960      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
2970      READ(5,*) NRX
2980      C
2990      C          SUBROUTINE TO READ AND INTERPOLATE THE DEPTH PROFILE
3000      C          RH = DEPTH

```

```

3010      C                      RHX = X DERIVATIVE OF DEPTH
3020      C
3030      DD 10 I = 1,NRX
3040      READ(5,*) A(I),B(I)
3050      A(I) = A(I)*1.0E+05
3060      10      B(I) = B(I)*100.
3070      DX = XMAX/FLOAT(NN-1)
3080      RH(1) = B(1)
3090      DD 20 N = 2,NN
3100      X = DX*FLOAT(N-1)
3110      IF (X.GT.A(NRX)) GO TO 15
3120      IC = 0
3130      DD 8 J = 2,NRX
3140      IF (IC.NE.0) GO TO 8
3150      I = J
3160      IF (X.GT.A(I)) GO TO 8
3170      IC = I
3180      8
3190      CONTINUE
3200      IM = I -1
3210      AA = (B(I) -B(IM))/(A(I) - A(IM))
3220      XX = X - A(IM)
3230      RH(N) = B(IM) +AA*XX
3240      GO TO 20
3250      15
3260      20      CONTINUE
3270      RHX(1) = (RH(2) - RH(1))/DX
3280      NM = NN -1
3290      RHX(NN) = (RH(NN) - RH(NM))/DX
3300      D2 = 2.*DX
3310      DD 30 N = 2,NM
3320      IP = N +1
3330      IM = N -1
3340      30      RHX(N) = (RH(IP) - RH(IM))/D2
3350      WRITE(6,903)
3360      WRITE(6,902) (RH(N),N=1,NN)
3370      902      FORMAT(10E13.5)
3380      903      FORMAT(//' DEPTH IN CM'//)
3390      RETURN
3400      END
3410      SUBROUTINE LGWV(NN,XMAX,F,WFF,WB,RL)
3420      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
3430      DIMENSION WFF(400)
3440      C
3450      C                      SUBROUTINE TO CALCULATE THE BN AND ANN IN THE LONG WAVE
3460      C                      LIMIT
3470      C
3480      DX = XMAX/FLOAT(NN -1)
3490      DXX = 2.*DX
3500      GF = 980./F
3510      CALL WFR(NN,XMAX,WFF)
3520      A(1) = 0.
3530      DD 100 N = 2,NN
3540      A(N) = (RHX(N)*B(N) + RHX(1)*B(1))/2.
3550      NX = N - 1
3560      IF (NX.LE.1) GO TO 90
3570      DD 20 I = 2,NX
3580      20      A(N) = A(N) + RHX(I)*B(I)
3590      90      A(N) = A(N)*DX
3600      A(N) = A(N) + RH(1)*B(1) -RH(N)*B(N)

```



```

3610 100 A(N) = -GF*A(N)
3620 NX = NN - 1
3630 WF = RHX(1)*(A(1)/RH(1))**2
3640 WF = WF + RHX(NN)*(A(NN)/RH(NN))**2
3650 IFZ = 0
3660 WF = WF/2.
3670 DO 150 N = 2,NX
3680 IF ( IFZ.NE.0) GO TO 150
3690 IF (RHX(N).NE.0.) GO TO 140
3700 IFZ = N
3710 140 WF = WF + RHX(N)*(A(N)/RH(N))**2
3720 150 CONTINUE
3730 WF = WF*DX
3740 WF = SQRT(WF*WF)
3750 WF = SQRT(WF)
3760 DD 160 N = 1,NN
3770 160 A(N) = A(N)/WF
3780 NH = IFZ - 1
3790 PX = (A(2) - A(1))/(DX*RH(1))
3800 WF = 0.5*PX*PX*WFF(1)
3810 BN = 0.5*RHX(NH)*A(NH)/(RH(NH)**2)
3820 NH = NH - 1
3830 DD 180 N = 2,NH
3840 BN = BN + RHX(N)*A(N)/(RH(N)**2)
3850 I1 = N - 1
3860 I2 = N + 1
3870 PX = (A(I2) - A(I1))/DXX
3880 PX = PX/RH(N)
3890 180 WF = WF + PX*PX*WFF(N)
3900 BN = BN*DX
3910 WF = WF*RH(1)*DX
3920 WRITE(6,905)
3930 WRITE(6,901) WF
3940 WRITE(6,903) BN
3950 WRITE(6,904)
3960 WRITE(6,902) (A(I),I=1,NN)
3970 C = WB/RL
3980 NX = NN - 1
3990 WF = RHX(1)*B(1)*B(1)*C.5
4000 DO 210 I = 2,NX
4010 210 WF = WF + RHX(I)*B(I)*B(I)
4020 WF = WF + 0.5*RHX(NN)*B(NN)*B(NN)
4030 WF = WF*DX
4040 WF = WF + RH(1)*B(1)*B(1)
4050 WF = SQRT(WF)
4060 DO 215 I = 1,NN
4070 215 B(I) = B(I)/WF
4080 BX = -F*B(1)/C
4090 WF = 0.5*WFF(1)*BX*BX
4100 DO 220 I = 2,NX
4110 IP = I + 1
4120 IM = I - 1
4130 BX = (B(IP) - B(IM))/DXX
4140 220 WF = WF + WFF(I)*BX*BX
4150 WF = WF*DX/F
4160 BN = B(1)
4170 WRITE(6,906)
4180 WRITE(6,907) WF
4190 WRITE(6,908) BN
4200 WRITE(6,909)

```

```

4210      WRITE(6,902) (B(I),I = 1,NN)
4220  901   FORMAT(//'  ANN =',E15.5,' 1/CM')
4230  902   FORMAT(10E13.5)
4240  903   FORMAT('/'  BN =',E15.5,' CM-1/2'/)
4250  904   FORMAT(//'  PHI ='/)
4260  905   FORMAT(' USING STREAM FUNCTION')
4270  906   FORMAT(//' USING PRESSURE')
4280  907   FORMAT(' ANN = ',E15.5,' SEC/CM2')
4290  908   FORMAT(' BN =',E15.5,' CM-1/2')
4300  909   FORMAT(' NORMALIZED PRESSURE (CM-1/2)')
4310      RETURN
4320      END
4330
4340      SUBROUTINE MUTS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
4350      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
4360      DOUBLE PRECISION A1,A2,A3,AL,DSQRT
4370  C
4380  C          SUBROUTINE TO SET UP AND SOLVE THE U EQUATION
4390  C
4400      NU = NN
4410      IF (IDD3.NE.1) GO TO 2
4420      NU = NN - 1
4430  2      IF (IDD4.NE.1) GO TO 4
4440      NU = NU - 1
4450  4      DX = XMAX/FLGAT(NN-1)
4460  5      GG = DD1/980.
4470      RLL = RL*RL
4480      DXX = DX*DX
4490      DDD = DX/2.
4500      DO 100 N = 1,NU
4510      N1 = N
4520      IF (IDD4.NE.1) GO TO 8
4530      N1 = N + 1
4540  8      B(N) = 0.
4550      WP = W + RL*V(N1)
4560      W2 = WP*WP
4570      RR = RH(N1)
4580      RX = RHX(N1)
4590      WX = RL*VX(N1)
4600      FP = F + VX(N1)
4610      A1 = RR*(GG*W2 - RR*RLL)
4620      A2 = RR*(RX*RLL - 2.0*WP*WX*GG)
4630      A3 = -GG*GG*W2*(F*FP - REPS*W2)
4640      A3 = A3 + REPS*RR*RR*RLL*RLL
4650      A3 = A3 + GG*RR*RLL*(-2.0*W2*REPS + F*FP + 2.0*FP*VX(N1))
4660      A3 = A3 - RR*RLL*RL*FP*RX/WP
4670      A3 = A3 - RR*RL*VXX(N1)*(GG*W2 - RR*RLL)/WP
4680      N1 = N + NU
4690      N2 = N + 2*NU
4700      A(N) = -2.*A1 + A3*DXX
4710      A(N1) = A1 - A2*DDD
4720      A(N2) = A1 + A2*DDD
4730      IF (N.EQ.1) GO TO 20
4740      IF (N.EQ.NU) GO TO 30
4750      GO TO 100
4760  20     IF (IDD4.NE.0) GO TO 25
4770      A(N2) = A(N1) + A(N2)
4780      A(N1) = 0.
4790      GO TO 100
4800  25     IF (IDD4.EQ.1) GO TO 28

```

```

4810      WRITE(6,*) A1,A2,A3,REPS,GG,RR,W2,RLL,FP,F,RX,WP,VX(N1),VXX(N1)
4820      AL = DSQRT(-A3/A1)
4830      A(N2) = A(N2) + A(N1)
4840      A(N) = A(N) -2.0*DX*AL*A(N1)
4850      28      A(N1) = 0.
4860      GO TO 100
4870      30      IF (IDD3.NE.0) GO TO 35
4880      A(N1) = A(N1) + A(N2)
4890      A(N2) = 0.
4900      GO TO 100
4910      35      IF (IDD3.EQ.1) GO TO 36
4920      AL = DSQRT(-A3/A1)
4930      A(N1) = A(N1) +A(N2)
4940      A(N) = A(N) -2.0*DX*AL*A(N2)
4950      36      A(N2) = 0.0
4960      100     CONTINUE
4970      B(5) = 1.
4980      CALL TRI(NU)
4990      IF (IDD4.NE.1) GO TO 140
5000      DO 120 I = 1,NU
5010      N = NU -I +1
5020      N1 = N + 1
5030      120     B(N1) = B(N)
5040      B(1) = 0.
5050      140     IF (IDD3.NE.1) GO TO 150
5060      B(NN) = 0.
5070      150     CALL CALC(NN,RI)
5080      RETURN
5090      END
5100
5110      SUBROUTINE VRD(NN,XMAX,F)
5120      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
5130      C
5140      C          SUBROUTINE TO READ AND INTERPOLATE THE MEAN V PROFILE
5150      C          V = ALONGSHORE VELOCITY
5160      C          VX = X DERIVATIVE OF V
5170      C          VXX = SECOND X DERIVATIVE OF V
5180      C
5190      DX = XMAX/FLOAT(NN-1)
5200      READ(5,*) NRD
5210      IF (NRD.EQ.0) GO TO 200
5220      DO 5 I = 1,NRD
5230      READ(5,*) A(I),B(I)
5240      C          A = DISTANCE FROM SHORE IN KM
5250      C          B = V IN CM/SEC
5260      5      A(I) = A(I)*1.0E+05
5270      10     DO 100 N = 1,NN
5280      X = DX*FLOAT(N-1)
5290      IF (X.GT.A(NRD)) GO TO 90
5300      IF (X.LT.A(1)) GO TO 80
5310      J = 0
5320      DO 20 I = 2,NRD
5330      IF (J.NE.0) GO TO 20
5340      IF (X.GT.A(I)) GO TO 20
5350      J = I
5360      20     CONTINUE
5370      JJ = J - 1
5380      XX = A(J) - X
5390      XY = A(J) - A(JJ)
5400      V(N) = B(J) -XX*(B(J) -B(JJ))/XY

```

```

5410          GO TO 100
5420      80    V(N) = 0.
5430          GO TO 100
5440      90    V(N) = B(NRD)
5450     100    CONTINUE
5460          VX(1) = (V(2) - V(1))/DX
5470          NQ = NN - 1
5480          VX(NN) = (V(NN) - V(NQ))/DX
5490          DXX = DX*DX
5500          DZ = 2.*DX
5510          VXX(1) = 0.
5520          VXX(NN) = 0.
5530          DO 120 N = 2,NQ
5540          I1 = N - 1
5550          I2 = N + 1
5560          VX(N) = (V(I2) - V(I1))/DZ
5570     120    VXX(N) = (V(I2) - 2.*V(N) + V(I1))/DXX
5580          WRITE(6,904)
5590          WRITE(6,903) (V(N),N=1,NN)
5600          IST = 0
5610          DO 150 N = 1,NN
5620          Q = -RHX(N)*(F+VX(N)) + RH(N)*VXX(N)
5630          A(N) = Q
5640          IF (IST.NE.0) GO TO 150
5650          IF (N.NE.1) GO TO 130
5660          PQ = Q
5670          GO TO 150
5680     130    IF (PQ.EQ.0.) GO TO 135
5690          RQ = Q/PQ
5700          IF (RQ.LT.0.) GO TO 140
5710          IF (Q.EQ.0.) GO TO 150
5720     135    PQ = Q
5730          GO TO 150
5740     140    IST = 1
5750     150    CONTINUE
5760          IF (IST.EQ.0) GO TO 250
5770          WRITE(6,905)
5780          WRITE(6,906)
5790          WRITE(6,905)
5800          WRITE(6,907)
5810          WRITE(6,903) (A(N),N=1,NN)
5820          GO TO 250
5830     200    DO 220 N = 1,NN
5840          V(N) = 0.
5850          VX(N) = 0.
5860     220    VXX(N) = 0.
5870     903    FORMAT(10E13.5)
5880     904    FORMAT('//° V IN CM/SEC°/)
5890     905    FORMAT('/° *****°/)
5900     906    FORMAT(' POSSIBILITY OF UNSTABLE MODES°)
5910     907    FORMAT('//° PBARX°/)
5920     250    RETURN
5930          END
5940
5950          SUBROUTINE MATS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
5960          COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
5970          DOUBLE PRECISION A1,A2,A3,B1,B2,B3,BB1,BB2,BX,DSQRT,AL
5980      C
5990      C          SUBROUTINE TO SET UP AND SOLVE THE PRESSURE EQUATION
6000      C

```

```

6010      DX = XMAX/FLOAT(NN - 1)
6020      GG = DD1/980.
6030      RLL = RL*RL*REPS
6040      DXX = DX*DX
6050      DDX = 2.*DX
6060      DDD = DX/2.
6070      DO 100 N = 1,NN
6080      B(N) = 0.
6090      WP = W + RL*V(N)
6100      FP = F + VX(N)
6110      FW = F*FP - WP*WP*REPS
6120      RR = RH(N)
6130      RX = RHX(N)
6140      WX = RL*VX(N)
6150      FLW = F*RL/WP
6160      A1 = FW*RR
6170      A2 = FW*RX - RR*(F*VXX(N) - 2.0*WP*WX*REPS)
6180      A3 = -GG*FW*FW + RX*FLW*FW - RR*FLW*(F*VXX(N) - 2.*WP*WX*REPS)
6190      A3 = A3 - RR*RLL*FW
6200      N1 = N + NN
6210      N2 = N + 2*NN
6220      A(N) = -2.*A1 + A3*DXX
6230      A(N1) = A1 - A2*DDD
6240      A(N2) = A1 + A2*DDD
6250      IF (N.EQ.1) GO TO 20
6260      IF (N.EQ.NN) GO TO 30
6270      GO TO 100
6280  20    IF (IDD4.NE.0) GO TO 25
6290      B1 = RL*RR*FP
6300      B2 = RLL*WP*RR + GG*WP*FW
6310      A(N) = A(N) + 2.0*DX*B2*A(N1)/B1
6320      A(N2) = A(N2) + A(N1)
6330      A(N1) = 0.
6340      GO TO 100
6350  25    IF (IDD4.NE.1) GO TO 27
6360      A(N2) = A(N2) + A(N1)
6370      A(N) = A(N) + DDX*FLW*A(N1)
6380      A(N1) = 0.
6390      GO TO 100
6400  27    AL = DSQRT(-A3/A1)
6410      A(N2) = A(N2) + A(N1)
6420      A(N) = A(N) - 2.0*DX*AL*A(N1)
6430      A(N1) = 0.
6440      GO TO 100
6450  30    IF (IDD3.EQ.1) GO TO 35
6460      IF (IDD3.EQ.2) GO TO 32
6470      B1 = -RR*WP*FW
6480      B2 = -FW*(RR*RL*F + RX*WP + RR*WX)
6490      B2 = B2 + RR*WP*(F*VXX(N) - 2.0*WP*WX*REPS)
6500      B3 = RL*F*(-RX*FW + RR*(F*VXX(N) - 2.*WP*WX*REPS))
6510      BX = 1. + DDD*B2/B1
6520      BB1 = -(-2. + B3*DXX/B1)/BX
6530      BB2 = -(1. - B2*DDD/B1)/BX
6540      A(N) = A(N) + A(N2)*BB1
6550      A(N1) = A(N1) + A(N2)*BB2
6560      A(N2) = 0.
6570      GO TO 100
6580  32    AL = DSQRT(-A3/A1)
6590      A(N1) = A(N1) + A(N2)
6600      A(N) = A(N) - 2.0*DX*AL*A(N2)

```

```

6610      A(N2) = 0.
6620      GO TO 100
6630      35      A(N1) = A(N1) + A(N2)
6640      A(N) = A(N) -DDX*FLW*A(N2)
6650      A(N2) = 0.
6660      10C     CONTINUE
6670      B(5) = C.00001
6680      EQQ = 1.0E-36
6690      IKK = 0
6700      DO 105 I = 1,NN
6710      AQ = ABS(A(I))
6720      IF (AQ.GT.EQQ) GO TO 105
6730      IKK = 1
6740      105     CONTINUE
6750      IF (IKK.EQ.1) GO TO 115
6760      CALL TRI(NN)
6770      CALL CALC(NN,RI)
6780      GO TO 110
6790      115     WRITE(6,901)
6800      901     FORMAT(' NUMERICAL PROBLEM: SMALL DIAGONAL ELEMENT')
6810      11C     RETURN
6820      END
6830
6840      SUBROUTINE CRLC(W,RL,NN,ICR)
6850      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
6860      C
6870      C          SUBROUTINE TO CHECK FOR CRITICAL LAYERS
6880      C
6890      ICR = 0
6900      DO 100 N = 1,NN
6910      WP = W + RL*V(N)
6920      IF (N.NE.1) GO TO 30
6930      PQ = WP
6940      GO TO 100
6950      30      PQ = WP/PQ
6960      IF (PQ.GT.C.) GO TO 40
6970      ICR = ICR +1
6980      40      PQ = WP
6990      100     CONTINUE
7000      RETURN
7010      END
7020
7030      SUBROUTINE TRI(N)
7040      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
7050      DOUBLE PRECISION AA,BB
7060      C
7070      C
7080      C          SUBROUTINE TO SOLVE A TRIDIAGONAL MATRIX BY GAUSSIAN
7090      C          ELIMINATION
7100      C
7110      C          STORE MAIN DIAGONAL FIRST
7120      C          0, LOWER DIAGONAL
7130      C          UPPER DIAGONAL, 0
7140      N3 = N +1
7150      N2 = 2*N
7160      NN = N - 1
7170      DO 5 I = 1,NN
7180      II = N3 + I
7190      AA = A(II)/A(I)
7200      I2 = I + 1

```

```

7210      I3 =N2 + I
7220      BB = A(I2) -AA*A(I3)
7230      A(I2) = BB
7240      BB = B(I2) -AA*B(I)
7250      5  B(I2) = BB
7260      DO 10 IJ = 1,NN
7270      I = N3 - IJ
7280      B(I) = B(I)/A(I)
7290      II = I + N2 -1
7300      IP = I -1
7310      BB = B(IP) -A(II)*B(I)
7320      10  B(IP) = BB
7330      B(I) = B(I)/A(I)
7340      RETURN
7350      END
7360
7370      SUBROUTINE WFR(NN,XMAX,WF)
7380      COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
7390      DIMENSION WF(400)
7400      C
7410      C          SUBROUTINE TO READ AND INTERPOLATE FRICTIONAL WEIGHT
7420      C          FUNCTIONS, WF
7430      C
7440      READ(5,*) NW
7450      IF (NW.EQ.0) GO TO 100
7460      DO 5 I = 1,NW
7470      II = 100 + I
7480      READ(5,*) A(I),A(II)
7490      5  A(I) = A(I)*1.0E+05
7500      DX = XMAX/FLOAT(NN-1)
7510      WF(1) = A(101)
7520      NMM = 100 + NW
7530      WFM = A(NMM)
7540      DO 50 N = 2,NN
7550      X = DX*FLOAT(N-1)
7560      IF (X.GT.A(NW)) GO TO 48
7570      IC = 0
7580      DO 46 J = 2,NW
7590      IF (IC.NE.0) GO TO 46
7600      I = J
7610      IF (X.GT.A(I)) GO TO 46
7620      IC = I
7630      46  CONTINUE
7640      II = 100 + I
7650      IIM = II - 1
7660      IM = I -1
7670      WX = (A(II) - A(IIM))/(A(I) -A(IM))
7680      XX =X -A(IM)
7690      WF(N) = A(IIM) + WX*XX
7700      GO TO 50
7710      48  WF(N) = WFM
7720      50  CONTINUE
7730      GO TO 200
7740      100 DO 150 I = 1,NN
7750      150 WF(I) = 1.0
7760      200 WRITE(6,901)
7770      WRITE(6,902) (WF(I),I=1,NN)
7780      901  FORMAT(/' FRICTION WEIGHT FUNCTION')
7790      902  FORMAT(10E12.5)
7800      RETURN
7810      END

```

Program BIGLOAD2

Listing



```

10      PROGRAM HYBD
20      COMMON F,DX,DT,NN,MM,NM,NMX
30      DOUBLE PRECISION AX,BX,XL
40      COMMON AX(425,53),BX(425),XL(11475)
50      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
60      DIMENSION WW(3), RII(3)
70      DIMENSION R(25)
80      DIMENSION BX8(425)
90      DIMENSION WH(10), CH(10),RLH(10)
100     C
110     C          MAIN PROGRAM TO CALL ALL OF THE OTHER PIECES
120     C
130     C          DIMENSION(XL) = NM*(NN+2)
140     C          DIM(AX) = NM,2NN+3
150     C
160     C          REVISED APRIL, 1985
170     READ(5,*) EPS,EST,DD1
180     READ(5,*) ICCM,NCALM,NITM,ISD
190     ICCM = 1
200     3  READ(5,*) F,XMAX
210     XMAX = XMAX*1.0E+05
220     F = F*1.0E-05
230     WRITE(6,907) F,XMAX
240     READ(5,*) BETA,ILWH
250     IF (ILWH.EQ.1) GO TO 4
260     IF (ILWH.NE.0) GO TO 140
270     NCALM = 1
280     WRITE(6,910)
290     GO TO 5
300     4  WRITE(6,911)
310     5  DDLW = FLOAT(ILWH)
320     WRITE(6,909) BETA
330     READ(5,*) NCAL,WH(1)
340     READ(5,*) IDIAG
350     READ(5,*) RLF,DRL
360     WH(1) = WH(1)*1.0E-06
370     RLF = RLF*1.0E-07
380     DRL = DRL*1.0E-07
390     WRITE(6,905) EPS,EST,DD1
400     MM = 17
410     NN = 25
420     NM = NN*MM
430     NMX = 2*NN + 3
440     DX = XMAX/FLOAT(NN-1)
450     DT = 1./FLOAT(MM-1)
460     DO 50 I = 1,NCALM
470     50  RLH(I) = RLF + DRL*FLOAT(I-1)
480     RL = RLH(NCAL)
490     READ(5,*) (WH(J),J=1,3)
500     DO 6 J = 1,3
510     6  WW(J) = WW(J)*1.0E-06
520     CALL DEP
530     CALL NSQ
540     1  RIB = 0.
550     I = 1
560     15  W = WW(I)
570     CALL MATS(RL,W,DD1,RI,IER,BETA,DDLW)
580     IF (IER.NE.0) GO TO 140
590     RII(I) = RI
600     IF (RI.LT.RIB) GO TO 18

```

```

610      RIB = RI
620      WB = W
630      IB = I
640      DO 10 IJ = 1,NM
650  10   BXB(IJ) = BX(IJ)
660  18   I = I+1
670      IF (I.EQ.2) GO TO 15
680      IF (I.EQ.4) GO TO 19
690      IF (RII(1).LT.RII(2)) GO TO 15
700      WW(3) = WW(1)*(1. -EST)
710      GO TO 15
720  19   NIT = 3
730      IGP = 0
740  20   CALL NGSW(WW,RII,WB,WN,ISUC,EPS,IN,IGP,EST,ISD)
750      IF (WN.GT.F) GO TO 140
760      IF (ISUC.EQ.2) GO TO 140
770      IF (ISUC.EQ.1) GO TO 100
780      CALL MATS(RL,WN,DDL,RI,IER,BETA,DDLW)
790      IF (IER.NE.0) GO TO 140
800      IF (IN.NE.0) GO TO 24
810      IF (RI.GT.RIB) GO TO 21
820      IF (WN.LT.WW(2)) GO TO 23
830  22   IN = 3
840      GO TO 24
850  23   IN = 1
860      GO TO 24
870  21   IF (WN.LT.WW(2)) GO TO 22
880      GO TO 23
890  24   WW(IN) = WN
900      RII(IN) = RI
910      IF (RI.LT.RIB) GO TO 30
920      WB = WN
930      RIB = RI
940      DO 25 I = 1,NM
950  25   BXB(I) = BX(I)
960  30   NIT = NIT +1
970      IF(NIT.LT.NITM) GO TO 20
980      WRITE(6,902) NIT
990      GO TO 140
1000  100  RN = SQRT(RIB)
1010      DO 110 I = 1,NM
1020  110  BXB(I) = BXB(I)/RN
1030      CC = WB/RL
1040      WRITE(6,903) WB,RL,CC,EPS,NIT
1050      IF (NCAL.NE.1) GO TO 125
1060      CALL LGWH(WB,RL,DDL,R,BETA,DDLW)
1070      IF (IDIAG.EQ.0) GO TO 125
1080      CALL UCAL(WB,RL,BETA,DDLW)
1090      CALL RHOC
1100  125  WRITE(6,908)
1110      DO 130 N = 1,NN
1120      X = DX*FLOAT(N-1)
1130      DZ = RH(N)*DT
1140      WRITE(6,904) X,RH(N),DZ
1150      ML = 1 +MM*(N-1)
1160      MH = MM*N
1170      WRITE(6,901) (BX(M),M=ML,MH)
1180  130  CONTINUE
1190      CALL DIAG(WB,RL,DDLW)
1200      WH(NCAL) = WB

```

```

1210      CH(NCAL) = CC
1220      NCAL = NCAL + 1
1230      IF (NCAL.GT.NCALM) GO TO 140
1240      RL = RLH(NCAL)
1250      IF (NCAL.GE.3) GO TO 200
1260      WW(2) = CC*RL
1270      GO TO 205
1280 200      I1 = NCAL - 2
1290          I2 = NCAL - 1
1300          CG = (WH(I2) - WH(I1))/(RLH(I2) - RLH(I1))
1310          WW(2) = WH(I2) + CG*(RLH(NCAL) - RLH(I2))
1320 205      WW(1) = WW(2)*(1.0 - EST)
1330          WW(3) = WW(2)*(1.0 + EST)
1340          IF (WW(1).LT.0.) GO TO 140
1350          IF (WW(3).GT.F) GO TO 140
1360          GO TO 1
1370 140      ICC = ICC + 1
1380          IF (ICC.LE.ICCM) GO TO 3
1390 901      FORMAT(2X,10E12.5)
1400 902      FORMAT(//' USED UP ',I5,' ITERATIONS'//)
1410 903      FORMAT(///' CONVERGED: W,L,C,EPS,NIT =',4E15.5,I10//)
1420 904      FORMAT(/' X,H,DZ =',3E15.5)
1430 905      FORMAT(/' EPS,EST,DD1 =', 2E15.5,F10.2/)
1440 906      FORMAT(3F10.5)
1450 907      FORMAT(///' F,XMAX =',2E15.5//)
1460 908      FORMAT(///' PRESSURE'/)
1470 909      FORMAT(/' BETA= ',E15.5)
1480 910      FORMAT(' LONG WAVE LIMIT')
1490 911      FORMAT(' GENERAL FREQUENCY AND WAVENUMBER')
1500          STOP
1510          END
1520
1530          SUBROUTINE NGSW(WW,RII,WB,WN,ISUC,EPS,IN,IGP,EST,ISD)
1540          COMMON F,DX,DT,NN,MM,NP,NMX
1550          DOUBLE PRECISION AX,BX,XL
1560          COMMON AX(425,53),BX(425),XL(11475)
1570          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
1580          DIMENSION WW(3),RII(3)
1590  C
1600  C          SUBROUTINE TO COMPUTE THE NEXT GUESS AT W
1610  C
1620  C          ISD = 0  NORMAL
1630  C          ISD = 1  SEARCH DOWN
1640  C          ISD = -1 SEARCH UP
1650          DEL = EST
1660  5          IC = 0
1670              DD 10 I = 1,2
1680              I1 = 1 + I
1690              IF (WW(I1).GT.WW(I)) GO TO 10
1700              IC = 1
1710              RX = RII(I)
1720              WX = WW(I)
1730              WW(I) = WW(I1)
1740              WW(I1) = WX
1750              RII(I) = RII(I1)
1760              RII(I1) = RX
1770 10          CONTINUE
1780              IF (IC.NE.0) GO TO 5
1790              IF (RII(3).GT.RII(2)) GO TO 150
1800              IF (RII(1).GT.RII(2)) GO TO 160

```

```

1810      ISD = 0
1820      WB1 = (WW(1) + WW(2))/2.
1830      WB2 = (WW(2) + WW(3))/2.
1840      RW1 = (RII(2) - RII(1))/(WW(2) - WW(1))
1850      RW2 = ( RII(3) - RII(2))/(WW(3) - WW(2))
1860      A = (RW2 - RW1)/(WB2 - WB1)
1870      B = RW1 - A*WB1
1880      WN = -B/A
1890      IF (WN.LE.WW(1)) GO TO 50
1900      IF (WN.GT.WW(3)) GO TO 45
1910      GO TO 20
1920      50      WN = WB1
1930      GO TO 20
1940      45      WN = WB2
1950      20      EP = WN - WB
1960      EP = ABS(EP)
1970      EP = EP/WB
1980      IF (EP.LE.EPS) GO TO 100
1990      IGP = IGP + 1
2000      ISUC = 0
2010      IN = 0
2020      GO TO 120
2030      100     ISUC = 1
2040      IF (IGP.NE.0) GO TO 120
2050      IGP = IGP + 1
2060      ISUC = 0
2070      IF (RII(3).GT.RII(1)) GO TO 110
2080      WN = WB2
2090      GO TO 20
2100      110     WN = WB1
2110      GO TO 20
2120      150     IF (RII(1).GT.RII(2)) GO TO 180
2130      155     IF (ISD.EQ.1) GO TO 165
2140      WN = WW(3)*(1. + DEL)
2150      ISUC = 0
2160      IGP = 0
2170      IN = 1
2180      GO TO 120
2190      160     IF (RII(3).GT.RII(2)) GO TO 180
2200      165     IF (ISD.EQ.-1) GO TO 155
2210      WN = WW(1)*(1. - DEL)
2220      ISUC = 0
2230      IGP = 0
2240      IN = 3
2250      GO TO 120
2260      180     IF (RII(3).GT.RII(1)) GO TO 155
2270      GO TO 165
2280      120     IF (WN.LE.0.) GO TO 125
2290      GO TO 130
2300      125     ISUC = 2
2310      130     RETURN
2320      END
2330
2340      SUBROUTINE NSQ
2350      COMMON F,DX,DT,NN,MM,NM,NMX
2360      DOUBLE PRECISION AX,BX,XL
2370      COMMON AX(425,53),BX(425),XL(11475)
2380      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
2390      C
2400      C      SUBROUTINE TC READ AND INTERPOLATE BUOYANCY FREQUENCY

```

```

2410 C          SQUARED PROFILE BV, AND Z DERIVATIVE BVZ
2420 C
2430          READ(5,*) NR,DZR,ALPH
2440          ALPH = ALPH*1.0E+05
2450          DZR = DZR*100.
2460          MX = MM -1
2470          READ(5,*) CMLT
2480          DO 20 I = 1,NR
2490          READ(5,*) XL(I)
2500 20          XL(I) = XL(I)*CMLT
2510          TF = XL(NR)
2520          ZZZ = DZR*FLOAT(NR-1)
2530          DZI = RH(NN) - ZZZ
2540          NL = NR +1
2550          NH = 600
2560          DO 30 I = NL,NH
2570          Z = DZR*FLOAT(I-1)
2580 30          XL(I) = TF*EXP((ZZZ -Z)/ALPH)
2590          DO 200 N = 1,NN
2600          DD = RH(N)
2610          DZ = DT*DD
2620          IF (DZ.GT.DZR) GO TO 110
2630          BV(N,MM) = XL(1)
2640          DO 50 M = 1,MX
2650          Z = DD - DZ*FLOAT(M-1)
2660          IBXL = 1 + IFIX(Z/DZR)
2670          IBXH = IBXL +1
2680          ZS = DZR*FLOAT(IBXL -1)
2690 50          BV(N,M) = XL(IBXL) + (Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
2700          Z = DD - DZ
2710          IBXL = 1 + IFIX(Z/DZR)
2720          IBXH = IBXL + 1
2730          ZS = DZR*FLOAT(IBXL-1)
2740          AQ = XL(IBXL) + (Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
2750          GO TO 145
2760 110          ZD = DZ/2.
2770          XBB = 0.
2780          NAVG = 0
2790          DO 120 I = 1,NR
2800          ZC = DZR*FLOAT(I-1)
2810          IF (ZC.GT.ZD) GO TO 120
2820          XBB = XBB + XL(I)
2830          NAVG = NAVG +1
2840 120          CONTINUE
2850          BV(N,MM) = XBB/FLOAT(NAVG)
2860          DO 140 MQ = 1,MM
2870          M = MQ -1
2880          Z = DD - DZ*FLOAT(M-1)
2890          ZS = Z - DZ/2.
2900          ZD = ZS + DZ
2910 125          XBB = 0.
2920          NAVG = 0
2930          DO 130 I = 1,NH
2940          ZC = DZR*FLOAT(I-1)
2950          IF ( ZC.LT.ZS) GO TO 130
2960          IF (ZC.GT.ZD) GO TO 130
2970          XBB = XBB + XL(I)
2980          NAVG = NAVG + 1
2990 130          CONTINUE
3000          IF (NAVG.NE.0) GO TO 135

```

```

1010      IBXL = 1 + IFIX(Z/DZR)
1020      IBXH = IBXL + 1
1030      ZSS = DZR*FLOAT(IBXL-1)
1040      IF (M.EQ.0) GO TO 133
1050      BV(N,M) = XL(IBXL) + (Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
1060      GO TO 140
1070  133      AQ = XL(IBXL) + (Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
1080      GO TO 140
1090  135      IF (M.EQ.0) GO TO 138
1100      BV(N,M) = XBB/FLOAT(NAVG)
1110      GO TO 140
1120  138      AQ = XBB/FLOAT(NAVG)
1130  140      CONTINUE
1140  145      DD 150 M = 2, MX
1150      IP = M + 1
1160      IM = M - 1
1170  150      BVZ(N,M) = (BV(N,IP) - BV(N,IM))/(2.*DZ)
1180      BVZ(N,1) = (BV(N,2) - AQ)/(2.C*DZ)
1190      BVZ(N,MM) = (BV(N,MM) - BV(N,MX))/DZ
1200  200      CONTINUE
1210      WRITE(6,901)
1220      WRITE(6,902) (BV(NN,J),J=1,MM)
1230  901      FORMAT(// ' NSQUARED AT XMAX '/')
1240  902      FORMAT(2X,10E12.5)
1250  250      RETURN
1260      END
1270
1280      SUBROUTINE MATS(RL,W,DD1,RI,IER,BETA,DDLW)
1290      COMMON F,DX,DT,NN,MM,NM,NMX
1300      DOUBLE PRECISION AX,BX,XL
1310      COMMON AX(425,53),BX(425),XL(11475)
1320      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
1330  C
1340  C          SUBROUTINE TO SET UP AND SOLVE THE PRESSURE EQUATION
1350  C
1360  C          DD1 = 0 RIGID LID
1370  C          DD1 = 1 FREE SURFACE
1380      GG = 1./980.
1390      DXDT = DX/(2.*DT)
1400      DXX = DX*DX
1410      DDD = DXX/(DT*DT)
1420      RLL = RL*RL*DDLW
1430      DDX = DX*DXDT
1440      DQ = 2.0*DX
1450      FLW = F*RL/W
1460      BLW=BETA*RL/W
1470      CCC1 = W*(F*F - DDLW*W*W)/DXX
1480      CCC2 = 0.5*F*(2.0*BETA*W + RL*(F*F - DDLW*W*W))/DX
1490      CCC3 = BETA*RL*(F*F + DDLW*W*W)
1500      AA5 = 2.0*F*BETA/(F*F - W*W*DDLW)
1510      RLQ = DDLW*RLL - BLW*(F*F + DDLW*W*W)/(F*F - DDLW*W*W)
1520      DO 5 I = 1,NM
1530  5      BX(I) = 0.
1540      DO 10 J = 1,NMX
1550      DO 10 I = 1,NM
1560  10      AX(I,J) = 0.
1570      BX(39) = 1.
1580      J = NN + 2
1590      IP1 = J + 1
1600      IM1 = J - 1

```

```

3610      IPP1 = 2*NN + 1
3620      IPP = 2*NN + 2
3630      IPPM = 2*NN + 3
3640      DD 100 M = 1,MM
3650      DD 90 N = 1,NN
3660      I = N + NN*(M-1)
3670      CALL AS(A1,A2,A3,W,N,M,C1,C2,C3,DDLW)
3680      AX(I,1) = A2*DXDT
3690      AX(I,IP1) = 1.-AA5*DX/2.0
3700      AX(I,IPM1) = -A2*DXDT
3710      AX(I,2) = A1*DDD - DDX*(A3+A2*AA5)
3720      AX(I,J) = -2. -2.*DDD*A1 - RLQ*DX
3730      AX(I,IPM) = A1*DDD + DDX*(A3+A2*AA5)
3740      AX(I,3) = -A2*DXDT
3750      AX(I,IP1) = 1.+AA5*DX/2.0
3760      AX(I,IPMM) = A2*DXDT
3770      IF ( N.EQ.1) GO TO 20
3780      IF (N.EQ.NN) GO TO 30
3790      IF (M.EQ.1) GO TO 40
3800      IF (M.EQ.MM) GO TO 50
3810      GO TO 90
3820  20      AX(I,1) = 0.
3830      AX(I,IP1) = 0.
3840      AX(I,IPM1) = 0.
3850      AX(I,3) = C.
3860      AX(I,IPMM) = 0.
3870      AX(I,IP1) = 2.
3880      TH = -1. + DT*FLOAT(M-1)
3890      B1 = -TH*RHX(N)/RH(N)
3900      AX(I,2) = AX(I,2) -A2*B1*DDD + A2*FLW*2.0*DDX - B1*2.0*DXDT
3910      AX(I,J) = AX(I,J) + 2.*DX*FLW + 2.*A2*B1*DDD
3920      AX(I,IPM) = AX(I,IPM) -A2*B1*DDD -A2*FLW*2.0*DDX + B1*2.0*DXDT
3930      IF (M.NE.MM) GO TO 25
3940      D5 = DD1*RH(N)*BV(N,M)*GG
3950      AX(I,2) = AX(I,2) + AX(I,IPM)
3960      AX(I,J) = AX(I,J) - 2.0*DT*D5*AX(I,IPM)
3970      AX(I,IPM) = 0.
3980      GO TO 90
3990  25      IF (M.NE.1) GO TO 90
4000      AX(I,IPM) = AX(I,IPM) + AX(I,2)
4010      AX(I,2) = 0.
4020      GO TO 90
4030  30      DZ = DT*RH(N)
4040      AX(I,3) = 0.
4050      AX(I,IPMM) = 0.
4060      AX(I,1) = 0.
4070      AX(I,3) = C.
4080      AX(I,J) = AX(I,J) + AX(I,IP1)*{(2.0*CCC1-CCC3)/(CCC1+CCC2)}
4090      AX(I,IP1) = AX(I,IP1) + AX(I,IP1)*{(CCC2-CCC1)/(CCC1+CCC2)}
4100      AX(I,IP1) = 0.
4110      IF (M.NE.MM) GO TO 35
4120      AX(I,2) = AX(I,2) + AX(I,IPM)
4130      AX(I,J) = AX(I,J) -2.0*DZ*DD1*GG*BV(N,M)*AX(I,IPM)
4140      AX(I,IPM) = 0.
4150      GO TO 90
4160  35      IF (M.NE.1) GO TO 90
4170      AX(I,IPM) = AX(I,IPM) + AX(I,2)
4180      AX(I,2) = C.
4190      GO TO 90
4200  40      D1 = 2.*C2*DT/(C1 +C2*C3)

```

```

4210      AX(I,IPM) = AX(I,IPM) + AX(I,2)
4220      AX(I,IP1) = AX(I,IP1) + D1*AX(I,2)/DQ
4230      AX(I,IM1) = AX(I,IM1) - D1*AX(I,2)/DQ
4240      AX(I,J) = AX(I,J) + D1*FLW*AX(I,2)
4250      NNN = N + 1
4260      IJJP = J + 2
4270      CALL AS(A1,A2,A3,W,NNN,M,C1,C2,C3,DDLW)
4280      D1 = 2.0*C2*DT/(C1 + C2*C3)
4290      AX(I,IPMM) = AX(I,IPMM) + AX(I,3)
4300      AX(I,IJJP) = AX(I,IJJP) + D1*AX(I,3)/DQ
4310      AX(I,J) = AX(I,J) - D1*AX(I,3)/DQ
4320      AX(I,IP1) = AX(I,IP1) + D1*FLW*AX(I,3)
4330      IF (N.EQ.2) GO TO 45
4340      IJJM = J - 2
4350      NNN = N - 1
4360      CALL AS(A1,A2,A3,W,NNN,M,C1,C2,C3,DDLW)
4370      D1 = 2.0*C2*DT/(C1 + C2*C3)
4380      AX(I,IPM1) = AX(I,IPM1) + AX(I,1)
4390      AX(I,J) = AX(I,J) + D1*AX(I,1)/DQ
4400      AX(I,IJJM) = AX(I,IJJM) - D1*AX(I,1)/DQ
4410      AX(I,IM1) = AX(I,IM1) + D1*FLW*AX(I,1)
4420      GO TO 48
4430      45      AX(I,IPM1) = AX(I,IPM1) + AX(I,1)
4440      48      AX(I,1) = 0.
4450      AX(I,2) = 0.
4460      AX(I,3) = 0.
4470      GO TO 90
4480      50      AX(I,1) = 0.
4490      AX(I,3) = 0.
4500      D5 = RH(N)*DD1*BV(N,M)*GG
4510      AX(I,2) = AX(I,2) + AX(I,IPM)
4520      AX(I,J) = AX(I,J) - D5*2.0*DT*AX(I,IPM)
4530      AX(I,IM1) = AX(I,IM1) - D5*2.0*DT*AX(I,IPM1)
4540      AX(I,IP1) = AX(I,IP1) - D5*DT*2.0*AX(I,IPMM)
4550      AX(I,IPM1) = 0.0
4560      AX(I,IPMM) = 0.0
4570      AX(I,IPM) = 0.0
4580      90      CONTINUE
4590      10C     CONTINUE
4600      NDD = NN + 1
4610      NNS = NM
4620      CALL LEQT1B(AX,NM,NDC,NDC,NM,BX,1,NM,0,XL,IER)
4630      C      LEQT1B IS AN IMSL ROUTINE
4640      CALL CALI(RI)
4650      CC = W/RL
4660      WRITE(6,901) W,RL,CC,RI,IER
4670      901    FORMAT(/'  W,L,C,RI,IER =' ,4E15.5,I10)
4680      RETURN
4690      END
4700
4710      SUBROUTINE AS(A1,A2,A3,W,N,M,C1,C2,C3,DDLW)
4720      COMMON F,DX,DT,NN,MM,NM,NMX
4730      DOUBLE PRECISION AX,BX,XL
4740      COMMON AX(425,53),BX(425),XL(11475)
4750      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
4760      C
4770      C      SUBROUTINE TO COMPUTE COEFFICIENTS FOR MATRIX
4780      C
4790      F2 = F*F
4800      W2 = W*W*DDLW

```



```

4810      TZ = 1./RH(N)
4820      TH = -1. +DT*FLOAT(M-1)
4830      TX = -RHX(N)*TH*TZ
4840      AA = (RHX(N)/RH(N))*2
4850      TXX = (2.*AA -RHXX(N)/RH(N))*TH
4860      BW = BV(N,M)
4870      BWW = BW*BW
4880      A1 = TX*TX + TZ*TZ*(F2-W2)/Bw
4890      A2 = TX
4900      A3 = TXX -TZ*BVZ(N,M)*(F2-W2)/BWW
4910      C1 = TZ
4920      C2 = RHX(N)*BV(N,1)/(F2-W2)
4930      C3 = C1*RHX(N)
4940      RETURN
4950      END

4960
4970      SUBROUTINE DEP
4980      COMMON F,DX,DT,NN,MM,NM,NMX
4990      DOUBLE PRECISION AX,BX,XL
5000      COMMON AX(425,53),BX(425),XL(11475)
5010      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
5020      C
5030      C          SUBROUTINE TO READ AND INTERPOLATE DEPTH PROFILE
5040      C          RH = DEPTH
5050      C          RHX = X DERIVATIVE OF DEPTH
5060      C          RHXX = SECOND X DERIVATIVE OF DEPTH
5070      C
5080      READ(5,*) NRX
5090      DO 5 I =1,NRX
5100      READ(5,*) XL(I),BX(I)
5110      BX(I) = BX(I)*100.
5120      5  XL(I) = XL(I)*1.0E+05
5130      RHMA = BX(NRX)
5140      RH(1) = BX(1)
5150      DO 20 N = 2,NN
5160      X = DX*FLOAT(N-1)
5170      IF (X.GT.XL(NRX)) GO TO 15
5180      IC = 0
5190      DO 8 J = 2,NRX
5200      IF (IC.NE.0) GO TO 8
5210      I = J
5220      IF (X.GT.XL(I)) GO TO 8
5230      IC = I
5240      8  CONTINUE
5250      IM = I-1
5260      RHX(N) = (BX(I) -BX(IM))/(XL(I)-XL(IM))
5270      XX = X - XL(IM)
5280      RH(N) = BX(IM) + RHX(N)*XX
5290      GO TO 20
5300      15  RH(N) = RHMA
5310      20  CONTINUE
5320      RHX(1) = (RH(2) - RH(1))/DX
5330      RHXX(1) = 0.
5340      D2 = 2.*DX
5350      DXX = DX*DX
5360      NX = NN -1
5370      DO 30 N = 2,NX
5380      IP = N +1
5390      IM = N -1
5400      RHX(N) =(RH(IP) - RH(IM))/D2

```

```

5410 30      RHXX(N) = (RH(IP) -2.*RH(N) + RH(IM))/DXX
5420      RHX(NN) = 0.
5430      RHXX(NN) = 0.
5440      RETURN
5450      END
5460
5470      SUBROUTINE CALI(RIX)
5480      COMMON F,DX,DT,NN,MM,NM,NMX
5490      DOUBLE PRECISION AX,BX,XL
5500      COMMON AX(425,53),BX(425),XL(11475)
5510      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
5520      DOUBLE PRECISION RI,RZ
5530  C
5540  C          SUBROUTINE TO CALCULATE THE INTEGRAL OF RESPONSE SQUARED
5550  C
5560      DO 2 M = 1,MM
5570      DO 2 N = 1,NN
5580          IN = N + NN*(M -1)
5590          II = M + MM*(N-1)
5600  2      XL(II) = BX(IN)
5610      DO 3 I = 1,NM
5620  3      BX(I) = XL(I)
5630      RI = 0.
5640      RR = 0.
5650      RZ = 0.
5660      DO 50 N = 1,NN
5670          IF (N.EQ.1) GO TO 5
5680          IF (N.NE.NN) GO TO 10
5690  5      DXX = DX/2.
5700          GO TO 15
5710  10      DXX = DX
5720  15      DO 50 M = 1,MM
5730          DZ = DT*RH(N)
5740          IF (M.EQ.1) GO TO 25
5750          IF (M.NE.MM) GO TO 30
5760  25      DZZ = DZ/2.
5770          GO TO 35
5780  30      DZZ = DZ
5790  35      I = (N-1)*MM +M
5800          RZ = RZ + DZZ*DXX
5810          RI = RI + DZZ*DXX*BX(I)**2
5820  50      CONTINUE
5830          RI = RI/RZ
5840          RIX = RI
5850          RETURN
5860          END
5870
5880      SUBROUTINE DIAG(W,RL,DDLW)
5890      COMMON F,DX,DT,NN,MM,NM,NMX
5900      DOUBLE PRECISION AX,BX,XL
5910      COMMON AX(425,53),BX(425),XL(11475)
5920      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
5930  C
5940  C          SUBROUTINE TO COMPUTE MOMENTUM DIAGNOSTICS
5950  C
5960      N = 5
5970      I = N*MM
5980      IP = I + MM
5990      IM = I -MM
6000      PX = (BX(IP) -BX(IM))/(2.*DX)

```

```

6010      F2 = F*F - W*W*DDLW
6020      VT = W*(F*PX + DDLW*RL*W*BX(I))/F2
6030      FU = -F*(RL*F*BX(I) + W*PX)/F2
6040      PY = RL*BX(I)
6050      X = DX*FLOAT(N-1)
6060      WRITE(6,901) X
6070      WRITE(6,902) VT,FU,PY
6080  901   FORMAT(/'  Y MOMENTUM TERMS AT Z = 0, X =',E15.5)
6090  902   FORMAT(/'  VT,FU,PY =',3E15.5/)
6100      RETURN
6110      END
6120
6130      SUBROUTINE LGWH(W,RL,DD1,R,BETA,DDLW)
6140      COMMON F,DX,DT,NN,MM,NM,NMX
6150      DOUBLE PRECISION AX,BX,XL
6160      COMMON AX(425,53),BX(425),XL(11475)
6170      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
6180      DOUBLE PRECISION RD,RP,RI,XINT,XP
6190      DIMENSION R(25)
6200  C
6210  C      SUBROUTINE TO COMPUTE LONG WAVE PARAMETERS BN,ANN
6220  C
6230      XINT = 0.
6240      RHO = 1.03
6250      F2 = F*F
6260      MX = MM - 1
6270  C      NORMALIZE
6280      NX = NN - 1
6290      RI = (BX(1)**2 + BX(MM)**2)/2.
6300      RD = (RHX(1)*BX(1)**2)/2.
6310      DO 10 I = 2, MX
6320  10     RI = RI + BX(I)**2
6330      DO 20 I = 2, NX
6340      IJ = MM*(I-1) + 1
6350  20     RD = RD + RHX(I)*BX(IJ)**2
6360      RD = RD *DX
6370      RI = RI*RH(1)*DT
6380      RD = RD + RI
6390      RD = DSQRT(RD*RD)
6400      RD = DSQRT(RD)
6410      DO 30 I = 1, NM
6420  30     BX(I) = BX(I)/RD
6430      CALL VCAL(W,RL,BETA,DDLW)
6440  C      COMPUTE WIND COUPLING
6450  40     BN = BX(MM)
6460  C      NOW GET BOTTOM FRICTION
6470  C      READ R(X)
6480      READ(5,*) NR
6490      DO 45 I = 1, NR
6500      READ(5,*) AX(I,1), AX(I,2)
6510  45     AX(I,1) = AX(I,1)*1.0E+05
6520      R(1) = AX(1,2)
6530      RMA = AX(NR,2)
6540      DO 50 N = 2, NN
6550      X = DX*FLOAT(N-1)
6560      IF (X.GT.AX(NR,1)) GO TO 48
6570      IC = 0
6580      DO 46 J = 2, NR
6590      I = J
6600      IF (X.GT.AX(I,1)) GO TO 46

```

```

6610      IC = I
6620      46      CONTINUE
6630      IM = I -1
6640      RX = (AX(I,2) -AX(IM,2))/(AX(I,1)-AX(IM,1))
6650      XX = X - AX(IM,1)
6660      R(N) = AX(IM,2) + RX*XX
6670      GO TO 50
6680      48      R(N) = RMA
6690      50      CONTINUE
6700      C          X=0 CONTRIBUTION
6710      RI = 0.5*R(1)*(RL*F*BX(1)/W)**2
6720      C          X = L CONTRIBUTION
6730      I = 1 + MM*(NN-1)
6740      BXP = F*XL(I)
6750      RI = RI + 0.5*R(NN)*(BXP)**2
6760      C          INTERMEDIATE X
6770      DO 150 N = 2,NX
6780      I = 1 + MM*(N-1)
6790      IP = I + MM
6800      IM = I - MM
6810      RX = 0.5*(BX(IP)-BX(IM))/DX
6820      150     RI = RI + R(N)*RX*F*XL(I)
6830      ANN = RI*DX/F
6840      C
6850      WRITE(6,902) BN,ANN
6860      WRITE(6,901)
6870      WRITE(6,905) (R(I),I=1,NN)
6880      CC = 0.5/(980.*RHO)
6890      DO 220 N = 1,NN
6900      XX = DX
6910      IF (N.EQ.1) GO TO 210
6920      IF (N.NE.NN) GO TO 215
6930      210     XX = DX/2.
6940      215     I = N*MM
6950      220     XINT = XINT + CC*XX*BX(I)*BX(I)
6960      WRITE(6,904) XINT
6970      901     FORMAT(/' R =')
6980      902     FORMAT('   BN,ANN =',2E15.5/)//)
6990      903     FORMAT(2F10.5)
7000      904     FORMAT(//' FREE SURFACE CONTRIBUTION TO PE =',E15.5/)
7010      905     FORMAT(10E12.5)
7020      500     RETURN
7030      END
7040
7050      SUBROUTINE VCAL(W,RL,BETA,DDLW)
7060      COMMON F,DX,DT,NN,MM,NM,NMX
7070      DOUBLE PRECISION AX,BX,XL
7080      COMMON AX(425,53),BX(425),XL(11475)
7090      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
7100      DOUBLE PRECISION XINT
7110      C
7120      C          SUBROUTINE TO COMPUTE THE V FIELD OF THE WAVE
7130      C
7140      RHO = 0.515
7150      XINT = 0.
7160      WRITE(6,905)
7170      FW = F*F - DDLW*W*W
7180      WL = DDLW*W*RL
7190      RLW = RL/W
7200      FLW = F*RL/W

```

```

7210      MX = MM -1
7220      DXX = 2.*DX
7230      W2 = W*W*DDLW
7240      DO 50 N = 1,NN
7250      XX = DX
7260      X = DX*FLOAT(N-1)
7270      D = RH(N)
7280      DD = RHX(N)/D
7290      DZ = DT*D
7300      RHZ = RHX(N)**2
7310      IF (N.EQ.1) GO TO 10
7320      IF (N.NE.NN) GO TO 20
7330      GO TO 18
7340  10    DO 15 M = 1,MM
7350      I = (N-1)*MM +M
7360  15    XL(I) = -RLW*BX(I)
7370      XX = DX/2.
7380      GO TO 40
7390  18    CC1 = W*FW/(DX*DX)
7400      CC2 = 0.5*(2.0*F*BETA*W + RL*F*FW)/DX
7410      CC3 = BETA*RL*(F*F + DDLW*W*W)
7420      DQ = 1./(F*DX)
7430      DO 19 M = 1,MM
7440      I = (N-1)*MM +M
7450      IM = I -MM
7460      XP = BX(I)*(2.0*CC1 -CC3)/(CC1+CC2)
7470      XP = XP + BX(IM)*(CC2-CC1)/(CC1+CC2)
7480      XPX = (XP - BX(IM))/(2.0*DX)
7490  19    XL(I) = (F*XPX + WL*BX(I))/FW
7500      XX = DX/2.
7510      GO TO 40
7520  20    I = (N-1)*MM +1
7530      IP = I + MM
7540      IM = I - MM
7550      CALL AS(A1,A2,A3,W,N,1,C1,C2,C3,DDLW)
7560      CQ = C2*C3/(C1+ C2*C3)
7570      XPX = (1.0 -CQ)*(BX(IP) -BX(IM))/DXX
7580      XPX = XPX -CQ*FLW*BX(I)
7590      XL(I) = (WL*BX(I) + F*XPX)/FW
7600      I = N*MM
7610      IP = I + MM
7620      IM = I -MM
7630      XPX = (BX(IP) - BX(IM))/DXX
7640      XL(I) =(WL*BX(I) + F*XPX)/FW
7650      DO 30 M = 2,MX
7660      I = (N-1)*MM +M
7670      IP = I + MM
7680      IM = I -MM
7690      I2 = I +1
7700      I1 = I -1
7710      XPX = (BX(IP) - BX(IM))/DXX
7720      T = -1. +DT*FLOAT(M-1)
7730      XPT = (BX(I2) -BX(I1))/(2.*DT)
7740      XPX = XPX - T*DD*XPT
7750  30    XL(I) = (WL*BX(I) + F*XPX)/FW
7760  40    DO 45 M = 1,MM
7770      I = (N-1)*MM + M
7780      ZZ = DZ
7790      IF (M.EQ.1) GO TO 42
7800      IF (M.NE.MM) GO TO 45

```

```

7810 42 ZZ = DZ/2.
7820 45 XINT = XINT + RHO*XL(I)*XL(I)*XX*ZZ
7830 WRITE(6,904) X,D,DZ
7840 IL = (N-1)*MM + 1
7850 IH = N*MM
7860 WRITE(6,901) (XL(IJ),IJ=IL,IH)
7870 50 CONTINUE
7880 WRITE(6,902) XINT
7890 901 FORMAT(2X,10E12.5)
7900 902 FORMAT(///' V CONTRIBUTION TO KE =',E15.5/)
7910 905 FORMAT(///// ' V'///)
7920 904 FORMAT(// ' X, H, DZ =',3E15.5)
7930 RETURN
7940 END
7950
7960 SUBROUTINE RHOC
7970 COMMON F,DX,DT,NN,MM,NM,NMX
7980 DOUBLE PRECISION AX,BX,XL
7990 COMMON AX(425,53),BX(425),XL(11475)
3000 COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
3010 DOUBLE PRECISION XINT
3020 C
3030 C SUBROUTINE TO COMPUTE THE DENSITY FIELD OF THE WAVE
3040 C
3050 G2 = 980.*980./2.06
3060 XINT = 0.
3070 WRITE(6,903)
3080 G = 980.
3090 DO 50 N = 1,NN
3100 DXX = DX
3110 IF (N.EQ.1) GO TO 2
3120 IF (N.NE.NN) GO TO 5
3130 2 DXX = DX/2.
3140 5 X = DX*FLOAT(N-1)
3150 DZ = DT*RH(N)
3160 GDZ = G*DZ
3170 DO 40 M = 1,MM
3180 DZZ = DZ
3190 I = (N-1)*MM + M
3200 IF (M.EQ.1) GO TO 10
3210 IF (M.NE.MM) GO TO 20
3220 DZZ = DZ/2.
3230 XL(M) = BV(N,M)*BX(I)/(G*G)
3240 GO TO 35
3250 10 IP = I + 1
3260 DZZ = DZ/2.
3270 XL(M) = -(BX(IP) - BX(I))/GDZ
3280 GO TO 35
3290 20 IP = I + 1
3300 IM = I - 1
3310 XL(M) = -(BX(IP) - BX(IM))/(2.*GDZ)
3320 35 XX = G2*XL(M)*XL(M)/BV(N,M)
3330 XINT = XINT + XX*DZZ*DXX
3340 40 CONTINUE
3350 WRITE(6,901) X,RH(N),DZ
3360 50 WRITE(6,902) (XL(M),M=1,MM)
3370 WRITE(6,904) XINT
3380 901 FORMAT(// ' X,D,DZ =',3E15.5)
3390 902 FORMAT(2X,10E12.5)
3400 903 FORMAT(/// ' RHO'//)

```

```

3410 904   FORMAT(// ' RHO CONTRIBUTION TO PE = ',E15.5/)
3420      RETURN
3430      END
3440
3450      SUBROUTINE UCAL(W,RL,BETA,DDLW)
3460      COMMON F,DX,DT,NN,PM,NP,NMX
3470      DOUBLE PRECISION AX,BX,XL
3480      COMMON AX(425,53),BX(425),XL(11475)
3490      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
3500      DOUBLE PRECISION XINT
3510      C
3520      C           SUBROUTINE TO COMPUTE THE U FIELD OF THE WAVE
3530      C
3540      RHO = 0.515
3550      XINT = 0.
3560      FW = F*F - DDLW*w*w
3570      FL = F*RL
3580      WRITE(6,903)
3590      DO 100 N = 1,NN
3600      DXX = DX
3610      X = DX*FLOAT(N-1)
3620      DZ = DT*RH(N)
3630      DO 5 I = 1,MM
3640      5      XL(I) = 0.
3650      IF (N.EQ.1) GO TO 90
3660      IF (N.EQ.NN) GO TO 110
3670      DD = RHX(N)/RH(N)
3680      DO 85 M = 1,MM
3690      I = (N-1)*MM + M
3700      IP = I + MM
3710      IM = I - MM
3720      T = -1. + DT*FLOAT(M-1)
3730      IF (M.EQ.1) GO TO 10
3740      IF (M.EQ.MM) GO TO 15
3750      GO TO 20
3760      10      I1 = I + 1
3770      XPT = (BX(I1) - BX(I))/DT
3780      GO TO 25
3790      15      XPT = 0.
3800      GO TO 25
3810      20      I2 = I + 1
3820      I1 = I - 1
3830      XPT = (BX(I2) - BX(I1))/(2.*DT)
3840      25      XPX = (BX(IP) - BX(IM))/(2.*DX)
3850      XPX = XPX - T*DD*XPT
3860      85      XL(M) = -(W*XPX + FL*BX(I))/FW
3870      GO TO 90
3880      110     CC1 = W*FW/(DX*DX)
3890      CC2 = 0.5*(2.0*F*BETA*w + RL*F*FW)/DX
3900      CC3 = BETA*RL*(F*F + DDLW*w*w)
3910      DXX = DX/2.
3920      DO 120 M = 1,MM
3930      I = (NN-1)*MM + M
3940      IM = I - MM
3950      XPIM = BX(I)*(2.0*CC1-CC3)/(CC1+CC2)
3960      XPIM = XPIM + BX(IM)*(CC2-CC1)/(CC1+CC2)
3970      120     XL(M) = -(FL*BX(I) + W*0.5*(XPIM - BX(IM))/DX)/FW
3980      90      DO 95 M = 1,MM
3990      DZZ = DZ
9000      IF (M.EQ.1) GO TO 92

```

```
010      IF (M.NE.MM) GO TO 95
020  92    DZZ = DZ/2.
030  95    XINT = XINT + RHO*XL(M)*XL(M)*DX*DX*DZZ
040      WRITE(6,901) X,RH(N),DZ
050      WRITE(6,902) (XL(M),M=1,MM)
060  100   CONTINUE
070      WRITE(6,904) XINT
080  901   FORMAT(/' X,D,DZ =' ,3E15.5)
090  902   FORMAT(2X,10E12.5)
100  903   FORMAT(///' U'//)
110  904   FORMAT(///' U CONTRIBUTION TO KE =' ,E15.5//)
120      RETURN
130      END
```



Program CROSS  
Listing

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
EXTERNAL FNA
COMMON S(101),P(101),G(101),T(101),X(101),VS(101),NV,DZ,SURF
DIMENSION RH(25)
```

```
C
C
C
C
C
```

```
MAIN PROGRAM TO CALL THE OTHER PIECES
```

```
REVISED APRIL,1985
```

```
READ(5,*)NV,DD1
READ(5,*)F,XMAX
XMAX=XMAX*1.E5
F=F*1.E-5
IF(DC1.EQ.C.)WRITE(6,908)
IF(DC1.EQ.1.)WRITE(6,909)
WRITE(6,907)F,XMAX
NN=25
DX=XMAX/(NN-1)
CALL DEP(DX,RH,NN)
DZ=RH(NN)/(NV-1)
K1=1
X(1)=C.0
X1=0.0
X2=0.0
H1=C.0
H2=RH(1)
DO 100 I=2,NV
Z=FLCAT(I-1)*DZ
IF(Z.GE.RH(NN))Z=RH(NN)
160 IF(Z.LE.H2)GO TO 150
K1=K1+1
H1=RH(K1-1)
H2=RH(K1)
X1=DX*FLOAT(K1-2)
X2=DX*FLOAT(K1-1)
GO TO 160
150 DH=H2-H1
SX=(X2-X1)/DH
BX=-(X2*H1-X1*H2)/DH
X(I)=SX*Z+BX
100 CONTINUE
CALL NSQ(RH(NN),F,DZ,VS,NV)
SURF=DD1*VS(1)*F*F/981.
READ(5,*)NRS
WRITE(6,901)DZ
WRITE(6,902)(X(I),I=1,NV)
WRITE(6,903)
WRITE(6,902)(VS(I),I=1,NV)
DO 200 IR=1,NRS
READ(5,*)XMIN,XMAX,XINC
XMIN=XMIN*1.E-7
XMAX=XMAX*1.E-7
XINC=XINC*1.E-7
CALL FOOT(XMIN,XMAX,XINC,FNA,X3,IRCOI)
IF(IRCOI.EQ.0)GO TO 200
DO 300 I=2,NV
300 P(I)=P(I)*DEXP(-X3*(X(I)+X(I-1))*0.5)
PSUM=C.0
DO 350 I=2,NV
350 PSUM=PSUM+P(I)**2
```

```

        PSLM=DSQRT(DZ*PSUM)
        DO 36C I=2,NV
36C P(I)=P(I)/PSLM
        CF=F/X3
        WRITE(6,904)X3,CF
        WRITE(6,905)
        WRITE(6,902)(P(I),I=2,NV)
200 CONTINUE
901 FORMAT(//' INVERSE TOPOGRAPHY, DZ=',D15.5/)
902 FORMAT(1X,10D12.5)
903 FORMAT(//' NSQUARED/FSQUARED'//)
904 FORMAT(////' DISPERSION CURVE CROSSES  $w=F$  AT  $L=$ ',D15.5,5X,
1 'PHASE SPEED=',D15.5/)
905 FORMAT(//' PRESSURE FIELD (SURFACE TO BOTTOM)'//)
907 FORMAT(//' F,XMAX=',2D15.5)
908 FORMAT(//' RIGID LIC'//)
909 FORMAT(//' FREE SURFACE'//)
        STOP
        END
        SUBROUTINE DEP(DX,RH,NN)
        IMPLICIT DOUBLE PRECISION (A-H,C-Z)
        DIMENSION XL(101),BX(101),RH(NN)

```

C  
C  
C  
C

```

        SUBROUTINE TO READ AND INTERPOLATE DEPTH PROFILE
                RH= DEPTH

```

```

        READ(5,*)NRX
        DO 5 I=1,NRX
        READ(5,*)XL(I),BX(I)
        BX(I)=BX(I)*10C.
5 XL(I)=XL(I)*1.E5
        IF(NRX.EQ.1)GO TO 7
        DO 6 I=2,NRX
6 IF(BX(I).LE.BX(I-1))GO TO 100
7 RHMA=BX(NRX)
        XMA=XL(NRX)
        RH(1)=BX(1)
        DO 2C N=2,25
        X=DX*FLOAT(N-1)
        IF(X.GT.XMA)GO TO 15
        IC=0
        DO 8 J=2,NRX
        IF(IC.NE.0)GO TO 8
        I=J
        IF(X.GT.XL(I))GO TO 8
        IC=I
8 CONTINUE
        IM=I-1
        RHX=(BX(I)-BX(IM))/(XL(I)-XL(IM))
        XX=X-XL(IM)
        RH(N)=BX(IM)+RHX*XX
        GO TO 20
15 RH(N)=RHMA
20 CONTINUE
        GO TO 200
100 WRITE(6,901)
901 FORMAT(//' TOPOGRAPHY IS NOT MONOTONIC'//)
200 RETURN
        END
        SUBROUTINE NSQ(RHMA,F,DZ,VS,NV)

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)  
DIMENSION XL(1000),VS(NV)

C  
C  
C

SUBROUTINE TO READ AND INTERPOLATE BUOYANCY FREQUENCY

```
FSC=F*F
READ(5,*)NR,DZR,ALPH
ALPH=ALPH*1.E5
DZR=DZR*100.
READ(5,*)CMLT
DO 20 I=1,NR
READ(5,*)XL(I)
20 XL(I)=XL(I)*CMLT
TF=XL(NR)
ZZZ=DZR*FLOAT(NR-1)
I=NR+1
50 Z=DZR*FLOAT(I-1)
XL(I)=TF*DEXP((ZZZ-Z)/ALPH)
IF(Z.GT.RHMA)GO TO 100
I=I+1
GO TO 50
100 K1=1
VS(1)=XL(1)/FSQ
H1=0.0
H2=DZR
DO 200 I=2,NV
Z=FLOAT(I-1)*DZ
IF(Z.GE.RHMA)Z=RHMA
160 IF(Z.LE.H2)GO TO 150
K1=K1+1
H1=FLOAT(K1-1)*DZR
H2=FLOAT(K1)*DZR
GO TO 160
150 DH=H2-H1
SN=(XL(K1+1)-XL(K1))/DH
BN=-(XL(K1+1)*H1-XL(K1)*H2)/DH
VS(I)=(SN*Z+BN)/FSQ
200 CONTINUE
RETURN
END
REAL*8 FUNCTION FNA(k)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON S(101),P(101),G(101),T(101),X(101),VS(101),NV,DZ,SURF
```

C  
C  
C

SUBFUNCTION TO INTEGRATE FROM Z=0 TO Z=-H

```
NVM=NV-2
W2=k*k
TW=-2.*W
DO 100 I=1,NV
EX=DEXP(TW*X(I))
T(I)=W2*EX
100 S(I)=EX/VS(I)
P(NV)=1.0
G(NV)=0.0
DO 200 I=1,NVM
J=NV-I
K=J+1
G(J)=G(K)-.5*DZ*(T(J)+T(K))*P(K)
200 P(J)=P(K)+DZ*G(J)/S(J)
```

```

G(1)=G(2)-.5*DZ*(T(1)+T(2))*P(2)
FNA=C(1)+S(1)*SURF*.5*(3.*P(2)-P(3))
RETURN
END
SUBROUTINE ROOT(X5,X6,X7,FNA,X3,IROOT)
IMPLICIT DOUBLE PRECISION (A-H,C-Z)

```

```

C
C          SUBROUTINE TO FIND ROOT OF FNA BETWEEN X5 AND X6
C

```

```

      IROOT=1
220  IQ=INT((X6-X5)/X7)-1
      I1=0
240  X1=X5+I1*X7
      X2=X1+X7
      Y1=FNA(X1)
      Y2=FNA(X2)
      IF(Y1*Y2.LT.C.0)GO TO 340
      I1=I1+1
      IF(I1.NE.IQ)GO TO 240
      WRITE(6,13)
      IROOT=C
13  FORMAT(//' NO ROOT FOUND'/)
      GO TO 400
340  IF(X1.NE.X5)GO TO 370
      X7=X7/2.
      GO TO 220
370  X2=X1
      X1=X1-C.1*X7
      Y1=FNA(X1)
      Y2=FNA(X2)
420  X3=(Y2*X1-Y1*X2)/(Y2-Y1)
      IF(X3.GT.X5.AND.X3.LT.X6)GO TO 200
      WRITE(6,11)X3
11  FORMAT('/' PROJECTION OUT OF INTERVAL:' ,D15.5//)
200  Y3=FNA(X3)
      IF(ABS((X3-X2)/X2).LT.1.E-7)GO TO 400
      X1=X2
      X2=X3
      Y1=Y2
      Y2=Y3
      GO TO 420
400  RETURN
      END

```

Program BIGDRV2

Listing

```

10      PROGRAM WDSTF
20      COMMON F,DX,DT,NN,MM,NM,NMX
30      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
40      COMMON XL(600)
50      DOUBLE COMPLEX AX(425,53),BX(425),BXB(425),DCMPLX
60      DIMENSION BBB(50),TX(25),TY(25),TYX(25),TXX(25)
70      DIMENSION R(25),RX(25)
80      DOUBLE PRECISION DREAL,DIMAG,DATAN2,A,B,AMT,THT,DSQRT
90      DOUBLE COMPLEX FFC(25)

100     C
110     C           MAIN PROGRAM TO CALL ALL OF THE OTHER PIECES
120     C
130     C           DIM(AX) = NM,2NN+3
140     READ(5,*) ICCM
150     ICCM = 1
160     READ(5,*) F,XMAX
170     FF = 180.0/3.14149
180     XMAX = XMAX*1.0E+05
190     F = F*1.0E-05
200     WRITE(6,907) F,XMAX
210     READ(5,*) ILW,IRL,IXY
220     IF (ILW.EQ.1) GO TO 5
230     ILW = 0
240     WRITE(6,910)
250     GO TO 10
260     5      WRITE(6,911)
270     10     DD2 = FLOAT(ILW)
280     IF (IRL.EQ.1) GO TO 15
290     IRL = 0
300     WRITE(6,912)
310     GO TO 25
320     15     WRITE(6,913)
330     25     DD1 = FLOAT(IRL)
340     IF (IXY.EQ.1) GO TO 27
350     IXY = 0
360     WRITE(6,915)
370     GO TO 29
380     27     IF (ILW.EQ.0) GO TO 28
390     WRITE(6,916)
400     GO TO 29
410     28     WRITE(6,917)
420     GO TO 200
430     29     MM = 17
440     NN = 25
450     NM = NN*MM
460     NMX = 2*NN + 3
470     DX = XMAX/FLGAT(NN-1)
480     DT = 1./FLGAT(MM-1)
490     CALL DEP(BBB,RQPP)
500     CALL NSQ
510     CALL FRIC(R,RX)
520     CALL WRD(TX,TY,TXX,TYX,IXY)
530     3      READ(5,*) W,RL
540     CALL MATS(RL,W,AX,BX,R,RX,DD1,DD2,TX,TY,TXX,TYX,FFC)
550     DO 20 M = 1,MM
560     DO 20 N = 1,NN
570     IN = N + NN*(M-1)
580     II = M + MM*(N-1)
590     20     BXB(II) = BX(IN)
600     DO 30 I = 1,NM

```

```

610 30 BX(I) = BXB(I)
620 WRITE(6,903) W,RL
630 CALL VCAL(W,RL,BX,BXB,R,RX,VKE,DD2,TX,TY)
640 CALL UCAL(W,RL,BX,BXB,R,UKE,DD2,TX,TY)
650 CALL RHOC(BX,BXB,RPE,DC1,DD2,FFC)
660 CALL PEC(BX,SPE,IRL)
670 RRRR = (UKE + VKE)/(RPE + SPE)
680 TE = UKE + VKE + RPE + SPE
690 WRITE(6,914) TE
700 WRITE(6,909) RRRR
710 125 WRITE(6,908)
720 126 DO 130 N = 1,NN
730 X = DX*FLOAT(N-1)
740 DZ = RH(N)*DT
750 WRITE(6,904) X,RH(N),DZ
760 ML = 1 +MM*(N-1)
770 MH = MM*N
780 DO 128 M = ML,MH
790 A = DREAL(BX(M))
800 B = DIMAG(BX(M))
810 AMT = DSQRT(A*A + B*B)
820 THT = FF*DATAN2(B,A)
830 128 BX(M) = DCPLX(AMT,T+T)
840 WRITE(6,901) (BX(M),M=ML,MH)
850 130 CONTINUE
860 140 ICCC = ICCC +1
870 IF (ICCC.LE.ICCM) GO TO 3
880 901 FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
890 903 FORMAT(////' W,L =',2E15.5)
900 904 FORMAT('/' X,H,DZ =',3E15.5)
910 907 FORMAT(///' F,XMAX =',2E15.5//)
920 908 FORMAT(///' PRESSURE'/)
930 909 FORMAT('/' KE/PE =',E15.5)
940 910 FORMAT(' LONG WAVE')
950 911 FORMAT(' GENERAL FREQUENCY AND WAVELENGTH')
960 912 FORMAT(' RIGID LID')
970 913 FORMAT(' FREE SURFACE')
980 914 FORMAT(' TCTAL ENERGY/LENGTH =',E15.5)
990 915 FORMAT(' TAU Y DRIVING')
1000 916 FORMAT(' TAU X DRIVING')
1010 917 FORMAT(' ERROR: TAU X DRIVING IN THE LONG WAVE LIMIT')
1020 200 STOP
1030 END
1040
1050 SUBROUTINE NSQ
1060 COMMON F,DX,DT,NN,MM,NM,NMX
1070 COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
1080 COMMON XL(600)
1090 C
1100 C
1110 C PROGRAM TO READ AND INTERPOLATE BUOYANCY FREQUENCY SQUARE
1120 C BV = BUOYANCY FREQUENCY SQUARED
1130 C BVZ = Z DERIVATIVE OF BV
1140 C
1150 READ(5,*) NR,DZR,ALPH
1160 ALPH = ALPH*1.0E+05
1170 DZR = DZR*100.
1180 MX = MM -1
1190 READ(5,*) CMLT
1200 DO 20 I = 1,NR

```



```

1210      READ(5,*) XL(I)
1220  20    XL(I) = XL(I)*CMLT
1230      TF = XL(NR)
1240      ZZZ = DZR*FLOCAT(NR-1)
1250      DZI = RH(NN) - ZZZ
1260      NL = NR +1
1270      NH = 600
1280      DO 30 I = NL,NH
1290      Z = DZR*FLOCAT(I-1)
1300  30    XL(I) = TF*EXP((ZZZ - Z)/ALPH)
1310      DO 200 N = 1,NN
1320      DD = RH(N)
1330      DZ = DT*DD
1340      IF (DZ.GT.DZR) GO TO 110
1350      BV(N,MM) = XL(1)
1360      DO 50 M = 1,MX
1370      Z = DD - DZ*FLOCAT(M-1)
1380      IBXL = 1 + IFIX(Z/DZR)
1390      IBXH = IBXL +1
1400      ZS = DZR*FLOCAT(IBXL -1)
1410  50    BV(N,M) = XL(IBXL) + (Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
1420      Z = DD - DZ
1430      IBXL = 1 + IFIX(Z/DZR)
1440      IBXH = IBXL + 1
1450      ZS = DZR*FLOCAT(IBXL-1)
1460      AQ = XL(IBXL) + (Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
1470      GO TO 145
1480  110   ZD = DZ/2.
1490      XBB = 0.
1500      NAVG = 0
1510      DO 120 I = 1,NR
1520      ZC = DZR*FLOCAT(I-1)
1530      IF (ZC.GT.ZD) GO TO 120
1540      XBB = XBB + XL(I)
1550      NAVG = NAVG +1
1560  120   CONTINUE
1570      BV(N,MM) = XBB/FLOCAT(NAVG)
1580      DO 140 MQ = 1,MM
1590      M = MQ -1
1600      Z = DD - DZ*FLOCAT(M-1)
1610      ZS = Z - DZ/2.
1620      ZD = ZS + DZ
1630  125   XBB = 0.
1640      NAVG = 0
1650      DO 130 I = 1,NH
1660      ZC = DZR*FLOCAT(I-1)
1670      IF ( ZC.LT.ZS) GO TO 130
1680      IF (ZC.GT.ZD) GO TO 130
1690      XBB = XBB + XL(I)
1700      NAVG = NAVG + 1
1710  130   CONTINUE
1720      IF (NAVG.NE.0) GO TO 135
1730      IBXL = 1 + IFIX(Z/DZR)
1740      IBXH = IBXL + 1
1750      ZSS = DZR*FLOCAT(IBXL-1)
1760      IF (M.EQ.0) GO TO 133
1770      BV(N,M) = XL(IBXL) + (Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
1780      GO TO 140
1790  133   AQ = XL(IBXL) + (Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
1800      GO TO 140

```

```

1810 135 IF (M.EQ.0) GO TO 138
1820 BV(N,M) = XBB/FLOAT(NAVG)
1830 GO TO 140
1840 138 AQ = XBB/FLOAT(NAVG)
1850 140 CONTINUE
1860 145 DO 150 M = 2, MX
1870 IP = M + 1
1880 IM = M - 1
1890 150 BVZ(N,M) = (BV(N,IP) - BV(N,IM))/(2.*DZ)
1900 BVZ(N,1) = (BV(N,2) - AQ)/(2.0*DZ)
1910 BVZ(N,MM) = (BV(N,MM) - BV(N,MX))/DZ
1920 200 CONTINUE
1930 WRITE(6,901)
1940 WRITE(6,902) (BV(NN,J),J=1,MM)
1950 901 FORMAT(/ / ' NSQUARED AT XMAX ' /)
1960 902 FORMAT(2X,10E12.5)
1970 250 RETURN
1980 END
1990
2000 SUBROUTINE MATS(RL,W,AX,BX,R,RX,DD1,DD2,TX,TY,TXX,TYX,FFC)
2010 COMMON F,DX,DT,NN,PM,NP,NMX
2020 COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
2030 COMMON XL(600)
2040 DOUBLE COMPLEX AX(425,53),BX(425),AB,DCMPLX,B3,B5
2050 DOUBLE COMPLEX AA3,AA4,AA5,DQ,AA1,AA2,FFC(25)
2060 DOUBLE COMPLEX GAM,B4,ABB
2070 DIMENSION R(25),RX(25),TX(25),TY(25),TYX(25),TXX(25)
2080 C
2090 C
2100 C SUBROUTINE TO SET AND SOLVE THE P EQUATION
2110 C
2120 GG = 1./980.
2130 CSS = GG*2.0*DT*DD1
2140 CALL FFCCAL(TX,TY,TXX,TYX,FFC,W,RL,DD2)
2150 DXDT = DX/(2.*DT)
2160 DXX = DX*DX
2170 DDD = DXX/(DT*DT)
2180 RLL = DD2*RL*RL
2190 DDX = DX*DXDT
2200 RQP = 2.0*DX
2210 DQ = DCMPLX(RQP,0.0)
2220 FLW = F*RL/W
2230 3 DO 5 I = 1,NM
2240 5 BX(I) = (0.0,0.0)
2250 DO 10 J = 1,NMX
2260 DO 10 I = 1,NM
2270 10 AX(I,J) = (0.0,0.0)
2280 J = NN + 2
2290 IP1 = J + 1
2300 IM1 = J - 1
2310 IJJP = J + 2
2320 IJJM = J - 2
2330 IPM1 = 2*NN + 1
2340 IPM = 2*NN + 2
2350 IPMM = 2*NN + 3
2360 DO 100 M = 1,MM
2370 DO 90 N = 1,NN
2380 I = N + NN*(M-1)
2390 CALL AS(A1,A2,A3,W,N,M,C1,C2,C3,DD2)
2400 RQP = A2*DXDT

```

```

2410      AB = DCMLPX(RQP,0.0)
2420      AX(I,1) = AB
2430      AX(I,IPM1) = -AB
2440      AX(I,3) = -AB
2450      AX(I,IPMM) = AB
2460      AX(I,IM1) = (1.0,0.0)
2470      AX(I,IP1) = (1.0,0.0)
2480      RQP = A1*DDD - A3*CDX
2490      AX(I,2) = DCMLPX(RQP,0.0)
2500      RQP = A1*DDD + A3*DDX
2510      AX(I,IPM) = DCMLPX(RQP,0.0)
2520      RQP = -2.0 - 2.0*A1*DDD - RLL*DXX
2530      AX(I,J) = DCMLPX(RQP,0.0)
2540      IF (N.EQ.1) GO TO 20
2550      IF (N.EQ.NN) GO TO 30
2560      IF (M.EQ.1) GO TO 40
2570      IF (M.EQ.MM) GO TO 50
2580      GO TO 90
2590  20    RQP = 2.0*DX
2600      FW = F*F - DD2*W*W
2610      CHH = RL*F*RH(N)
2620      CH = R(N)*DD2*2.0*F*W*RL/FW
2630      AA1 = DCMLPX(CH,CHH)
2640      CH = R(N)*(F*F + DD2*W*W)/FW
2650      CHH = W*RH(N)
2660      AA2 = DCMLPX(CH,CHH)
2670      CH = F*TY(N)
2680      CHH = W*TX(N)*DD2
2690      ABB = DCMLPX(CH,CHH)
2700      ABB = 2.0*DX*ABB/AA2
2710      BX(I) = BX(I) + AX(I,IM1)*ABB
2720      BX(I) = BX(I) + AX(I,IPM1)*ABB
2730      BX(I) = BX(I) + AX(I,1)*ABB
2740      AX(I,IP1) = AX(I,IP1) + AX(I,IM1)
2750      AX(I,J) = AX(I,J) + RQP*AA1*AX(I,IM1)/AA2
2760      AX(I,IPM) = AX(I,IPM) + AX(I,IM1)*A2*DX/DT
2770      AX(I,2) = AX(I,2) - AX(I,IM1)*A2*DX/DT
2780      AX(I,IPMM) = AX(I,IPMM) + AX(I,IPM1)
2790      AX(I,IPM) = AX(I,IPM) + AX(I,IPM1)*RQP*(AA1/AA2 + A2/DT)
2800      AX(I,J) = AX(I,J) - AX(I,IPM1)*A2*RQP/DT
2810      AX(I,3) = AX(I,3) + AX(I,1)
2820      AX(I,2) = AX(I,2) + AX(I,1)*RQP*(AA1/AA2 - A2/DT)
2830      AX(I,J) = AX(I,J) + AX(I,1)*A2*RQP/DT
2840      AX(I,1) = (0.0,0.0)
2850      AX(I,IPM1) = (0.0,0.0)
2860      AX(I,IM1) = (0.0,0.0)
2870  24    IF (M.NE.MM) GO TO 25
2880      AX(I,2) = AX(I,2) + AX(I,IPM)
2890      AX(I,3) = AX(I,3) + AX(I,IPMM)
2900      CZQ = RH(N)*BV(N,M)*CSS
2910      AX(I,IP1) = AX(I,IP1) - AX(I,IPMM)*CZQ
2920      AX(I,J) = AX(I,J) - AX(I,IPM)*CZQ
2930      BX(I) = BX(I) - AX(I,IPM)*2.0*RH(1)*DT*FFC(1)
2940      BX(I) = BX(I) - AX(I,IPMM)*2.0*RH(1)*DT*FFC(1)
2950      AX(I,IPM) = (0.0,0.0)
2960      AX(I,IPMM) = (0.0,0.0)
2970      GO TO 90
2980  25    IF (M.NE.1) GO TO 90
2990      CH = R(N)*DD2*2.0*F*W*RL/FW
3000      CHH = F*RL*RH(N)

```

```

3010      B3 = DCMLX(CH,CHH)
3020      CH = R(N)*(F*F + DD2*W*W)/FW
3030      CHH = W*RH(N)
3040      B4 = DCMLX(CH,CHH)
3050      CH = F*TY(N)
3060      CHF = W*DD2*TX(N)
3070      B5 = DCMLX(CH,CHH)
3080      CH = -2.0*W*RL*DD2*F*RX(N)/FW
3090      CHH = -RL*F*RHX(N)
3100      AA1 = DCMLX(CH,CHH)
3110      CH = (F*F + DD2*W*W)*RX(N)/FW
3120      CHH = W*RHX(N)
3130      AA2 = DCMLX(CH,CHH)
3140      AA1 = AA1 + AA2*B3/B4
3150      CH = -R(N)*(F*F + DD2*W*W)*BVZ(N,M)/(BV(N,M)*BV(N,M))
3160      CH = CH + 2.0*W*RL*DD2*F*R(N)*RHX(N)/FW
3170      CHH = -W*FW/BV(N,M)
3180      AA2 = DCMLX(CH,CHH)
3190      CH = (F*F + DD2*W*W)*(-R(N)*RHX(N))/FW
3200      AA2 = AA2 + CH*B3/B4
3210      CH = R(N)*(F*F + DD2*W*W)/BV(N,M)
3220      AA3 = DCMLX(CH,0.0)
3230      CH = -(F*F + DD2*W*W)*RX(N)/FW
3240      CHF = -W*RHX(N)
3250      AA4 = DCMLX(CH,CHF)
3260      AA4 = AA4/B4
3270      DZ = DT*RH(N)
3280      ABB = (0.5*AA2 - AA3/DZ)/DZ
3290      AX(I,J) = AX(I,J) + AX(I,2)*(AA1 - 2.0*AA3/(DZ*DZ))/ABB
3300      AX(I,IPM) = AX(I,IPM) + AX(I,2)*(0.5*AA2 + AA3/DZ)/(DZ*ABB)
3310      BX(I) = BX(I) - AX(I,2)*AA4*B5/ABB
3320      RQP = 2.0*DX
3330      CALL AACAL(W,RL,R,RX,N,M,DD2,AA1,AA2,AA3,AA4,AA5)
3340      ABB = AA1*0.5/DT - AA4/(DT*DT)
3350      AX(I,J) = AX(I,J) + AX(I,3)*(-AA3/RQP + AA5/(DX*DT))/ABB
3360      B3 = AX(I,3)*(AA2 - 2.0*AA4/(DT*DT) - AA5/(DX*DT))/ABB
3370      AX(I,IP1) = AX(I,IP1) + B3
3380      AX(I,IJJP) = AX(I,IJJP) + AX(I,3)*AA3/(RQP*ABB)
3390      AX(I,IPM) = AX(I,IPM) - AX(I,3)*AA5/(DX*DT*ABB)
3400      B3 = AX(I,3)*(AA1*0.5/DT + AA4/(DT*DT) + AA5/(DX*DT))/ABB
3410      AX(I,IPMM) = AX(I,IPMM) + B3
3420      AX(I,3) = (0.0,0.0)
3430      AX(I,2) = (0.0,0.0)
3440      GO TO 90
3450 3C      DZ = DT*RH(N)
3460      AX(I,IPM1) = (0.0,0.0)
3470      AX(I,3) = (0.0,0.0)
3480      AX(I,IPMM) = (0.0,0.0)
3490      AX(I,1) = (0.0,0.0)
3500      C1 = 1. + FLW*DX/2.
3510      C2 = 1. - FLW*DX/2.
3520      AX(I,J) = AX(I,J) + 2.0*AX(I,IP1)/C1
3530      AX(I,IM1) = AX(I,IM1) - C2*AX(I,IP1)/C1
3540      AX(I,IP1) = (0.0,0.0)
3550      IF (M.NE.MM) GO TO 35
3560      AX(I,2) = AX(I,2) + AX(I,IPM)
3570      CZQ = RH(N)*BV(N,M)*CSS
3580      AX(I,J) = AX(I,J) - AX(I,IPM)*CZQ
3590      BX(I) = BX(I) - AX(I,IPM)*2.0*RH(N)*DT*FFC(N)
3600      AX(I,IPM) = (0.0,0.0)

```

```

3610      GO TO 90
3620 35    IF (M.NE.1) GO TO 90
3630      CA = R(N)*(F*F + DD2*h*h)/(F*F - DD2*h*h)
3640      CH = -CA*BVZ(N,M)/BV(N,M)
3650      CHH = -w
3660      ABB = DCMPLX(CH,CHH)
3670      ABB = ABB/CA
3680      AA1 = 1.0 - ABB*RH(N)*DT/2.0
3690      AA2 = 1.0 + ABB*DT*RH(N)/2.0
3700      AX(I,J) = AX(I,J) + AX(I,2)*2.0/AA1
3710      AX(I,IPM) = AX(I,IPM) - AX(I,2)*AA2/AA1
3720      AX(I,2) = (0.0,0.0)
3730      GO TO 90
3740 40    CALL AACAL(W,RL,R,RX,N,M,DD2,AA1,AA2,AA3,AA4,AA5)
3750      DTQ = 2.0*DT*DX
3760      RQP = 2.0*DX
3770      ABB = AA1*0.5/DT - AA4/(DT*DT)
3780      AX(I,IM1) = AX(I,IM1) + AX(I,2)*(-AA3/RQP + AA5/DTQ)/ABB
3790      AX(I,J) = AX(I,J) + AX(I,2)*(AA2 - AA4*2.0/(DT*DT))/ABB
3800      AX(I,IP1) = AX(I,IP1) + AX(I,2)*(AA3/RQP - AA5/DTQ)/ABB
3810      AX(I,IPM1) = AX(I,IPM1) + AX(I,2)*(-AA5/DTQ)/ABB
3820      AX(I,IPM) = AX(I,IPM) + AX(I,2)*(AA1*0.5/DT + AA4/(DT*DT))/ABB
3830      AX(I,IPMM) = AX(I,IPMM) + AX(I,2)*AA5/(DTQ*ABB)
3840      IF (N.EQ.2) GO TO 42
3850      AX(I,IJJM) = AX(I,IJJM) - AX(I,1)*AA3/(RQP*ABB)
3860      B3 = AX(I,1)*(AA2 - 2.0*AA4/(DT*DT) + AA5*2.0/DTQ)/ABB
3870      AX(I,IM1) = AX(I,IM1) + B3
3880      AX(I,J) = AX(I,J) + AX(I,1)*(AA3/RQP - AA5*2.0/DTQ)/ABB
3890      B3 = AX(I,1)*(AA1*0.5/DT + AA4/(DT*DT) - AA5*2.0/DTQ)/ABB
3900      AX(I,IPM1) = AX(I,IPM1) + B3
3910      AX(I,IPM) = AX(I,IPM) + AX(I,1)*AA5*2.0/(DTQ*ABB)
3920 41    AX(I,J) = AX(I,J) + AX(I,3)*(-AA3/RQP + AA5*2.0/DTQ)/ABB
3930      B3 = AX(I,3)*(AA2 - AA4*2.0/(DT*DT) - AA5*2.0/DTQ)/ABB
3940      AX(I,IP1) = AX(I,IP1) + B3
3950      AX(I,IJJP) = AX(I,IJJP) + AX(I,3)*AA3/(RQP*ABB)
3960      AX(I,IPM) = AX(I,IPM) - AX(I,3)*AA5*2.0/(DTQ*ABB)
3970      B3 = AX(I,3)*(AA1*0.5/DT + AA4/(DT*DT) + AA5*2.0/DTQ)/ABB
3980      AX(I,IPMM) = AX(I,IPMM) + B3
3990      AX(I,3) = (0.0,0.0)
4000      AX(I,1) = (0.0,0.0)
4010      AX(I,2) = (0.0,0.0)
4020      GO TO 90
4030 42    B3 = AX(I,1)*(AA2 - AA3/DX - 2.0*AA4/(DT*DT) + AA5*2.0/DTQ)/ABB
4040      AX(I,IM1) = AX(I,IM1) + B3
4050      AX(I,J) = AX(I,J) + AX(I,1)*(AA3/DX - AA5*2.0/DTQ)/ABB
4060      B3 = AX(I,1)*(0.5*AA1/DT + AA4/(DT*DT) - AA5*2.0/DTQ)/ABB
4070      AX(I,IPM1) = AX(I,IPM1) + B3
4080      AX(I,IPM) = AX(I,IPM) + AX(I,1)*AA5*2.0/(DTQ*ABB)
4090      GO TO 41
4100 50    AX(I,1) = (0.0,0.0)
4110      AX(I,3) = (0.0,0.0)
4120      AX(I,IPM1) = (0.0,0.0)
4130      AX(I,IPMM) = (0.0,0.0)
4140      AX(I,2) = AX(I,2) + AX(I,IPM)
4150      CZQ = RH(N)*BV(N,M)*CSS
4160      AX(I,J) = AX(I,J) - AX(I,IPM)*CZQ
4170      BX(I) = BX(I) - AX(I,IPM)*2.0*RH(N)*DT*FFC(N)
4180      AX(I,IPM) = (0.0,0.0)
4190 90    CONTINUE
4200 100   CONTINUE

```

```

4210      CALL BANDG(AX,BX)
4220      RETURN
4230      END
4240
4250      SUBROUTINE AS(A1,A2,A3,W,N,M,C1,C2,C3,CD2)
4260      COMMON F,DX,DT,NN,PM,NP,NMX
4270      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
4280      COMMON XX(600)
4290      C
4300      C          SUBROUTINE TO CALCULATE COEFFICIENTS FOR MATS
4310      C
4320      F2 = F*F
4330      W2 = DD2*W*W
4340      TZ = 1./RH(N)
4350      TH = -1. +DT*FLOAT(M-1)
4360      TX = -RHX(N)*TH*TZ
4370      AA = (RHX(N)/RH(N))**2
4380      TXX = (2.*AA -RHXX(N)/RH(N))*TH
4390      BW = BV(N,M)
4400      BWh = BW*Bh
4410      A1 = TX*TX + TZ*TZ*(F2-W2)/Bh
4420      A2 = TX
4430      A3 = TXX -TZ*BVZ(N,M)*(F2-W2)/BWh
4440      C1 = TZ
4450      C2 = RHX(N)*BV(N,1)/(F2-W2)
4460      C3 = C1*RHX(N)
4470      RETURN
4480      END
4490
4500      SUBROUTINE DEP(BX,RQPP)
4510      COMMON F,DX,DT,NN,PM,NP,NMX
4520      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
4530      COMMON XL(600)
4540      DIMENSION BX(50)
4550      C
4560      C          SUBROUTINE TO READ AND INTERPOLATE DEPTH PROFILE
4570      C          RH = DEPTH
4580      C          RHX = X DERIVATIVE OF RH
4590      C          RHXX = SECOND X DERIVATIVE OF RH
4600      C
4610      READ(5,*) NRX
4620      DO 5 I =1,NRX
4630      READ(5,*) XL(I),BX(I)
4640      BX(I) = BX(I)*100.
4650      5  XL(I) = XL(I)*1.0E+05
4660      RHMA = BX(NRX)
4670      RQPP = XL(NRX)
4680      RH(1) = BX(1)
4690      DO 20 N = 2,NN
4700      X = DX*FLOAT(N-1)
4710      IF (X.GT.XL(NRX)) GO TO 15
4720      IC = 0
4730      DO 8 J = 2,NRX
4740      IF (IC.NE.0) GO TO 8
4750      I = J
4760      IF (X.GT.XL(I)) GO TO 8
4770      IC = I
4780      8  CONTINUE
4790      IM = I-1
4800      RHX(N) = (BX(I) -BX(IM))/(XL(I)-XL(IM))

```

```

4810      XX = X - XL(IM)
4820      RH(N) = BX(IM) + RHX(N)*XX
4830      GO TO 20
4840      15      RH(N) = RHMA
4850      20      CONTINUE
4860      RHX(1) = (RH(2) - RH(1))/DX
4870      RHXX(1) = 0.
4880      D2 = 2.*DX
4890      DXX = DX*DX
4900      NX = NN - 1
4910      DO 30 N = 2,NX
4920      IP = N + 1
4930      IM = N - 1
4940      RHX(N) = (RH(IP) - RH(IM))/D2
4950      30      RHXX(N) = (RH(IP) - 2.*RH(N) + RH(IM))/DXX
4960      RHX(NN) = 0.
4970      RHXX(NN) = 0.
4980      RETURN
4990      END
5000
5010      SUBROUTINE VCAL(W,RL,BX,XL,R,RX,XINT,DD2,TX,TY)
5020      COMMON F,DX,DT,NN,MM,NM,NMX
5030      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
5040      COMMON XB(600)
5050      DOUBLE COMPLEX BX(425),XL(425),XPX,XPT,DCMPLX,RMU,T3,B3,B5,AA1,AA2
5060      DOUBLE COMPLEX GAM,B4,AA3,AA4,AA5
5070      DOUBLE PRECISION DREAL,DIMAG,DATAN2,DSCRT,A,B
5080      DIMENSION R(25),RX(25),TX(25),TY(25)
5090      DOUBLE PRECISION XINT
5100      C
5110      C          SUBROUTINE TO CALCULATE V FROM P
5120      C
5130      RHO = 0.5015
5140      FF = 180.0/3.14159
5150      XINT = 0.
5160      WRITE(6,905)
5170      FW = F*F - DD2*W*W
5180      WL = DD2*W*RL
5190      RLW = RL/W
5200      FLW = F*RL/W
5210      MX = MM - 1
5220      DXX = 2.*DX
5230      W2 = DD2*W*W
5240      DO 50 N = 1,NN
5250      XX = DX
5260      X = DX*FLOAT(N-1)
5270      D = RH(N)
5280      DD = RHX(N)/D
5290      DZ = DT*D
5300      RHZ = RHX(N)**2
5310      IF (N.EQ.1) GO TO 2
5320      IF (N.NE.NN) GO TO 20
5330      GO TO 18
5340      2      CH = R(N)*(F*F + W2)/FW
5350      CHH = W*RH(N)
5360      GAM = DCMPLX(CH,CHH)
5370      CH = F*TY(N)
5380      CHH = W*TX(N)*DD2
5390      XPX = DCMPLX(CH,CHH)
5400      XPX = XPX/GAM

```

```

5410      XPT = XPX
5420      CH = R(N)*DD2*2.0*W*F*RL/FW
5430      CHH = RL*F*RH(N)
5440      B4 = DCMPLX(CH,CHH)
5450      DO 5 I = 1,MM
5460      XPX = XPT -B4*BX(I)/GAM
5470      5      XL(I) = (F*XPX + WL*BX(I))/FW
5480      XX = DX/2.0
5490      GO TO 40
5500      18      C1 = 1. + FLW*DX/2.
5510      C2 = 1. - FLW*DX/2.
5520      DQ = 1./(F*DX)
5530      DO 19 M = 1,MM
5540      I = (N-1)*MM +M
5550      IM = I -MM
5560      XPX = (BX(I)*2.0/C1 - BX(IM)*(1. + C2/C1))/DXX
5570      19      XL(I) = (F*XPX + WL*BX(I))/FW
5580      XX = DX/2.
5590      GO TO 40
5600      20      I = (N-1)*MM +1
5610      M = 1
5620      IP = I + MM
5630      IM = I - MM
5640      I2 = I + 1
5650      I1 = I - 1
5660      IQQ = I -MM + 1
5670      IQR = I + MM + 1
5680      CALL AACAL(W,RL,R,RX,N,M,DD2,AA1,AA2,AA3,AA4,AA5)
5690      DTT = 2.0*DT
5700      GAM = AA1/DTT - AA4/(DT*DT)
5710      B3 = BX(IM)*(-AA3/DXX +AA5/(DXX*DT))
5720      B3 = B3 +BX(I)*(AA2-2.0*AA4/(DT*DT))
5730      B3 = B3 + BX(IP)*(AA3/DXX -AA5/(DXX*DT))
5740      B3 = B3 + BX(IQQ)*(-AA5/(DXX*DT))
5750      B3 = B3 + BX(I2)*(AA1/DTT + AA4/(DT*DT))
5760      B3 = B3 + BX(IQR)*(AA5/(DTT*DX))
5770      B3 = B3/GAM
5780      XPX = (BX(IP) - BX(IM))/DXX
5790      T = -1.0
5800      XPT = (BX(I2)-B3)/DTT
5810      XPX = XPX - T*DD*XPT
5820      XL(I) = (F*XPX + WL*EX(I))/FW
5830      I = N*MM
5840      IP = I + MM
5850      IM = I -MM
5860      XPX = (BX(IP) - BX(IM))/DXX
5870      XL(I) =(WL*BX(I) + F*XPX)/FW
5880      DO 30 M = 2,MX
5890      I = (N-1)*MM +M
5900      IP = I + MM
5910      IM = I -MM
5920      I2 = I +1
5930      I1 = I -1
5940      XPX = (BX(IP) - BX(IM))/DXX
5950      T = -1. +DT*FLOAT(M-1)
5960      XPT = (BX(I2) -BX(I1))/(2.*DT)
5970      XPX = XPX - T*DD*XPT
5980      30      XL(I) = (WL*BX(I) + F*XPX)/FW
5990      40      DO 45 M = 1,MM
6000      I = (N-1)*MM + M

```



```

5010      ZZ = DZ
5020      A = DREAL(XL(I))
5030      B = DIMAG(XL(I))
5040      CQ = A*A + B*B
5050      IF (M.EQ.1) GO TO 42
5060      IF (M.NE.MM) GO TO 45
5070      42      ZZ = DZ/2.
5080      45      XINT = XINT + RHO*XX*ZZ*CQ
5090      WRITE(6,904) X,D,DZ
5100      IL = (N-1)*MM + 1
5110      IH = N*MM
5120      DO 46 IJ = IL,IH
5130      A = DREAL(XL(IJ))
5140      B = DIMAG(XL(IJ))
5150      AMT = DSQRT(A*A + B*B)
5160      THT = FF*DATAN2(B,A)
5170      46      XL(IJ) = DCMPLX(AMT,THT)
5180      WRITE(6,901) (XL(IJ),IJ=IL,IH)
5190      50      CONTINUE
5200      WRITE(6,902) XINT
5210      DO 110 N = 1,NN
5220      IJ = 1 + MM*(N-1)
5230      A = DREAL(XL(IJ))
5240      B = DIMAG(XL(IJ))
5250      AMT = R(N)*A
5260      THT = B
5270      110     XL(N) = DCMPLX(AMT,THT)
5280      WRITE(6,906)
5290      WRITE(6,901)(XL(N),N=1,NN)
5300      901     FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
5310      902     FORMAT(///'  Y CONTRIBUTION TO KE =' ,E15.5/)
5320      905     FORMAT(////'  V'///)
5330      904     FORMAT('/'      X, H, DZ =' ,3E15.5)
5340      906     FORMAT(///'  Y BOTTOM STRESS (DYNES/CM2)='/)
5350      RETURN
5360      END
5370
5380      SUBROUTINE RHDC(BX,XL,XINT,DD1,DD2,FFC)
5390      COMMON F,DX,DT,NN,PM,NM,NMX
5400      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
5410      COMMON XQ(600)
5420      DOUBLE COMPLEX BX(425),XL(425),DCMPLX,FFC(25)
5430      DOUBLE PRECISION DREAL,DIMAG,DATAN2,DCSRT,A,B,CDABS
5440      DOUBLE PRECISION XINT
5450      C
5460      C      SUBROUTINE TO CALCULATE DENSITY FROM P
5470      C
5480      G2 = 980.*980./2.06
5490      GRQ = (1.0/980.0)**2
5500      FF = 180.0/3.14159
5510      XINT = 0.
5520      WRITE(6,903)
5530      G = 980.
5540      DO 50 N = 1,NN
5550      DXX = DX
5560      IF (N.EQ.1) GO TO 2
5570      IF (N.NE.NN) GO TO 5
5580      2      DXX = DX/2.
5590      5      X = DX*FLOAT(N-1)
5600      DZ = DT*RH(N)

```

```

6610      GDZ = G*DZ
6620      DO 40 M = 1,MM
6630      DZZ = DZ
6640      I = (N-1)*MM +M
6650      IF (M.EQ.1) GO TO 10
6660      IF (M.NE.MM) GO TO 20
6670      DZZ = DZ/2.
6680      XL(M) = DD1*BV(N,M)*CRC*BX(I) -FFC(N)/98C.
6690      GO TO 35
6700      10      IP = I + 1
6710      DZZ = DZ/2.
6720      XL(M) = -(BX(IP) - BX(I))/GDZ
6730      GO TO 35
6740      20      IP = I +1
6750      IM = I -1
6760      XL(M) = -(BX(IP) -BX(IM))/(2.*GDZ)
6770      35      XX = G2*CDABS(XL(M))*CDABS(XL(M))/BV(N,M)
6780      XINT = XINT + XX*DZZ*DXX
6790      40      CONTINUE
6800      WRITE(6,901) X,RH(N),DZ
6810      DO 45 M = 1,MM
6820      A = DREAL(XL(M))
6830      B = DIMAG(XL(M))
6840      AMT = 1000.C*DSQRT(A*A + B*B)
6850      IF (AMT.NE.C.0) GO TO 42
6860      41      THT = 0.0
6870      GO TO 45
6880      42      THT = FF*DATAN2(B,A)
6890      45      XL(M) = DCMPLX(AMT,THT)
6900      50      WRITE(6,902) (XL(M),M=1,MM)
6910      WRITE(6,904) XINT
6920      901      FORMAT(/'  X,D,DZ =',3E15.5)
6930      902      FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
6940      903      FORMAT(///'  RHO (SIGMA-T UNITS)'/)
6950      904      FORMAT(///'  RHO CONTRIBUTION TO PE =',E15.5/)
6960      RETURN
6970      END
6980
6990      SUBROUTINE UCAL(W,RL,BX,XL,R,XINT,DD2,TX,TY)
7000      COMMON F,DX,DT,NN,MM,NP,NMX
7010      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
7020      COMMON XX(600)
7030      DIMENSION R(25),TX(25),TY(25)
7040      DOUBLE COMPLEX BX(425),XL(425),DCMPLX,XPT,XPX,XPIM,GAM,B4
7050      DOUBLE PRECISION XINT
7060      DOUBLE PRECISION CDABS,DREAL,DIMAG,DATAN2,DSQRT,A,B
7070      C
7080      C      SUBROUTINE TO CALCULATE U FROM P
7090      C
7100      RHO = 0.5015
7110      FF = 180.0/3.14159
7120      XINT = 0.
7130      FW = F*F - DD2*W*W
7140      FL = F*RL
7150      WRITE(6,903)
7160      DO 100 N = 1,NN
7170      DXX = DX
7180      X = DX*FLOAT(N-1)
7190      DZ = DT*RH(N)
7200      IF (N.EQ.1) GO TO 86

```

```

7210      IF (N.EQ.NN) GO TO 110
7220      DD = RHX(N)/RH(N)
7230      DO 85 M = 1,MM
7240      I = (N-1)*MM + M
7250      IP = I + MM
7260      IM = I - MM
7270      T = -1. + DT*FLOAT(M-1)
7280      IF (M.EQ.1) GO TO 10
7290      IF (M.EQ.MM) GO TO 15
7300      GO TO 20
7310  10    I1 = I + 1
7320      XPT = (BX(I1) - BX(I))/DT
7330      GO TO 25
7340  15    XPT = (0.0,0.0)
7350      GO TO 25
7360  20    I2 = I+1
7370      I1 = I-1
7380      XPT = (BX(I2) - BX(I1))/(2.*DT)
7390  25    XPX = (BX(IP) - BX(IM))/(2.*DX)
7400      XPX = XPX - T*DD*XPX
7410  85    XL(M) = -(0.0,1.0)*(W*XPX + FL*BX(I))/FW
7420      GO TO 90
7430  86    CH = R(N)*(F*F + DD2*W*W)/FW
7440      CHF = W*RH(N)
7450      GAM = DCMPLX(CH,CHF)
7460      CH = F*TY(N)
7470      CHF = W*TX(N)*DD2
7480      XPX = DCMPLX(CH,CHF)
7490      XPX = XPX/GAM
7500      XPT = XPX
7510      CH = R(N)*DD2*2.0*W*F*RL/FW
7520      CHF = RL*F*RH(N)
7530      B4 = DCMPLX(CH,CHF)
7540      DO 87 M = 1,MM
7550      XPX = XPT - B4*BX(M)/GAM
7560  87    XL(M) = -(0.0,1.0)*(FL*BX(M) + W*XPX)/FW
7570      DXX = DX/2.0
7580      GO TO 90
7590  110   FLW = 0.5*FL/W
7600      DXX = DX/2.
7610      DO 120 M = 1,MM
7620      I = (NN-1)*MM + M
7630      IM = I - MM
7640      XPIM = 2.*BX(I)/DX + BX(IM)*(FLW - 1./DX)
7650      XPIM = XPIM/(FLW + 1./DX)
7660  120   XL(M) = -(0.0,1.0)*(FL*BX(I) + W*0.5*(XPIM-BX(IP))/DX)/FW
7670  90    DO 95 M = 1,MM
7680      DZZ = DZ
7690      IF (M.EQ.1) GO TO 92
7700      IF (M.NE.MM) GO TO 95
7710  92    DZZ = DZ/2.
7720  95    XINT = XINT + RHO*CDABS(XL(M))*CDABS(XL(M))*DXX*DZZ
7730      DO 98 M = 1,MM
7740      A = DREAL(XL(M))
7750      B = DIMAG(XL(M))
7760      AMT = CDABS(XL(M))
7770      THT = FF*DATAN2(B,A)
7780  98    XL(M) = DCMPLX(AMT,THT)
7790      WRITE(6,901) X,RH(N),DZ
7800      WRITE(6,902) (XL(M),M=1,MM)

```

```

7810 100 CONTINUE
7820 WRITE(6,904) XINT
7830 901 FORMAT(/' X,D,DZ =',3E15.5)
7840 902 FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
7850 903 FORMAT(///' U'/)
7860 904 FORMAT(///' U CONTRIBUTION TO KE =',E15.5/)
7870 RETURN
7880 END
7890
7900 SUBROUTINE BANDG(A,BB)
7910 COMMON F,DX,DT,NN,MM,NM,NMX
7920 COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
7930 DOUBLE COMPLEX A(425,53),BB(425)
7940 DOUBLE COMPLEX R
7950 C
7960 C SUBROUTINE TO DO BAND GAUSSIAN ELIMINATION
7970 C
7980 MBW = NN + 1
7990 NH = NM - 1
8000 MDIAG = MBW + 1
8010 DO 50 N = 1,NH
8020 IP = N + 1
8030 MH = N + MBW
8040 IF (MH.LE.NM) GO TO 5
8050 MH = NM
8060 5 DO 50 IR = IP,MH
8070 ICD = MBW + N + 1 - IR
8080 R = A(IR,ICD)/A(N,MDIAG)
8090 BB(IR) = BB(IR) - R*BB(N)
8100 DO 50 IC = IP,MH
8110 ICD = MBW + IC + 1 - IR
8120 ICB = MBW + IC + 1 - N
8130 A(IR,ICD) = A(IR,ICD) - R*A(N,ICB)
8140 50 CONTINUE
8150 DO 100 I = 1,NH
8160 N = NM - I + 1
8170 BB(N) = BB(N)/A(N,MDIAG)
8180 IL = N - MBW
8190 IH = N - 1
8200 IF (IL.GE.1) GO TO 60
8210 IL = 1
8220 60 DO 100 IR = IL,IH
8230 ICD = MBW + N + 1 - IR
8240 BB(IR) = BB(IR) - A(IR,ICD)*BB(N)
8250 100 CONTINUE
8260 BB(1) = BB(1)/A(1,MDIAG)
8270 RETURN
8280 END
8290
8300 SUBROUTINE FRIC(R,RX)
8310 COMMON F,DX,DT,NN,MM,NM,NMX
8320 COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
8330 DIMENSION R(25),RX(25),A(25)
8340 C
8350 C SUBROUTINE TO READ AND INTERPOLATE BOTTOM RESISTANCE
8360 C COEFFICIENT R
8370 C
8380 C RX = X DERIVATIVE OF R
8390 C
8400 READ(5,*) NF

```

```

8410      DD 5 I = 1,NF
8420      READ(5,*) A(I),XL(I)
8430      5  A(I) = A(I)*1.CE+05
8440      RMA = XL(NF)
8450      R(1) = XL(1)
8460      DD 20 N = 2,NN
8470      X = CX*FLOAT(N-1)
8480      IF (X.GT.A(NF)) GO TO 15
8490      IC = 0
8500      DD 8 J = 2,NF
8510      IF (IC.NE.0) GO TO 8
8520      I = J
8530      IF (X.GT.A(I)) GO TO 8
8540      IC = I
8550      8  CONTINUE
8560      IM = I - 1
8570      RQ = (XL(I) - XL(IM))/(A(I) - A(IM))
8580      XX = X - A(IM)
8590      R(N) = XL(IM) + XX*RQ
8600      GO TO 20
8610      15  R(N) = RMA
8620      20  CONTINUE
8630      RX(1) = (R(2) - R(1))/CX
8640      D2 = 2.0*DX
8650      NX = NN - 1
8660      DD 30 N = 2,NX
8670      IP = N + 1
8680      IM = N - 1
8690      30  RX(N) = (R(IP) - R(IM))/D2
8700      RX(NN) = 0.
8710      WRITE(6,901)
8720      WRITE(6,902) (R(I),I = 1,NN)
8730      WRITE(6,903)
8740      901  FORMAT(//' R(CM/SEC)'/)
8750      902  FORMAT(10E12.5)
8760      903  FORMAT(///)
8770      RETURN
8780      END
8790
8800      SUBROUTINE FFCCAL(TX,TY,TXX,TYX,FFC,h,RL,DD2)
8810      COMMON F,DX,DT,NN,MM,NM,NMX
8820      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
8830      DIMENSION TX(25),TY(25),TXX(25),TYX(25)
8840      DOUBLE COMPLEX FFC(25),DCMPLX
8850      C
8860      C          SUBROUTINE TO CALCULATE WIND FORCING TERMS
8870      C
8880      FW = W*(F*F - DD2*h*h)
8890      FW = 1.0/FW
8900      M = MM
8910      DD 20 N = 1,NN
8920      CC = BV(N,M)*FW
8930      CR = DD2*(-W*TXX(N) + RL*F*TX(N))
8940      CI = F*TYX(N) - DD2*RL*W*TY(N)
8950      FFC(N) = DCMPLX(CR,CI)
8960      20  FFC(N) = CC*FFC(N)
8970      RETURN
8980      END
8990
9000      SUBROUTINE PEC(BX,SPE,IRL)

```

```

9010      COMMON F,DX,DT,NN,MM,NM,NMX
9020      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
9030      DOUBLE COMPLEX BX(425)
9040      DOUBLE PRECISION XINT,CDABS
9050      C
9060      C          SUBROUTINE TO CALCULATE THE POTENTIAL ENERGY ASSOCIATED
9070      C          WITH FREE SURFACE ELEVATION
9080      C
9090      IF (IRL.EQ.0) GO TO 50
9100      GG = 1.0/(2.0*980.*1.03)
9110      XINT = 0.
9120      DO 20 N = 1,NN
9130      I = N*MM
9140      XX = DX
9150      IF (N.EQ.1) GO TO 5
9160      IF (N.EQ.NN) GO TO 5
9170      GO TO 20
9180      5      XX = DX/2.
9190      20      XINT = XINT + CDABS(BX(I))*CDABS(BX(I))*XX
9200      SPE = GG*XINT
9210      GO TO 60
9220      50      SPE = 0.
9230      60      WRITE(6,901) SPE
9240      901      FORMAT(/' FREE SURFACE POTENTIAL ENERGY =',E15.5)
9250      RETURN
9260      END
9270
9280      SUBROUTINE WRD(TX,TY,TXX,TYX,IXY)
9290      COMMON F,DX,DT,NN,MM,NM,NMX
9300      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
9310      DIMENSION TX(25),TY(25),TXX(25),TYX(25)
9320      C
9330      C          SUBROUTINE TO READ AND INTERPOLATE WIND STRESS PROFILES
9340      C
9350      C          TX = X WIND STRESS
9360      C          TXX = X DERIVATIVE OF TX
9370      C          TY = Y WIND STRESS
9380      C          TYX = X DERIVATIVE OF TY
9390      C
9400      READ(5,*) NRD
9410      IF (NRD.EQ.0) GO TO 100
9420      DO 5 I = 1,NRD
9430      READ(5,*) TXX(I),XL(I)
9440      5      TXX(I) = TXX(I)*1.0E+05
9450      RMA = XL(NRD)
9460      TX(1) = XL(1)
9470      DO 20 N = 2,NN
9480      X = DX*FLOAT(N-1)
9490      IF (X.GT.TXX(NRD)) GO TO 15
9500      IC = 0
9510      DO 8 J = 2,NRD
9520      IF (IC.NE.0) GO TO 8
9530      I = J
9540      IF (X.GT.TXX(I)) GO TO 8
9550      IC = I
9560      8      CONTINUE
9570      IM = I -1
9580      RQ = (XL(I) -XL(IM))/(TXX(I) -TXX(IM))
9590      XX = X - TXX(IM)
9600      TX(N) = XL(IM) + XX*RQ

```

```

9610      GO TO 20
9620      15      TX(N) = RMA
9630      20      CONTINUE
9640      GO TO 110
9650      100     DO 105 I = 1,NN
9660      105     TX(I) = 1.0
9670      110     TXX(1) = (TX(2) - TX(1))/DX
9680      D2 = 2.0*DX
9690      NX = NN - 1
9700      DO 30 N = 2,NX
9710      IP = N + 1
9720      IM = N - 1
9730      30      TXX(N) = (TX(IP) - TX(IM))/D2
9740      TXX(NN) = 0.
9750      IF (IXY.EQ.0) GO TO 140
9760      WRITE(6,901)
9770      GO TO 150
9780      140     WRITE(6,902)
9790      DO 145 I = 1,NN
9800      TY(I) = TX(I)
9810      TYX(I) = TXX(I)
9820      TX(I) = 0.0
9830      145     TXX(I) = 0.
9840      WRITE(6,903) (TY(I),I = 1,NN)
9850      GO TO 160
9860      150     DO 155 I = 1,NN
9870      TY(I) = 0.
9880      155     TYX(I) = 0.
9890      WRITE(6,903) (TX(I),I = 1,NN)
9900      901     FORMAT(' TAUX (DYNE/CM2)')
9910      902     FORMAT(' TAUY (DYNE/CM2)')
9920      903     FORMAT(10E12.5)
9930      160     RETURN
9940      END
9950
9960      SUBROUTINE AACAL(W,RL,R,RX,N,P,DD2,AA1,AA2,AA3,AA4,AA5)
9970      COMMON F,DX,DT,NN,MM,NM,NMX
9980      COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
9990      DIMENSION R(25),RX(25)
0000      DOUBLE COMPLEX AA1,AA2,AA3,AA4,AA5,DCMPLX
0010      C
0020      C          SUBROUTINE TO CALCULATE COEFFICIENTS FOR MATS
0030      C
0040      CALL AS(A1,A2,A3,W,N,M,C1,C2,C3,DD2)
0050      CB = F*F + DD2*W*W
0060      CC = F*F - DD2*W*W
0070      CA = BV(N,M)/(CC**2)
0080      CH = (-RX(N)*C3-R(N)*C3*C3)*CB-R(N)*CC*CB*BVZ(N,M)*C1/(BV(N,M)**2)
0090      CH = -CA*(CH + 2.0*DD2*W*RL*F*R(N)*C3)
0100      CHH = W*(C1 + C2*C3)
0110      AA1 = DCMPLX(CH,CHH)
0120      CH = CA*2.0*DD2*W*RL*F*R(N)
0130      CHH = C2*F*RL
0140      AA2 = DCMPLX(CH,CHH)
0150      CH = CA*CB*R(N)
0160      CHH = W*C2
0170      AA3 = DCMPLX(CH,CHH)
0180      CH = CB*R(N)*C3*C3 + R(N)*CB*CC*C1*C1/BV(N,M)
0190      CH = -CA*CH
0200      CHH = 0.

```

```
0210      AA4 = DCMLPX(CH,CHH)
0220      CH = -CA*CB*R(N)*C3
0230      AA5 = DCMLPX(CH,CHF)
0240      RETURN
0250      END
```





<b>REPORT DOCUMENTATION PAGE</b>	<b>1. REPORT NO.</b> WHOI-85-17	<b>2.</b>	<b>3. Recipient's Accession No.</b>
<b>4. Title and Subtitle</b> Programs for Computing Properties of Coastal-Trapped Waves and Wind-Driven Motions Over the Continental Shelf and Slope		<b>5. Report Date</b> April 1985	
<b>7. Author(s)</b> Kenneth H. Brink, David C. Chapman		<b>8. Performing Organization Rept. No.</b> WHOI-85-17	
<b>9. Performing Organization Name and Address</b> Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543		<b>10. Project/Task/Work Unit No.</b>	
		<b>11. Contract(C) or Grant(G) No.</b> (C) (G)OCE 84-08563	
<b>12. Sponsoring Organization Name and Address</b> National Science Foundation		<b>13. Type of Report &amp; Period Covered</b> Technical	
		<b>14.</b>	
<b>15. Supplementary Notes</b> This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-85-17.			
<b>16. Abstract (Limit: 200 words)</b>  Documentation and listings are presented for a sequence of computer programs to be used for problems in continental shelf dynamics. Three of the programs are to be used for computing properties of free and forced coastal-trapped waves. A final program may be used to compute wind-driven fluctuations over the continental shelf and slope.			
<b>17. Document Analysis a. Descriptors</b> 1. coastal-trapped waves 2. wind-driven shelf currents 3. continental shelf  <b>b. Identifiers/Open-Ended Terms</b>   <b>c. COSATI Field/Group</b>			
<b>18. Availability Statement:</b> Approved for publication; distribution unlimited.		<b>19. Security Class (This Report)</b> UNCLASSIFIED	<b>21. No. of Pages</b> 98
		<b>20. Security Class (This Page)</b>	<b>22. Price</b>

# DOCUMENT LIBRARY

April 9, 1985

## DISTRIBUTION LIST FOR TECHNICAL REPORT EXCHANGE

Institute of Marine Sciences Library  
University of Alaska  
O'Neill Building  
905 Koyukuk Ave., North  
Fairbanks, AK

Attn: Stella Sanchez-Wade  
Documents Section  
Scripps Institution of Oceanography  
Library, Mail Code C-075C  
La Jolla, CA 92093

Hancock Library of Biology & Oceanography  
Alan Hancock Laboratory  
University of Southern California  
University Park  
Los Angeles, CA 90089-0371

Gifts & Exchanges  
Library  
Bedford Institute of Oceanography  
P.O. Box 1006  
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International  
Ice Patrol  
c/o Coast Guard R & D Center  
Avery Point  
Groton, CT 06340

Library  
Physical Oceanographic Laboratory  
Nova University  
8000 N. Ocean Drive  
Dania, FL 33304

NOAA/EDIS Miami Library Center  
4301 Rickenbacker Causeway  
Miami, FL 33149

Library  
Skidaway Institute of Oceanography  
P.O. Box 13687  
Savannah, GA 31416

Institute of Geophysics  
University of Hawaii  
Library Room 252  
2525 Correa Road  
Honolulu, HI 96822

Library  
Chesapeake Bay Institute  
4800 Atwell Road  
Shady Side, MD 20876

MIT Libraries  
Serial Journal Room 14E-210  
Cambridge, MA 02139

Director, Ralph M. Parsons Laboratory  
Room 48-311  
MIT  
Cambridge, MA 02139

Marine Resources Information Center  
Bldg. E38-320  
MIT  
Cambridge, MA 02139

Library  
Lamont-Doherty Geological Observatory  
Columbia University  
Palisades, NY 10964

Library  
Serials Department  
Oregon State University  
Corvallis, OR 97331

Pell Marine Science Library  
University of Rhode Island  
Narragansett Bay Campus  
Narragansett, RI 02882

Working Collection  
Texas A&M University  
Dept. of Oceanography  
College Station, TX 77843

Library  
Virginia Institute of Marine Science  
Gloucester Point, VA 23062

Fisheries-Oceanography Library  
151 Oceanography Teaching Bldg.  
University of Washington  
Seattle, WA 98195

Library  
R.S.M.A.S.  
University of Miami  
4600 Rickenbacker Causeway  
Miami, FL 33149