*WHOI-87-24*

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

**-Second Edition-**

by

Kenneth H. Brink

David C. Chapman

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543



June 1987

## Technical Report

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



**Approved for Distribution:**

Robert C Beardsley

**Robert C. Beardsley, 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. This second edition includes several minor revisions and corrections in the computer code and the documentation.

## COMMENTS ON THE SECOND EDITION

In May 1987, we found that we had run out of copies of the original report. Rather than simply make more copies of the original report, we chose to create a revised version with a few improvements in the programs and in the documentation.

Only programs BTCSW and BIGLOAD2 have been modified, and most of the changes in these are either minor corrections or cosmetic improvements in the output. One set of changes in BTCSW will make a substantial difference in estimating the decay time for barotropic Kelvin waves. These corrections involve the statements

```
CALL LGWV ----        (main program)
SUBROUTINE LGWV ---- (subroutine LGWV)
```

and three lines following statement 220 in subroutine LGWV. In all cases, statements that have been changed from the original version are bracketed by a line of 10 "C"s, e.g.

```
CCCCCCCCCC
200  C = WB/RL
CCCCCCCCCC
```

in subroutine LGWV, program BTCSW.

Also, note that the line numbers for the program listings are now consecutive within each subroutine rather than for the entire listing.

As in the past, please feel free to contact either of us if programming bugs or inconsistencies should be detected.

## Table of Contents


| | | Page |
|---|---|---|
| 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 | 26 |
| 5. | Wind-Driven Motions | 32 |
| | Word of Caution | 44 |
| | Acknowledgements | 45 |
| | References | 46 |
| | Program Listings | |
| | BTCSW | 47 |
| | BIGLOAD2 | 67 |
| | CROSS | 89 |
| | BIGDRV2 | 96 |

# 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.

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

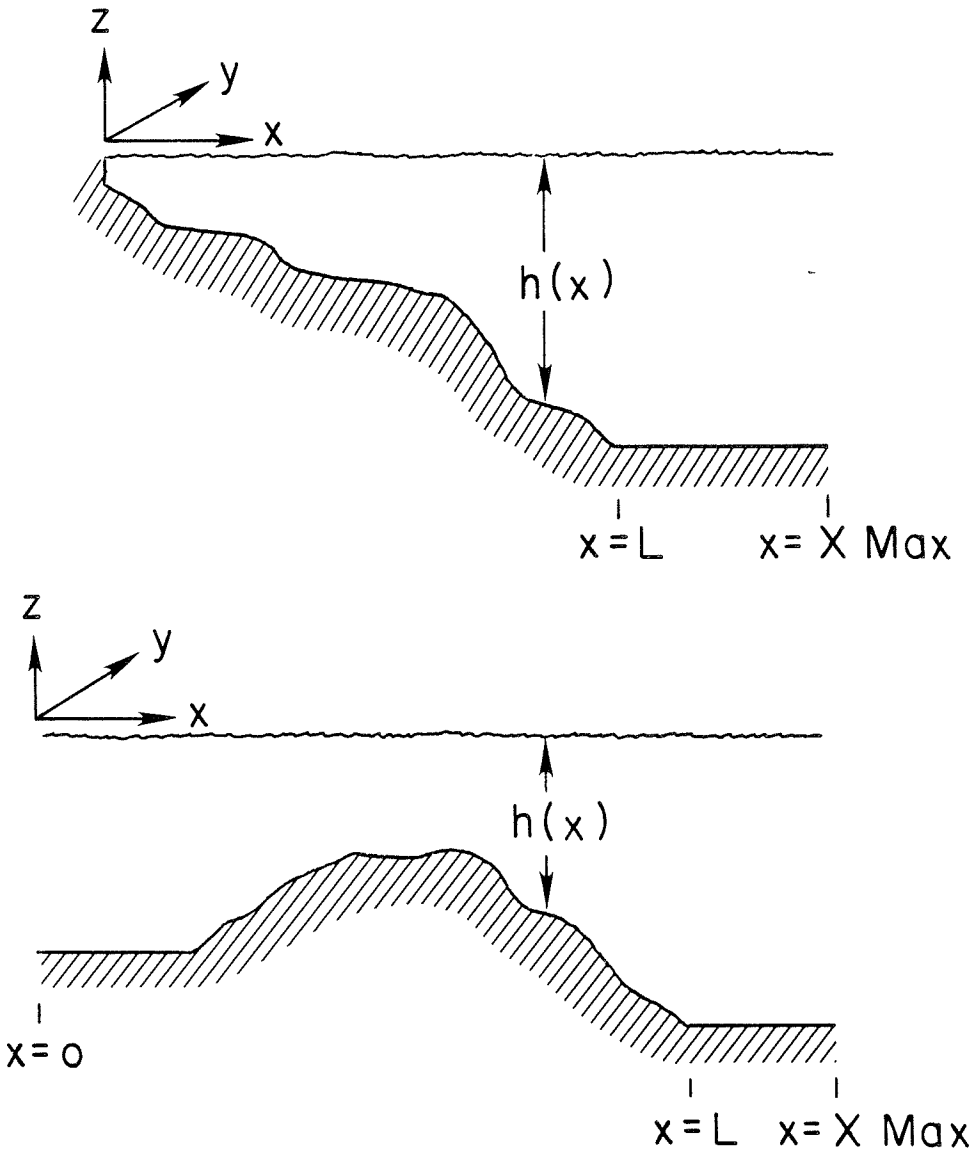


Figure 1: Topography and coordinate system definitions used in all programs: (upper) with a coast, (lower) without a coast.

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 <u>monotonic</u> 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 $\underline{\hat{\tau}}(x)\ \exp[i(\omega t + \ell y)]$ 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:

$$\varepsilon U_t + \varepsilon v_o U_y - fV = -gh\zeta_x \ , \tag{2.1a}$$

$$V_t + v_o V_y + U v_{ox} + fU = -gh\zeta_y \ , \tag{2.1b}$$

$$\delta\zeta_t + v_o\zeta_y + U_x + V_y = 0 \ , \tag{2.1c}$$

where the onshore, and alongshelf directions are $x$ and $y$, respectively, and there are no alongshelf variations in the mean flow $v_o(x)$ 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 $\varepsilon$ and $\delta$ are defined as follows:

$\varepsilon = 0$ for the long-wave approximation,
$\varepsilon = 1$ 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 $\exp[i(\omega t + \ell y)]$, (2.1) become

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

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

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

where $\omega' = \omega + \ell v_o$

and $f' = f + v_{ox}$ .

These can be reduced to either:

$$0 = U_{xx}[\delta \frac{\omega'^2}{g} - h\ell^2]h$$

$$+ U_x[h_x \ell^2 - \frac{2\omega'\ell v_{ox}\delta}{g}]h$$

$$+ U[-\frac{\delta^2\omega'^2}{g^2}(ff' - \varepsilon\omega'^2) + \varepsilon h^2\ell^4$$

$$+ \frac{\delta}{g} h\ell^2(ff' - 2\omega'^2\varepsilon + 2f'v_{ox})$$

$$- \frac{h\ell^3 f' h_x}{\omega'} - h\frac{\ell}{\omega'} v_{oxx}(\delta\frac{\omega'^2}{g} - h\ell^2)] \qquad (2.2)$$

or

$$0 = \zeta_{xx}[ff' - \varepsilon\omega'^2]h$$

$$+ \zeta_x[(ff' - \omega'^2\varepsilon)h_x - h(f v_{oxx} - 2\omega'\ell v_{ox}\varepsilon)]$$

$$+ \zeta[-\frac{\delta}{g}(ff' - \varepsilon\omega'^2)^2 + h_x\frac{f\ell}{\omega'}(ff' - \varepsilon\omega'^2)$$

$$- h\frac{f\ell}{\omega'}(fv_{oxx} - 2\varepsilon\omega'\ell v_{ox})$$

$$-h\ell^2\varepsilon(ff' - \omega'^2\varepsilon)] \ . \qquad (2.3)$$

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, <u>but</u> it is only valid if $h$ and $v_o$ 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, <u>but</u> it is only valid for $h$ and $v_o$ 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:

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

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 $\ell = 0.5 \times 10^{-8} cm^{-1}$ and the others will be (1.5, 2.5, 3.5,...) $\times 10^{-8} cm^{-1}$.

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 $f = 7.5 \times 10^{-5} s^{-1}$.

XMAX is the distance (in km) from x = 0 to the offshore boundary of the grid (Figure 1). Typically, XMAX ~ 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 programs 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 $\times 10^{-8} cm^{-1}$). The program multiplies WW(I) by $10^{-5}$, so WW(1) = 0.5 corresponds to $\omega = 0.5 \times 10^{-5} s^{-1}$.

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.

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

where $E_o$ 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

$$\left(\frac{f + v_{ox}}{h}\right)_x$$

changes sign, a warning is given.

E. Input Summary

The input file should include:

```
IMDM      NN
NITM      ISD      EPS      DEL
IUP       ILLW
IDD1      IDD3     IDD4
NCALM     ILW
RLF       DRL
IPC
F         XMAX
NRX
X         H   ┐
              │
              ├──  NRX times
              │
              ┘
NRD
X         V   ┐
              │
              ├──  NRD times
              │
              ┘
WW(1)     WW(2)    WW(3)
NW
X         WF  ┐
              │
              ├──  NW times
              │
              ┘
```

F. Program Output

The output (file 6) includes statements about which boundary conditions were used, and listings of $h(x)$, $v_o(x)$, $U_n(x)$, $\zeta_n(x)$, $WF(x)$ and $\phi_n(x)$. The functions of $x(h, v_o, U, \zeta, WF, \phi)$ are listed at $\Delta x$ increments across the

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 $v_o = 0$, 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

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

and $Y_n$ obeys

$$b_n' \tau_o^y = Y_{ny} - \frac{1}{c_n} Y_{nt} - r_o \sum_m a_{nm}' Y_m \ .$$

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

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

where WF is as above, $r_o$ 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, $\omega = 0.6867 \times 10^{-5} s^{-1}$, $a_{11} = 0.19865 \times 10^{-7} cm^{-1}$ and $b_1 = 0.1428 \times 10^{-1} cm^{-1/2}$. 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

$$\left.\begin{aligned} \varepsilon u_t - fv &= -\frac{1}{\rho_o} p_x \\ v_t + fu &= -\frac{1}{\rho_o} p_y \\ 0 &= -p_z - g\rho \\ u_x + v_y + w_z &= 0 \end{aligned}\right\} \quad (3.1)$$

and

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

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 $\varepsilon$ 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:

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

subject to

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

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

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

and

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

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

$$p \ \text{bounded} \qquad \text{as} \quad x \to \infty \ ,$$

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

line 3: F XMAX

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

XMAX is the offshore extent of the grid in km. Typically, XMAX ~ 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 $s^{-1}cm^{-1}$.

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 $10^{-6}s^{-1}$, so that WH(1) = 0.5 corresponds to $\omega = 0.5 \times 10^{-6}s^{-1}$.

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,

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

for n = 1, 2, 3,... 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 $10^{-6}s^{-1}$, so a value of 0.5 corresponds to $\omega = 0.5 \times 10^{-6}s^{-1}$.

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:

$$N^2 = N_o^2 \exp ( \zeta_o - \zeta )/ALPH)$$

where

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

$\zeta_o$ 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 $(rad/s)^2$. Specifically,

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

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 cm $s^{-1}$ defined by

$$\frac{1}{\rho_o} \underline{\tau}_B = r(x)\, \underline{v}(x, -h) \ .$$

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

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

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 a double precision external (IMSL) subroutine to solve the matrix equation.

vii.) Some users of this program may not have access to the IMSL package. In this case, subroutine BANDG from program BIGDRV2 can be modified to replace the IMSL routine LEQT1B.

Simply:

- replace

```
CALL LEQT1B(AX,NM,NDD,NDD,NM,BX,1,NM,0,XL,IER)
```

  in subroutine MATS with

```
CALL BANDG(AX,BX)
```

- transfer subroutine BANDG from BIGDRV2 into BIGLOAD2 and replace its fourth and fifth lines with

```
DOUBLE PRECISION A(425,53),BB(425)
DOUBLE PRECISION R.
```

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

CML T ┐
$N^2$ │
└ 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 dyne/cm$^2$, then v would be in cm/s.

After the first ($\omega$, $\ell$ ) point on a dispersion curve, only p will be listed, and in this case it is <u>not</u> 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 (1986). 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.
1.0E-06
1.25
1.25
1
0.0         0.05
```

This represents a uniform $N^2$ and a uniformly sloping shelf. After six iterations, $\omega/\ell$ = 312.49 cm/s for the n = 1 mode. The coupling coefficients are

$$b_n = 0.368 \times 10^{-2} cm^{-1/2}$$

$$a_{nn} = 0.18543 \times 10^{-8} \ cm^{-1} \ .$$

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 $\gamma << 1$, 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

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

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

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

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

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) $\ell$ 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 $\ell$ 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 $\ell$ 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 $10^{-5}s^{-1}$ within the program. Thus, F = 5.0 corresponds to $f = 5. \times 10^{-5}s^{-1}$.

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:

$$N^2 = N_o^2 \exp((\zeta_o - \zeta)/ALPH)$$

where

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

$\zeta_o$ 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 $(rad/s)^2$. Specifically,

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

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 ( $\ell_{minimum}$, $\ell_{maximum}$, $\Delta\ell$ ) 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 $cm^{-1}$. Thus, (X5, X6, X7) = (2., 3., .1) corresponds to $\ell_{min} = 2. \times 10^{-7} cm^{-1}$, $\ell_{max} = 3. \times 10^{-7} cm^{-1}$, $\Delta\ell = 0.1 \times 10^{-7} cm^{-1}$. The program then locates a root by shooting using $\ell = \ell_{min} + n\Delta\ell$ (n = 0, 1, 2...) up to $\ell = \ell_{max}$. If a root is located, Newton's method is used to home in on it. If no root is found between $\ell_{min}$ and $\ell_{max}$, 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 $\omega \tilde{>} .9f$, 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.

iv.) If X5 and X6 are chosen such that two or more solutions for $\ell$ lie between the two values, then the program will only find one of the solutions.

## E. Input Summary

NV DD1
F XMAX
NRX
X H ┐
└ NRX times
NR DZR ALPH
CMLT
$N^2$ ┐
└ NR times
NRS
X5 X6 X7 ┐
└ NRS times

## 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 $n = 0, 1, 2 \ldots$ 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 $(p_o e^{-\ell x})$ normalized by $\left(\int_{-H}^{0} p_o^2 dz\right)^{1/2}$. The pressure is listed at $z = -(n + 1/2)\,\Delta z$ where n = 1,2,3...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.
  1.0E-6
   56.25
   56.25
       2
       1.   2.   .1
       2.   3.   .1
```

This represents a uniformly sloping shelf with uniform stratification. The analytical solution of (4.1,4.2) is $\ell = \frac{n\pi}{L}\left[\left(\frac{NH}{fL}\right)^2 - 1\right]^{-1/2}$ (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:

$$\varepsilon u_t - fv = \frac{-1}{\rho_o} p_x \tag{5.1a}$$

$$v_t + fu = \frac{-1}{\rho_o} p_y \tag{5.1b}$$

$$0 = - p_z - g\rho \tag{5.1c}$$

$$u_x + v_y + w_z = 0 \tag{5.1d}$$

$$0 = \rho_t + w\rho_{oz} \ . \tag{5.1e}$$

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

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

The Boussinesq approximation is made throughout. Finally, subscripts x, y, z and t represent partial differentiation. The quantity $\varepsilon$ 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,

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

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

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

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

or

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

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

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

The boundary conditions are

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

$$\text{at} \quad z = -h(x),$$

$$0 = -\rho_o w + i\omega\delta g^{-1} p + (f^2 - \varepsilon\omega^2)^{-1}[(i\omega\varepsilon T^x + fT^y)_x + \varepsilon \ell (-ifT^x - \omega T^y)] \tag{5.4b}$$

$$\text{at} \quad z = 0,$$

$$0 = u_x \qquad \text{at} \quad x = \text{XMAX} , \tag{5.4c}$$

and

$$0 = -i(\ell f_p + \omega p_x)h + f(T^y - \rho_o r v_B) + i\omega\varepsilon(T^x - \rho_o r u_B) \tag{5.4d}$$

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

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

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

and

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

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 (1986) suggests that the "coast" be placed such that

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

where $r(x)$ is defined by

$$\underline{\tau}_B = \rho_0 r \underline{v}_B .$$

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

$$\delta = 0$$

$$\varepsilon = 0$$

$$T^x = 0$$

$$T_x^{\ y} = 0 .$$

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 $10^{-5}s^{-1}$ within the program. For example, F = 5. represents f = 5. x $10^{-5}s^{-1}$.

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 $\varepsilon$ 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:

$$N^2 = N_o^2 \exp((\zeta_o - \zeta)/ALPH)$$

where

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

$\zeta_o$ 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 $(rad/s)^2$. Specifically,

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

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/$cm^2$ for all x.

following lines: X T

These are values of offshore distance (x) in km and wind stress amplitude in dyne/$cm^2$. 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 $\ell = 0$ 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

ICCM

F XMAX

ILW IRL IXY

NRX

X H — NRX times

NR DZR ALPH

CMLT

$N^2$ — NR times

NF

X R — NF times

NW

X T — NW times

W RL — ICCM times

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.) $N^2$ $(rad/s)^2$ 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
1.0E-05             2.0E-08
```

The resulting output has the following energy components:

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

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

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

$p \rightarrow 0.45 \times 10^{12}$ 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 $-9°$. The $v$ maximum is at the surface at $x = 8.33$ km $(85, -48°)$.

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

The maximum in density is at the bottom at $x = 8.33$ km with $\rho = 0.032\ \sigma_t$ and a phase of $-39°$. 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.

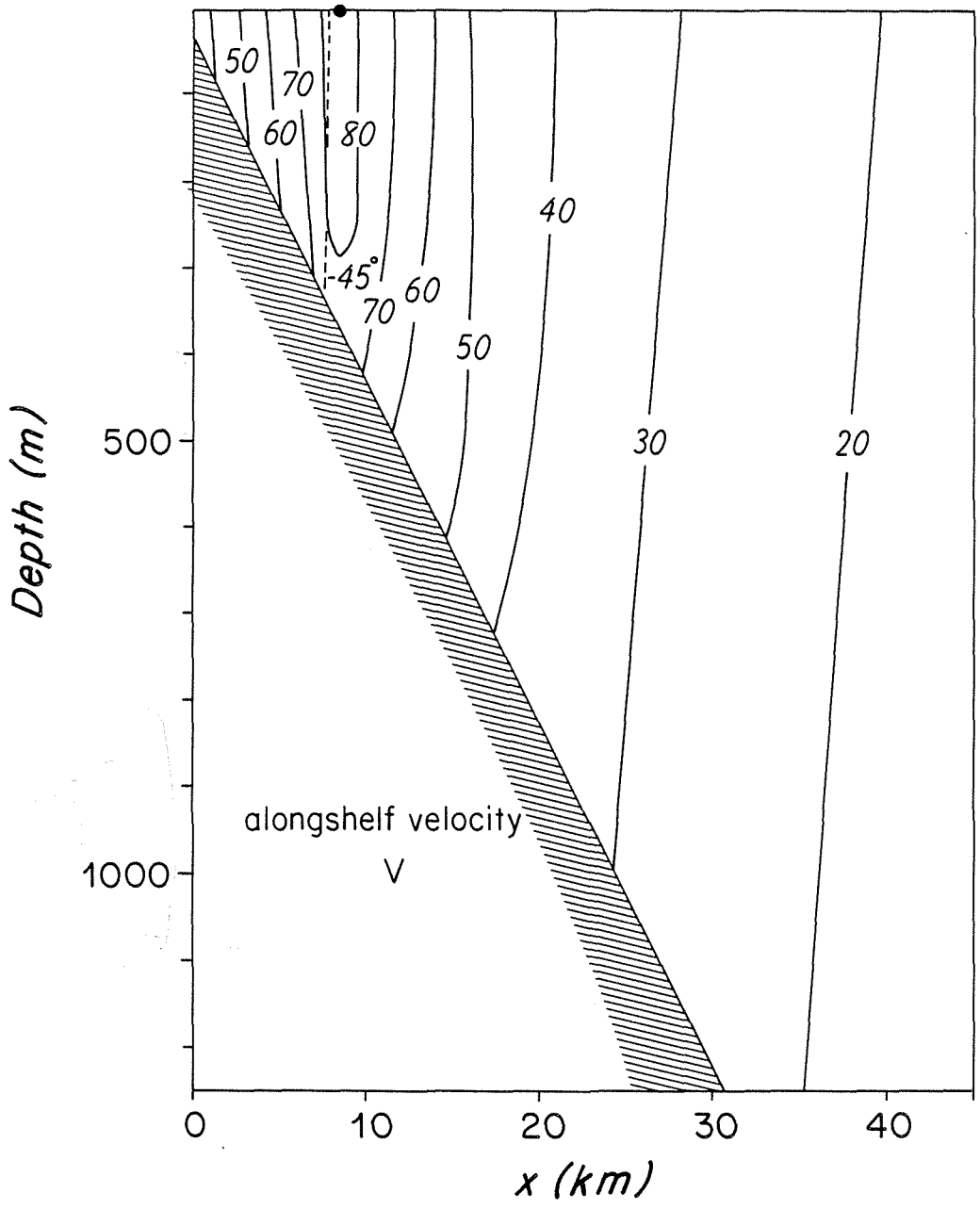


Figure 2: Alongshelf velocity for the example in section 5G. Amplitude (cm/s) is shown in solid lines and phase is shown by dashed contours. Only the upper 1250 m is shown.

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

All fields become weaker far offshore and at great depth. The v and p fields have a roughly 180° 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. The publication of the second edition was supported by National Science Foundation (NSF) grant OCE84-17769.

## 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, 15, 439-453.

Clarke, A. J. and S. Van Gorder, 1986. A method for estimating wind-driven, time-dependent, stratified shelf and slope water flow. Journal of Physical Oceanography, 16, 1013-1028.

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, 1986. The frictional nearshore response to forcing by synoptic scale winds. Journal of Physical Oceanography, 16, 934-946.

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.

# 1 Program BTCSW Listing

```
0001            PROGRAM BTSW
0002            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0003            DIMENSION BH(400),WW(3),RX(3),RLH(400),WH(400),CH(400)
0004            DIMENSION WF(400)
0005    C
0006    C                MAIN PROGRAM CALLS ALL OF THE PIECES
0007    C
0008    C                NN LESS THAN OR EQUAL 100
0009            IMD = 1
0010            READ(5,*) IMDM,NN
0011    C                DD1 = 1.  FREE SURFACE
0012    C                DD1 = 0.  RIGID LID
0013    C                IDD3 = 1  U = 0 AT X = XMAX
0014    C                IDD3 = 0  UX = 0 AT X = XMAX
0015    C                IDD3 = 2 REAL B.C. AT X = XMAX FOR VX,HX = 0. THERE
0016    C                IDD4 =1   U = 0 AT X = 0
0017    C                IDD4 = 0  UX = 0 AT X = 0
0018    C                IDD4 = 2  DECAY B.C. AT X = 0 FOR VX,HX = 0 THERE
0019    C                IPC = 0  REDUCED PRINT OUT
0020    C                IUP = 0 SEARCH IN U
0021    C                IUP = 1 SEARCH IN P
0022    C                ILLW = 0 STRICTLY LONG WAVE
0023    C                ILLW = 1 GENERAL FREQUENCY AND WAVENUMBER
0024    C                        REVISED JUNE,1987
0025    2       READ(5,*) NITM,ISD,EPS,DEL
0026            READ(5,*) IUP,ILLW
0027            READ(5,*) IDD1,IDD3,IDD4
0028            DD1 = FLOAT(IDD1)
0029            NCAL = 1
0030            READ(5,*) NCALM,ILW
0031            READ(5,*) RLF,DRL
0032            READ(5,*) IPC
0033            DRL = DRL*1.0E-08
0034            RLF = RLF*1.0E-08
0035            READ(5,*) F,XMAX
0036            XMAX = XMAX*1.0E+05
0037            F = F*1.0E-05
0038            DX = XMAX/FLOAT(NN-1)
0039            WRITE(6,907) F,XMAX,DX
0040            IF (IUP.NE.0) GO TO 315
0041    312     WRITE(6,924)
0042            IUP = 0
0043            GO TO 320
0044    315     IF (IUP.NE.1) GO TO 312
```

```
0045            WRITE(6,925)
0046    320     IF (IDD4.NE.0) GO TO 305
0047            WRITE(6,921)
0048            GO TO 310
0049    305     IF (IDD4.EQ.2) GO TO 306
0050            IDD4 = 1
0051            WRITE(6,922)
0052            GO TO 310
0053    306     WRITE(6,928)
0054    310     IF (IDD1.EQ.1) GO TO 3
0055            DD1 = 0.
0056            WRITE(6,917)
0057            GO TO 4
0058    3       WRITE(6,918)
0059    4       IF (IDD3.EQ.1) GO TO 6
0060            IF (IDD3.EQ.2) GO TO 7
0061            IDD3 = 0
0062            WRITE(6,919)
0063            GO TO 8
0064    6       WRITE(6,920)
0065            GO TO 8
0066    7       WRITE(6,923)
0067    8       IF (ILLW.EQ.1) GO TO 11
0068            ILLW = 0
0069            NCALM = 1
0070            WRITE(6,926)
0071            GO TO 12
0072    11      WRITE(6,927)
0073    12      REPS = FLOAT(ILLW)
0074            CALL DEP(NN,XMAX)
0075            CALL VRD(NN,XMAX,F)
0076            DO 10 I = 1,NCALM
0077    10      RLH(I) = DRL*FLOAT(I-1) + RLF
0078            RL = RLH(NCAL)
0079            READ(5,*) (WW(J),J=1,3)
0080            DO 5 J = 1,3
0081    5       WW(J) = WW(J)*1.0E-05
0082    1       RB = 0.
0083            WRITE(6,908)
0084            DO 18 J = 1,3
0085            W = WW(J)
0086            IF (IUP.EQ.1) GO TO 330
0087            CALL MUTS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
0088            GO TO 9
0089    330     CALL MATS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
0090    9       CP = W/RL
```

```
0091            CALL CRLC(W,RL,NN,ICR)
0092            WRITE(6,905) W,RL,CP,RI,ICR
0093            RX(J) = RI
0094            IF ( RI.LT.RB) GO TO  18
0095            RB = RI
0096            ICRH = ICR
0097            WB = W
0098            DO 15 I = 1,NN
0099    15      BH(I) = B(I)
0100    18      CONTINUE
0101            NIT = 3
0102            IGP = 0
0103    20      CALL NGW(WW,RX,WB,WN,ISUC,EPS,DEL,IN,ISD,IGP)
0104            IF (ISUC.EQ.1) GO TO 100
0105            IF (IUP.EQ.1) GO TO 340
0106            CALL MUTS(NN,XMAX,DD1,F,WN,RL,IDD3,RI,IDD4,REPS)
0107            GO TO 345
0108    340     CALL MATS(NN,XMAX,DD1,F,WN,RL,IDD3,RI,IDD4,REPS)
0109    345     IF (IN.NE.0) GO TO 29
0110            IF (RI.GT.RB) GO TO 21
0111            IF (WN.LT.WW(2)) GO TO 23
0112    22      IN = 3
0113            GO TO 29
0114    23      IN = 1
0115            GO TO 29
0116    21      IF (WN.GT.WW(2)) GO TO 23
0117            GO TO 22
0118    29      RX(IN) = RI
0119            WW(IN) = WN
0120            CP = WN/RL
0121            CALL CRLC(WN,RL,NN,ICR)
0122            WRITE(6,905) WN,RL,CP,RI,ICR
0123            IF ( RI.LT.RB) GO TO 30
0124            WB = WN
0125            ICRH = ICR
0126            RB = RI
0127            DO 25 I = 1,NN
0128    25      BH(I) = B(I)
0129    30      NIT = NIT + 1
0130            IF ( NIT.LT.NITM) GO TO 20
0131            WRITE(6,902) NIT
0132            GO TO 140
0133    100     DO 110 I = 1,NN
0134    110     BH(I) = BH(I)/RB
0135            CC = WB/RL
0136            IF ( ICRH.EQ.0) GO TO 115
```

```
0137            WRITE(6,911) ICRH
0138    115     WRITE(6,903) WB,RL,CC,EPS,DX,NIT
0139            IF (NCAL.EQ.1) GO TO 120
0140            IF (IPC.EQ.0) GO TO 150
0141    120     IF (IUP.EQ.1) GO TO 350
0142            WRITE(6,913)
0143            GO TO 360
0144    350     WRITE(6,912)
0145    360     WRITE(6,904) (BH(I),I=1,NN)
0146            IF (IUP.EQ.0) GO TO 135
0147            CALL MUTS(NN,XMAX,DD1,F,WB,RL,IDD3,RI,IDD4,REPS)
0148            DO 143 I = 1,NN
0149    143     B(I) = B(I)/RI
0150            WRITE(6,913)
0151            WRITE(6,904) (B(I),I = 1,NN)
0152    135     IF (RH(1).EQ.0.0) GO TO 150
0153            CALL MATS(NN,XMAX,DD1,F,WB,RL,IDD3,RI,IDD4,REPS)
0154            DO 130 I = 1,NN
0155    130     B(I) = B(I)/RI
0156            IF (IUP.EQ.1) GO TO 145
0157            WRITE(6,912)
0158            WRITE(6,904) (B(I),I=1,NN)
0159    145     IF (ILW.EQ.0) GO TO 150
0160            IF (NCAL.NE.1) GO TO 150
0161            IF ( RH(1).EQ.0.) GO TO 150
0162            IF (IDD4.NE.1) GO TO 150
0163            WRITE(6,908)
0164    CCCCCCCCCC
0165            CALL LGWV(NN,XMAX,F,WF,WB,RL,DD1)
0166    CCCCCCCCCC
0167    150     WH(NCAL) = WB
0168            CH(NCAL) =CC
0169            NCAL = NCAL + 1
0170            IF ( NCAL.GT.NCALM) GO TO 250
0171            RL = RLH(NCAL)
0172            IF (NCAL.GE.3) GO TO 200
0173            WW(2) = CC*RL
0174            GO TO 205
0175    200     I1 = NCAL - 2
0176            I2 = NCAL - 1
0177            CG = (WH(I2) - WH(I1))/(RLH(I2) - RLH(I1))
0178            WW(2)=  WH(I2) + CG*(RLH(NCAL)-RLH(I2))
0179    205     WW(1) = WW(2)*(1. - DEL)
0180            WW(3) = WW(2)*(1. +DEL)
0181            GO TO 1
0182    250     WRITE(6,909)
```

```
0183            DO 260 I = 1,NCALM
0184    260     WRITE(6,910) WH(I),RLH(I),CH(I)
0185            IMD = IMD +1
0186            IF (IMD.LE.IMDM) GO TO 2
0187    902     FORMAT(//'  USED UP',I3,' ITERATIONS'/)
0188    903     FORMAT(//'  CONVERGED: W,L,C,EPS,DX,NIT =',5E15.5,I5)
0189    904     FORMAT(10E13.5)
0190    905     FORMAT(/'  W,RL,CP,RI,ICR =',4E15.5,I10)
0191    907     FORMAT(//'  F,XMAX,DX =',3E15.5/)
0192    908     FORMAT(///)
0193    909     FORMAT(///'      W                L                   C'/)
0194    910     FORMAT(3E15.5)
0195    911     FORMAT(/'  SOLUTION HAS ',I3,' CRITICAL LAYERS'/)
0196    912     FORMAT(/'   ZETA'/)
0197    913     FORMAT(/'    U'/)
0198    917     FORMAT(/'  RIGID LID'/)
0199    918     FORMAT(/'  FREE SURFACE'/)
0200    919     FORMAT('  UX = 0 AT X = XMAX'/)
0201    920     FORMAT('  U = 0 AT X = XMAX'/)
0202    921     FORMAT('  UX = 0 AT X = 0'/)
0203    922     FORMAT('  U = 0 AT X = 0'/)
0204    923     FORMAT('  REAL B.C.AT X = XMAX')
0205    924     FORMAT(/' SEARCH IN U')
0206    925     FORMAT(/' SEARCH IN P')
0207    926     FORMAT(' LONG WAVE EXACTLY')
0208    927     FORMAT(' GENERAL FREQUENCY AND WAVENUMBER')
0209    928     FORMAT(' DECAY CONDITION AT X = 0')
0210    140     STOP
0211            END


0001
0002            SUBROUTINE NGW(WW,RX,WB,WN,ISUC,EPS,DEL,IN,ISD,IGP)
0003            DIMENSION WW(3),RX(3)
0004    C
0005    C                SUBROUTINE TO GUESS THE NEXT W
0006    C
0007    5       IC = 0
0008            DO 10 I = 1,2
0009            I1 = I +1
0010            AA = ABS(WW(I1))
0011            BB = ABS(WW(I))
0012            IF (AA.GT.BB) GO TO 10
0013            IC = 1
0014            RI = RX(I)
0015            WX = WW(I)
```

```
0016            RX(I) = RX(I1)
0017            WW(I) = WW(I1)
0018            WW(I1) = WX
0019            RX(I1) = RI
0020    10      CONTINUE
0021            IF (IC.NE.0) GO TO 5
0022            ISUC = 0
0023            IF ( RX(3).GT.RX(2)) GO TO 150
0024            IF (RX(1).GT.RX(2)) GO TO 160
0025            IL = 1
0026            IH = 3
0027            IF (RX(3).GT.RX(1)) GO TO 9
0028            IL = 3
0029            IH = 1
0030    9       WB1 = (WW(1) +WW(2))/2.
0031            WB2 = (WW(2) + WW(3))/2.
0032            RW1 = (RX(2) - RX(1))/(WW(2) - WW(1))
0033            RW2 = (RX(3) - RX(2))/(WW(3) -WW(2))
0034            A = (RW2 -RW1)/(WB2 - WB1)
0035            B = RW1  - A*WB1
0036            WN = -B/A
0037            EP= (WN -WB)/WB
0038            EP = ABS(EP)
0039            IF ( EP.LT.EPS) GO TO 100
0040            IGP = IGP + 1
0041            IF (WN.LT.WW(1)) GO TO 15
0042            IF (WN.LT.WW(3)) GO TO 20
0043    15      WN = (WW(2) +WW(IH))/2.
0044    20      IN = 0
0045            GO TO 130
0046    100     ISUC = 1
0047            IF (IGP.NE.0) GO TO 130
0048            IGP = IGP + 1
0049            ISUC = 0
0050            WN = (WW(2) + WW(IL))/2.
0051            GO TO 20
0052    150     IF (RX(1).GT.RX(2)) GO TO 180
0053    155     IF (ISD.EQ.1) GO TO 165
0054            WN = WW(3)*(1. + DEL)
0055            WW(1) = WN
0056            RX(1) = 0.
0057            IN = 1
0058            GO TO 130
0059    160     IF (RX(3).GT.RX(2)) GO TO 180
0060    165     IF (ISD.EQ.-1) GO TO 155
0061            WN = WW(1)*(1. - DEL)
```

```
0062            WW(3) = WN
0063            RX(3) = 0.
0064            IN = 3
0065            GO TO 130
0066    180     IF (RX(3).GT.RX(1)) GO TO 155
0067            GO TO 165
0068    130     RETURN
0069            END
```

```
0001
0002            SUBROUTINE CALC(NN,RX)
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004            DOUBLE PRECISION RI,DSQRT
0005    C
0006    C                   SUBROUTINE TO CALCULATE THE INTEGRAL OF RESPONSE SQUARED
0007    C
0008            RI = (B(1)**2 + B(NN)**2)/2.
0009            NX = NN - 1
0010            DO 5 I = 2,NX
0011    5       RI = RI + B(I)**2
0012            RI = DSQRT(RI)
0013            RX = RI
0014            RETURN
0015            END
```

```
0001
0002            SUBROUTINE DEP(NN,XMAX)
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004            READ(5,*) NRX
0005    C
0006    C                   SUBROUTINE TO READ AND INTERPOLATE THE DEPTH PROFILE
0007    C                           RH = DEPTH
0008    C                           RHX = X DERIVATIVE OF DEPTH
0009    C
0010            DO 10 I = 1,NRX
0011            READ(5,*) A(I),B(I)
0012            A(I) = A(I)*1.0E+05
0013    10      B(I) = B(I)*100.
0014            DX = XMAX/FLOAT(NN-1)
0015            RH(1) = B(1)
0016            DO 20 N = 2,NN
0017            X = DX*FLOAT(N-1)
```

```
0018            IF (X.GT.A(NRX)) GO TO 15
0019            IC = 0
0020            DO 8 J = 2,NRX
0021            IF (IC.NE.0) GO TO 8
0022            I = J
0023            IF (X.GT.A(I)) GO TO 8
0024            IC = I
0025    8       CONTINUE
0026            IM = I -1
0027            AA = (B(I) -B(IM))/(A(I) - A(IM))
0028            XX = X - A(IM)
0029            RH(N) = B(IM) +AA*XX
0030            GO TO 20
0031    15      RH(N) = B(NRX)
0032    20      CONTINUE
0033            RHX(1) = (RH(2) - RH(1))/DX
0034            NM = NN -1
0035            RHX(NN) = (RH(NN) - RH(NM))/DX
0036            D2 = 2.*DX
0037            DO 30 N = 2,NM
0038            IP = N +1
0039            IM = N -1
0040    30      RHX(N) = (RH(IP) - RH(IM))/D2
0041            WRITE(6,903)
0042            WRITE(6,902) (RH(N),N=1,NN)
0043    902     FORMAT(10E13.5)
0044    903     FORMAT(//'  DEPTH IN CM'/)
0045            RETURN
0046            END


0001
0002    CCCCCCCCCC
0003            SUBROUTINE LGWV(NN,XMAX,F,WFF,WB,RL,DD1)
0004    CCCCCCCCCC
0005            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0006            DIMENSION WFF(400)
0007    C
0008    C                SUBROUTINE TO CALCULATE THE BN AND ANN IN THE LONG WAVE
0009    C                        LIMIT
0010    C
0011            DX = XMAX/FLOAT(NN -1)
0012            DXX = 2.*DX
0013            GF = 980./F
0014            CALL WFR(NN,XMAX,WFF)
0015    CCCCCCCCCC
```

```
0016            IF (DD1.NE.0.) GO TO 200
0017    CCCCCCCCCC
0018            A(1) = 0.
0019            DO 100 N = 2,NN
0020            A(N) = (RHX(N)*B(N) + RHX(1)*B(1))/2.
0021            NX = N - 1
0022            IF (NX.LE.1) GO TO 90
0023            DO 20 I = 2,NX
0024    20      A(N) = A(N) + RHX(I)*B(I)
0025    90      A(N) = A(N)*DX
0026            A(N) = A(N) + RH(1)*B(1) -RH(N)*B(N)
0027    100     A(N) = -GF*A(N)
0028            NX = NN - 1
0029            WF = RHX(1)*(A(1)/RH(1))**2
0030            WF = WF + RHX(NN)*(A(NN)/RH(NN))**2
0031            IFZ = 0
0032            WF = WF/2.
0033            DO 150 N = 2,NX
0034            IF ( IFZ.NE.0) GO TO 150
0035            IF (RHX(N).NE.0.) GO TO 140
0036            IFZ = N
0037    140     WF = WF + RHX(N)*(A(N)/RH(N))**2
0038    150     CONTINUE
0039            WF = WF*DX
0040            WF = SQRT(WF*WF)
0041            WF = SQRT(WF)
0042            DO 160 N = 1,NN
0043    160     A(N) = A(N)/WF
0044            NH = IFZ - 1
0045            PX = (A(2) -A(1))/(DX*RH(1))
0046            WF = 0.5*PX*PX*WFF(1)
0047            BN = 0.5*RHX(NH)*A(NH)/(RH(NH)**2)
0048            NH = NH -1
0049            DO 180 N = 2,NH
0050            BN = BN + RHX(N)*A(N)/(RH(N)**2)
0051            I1 = N - 1
0052            I2 = N + 1
0053            PX = (A(I2) -A(I1))/DXX
0054            PX = PX/RH(N)
0055    180     WF = WF+ PX*PX*WFF(N)
0056            BN = BN*DX
0057            WF = WF*RH(1)*DX
0058            WRITE(6,905)
0059            WRITE(6,901) WF
0060            WRITE(6,903) BN
0061            WRITE(6,904)
```

```
0062            WRITE(6,902) (A(I),I=1,NN)
0063    CCCCCCCCCC
0064    200     C = WB/RL
0065    CCCCCCCCCC
0066            NX = NN - 1
0067            WF = RHX(1)*B(1)*B(1)*0.5
0068            DO 210 I = 2,NX
0069    210     WF = WF +RHX(I)*B(I)*B(I)
0070            WF = WF + 0.5*RHX(NN)*B(NN)*B(NN)
0071            WF = WF*DX
0072            WF = WF + RH(1)*B(1)*B(1)
0073            WF = SQRT(WF)
0074            DO 215 I = 1,NN
0075    215     B(I) = B(I)/WF
0076            BX = -F*B(1)/C
0077            WF = 0.5*WFF(1)*BX*BX
0078            DO 220 I = 2,NX
0079            IP = I + 1
0080            IM = I - 1
0081            BX = (B(IP) - B(IM))/DXX
0082    220     WF = WF + WFF(I)*BX*BX
0083            WF = WF*DX/F
0084    CCCCCCCCCC
0085            ALX =DD1/(980.0*RH(NN))
0086            ALX = SQRT(ALX)
0087            WF = WF + 0.5*ALX*B(NN)*B(NN)*WFF(NN)
0088    CCCCCCCCCC
0089            BN = B(1)
0090            WRITE(6,906)
0091            WRITE(6,907) WF
0092            WRITE(6,908) BN
0093            WRITE(6,909)
0094            WRITE(6,902) (B(I),I = 1,NN)
0095    901     FORMAT(//'  ANN =',E15.5,' 1/CM')
0096    902     FORMAT(10E13.5)
0097    903     FORMAT(/'  BN =',E15.5,' CM-1/2'/)
0098    904     FORMAT(//'  PHI ='/)
0099    905     FORMAT(' USING STREAM FUNCTION')
0100    906     FORMAT(//' USING PRESSURE')
0101    907     FORMAT(' ANN = ',E15.5,' SEC/CM2')
0102    908     FORMAT(' BN =',E15.5,' CM-1/2')
0103    909     FORMAT(' NORMALIZED PRESSURE (CM-1/2)')
0104            RETURN
0105            END
```

```
0001
0002            SUBROUTINE MUTS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004            DOUBLE PRECISION A1,A2,A3,AL,DSQRT
0005    C
0006    C                    SUBROUTINE TO SET UP AND SOLVE THE U EQUATION
0007    C
0008            NU = NN
0009            IF (IDD3.NE.1) GO TO 2
0010            NU = NN - 1
0011    2       IF (IDD4.NE.1) GO TO 4
0012            NU = NU - 1
0013    4       DX = XMAX/FLOAT(NN-1)
0014    5       GG = DD1/980.
0015            RLL = RL*RL
0016            DXX = DX*DX
0017            DDD = DX/2.
0018            DO 100 N = 1,NU
0019            N1 = N
0020            IF (IDD4.NE.1) GO TO 8
0021            N1 = N + 1
0022    8       B(N) = 0.
0023            WP = W + RL*V(N1)
0024            W2 = WP*WP
0025            RR = RH(N1)
0026            RX = RHX(N1)
0027            WX = RL*VX(N1)
0028            FP = F + VX(N1)
0029            A1 = RR*(GG*W2 - RR*RLL)
0030            A2 = RR*(RX*RLL -2.0*WP*WX*GG)
0031            A3 = -GG*GG*W2*(F*FP -REPS*W2)
0032            A3 = A3 + REPS*RR*RR*RLL*RLL
0033            A3 = A3 + GG*RR*RLL*(-2.0*W2*REPS + F*FP + 2.0*FP*VX(N1))
0034            A3 = A3 - RR*RLL*RL*FP*RX/WP
0035            A3 = A3 - RR*RL*VXX(N1)*(GG*W2 - RR*RLL)/WP
0036            N1 = N + NU
0037            N2 = N + 2*NU
0038            A(N) = -2.*A1 + A3*DXX
0039            A(N1) = A1 -A2*DDD
0040            A(N2) = A1 + A2*DDD
0041            IF (N.EQ.1) GO TO 20
0042            IF (N.EQ.NU) GO TO 30
0043            GO TO 100
0044    20      IF (IDD4.NE.0) GO TO 25
```

```
0045            A(N2) = A(N1) + A(N2)
0046            A(N1) = 0.
0047            GO TO 100
0048    25      IF (IDD4.EQ.1) GO TO 28
0049            WRITE(6,*) A1,A2,A3,REPS,GG,RR,W2,RLL,FP,F,RX,WP,VX(N1),VXX(N1)
0050            AL = DSQRT(-A3/A1)
0051            A(N2) = A(N2) + A(N1)
0052            A(N) = A(N) -2.0*DX*AL*A(N1)
0053    28      A(N1) = 0.
0054            GO TO 100
0055    30      IF (IDD3.NE.0) GO TO 35
0056            A(N1) = A(N1) + A(N2)
0057            A(N2) = 0.
0058            GO TO 100
0059    35      IF (IDD3.EQ.1) GO TO 36
0060            AL = DSQRT(-A3/A1)
0061            A(N1) = A(N1) +A(N2)
0062            A(N) = A(N) -2.0*DX*AL*A(N2)
0063    36      A(N2) = 0.0
0064    100     CONTINUE
0065            B(5) = 1.
0066            CALL TRI(NU)
0067            IF (IDD4.NE.1) GO TO 140
0068            DO 120 I = 1,NU
0069            N = NU -I +1
0070            N1 = N + 1
0071    120     B(N1) = B(N)
0072            B(1) = 0.
0073    140     IF (IDD3.NE.1) GO TO 150
0074            B(NN) = 0.
0075    150     CALL CALC(NN,RI)
0076            RETURN
0077            END



0001
0002            SUBROUTINE VRD(NN,XMAX,F)
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004    C
0005    C               SUBROUTINE TO READ AND INTERPOLATE THE MEAN V PROFILE
0006    C                       V = ALONGSHORE VELOCITY
0007    C                       VX = X DERIVATIVE OF V
0008    C                       VXX = SECOND X DERIVATIVE OF V
0009    C
0010            DX = XMAX/FLOAT(NN-1)
```

```
0011            READ(5,*) NRD
0012            IF (NRD.EQ.0) GO TO  200
0013            DO 5 I = 1,NRD
0014            READ(5,*) A(I),B(I)
0015    C                 A = DISTANCE FROM SHORE IN KM
0016    C                 B = V IN CM/SEC
0017    5       A(I) = A(I)*1.0E+05
0018    10      DO 100 N = 1,NN
0019            X = DX*FLOAT(N-1)
0020            IF (X.GT.A(NRD)) GO TO 90
0021            IF (X.LT.A(1)) GO TO 80
0022            J = 0
0023            DO 20 I = 2,NRD
0024            IF (J.NE.0) GO TO 20
0025            IF (X.GT.A(I)) GO TO 20
0026            J = I
0027    20      CONTINUE
0028            JJ = J - 1
0029            XX = A(J) - X
0030            XY = A(J) - A(JJ)
0031            V(N) = B(J) -XX*(B(J) -B(JJ))/XY
0032            GO TO 100
0033    80      V(N) =  0.
0034            GO TO 100
0035    90      V(N) = B(NRD)
0036    100     CONTINUE
0037            VX(1) = (V(2) - V(1))/DX
0038            NQ = NN - 1
0039            VX(NN) = (V(NN) - V(NQ))/DX
0040            DXX = DX*DX
0041            DZ = 2.*DX
0042            VXX(1) = 0.
0043            VXX(NN) = 0.
0044            DO 120 N = 2,NQ
0045            I1 = N - 1
0046            I2 = N + 1
0047            VX(N) = (V(I2) - V(I1))/DZ
0048    120     VXX(N) = (V(I2) - 2.*V(N) + V(I1))/DXX
0049            WRITE(6,904)
0050            WRITE(6,903) (V(N),N=1,NN)
0051            IST = 0
0052            DO 150 N = 1,NN
0053            Q = -RHX(N)*(F+VX(N)) + RH(N)*VXX(N)
0054            A(N) = Q
0055            IF (IST.NE.0) GO TO 150
0056            IF (N.NE.1) GO TO 130
```

```
0057            PQ = Q
0058            GO TO 150
0059    130     IF (PQ.EQ.0.) GO TO 135
0060            RQ = Q/PQ
0061            IF (RQ.LT.0.) GO TO 140
0062            IF (Q.EQ.0.) GO TO 150
0063    135     PQ = Q
0064            GO TO 150
0065    140     IST = 1
0066    150     CONTINUE
0067            IF (IST.EQ.0) GO TO 250
0068            WRITE(6,905)
0069            WRITE(6,906)
0070            WRITE(6,905)
0071            WRITE(6,907)
0072            WRITE(6,903) (A(N),N=1,NN)
0073            GO TO 250
0074    200     DO 220 N = 1,NN
0075            V(N) = 0.
0076            VX(N) = 0.
0077    220     VXX(N) = 0.
0078    903     FORMAT(10E13.5)
0079    904     FORMAT(//'  V IN CM/SEC'/)
0080    905     FORMAT(/' ************************************************'/)
0081    906     FORMAT('  POSSIBILITY OF UNSTABLE MODES')
0082    907     FORMAT(//'  PBARX'/)
0083    250     RETURN
0084            END



0001
0002            SUBROUTINE MATS(NN,XMAX,DD1,F,W,RL,IDD3,RI,IDD4,REPS)
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004            DOUBLE PRECISION A1,A2,A3,B1,B2,B3,BB1,BB2,BX,DSQRT,AL
0005    C
0006    C                SUBROUTINE TO SET UP AND SOLVE THE PRESSURE EQUATION
0007    C
0008            DX = XMAX/FLOAT(NN - 1)
0009            GG = DD1/980.
0010            RLL = RL*RL*REPS
0011            DXX = DX*DX
0012            DDX = 2.*DX
0013            DDD = DX/2.
0014            DO 100 N = 1,NN
0015            B(N) = 0.
```

```
0016            WP = W + RL*V(N)
0017            FP = F + VX(N)
0018            FW = F*FP - WP*WP*REPS
0019            RR = RH(N)
0020            RX = RHX(N)
0021            WX = RL*VX(N)
0022            FLW = F*RL/WP
0023            A1 = FW*RR
0024            A2 = FW*RX - RR*(F*VXX(N) -2.0*WP*WX*REPS)
0025            A3 = -GG*FW*FW +RX*FLW*FW -RR*FLW*(F*VXX(N) -2.*WP*WX*REPS)
0026            A3 = A3 -RR*RLL*FW
0027            N1 = N + NN
0028            N2 = N + 2*NN
0029            A(N) = -2.*A1 + A3*DXX
0030            A(N1) = A1 - A2*DDD
0031            A(N2) = A1 + A2*DDD
0032            IF (N.EQ.1) GO TO 20
0033            IF (N.EQ.NN) GO TO 30
0034            GO TO 100
0035    20      IF (IDD4.NE.0) GO TO 25
0036            B1 = RL*RR*FP
0037            B2 = RLL*WP*RR +GG*WP*FW
0038            A(N) = A(N) + 2.0*DX*B2*A(N1)/B1
0039            A(N2) = A(N2) + A(N1)
0040            A(N1) = 0.
0041            GO TO 100
0042    25      IF (IDD4.NE.1) GO TO 27
0043            A(N2) = A(N2) + A(N1)
0044            A(N) = A(N) + DDX*FLW*A(N1)
0045            A(N1) = 0.
0046            GO TO 100
0047    27      AL = DSQRT(-A3/A1)
0048            A(N2) = A(N2) + A(N1)
0049            A(N) = A(N) - 2.0*DX*AL*A(N1)
0050            A(N1) = 0.
0051            GO TO 100
0052    30      IF (IDD3.EQ.1) GO TO 35
0053            IF (IDD3.EQ.2) GO TO 32
0054            B1 = -RR*WP*FW
0055            B2 = -FW*(RR*RL*F + RX*WP + RR*WX)
0056            B2 = B2 + RR*WP*(F*VXX(N) -2.0*WP*WX*REPS)
0057            B3 = RL*F*(-RX*FW + RR*(F*VXX(N) -2.*WP*WX*REPS))
0058            BX = 1. + DDD*B2/B1
0059            BB1 = -(-2. +B3*DXX/B1)/BX
0060            BB2 = -(1. -B2*DDD/B1)/BX
0061            A(N) = A(N) + A(N2)*BB1
```

```
0062            A(N1) = A(N1) + A(N2)*BB2
0063            A(N2) = 0.
0064            GO TO 100
0065    32      AL = DSQRT(-A3/A1)
0066            A(N1) = A(N1) + A(N2)
0067            A(N) = A(N) -2.0*DX*AL*A(N2)
0068            A(N2) = 0.
0069            GO TO 100
0070    35      A(N1) = A(N1) + A(N2)
0071            A(N) = A(N) -DDX*FLW*A(N2)
0072            A(N2) = 0.
0073    100     CONTINUE
0074            B(5) = 0.00001
0075            EQQ = 1.0E-36
0076            IKK = 0
0077            DO 105 I = 1,NN
0078            AQ = ABS(A(I))
0079            IF (AQ.GT.EQQ) GO TO 105
0080            IKK = 1
0081    105     CONTINUE
0082            IF (IKK.EQ.1) GO TO 115
0083            CALL TRI(NN)
0084            CALL CALC(NN,RI)
0085            GO TO 110
0086    115     WRITE(6,901)
0087    901     FORMAT(' NUMERICAL PROBLEM: SMALL DIAGONAL ELEMENT')
0088    110     RETURN
0089            END



0001
0002            SUBROUTINE CRLC(W,RL,NN,ICR)
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004    C
0005    C                   SUBROUTINE TO CHECK FOR CRITICAL LAYERS
0006    C
0007            ICR = 0
0008            DO 100 N = 1,NN
0009            WP = W + RL*V(N)
0010            IF (N.NE.1) GO TO 30
0011            PQ = WP
0012            GO TO 100
0013    30      PQ = WP/PQ
0014            IF (PQ.GT.0.) GO TO 40
0015            ICR = ICR +1
```

```
0016    40      PQ = WP
0017    100     CONTINUE
0018            RETURN
0019            END



0001
0002            SUBROUTINE TRI(N)
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004            DOUBLE PRECISION AA,BB
0005    C
0006    C
0007    C               SUBROUTINE TO SOLVE A TRIDIAGONAL MATRIX BY GAUSSIAN
0008    C                       ELIMINATION
0009    C
0010    C               STORE MAIN DIAGONAL FIRST
0011    C               0,LOWER DIAGONAL
0012    C               UPPER DIAGONAL,0
0013            N3 = N +1
0014            N2 = 2*N
0015            NN = N - 1
0016            DO 5 I = 1,NN
0017            II = N3 + I
0018            AA = A(II)/A(I)
0019            I2 = I + 1
0020            I3 =N2 + I
0021            BB = A(I2) -AA*A(I3)
0022            A(I2) = BB
0023            BB = B(I2) -AA*B(I)
0024    5       B(I2) = BB
0025            DO 10 IJ = 1,NN
0026            I = N3 - IJ
0027            B(I) = B(I)/A(I)
0028            II = I + N2 -1
0029            IP = I -1
0030            BB = B(IP) -A(II)*B(I)
0031    10      B(IP) = BB
0032            B(1) = B(1)/A(1)
0033            RETURN
0034            END



0001
0002            SUBROUTINE WFR(NN,XMAX,WF)
```

```
0003            COMMON RH(400),RHX(400),A(1200),B(400),V(400),VX(400),VXX(400)
0004            DIMENSION WF(400)
0005    C
0006    C                   SUBROUTINE TO READ AND INTERPOLATE FRICTIONAL WEIGHT
0007    C                           FUNCTIONS, WF
0008    C
0009            READ(5,*) NW
0010            IF (NW.EQ.0) GO TO 100
0011            DO 5 I = 1,NW
0012            II = 100 + I
0013            READ(5,*) A(I),A(II)
0014    5       A(I) = A(I)*1.0E+05
0015            DX = XMAX/FLOAT(NN-1)
0016            WF(1) = A(101)
0017            NMM = 100 + NW
0018            WFM = A(NMM)
0019            DO 50 N = 2,NN
0020            X = DX*FLOAT(N-1)
0021            IF (X.GT.A(NW)) GO TO 48
0022            IC = 0
0023            DO 46 J = 2,NW
0024            IF (IC.NE.0) GO TO 46
0025            I = J
0026            IF (X.GT.A(I)) GO TO 46
0027            IC = I
0028    46      CONTINUE
0029            II = 100 + I
0030            IIM = II - 1
0031            IM = I -1
0032            WX = (A(II) - A(IIM))/(A(I) -A(IM))
0033            XX =X -A(IM)
0034            WF(N) = A(IIM) + WX*XX
0035            GO TO 50
0036    48      WF(N) = WFM
0037    50      CONTINUE
0038            GO TO 200
0039    100     DO 150 I = 1,NN
0040    150     WF(I) = 1.0
0041    CCCCCCCCCC
0042    200     IF (NW.EQ.0) GO TO 250
0043            WRITE(6,901)
0044    CCCCCCCCCC
0045            WRITE(6,902) (WF(I),I=1,NN)
0046    CCCCCCCCCC
0047            GO TO 300
0048    250     WRITE(6,903)
```

```
0049    CCCCCCCCCC
0050    901     FORMAT(/' FRICTION WEIGHT FUNCTION')
0051    902     FORMAT(10E12.5)
0052    CCCCCCCCCC
0053    903     FORMAT(/' FRICTION WEIGHT FUNCTION = 1.0 EVERYWHERE')
0054    300     RETURN
0055    CCCCCCCCCC
0056            END
```

# 1 Program BIGLOAD2 Listing

```
0001            PROGRAM HYBD
0002            COMMON F,DX,DT,NN,MM,NM,NMX
0003            DOUBLE PRECISION AX,BX,XL
0004            COMMON AX(425,53),BX(425),XL(11475)
0005            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0006            DIMENSION WW(3), RII(3)
0007            DIMENSION R(25)
0008            DIMENSION BXB(425)
0009            DIMENSION WH(10), CH(10),RLH(10)
0010    C
0011    C                MAIN PROGRAM TO CALL ALL OF THE OTHER PIECES
0012    C
0013    C                DIMENSION(XL) = NM*(NN+2)
0014    C                DIM(AX) = NM,2NN+3
0015    C
0016    C                REVISED JUNE,1987
0017            READ(5,*) EPS,EST,DD1
0018            READ(5,*) ICCM,NCALM,NITM,ISD
0019            ICCC =1
0020    3       READ(5,*) F,XMAX
0021            XMAX = XMAX*1.0E+05
0022            F =F*1.0E-05
0023            WRITE(6,907) F,XMAX
0024            READ(5,*) BETA,ILWW
0025            IF (ILWW.EQ.1) GO TO 4
0026            IF (ILWW.NE.0) GO TO 140
0027            NCALM = 1
0028            WRITE(6,910)
0029            GO TO 5
0030    4       WRITE(6,911)
0031    5       DDLW = FLOAT(ILWW)
0032            WRITE(6,909) BETA
0033            READ(5,*) NCAL,WH(1)
0034            READ(5,*) IDIAG
0035            READ(5,*) RLF,DRL
0036            WH(1) = WH(1)*1.0E-06
0037            RLF = RLF*1.0E-07
0038            DRL = DRL*1.0E-07
0039            WRITE(6,905) EPS,EST,DD1
0040            MM = 17
0041            NN = 25
0042            NM = NN*MM
0043            NMX = 2*NN +3
0044            DX = XMAX/FLOAT(NN-1)
```

```
0045            DT = 1./FLOAT(MM-1)
0046            DO 50 I = 1,NCALM
0047    50      RLH(I) = RLF + DRL*FLOAT(I-1)
0048            RL = RLH(NCAL)
0049            READ(5,*) (WW(J),J=1,3)
0050            DO 6 J = 1,3
0051    6       WW(J) = WW(J)*1.0E-06
0052            CALL DEP
0053            CALL NSQ
0054    1       RIB = 0.
0055            I =1
0056    15      W = WW(I)
0057            CALL MATS(RL,W,DD1,RI,IER,BETA,DDLW)
0058            IF (IER.NE.0) GO TO 140
0059            RII(I) = RI
0060            IF (RI.LT.RIB) GO TO 18
0061            RIB = RI
0062            WB = W
0063            IB = I
0064            DO 10 IJ = 1,NM
0065    10      BXB(IJ) = BX(IJ)
0066    18      I = I+1
0067            IF (I.EQ.2) GO TO 15
0068            IF (I.EQ.4) GO TO 19
0069            IF (RII(1).LT.RII(2)) GO TO 15
0070            WW(3) = WW(1)*(1. -EST)
0071            GO TO 15
0072    19      NIT = 3
0073            IGP = 0
0074    20      CALL NGSW(WW,RII,WB,WN,ISUC,EPS,IN,IGP,EST,ISD)
0075            IF (WN.GT.F) GO TO 140
0076            IF (ISUC.EQ.2) GO TO 140
0077            IF (ISUC.EQ.1) GO TO 100
0078            CALL MATS(RL,WN,DD1,RI,IER,BETA,DDLW)
0079            IF (IER.NE.0) GO TO 140
0080            IF (IN.NE.0) GO TO 24
0081            IF (RI.GT.RIB) GO TO 21
0082            IF (WN.LT.WW(2)) GO TO 23
0083    22      IN = 3
0084            GO TO 24
0085    23      IN = 1
0086            GO TO 24
0087    21      IF (WN.LT.WW(2)) GO TO 22
0088            GO TO 23
0089    24      WW(IN) = WN
0090            RII(IN) = RI
```

```
0091            IF (RI.LT.RIB) GO TO 30
0092            WB = WN
0093            RIB = RI
0094            DO 25 I = 1,NM
0095    25      BXB(I) = BX(I)
0096    30      NIT = NIT +1
0097            IF(NIT.LT.NITM) GO TO 20
0098            WRITE(6,902) NIT
0099            GO TO 140
0100    100     RN = SQRT(RIB)
0101    CCCCCCCCCC
0102            RN = RN*BXB(MM)/ABS(BXB(MM))
0103    CCCCCCCCCC
0104            DO 110 I = 1,NM
0105    110     BX(I) = BXB(I)/RN
0106            CC = WB/RL
0107            WRITE(6,903) WB,RL,CC,EPS,NIT
0108            IF (NCAL.NE.1) GO TO 125
0109            CALL LGWH(WB,RL,DD1,R,BETA,DDLW)
0110            IF (IDIAG.EQ.0) GO TO 125
0111            CALL UCAL(WB,RL,BETA,DDLW)
0112    CCCCCCCCCC
0113            CALL RHOC(DD1)
0114    CCCCCCCCCC
0115    125     WRITE(6,908)
0116            DO 130 N = 1,NN
0117            X = DX*FLOAT(N-1)
0118            DZ = RH(N)*DT
0119            WRITE(6,904) X,RH(N),DZ
0120            ML = 1 +MM*(N-1)
0121            MH = MM*N
0122            WRITE(6,901) (BX(M),M=ML,MH)
0123    130     CONTINUE
0124            CALL DIAG(WB,RL,DDLW)
0125            WH(NCAL) = WB
0126            CH(NCAL) = CC
0127            NCAL = NCAL +1
0128            IF (NCAL.GT.NCALM) GO TO 140
0129            RL = RLH(NCAL)
0130            IF (NCAL.GE.3) GO TO 200
0131            WW(2) = CC*RL
0132            GO TO 205
0133    200     I1 = NCAL - 2
0134            I2 = NCAL - 1
0135            CG = (WH(I2) -WH(I1))/(RLH(I2) - RLH(I1))
0136            WW(2) = WH(I2) + CG*(RLH(NCAL) -RLH(I2))
```

```
0137    205     WW(1) = WW(2)*(1.0 - EST)
0138            WW(3) = WW(2)*(1.0 + EST)
0139            IF (WW(1).LT.0.) GO TO 140
0140            IF (WW(3).GT.F) GO TO 140
0141            GO TO 1
0142    140     ICCC = ICCC +1
0143            IF (ICCC.LE.ICCM) GO TO 3
0144    901     FORMAT(2X,10E12.5)
0145    902     FORMAT(//'  USED UP ',I5,'  ITERATIONS'//)
0146    903     FORMAT(///'   CONVERGED:  W,L,C,EPS,NIT =',4E15.5,I10//)
0147    904     FORMAT(/'  X,H,DZ =',3E15.5)
0148    905     FORMAT(/'  EPS,EST,DD1 =', 2E15.5,F10.2/)
0149    906     FORMAT(3F10.5)
0150    907     FORMAT(//'  F,XMAX =',2E15.5//)
0151    908     FORMAT(//'  PRESSURE'/)
0152    909     FORMAT(/' BETA= ',E15.5)
0153    910     FORMAT(' LONG WAVE LIMIT')
0154    911     FORMAT(' GENERAL FREQUENCY AND WAVENUMBER')
0155            STOP
0156            END
```

```
0001
0002            SUBROUTINE NGSW(WW,RII,WB,WN,ISUC,EPS,IN,IGP,EST,ISD)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007            DIMENSION  WW(3),RII(3)
0008    C
0009    C                SUBROUTINE TO COMPUTE THE NEXT GUESS AT W
0010    C
0011    C                ISD = 0  NORMAL
0012    C                ISD = 1  SEARCH DOWN
0013    C                ISD = -1 SEARCH UP
0014            DEL = EST
0015    5       IC = 0
0016            DO 10 I = 1,2
0017            I1 = 1 +I
0018            IF (WW(I1).GT.WW(I)) GO TO 10
0019            IC = 1
0020            RX = RII(I)
0021            WX = WW(I)
0022            WW(I) = WW(I1)
0023            WW(I1) = WX
```

```
0024            RII(I) = RII(I1)
0025            RII(I1) =RX
0026    10      CONTINUE
0027            IF (IC.NE.0) GO TO 5
0028            IF (RII(3).GT.RII(2)) GO TO 150
0029            IF (RII(1).GT.RII(2)) GO TO 160
0030            ISD = 0
0031            WB1 = (WW(1) +WW(2))/2.
0032            WB2 = (WW(2) + WW(3))/2.
0033            RW1 = (RII(2) - RII(1))/(WW(2) -WW(1))
0034            RW2 = ( RII(3) - RII(2))/(WW(3) - WW(2))
0035            A = (RW2 -RW1)/(WB2 -WB1)
0036            B = RW1 - A*WB1
0037            WN = -B/A
0038            IF (WN.LE.WW(1)) GO TO 50
0039            IF (WN.GT.WW(3)) GO TO 45
0040            GO TO 20
0041    50      WN = WB1
0042            GO TO 20
0043    45      WN = WB2
0044    20      EP = WN - WB
0045            EP = ABS(EP)
0046            EP = EP/WB
0047            IF (EP.LE.EPS) GO TO 100
0048            IGP = IGP +1
0049            ISUC = 0
0050            IN = 0
0051            GO TO 120
0052    100     ISUC = 1
0053            IF (IGP.NE.0) GO TO 120
0054            IGP = IGP +1
0055            ISUC = 0
0056            IF (RII(3).GT.RII(1)) GO TO 110
0057            WN = WB2
0058            GO TO 20
0059    110     WN = WB1
0060            GO TO 20
0061    150     IF (RII(1).GT.RII(2)) GO TO 180
0062    155     IF (ISD.EQ.1) GO TO 165
0063            WN = WW(3)*(1. + DEL)
0064            ISUC = 0
0065            IGP = 0
0066            IN = 1
0067            GO TO 120
0068    160     IF (RII(3).GT.RII(2)) GO TO 180
0069    165     IF (ISD.EQ.-1) GO TO 155
```

```
0070            WN = WW(1)*(1. - DEL)
0071            ISUC = 0
0072            IGP = 0
0073            IN = 3
0074            GO TO 120
0075    180     IF (RII(3).GT.RII(1)) GO TO 155
0076            GO TO 165
0077    120     IF (WN.LE.0.) GO TO 125
0078            GO TO 130
0079    125     ISUC = 2
0080    130     RETURN
0081            END
```

```
0001
0002            SUBROUTINE NSQ
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007    C
0008    C                SUBROUTINE TO READ AND INTERPOLATE BUOYANCY FREQUENCY
0009    C                        SQUARED PROFILE BV, AND Z DERIVATIVE BVZ
0010    C
0011            READ(5,*) NR,DZR,ALPH
0012            ALPH = ALPH*1.0E+05
0013            DZR = DZR*100.
0014            MX = MM -1
0015            READ(5,*) CMLT
0016            DO 20 I = 1,NR
0017            READ(5,*) XL(I)
0018    20      XL(I) = XL(I)*CMLT
0019            TF = XL(NR)
0020            ZZZ = DZR*FLOAT(NR-1)
0021            DZI = RH(NN) - ZZZ
0022            NL = NR +1
0023            NH = 600
0024            DO 30 I = NL,NH
0025            Z = DZR*FLOAT(I-1)
0026    30      XL(I) = TF*EXP((ZZZ -Z)/ALPH)
0027            DO 200 N = 1,NN
0028            DD = RH(N)
0029            DZ = DT*DD
0030            IF (DZ.GT.DZR) GO TO 110
```

```
0031            BV(N,MM) = XL(1)
0032            DO 50 M = 1,MX
0033            Z = DD - DZ*FLOAT(M-1)
0034            IBXL = 1 + IFIX(Z/DZR)
0035            IBXH = IBXL +1
0036            ZS = DZR*FLOAT(IBXL -1)
0037    50      BV(N,M) = XL(IBXL) +(Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
0038            Z = DD - DZ
0039            IBXL = 1 + IFIX(Z/DZR)
0040            IBXH = IBXL + 1
0041            ZS = DZR*FLOAT(IBXL-1)
0042            AQ = XL(IBXL) +(Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
0043            GO TO 145
0044    110     ZD = DZ/2.
0045            XBB = 0.
0046            NAVG = 0
0047            DO 120 I = 1,NR
0048            ZC = DZR*FLOAT(I-1)
0049            IF (ZC.GT.ZD) GO TO 120
0050            XBB = XBB + XL(I)
0051            NAVG = NAVG +1
0052    120     CONTINUE
0053            BV(N,MM) = XBB/FLOAT(NAVG)
0054            DO 140 MQ = 1,MM
0055            M = MQ -1
0056            Z = DD - DZ*FLOAT(M-1)
0057            ZS = Z - DZ/2.
0058            ZD = ZS + DZ
0059    125     XBB = 0.
0060            NAVG = 0
0061            DO 130 I = 1,NH
0062            ZC = DZR*FLOAT(I-1)
0063            IF ( ZC.LT.ZS) GO TO 130
0064            IF (ZC.GT.ZD) GO TO 130
0065            XBB = XBB + XL(I)
0066            NAVG = NAVG + 1
0067    130     CONTINUE
0068            IF (NAVG.NE.0) GO TO 135
0069            IBXL = 1 + IFIX(Z/DZR)
0070            IBXH = IBXL + 1
0071            ZSS = DZR*FLOAT(IBXL-1)
0072            IF (M.EQ.0) GO TO 133
0073            BV(N,M) = XL(IBXL) +(Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
0074            GO TO 140
0075    133     AQ = XL(IBXL) +(Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
0076            GO TO 140
```

```
0077    135     IF (M.EQ.0) GO TO 138
0078            BV(N,M) = XBB/FLOAT(NAVG)
0079            GO TO 140
0080    138     AQ = XBB/FLOAT(NAVG)
0081    140     CONTINUE
0082    145     DO 150 M = 2,MX
0083            IP = M +1
0084            IM = M -1
0085    150     BVZ(N,M) = (BV(N,IP) - BV(N,IM))/(2.*DZ)
0086            BVZ(N,1) = (BV(N,2) -AQ)/(2.0*DZ)
0087            BVZ(N,MM) = (BV(N,MM) - BV(N,MX))/DZ
0088    200     CONTINUE
0089            WRITE(6,901)
0090            WRITE(6,902) (BV(NN,J),J=1,MM)
0091    901     FORMAT(//'   NSQUARED AT XMAX '/)
0092    902     FORMAT(2X,10E12.5)
0093    250     RETURN
0094            END
```

```
0001
0002            SUBROUTINE MATS(RL,W,DD1,RI,IER,BETA,DDLW)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007    C
0008    C               SUBROUTINE TO SET UP AND SOLVE THE PRESSURE EQUATION
0009    C
0010    C               DD1 = 0  RIGID LID
0011    C               DD1 = 1  FREE SURFACE
0012            GG = 1./980.
0013            DXDT = DX/(2.*DT)
0014            DXX = DX*DX
0015            DDD = DXX/(DT*DT)
0016            RLL = RL*RL*DDLW
0017            DDX = DX*DXDT
0018            DQ = 2.0*DX
0019            FLW = F*RL/W
0020            BLW=BETA*RL/W
0021            CCC1 = W*(F*F - DDLW*W*W)/DXX
0022            CCC2 = 0.5*F*(2.0*BETA*W + RL*(F*F - DDLW*W*W))/DX
0023            CCC3 = BETA*RL*(F*F + DDLW*W*W)
0024            AA5 = 2.0*F*BETA/(F*F -W*W*DDLW)
0025            RLQ = DDLW*RLL - BLW*(F*F + DDLW*W*W)/(F*F -DDLW*W*W)
```

```
0026            DO 5 I = 1,NM
0027    5       BX(I) = 0.
0028            DO 10 J = 1,NMX
0029            DO 10 I = 1,NM
0030    10      AX(I,J) = 0.
0031            BX(39) = 1.
0032            J = NN + 2
0033            IP1 = J + 1
0034            IM1 = J - 1
0035            IPM1 = 2*NN +1
0036            IPM = 2*NN + 2
0037            IPMM = 2*NN + 3
0038            DO 100 M = 1,MM
0039            DO 90 N = 1,NN
0040            I = N + NN*(M-1)
0041            CALL AS(A1,A2,A3,W,N,M,C1,C2,C3,DDLW)
0042            AX(I,1) = A2*DXDT
0043            AX(I,IM1) = 1.-AA5*DX/2.0
0044            AX(I,IPM1) = -A2*DXDT
0045            AX(I,2) = A1*DDD - DDX*(A3+A2*AA5)
0046            AX(I,J) = -2. -2.*DDD*A1 - RLQ*DXX
0047            AX(I,IPM) = A1*DDD + DDX*(A3+A2*AA5)
0048            AX(I,3) = -A2*DXDT
0049            AX(I,IP1) = 1.+AA5*DX/2.0
0050            AX(I,IPMM) = A2*DXDT
0051            IF ( N.EQ.1) GO TO 20
0052            IF (N.EQ.NN) GO TO 30
0053            IF (M.EQ.1) GO TO 40
0054            IF (M.EQ.MM) GO TO 50
0055            GO TO 90
0056    20      AX(I,1) = 0.
0057    C                X = 0 BOUNDARY
0058            AX(I,IM1) = 0.
0059            AX(I,IPM1) = 0.
0060            AX(I,3) = 0.
0061            AX(I,IPMM) = 0.
0062            AX(I,IP1) = 2.
0063            TH = -1. + DT*FLOAT(M-1)
0064            B1 = -TH*RHX(N)/RH(N)
0065            AX(I,2) = AX(I,2) -A2*B1*DDD + A2*FLW*2.0*DDX - B1*2.0*DXDT
0066            AX(I,J) = AX(I,J) + 2.*DX*FLW + 2.*A2*B1*DDD
0067            AX(I,IPM) =AX(I,IPM)-A2*B1*DDD-A2*FLW*2.0*DDX + B1*2.0*DXDT
0068            IF (M.NE.MM) GO TO 25
0069            D5 = DD1*RH(N)*BV(N,M)*GG
0070            AX(I,2) = AX(I,2) + AX(I,IPM)
0071            AX(I,J) = AX(I,J) - 2.0*DT*D5*AX(I,IPM)
```

```
0072            AX(I,IPM) = 0.
0073            GO TO 90
0074    25      IF (M.NE.1) GO TO 90
0075            AX(I,IPM) = AX(I,IPM) + AX(I,2)
0076            AX(I,2) = 0.
0077            GO TO 90
0078    30      DZ = DT*RH(N)
0079    C                X = XMAX BOUNDARY
0080            AX(I,3) = 0.
0081            AX(I,IPMM) = 0.
0082            AX(I,1) = 0.
0083            AX(I,3) = 0.
0084            AX(I,J) = AX(I,J) +AX(I,IP1)*(2.0*CCC1-CCC3)/(CCC1+CCC2)
0085            AX(I,IM1) = AX(I,IM1) +AX(I,IP1)*(CCC2-CCC1)/(CCC1+CCC2)
0086            AX(I,IP1) = 0.
0087            IF (M.NE.MM) GO TO 35
0088            AX(I,2) = AX(I,2) +AX(I,IPM)
0089            AX(I,J) = AX(I,J) -2.0*DZ*DD1*GG*BV(N,M)*AX(I,IPM)
0090            AX(I,IPM) = 0.
0091            GO TO 90
0092    35      IF (M.NE.1) GO TO 90
0093            AX(I,IPM) = AX(I,IPM) + AX(I,2)
0094            AX(I,2) = 0.
0095            GO TO 90
0096    40      D1 = 2.*C2*DT/(C1 +C2*C3)
0097    C                Z = -H BOUNDARY
0098            AX(I,IPM) = AX(I,IPM) + AX(I,2)
0099            AX(I,IP1) = AX(I,IP1) + D1*AX(I,2)/DQ
0100            AX(I,IM1) = AX(I,IM1) -D1*AX(I,2)/DQ
0101            AX(I,J) = AX(I,J) +D1*FLW*AX(I,2)
0102            NNN = N +1
0103            IJJP = J + 2
0104            CALL AS(A1,A2,A3,W,NNN,M,C1,C2,C3,DDLW)
0105            D1 = 2.0*C2*DT/(C1 + C2*C3)
0106            AX(I,IPMM) = AX(I,IPMM) + AX(I,3)
0107            AX(I,IJJP) = AX(I,IJJP) + D1*AX(I,3)/DQ
0108            AX(I,J) = AX(I,J) -D1*AX(I,3)/DQ
0109            AX(I,IP1) = AX(I,IP1) + D1*FLW*AX(I,3)
0110            IF (N.EQ.2) GO TO 45
0111            IJJM = J - 2
0112            NNN = N - 1
0113            CALL AS(A1,A2,A3,W,NNN,M,C1,C2,C3,DDLW)
0114            D1 = 2.0*C2*DT/(C1 + C2*C3)
0115            AX(I,IPM1) = AX(I,IPM1) + AX(I,1)
0116            AX(I,J) = AX(I,J) + D1*AX(I,1)/DQ
0117            AX(I,IJJM) = AX(I,IJJM) -D1*AX(I,1)/DQ
```

```
0118            AX(I,IM1) = AX(I,IM1) + D1*FLW*AX(I,1)
0119            GO TO 48
0120    45      AX(I,IPM1) = AX(I,IPM1) + AX(I,1)
0121    48      AX(I,1) = 0.
0122            AX(I,2) = 0.
0123            AX(I,3) = 0.
0124            GO TO 90
0125    50      AX(I,1) = 0.
0126    C                Z = 0 BOUNDARY
0127            AX(I,3) = 0.
0128            D5 = RH(N)*DD1*BV(N,M)*GG
0129            AX(I,2) = AX(I,2) + AX(I,IPM)
0130            AX(I,J) = AX(I,J) - D5*2.0*DT*AX(I,IPM)
0131            AX(I,IM1) = AX(I,IM1) -D5*2.0*DT*AX(I,IPM1)
0132            AX(I,IP1) = AX(I,IP1) -D5*DT*2.0*AX(I,IPMM)
0133            AX(I,IPM1) = 0.0
0134            AX(I,IPMM) = 0.0
0135            AX(I,IPM) = 0.0
0136    90      CONTINUE
0137    100     CONTINUE
0138            NDD = NN +1
0139            NNS = NM
0140            CALL LEQT1B(AX,NM,NDD,NDD,NM,BX,1,NM,0,XL,IER)
0141    C                LEQT1B IS AN IMSL ROUTINE
0142            CALL CALI(RI)
0143            CC = W/RL
0144            WRITE(6,901) W,RL,CC,RI,IER
0145    901     FORMAT(/'  W,L,C,RI,IER =',4E15.5,I10)
0146            RETURN
0147            END



0001
0002            SUBROUTINE AS(A1,A2,A3,W,N,M,C1,C2,C3,DDLW)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007    C
0008    C                SUBROUTINE TO COMPUTE COEFFICIENTS FOR MATRIX
0009    C
0010            F2 = F*F
0011            W2 = W*W*DDLW
0012            TZ = 1./RH(N)
0013            TH = -1. +DT*FLOAT(M-1)
```

```
0014            TX = -RHX(N)*TH*TZ
0015            AA = (RHX(N)/RH(N))**2
0016            TXX = (2.*AA -RHXX(N)/RH(N))*TH
0017            BW = BV(N,M)
0018            BWW = BW*BW
0019            A1 = TX*TX + TZ*TZ*(F2-W2)/BW
0020            A2 = TX
0021            A3 = TXX -TZ*BVZ(N,M)*(F2-W2)/BWW
0022            C1 = TZ
0023            C2 = RHX(N)*BV(N,1)/(F2-W2)
0024            C3 = C1*RHX(N)
0025            RETURN
0026            END



0001
0002            SUBROUTINE DEP
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007    C
0008    C               SUBROUTINE TO READ AND INTERPOLATE DEPTH PROFILE
0009    C                       RH = DEPTH
0010    C                       RHX = X DERIVATIVE OF DEPTH
0011    C                       RHXX = SECOND X DERIVATIVE OF DEPTH
0012    C
0013            READ(5,*) NRX
0014            DO 5 I =1,NRX
0015            READ(5,*) XL(I),BX(I)
0016            BX(I) = BX(I)*100.
0017    5       XL(I) = XL(I)*1.0E+05
0018            RHMA = BX(NRX)
0019            RH(1) = BX(1)
0020            DO 20 N = 2,NN
0021            X = DX*FLOAT(N-1)
0022            IF (X.GT.XL(NRX)) GO TO 15
0023            IC = 0
0024            DO 8 J = 2,NRX
0025            IF (IC.NE.0) GO TO 8
0026            I = J
0027            IF (X.GT.XL(I)) GO TO 8
0028            IC = I
0029    8       CONTINUE
0030            IM = I-1
```

```
0031            RHX(N) = (BX(I) -BX(IM))/(XL(I)-XL(IM))
0032            XX = X - XL(IM)
0033            RH(N) = BX(IM) + RHX(N)*XX
0034            GO TO 20
0035    15      RH(N) = RHMA
0036    20      CONTINUE
0037            RHX(1) = (RH(2) - RH(1))/DX
0038            RHXX(1) = 0.
0039            D2 = 2.*DX
0040            DXX = DX*DX
0041            NX = NN -1
0042            DO 30 N = 2,NX
0043            IP = N +1
0044            IM = N -1
0045            RHX(N) =(RH(IP) - RH(IM))/D2
0046    30      RHXX(N) = (RH(IP) -2.*RH(N) + RH(IM))/DXX
0047            RHX(NN) = 0.
0048            RHXX(NN) = 0.
0049            RETURN
0050            END
```

```
0001
0002            SUBROUTINE CALI(RIX)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007            DOUBLE PRECISION RI,RZ
0008    C
0009    C                SUBROUTINE TO CALCULATE THE INTEGRAL OF RESPONSE SQUARED
0010    C
0011    C                FIRST, REORDER THE MATRIX
0012            DO 2 M = 1,MM
0013            DO 2 N = 1,NN
0014            IN = N + NN*(M -1)
0015            II = M + MM*(N-1)
0016    2       XL(II) = BX(IN)
0017            DO 3 I = 1,NM
0018    3       BX(I) = XL(I)
0019            RI = 0.
0020            RR = 0.
0021            RZ = 0.
0022            DO 50 N = 1,NN
0023            IF (N.EQ.1) GO TO 5
```

```
0024            IF (N.NE.NN) GO TO 10
0025     5      DXX = DX/2.
0026            GO TO 15
0027     10     DXX = DX
0028     15     DO 50 M = 1,MM
0029            DZ = DT*RH(N)
0030            IF (M.EQ.1) GO TO 25
0031            IF (M.NE.MM) GO TO 30
0032     25     DZZ = DZ/2.
0033            GO TO 35
0034     30     DZZ = DZ
0035     35     I = (N-1)*MM +M
0036            RZ = RZ + DZZ*DXX
0037            RI = RI + DZZ*DXX*BX(I)**2
0038     50     CONTINUE
0039            RI = RI/RZ
0040            RIX = RI
0041            RETURN
0042            END
```

```
0001
0002            SUBROUTINE DIAG(W,RL,DDLW)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007     C
0008     C                    SUBROUTINE TO COMPUTE MOMENTUM DIAGNOSTICS
0009     C
0010            N = 5
0011            I = N*MM
0012            IP = I + MM
0013            IM = I -MM
0014            PX = (BX(IP) -BX(IM))/(2.*DX)
0015            F2 = F*F - W*W*DDLW
0016            VT = W*(F*PX + DDLW*RL*W*BX(I))/F2
0017            FU = -F*(RL*F*BX(I) + W*PX)/F2
0018            PY = RL*BX(I)
0019            X = DX*FLOAT(N-1)
0020            WRITE(6,901) X
0021            WRITE(6,902) VT,FU,PY
0022     901    FORMAT(//'  Y MOMENTUM TERMS AT Z = 0, X =',E15.5)
0023     902    FORMAT(/'  VT,FU,PY =',3E15.5/)
0024            RETURN
0025            END
```

```
0001
0002            SUBROUTINE LGWH(W,RL,DD1,R,BETA,DDLW)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007            DOUBLE PRECISION RD,RP,RI,XINT,XP
0008            DIMENSION R(25)
0009    C
0010    C                   SUBROUTINE TO COMPUTE LONG WAVE PARAMETERS BN,ANN
0011    C
0012            XINT = 0.
0013            RHO = 1.03
0014            F2 = F*F
0015            MX = MM - 1
0016    C                   NORMALIZE
0017            NX = NN -1
0018            RI = (BX(1)**2 + BX(MM)**2)/2.
0019            RD = (RHX(1)*BX(1)**2)/2.
0020            DO 10 I = 2,MX
0021    10      RI = RI + BX(I)**2
0022            DO 20 I = 2,NX
0023            IJ = MM*(I-1) +1
0024    20      RD = RD + RHX(I)*BX(IJ)**2
0025            RD = RD *DX
0026            RI = RI*RH(1)*DT
0027            RD = RD + RI
0028            RD = DSQRT(RD*RD)
0029            RD = DSQRT(RD)
0030            DO 30 I = 1,NM
0031    30      BX(I) = BX(I)/RD
0032            CALL VCAL(W,RL,BETA,DDLW)
0033    C                   COMPUTE WIND COUPLING
0034    40      BN = BX(MM)
0035    C                   NOW GET BOTTOM FRICTION
0036    C                   READ R(X)
0037            READ(5,*) NR
0038            DO 45 I = 1,NR
0039            READ(5,*) AX(I,1),AX(I,2)
0040    45      AX(I,1) = AX(I,1)*1.0E+05
0041            R(1) = AX(1,2)
0042            RMA = AX(NR,2)
0043            DO 50 N = 2,NN
```

```
0044            X = DX*FLOAT(N-1)
0045            IF (X.GT.AX(NR,1)) GO TO 48
0046            IC = 0
0047            DO 46 J = 2,NR
0048            I = J
0049            IF (X.GT.AX(I,1)) GO TO 46
0050            IC = I
0051    46      CONTINUE
0052            IM = I -1
0053            RX = (AX(I,2) -AX(IM,2))/(AX(I,1)-AX(IM,1))
0054            XX = X - AX(IM,1)
0055            R(N) = AX(IM,2) + RX*XX
0056            GO TO 50
0057    48      R(N) = RMA
0058    50      CONTINUE
0059    C                 X=0 CONTRIBUTION
0060            RI = 0.5*R(1)*(RL*F*BX(1)/W)**2
0061    C                 X = L CONTRIBUTION
0062            I = 1 + MM*(NN-1)
0063            BXP = F*XL(I)
0064            RI = RI + 0.5*R(NN)*(BXP)**2
0065    C                 INTERMEDIATE X
0066            DO 150 N = 2,NX
0067            I = 1 + MM*(N-1)
0068            IP = I + MM
0069            IM = I - MM
0070            RX = 0.5*(BX(IP)-BX(IM))/DX
0071    150     RI = RI + R(N)*RX*F*XL(I)
0072            ANN = RI*DX/F
0073    C
0074            WRITE(6,902) BN,ANN
0075            WRITE(6,901)
0076            WRITE(6,905) (R(I),I=1,NN)
0077            CC = 0.5/(980.*RHO)
0078            DO 220 N = 1,NN
0079            XX = DX
0080            IF (N.EQ.1) GO TO 210
0081            IF (N.NE.NN) GO TO 215
0082    210     XX = DX/2.
0083    215     I = N*MM
0084    220     XINT = XINT + CC*XX*BX(I)*BX(I)
0085            WRITE(6,904) XINT
0086    901     FORMAT(/' R =')
0087    902     FORMAT('   BN,ANN =',2E15.5///)
0088    903     FORMAT(2F10.5)
0089    904     FORMAT(//'  FREE SURFACE CONTRIBUTION TO PE =',E15.5/)
```

```
0090    905     FORMAT(10E12.5)
0091    500     RETURN
0092            END


0001
0002            SUBROUTINE VCAL(W,RL,BETA,DDLW)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007            DOUBLE PRECISION XINT
0008    C
0009    C                SUBROUTINE TO COMPUTE THE V FIELD OF THE WAVE
0010    C
0011            RHO = 0.515
0012            XINT = 0.
0013            WRITE(6,905)
0014            FW = F*F - DDLW*W*W
0015            WL = DDLW*W*RL
0016            RLW = RL/W
0017            FLW = F*RL/W
0018            MX = MM -1
0019            DXX = 2.*DX
0020            W2 = W*W*DDLW
0021            DO 50 N = 1,NN
0022            XX = DX
0023            X = DX*FLOAT(N-1)
0024            D = RH(N)
0025            DD = RHX(N)/D
0026            DZ = DT*D
0027            RHZ = RHX(N)**2
0028            IF (N.EQ.1) GO TO 10
0029            IF (N.NE.NN) GO TO 20
0030            GO TO 18
0031    10      DO 15 M = 1,MM
0032            I = (N-1)*MM +M
0033    15      XL(I) = -RLW*BX(I)
0034            XX = DX/2.
0035            GO TO 40
0036    18      CC1 = W*FW/(DX*DX)
0037            CC2 = 0.5*(2.0*F*BETA*W +RL*F*FW)/DX
0038            CC3 = BETA*RL*(F*F +DDLW*W*W)
0039            DQ = 1./(F*DX)
0040            DO 19 M = 1,MM
0041            I = (N-1)*MM +M
```

```
0042            IM = I -MM
0043            XP = BX(I)*(2.0*CC1 -CC3)/(CC1+CC2)
0044            XP = XP + BX(IM)*(CC2-CC1)/(CC1+CC2)
0045            XPX = (XP - BX(IM))/(2.0*DX)
0046    19      XL(I) = (F*XPX + WL*BX(I))/FW
0047            XX = DX/2.
0048            GO TO 40
0049    20      I = (N-1)*MM +1
0050            IP = I + MM
0051            IM = I - MM
0052            CALL AS(A1,A2,A3,W,N,1,C1,C2,C3,DDLW)
0053            CQ = C2*C3/(C1+ C2*C3)
0054            XPX = (1.0 -CQ)*(BX(IP) -BX(IM))/DXX
0055            XPX = XPX -CQ*FLW*BX(I)
0056            XL(I) = (WL*BX(I) + F*XPX)/FW
0057            I = N*MM
0058            IP = I + MM
0059            IM = I -MM
0060            XPX = (BX(IP) - BX(IM))/DXX
0061            XL(I) =(WL*BX(I) + F*XPX)/FW
0062            DO 30 M = 2,MX
0063            I = (N-1)*MM +M
0064            IP = I + MM
0065            IM = I -MM
0066            I2 = I +1
0067            I1 = I -1
0068            XPX = (BX(IP) - BX(IM))/DXX
0069            T = -1. +DT*FLOAT(M-1)
0070            XPT = (BX(I2) -BX(I1))/(2.*DT)
0071            XPX = XPX - T*DD*XPT
0072    30      XL(I) = (WL*BX(I) + F*XPX)/FW
0073    40      DO 45 M = 1,MM
0074            I = (N-1)*MM + M
0075            ZZ = DZ
0076            IF (M.EQ.1) GO TO 42
0077            IF (M.NE.MM) GO TO 45
0078    42      ZZ = DZ/2.
0079    45      XINT = XINT + RHO*XL(I)*XL(I)*XX*ZZ
0080            WRITE(6,904) X,D,DZ
0081            IL = (N-1)*MM +1
0082            IH = N*MM
0083            WRITE(6,901) (XL(IJ),IJ=IL,IH)
0084    50      CONTINUE
0085            WRITE(6,902) XINT
0086    901     FORMAT(2X,10E12.5)
0087    902     FORMAT(//'  V CONTRIBUTION TO KE =',E15.5/)
```

```
0088    905     FORMAT(////'  V'//)
0089    904     FORMAT(/'    X, H, DZ =',3E15.5)
0090            RETURN
0091            END


0001
0002    CCCCCCCCCC
0003            SUBROUTINE RHOC(DD1)
0004    CCCCCCCCCC
0005            COMMON F,DX,DT,NN,MM,NM,NMX
0006            DOUBLE PRECISION AX,BX,XL
0007            COMMON AX(425,53),BX(425),XL(11475)
0008            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0009            DOUBLE PRECISION XINT
0010    C
0011    C                SUBROUTINE TO COMPUTE THE DENSITY FIELD OF THE WAVE
0012    C
0013            G2 = 980.*980./2.06
0014            XINT = 0.
0015            WRITE(6,903)
0016            G = 980.
0017            DO 50 N = 1,NN
0018            DXX = DX
0019            IF (N.EQ.1) GO TO 2
0020            IF (N.NE.NN) GO TO 5
0021    2       DXX = DX/2.
0022    5       X = DX*FLOAT(N-1)
0023            DZ = DT*RH(N)
0024            GDZ = G*DZ
0025            DO 40 M = 1,MM
0026            DZZ = DZ
0027            I = (N-1)*MM +M
0028            IF (M.EQ.1) GO TO 10
0029            IF (M.NE.MM) GO TO 20
0030            DZZ = DZ/2.
0031    CCCCCCCCCC
0032            XL(M) = BV(N,M)*BX(I)*DD1/(G*G)
0033    CCCCCCCCCC
0034            GO TO 35
0035    10      IP = I + 1
0036            DZZ = DZ/2.
0037            XL(M) = -(BX(IP) - BX(I))/GDZ
0038            GO TO 35
0039    20      IP = I +1
0040            IM = I -1
```

```
0041            XL(M) = -(BX(IP) -BX(IM))/(2.*GDZ)
0042    35      XX = G2*XL(M)*XL(M)/BV(N,M)
0043            XINT = XINT + XX*DZZ*DXX
0044    40      CONTINUE
0045            WRITE(6,901) X,RH(N),DZ
0046    50      WRITE(6,902) (XL(M),M=1,MM)
0047            WRITE(6,904) XINT
0048    901     FORMAT(/'  X,D,DZ =',3E15.5)
0049    902     FORMAT(2X,10E12.5)
0050    903     FORMAT(///'  RHO'/)
0051    904     FORMAT(//'  RHO CONTRIBUTION TO PE =',E15.5/)
0052            RETURN
0053            END


0001
0002            SUBROUTINE UCAL(W,RL,BETA,DDLW)
0003            COMMON F,DX,DT,NN,MM,NM,NMX
0004            DOUBLE PRECISION AX,BX,XL
0005            COMMON AX(425,53),BX(425),XL(11475)
0006            COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0007            DOUBLE PRECISION XINT
0008    C
0009    C                   SUBROUTINE TO COMPUTE THE U FIELD OF THE WAVE
0010    C
0011            RHO = 0.515
0012            XINT = 0.
0013            FW = F*F - DDLW*W*W
0014            FL = F*RL
0015            WRITE(6,903)
0016            DO 100 N = 1,NN
0017            DXX = DX
0018            X = DX*FLOAT(N-1)
0019            DZ = DT*RH(N)
0020            DO 5 I = 1,MM
0021    5       XL(I) = 0.
0022            IF (N.EQ.1) GO TO 90
0023            IF (N.EQ.NN) GO TO 110
0024            DD = RHX(N)/RH(N)
0025            DO 85 M = 1,MM
0026            I = (N-1)*MM +M
0027            IP = I + MM
0028            IM = I -MM
0029            T = -1. + DT*FLOAT(M-1)
0030            IF (M.EQ.1) GO TO 10
0031            IF (M.EQ.MM) GO TO 15
```

```
0032            GO TO 20
0033    10      I1 = I +1
0034            XPT = (BX(I1) - BX(I))/DT
0035            GO TO 25
0036    15      XPT = 0.
0037            GO TO 25
0038    20      I2 = I+1
0039            I1 = I-1
0040            XPT = (BX(I2) - BX(I1))/(2.*DT)
0041    25      XPX = (BX(IP) - BX(IM))/(2.*DX)
0042            XPX = XPX -T*DD*XPT
0043    85      XL(M) = -(W*XPX + FL*BX(I))/FW
0044            GO TO 90
0045    110     CC1 = W*FW/(DX*DX)
0046            CC2 = 0.5*(2.0*F*BETA*W +RL*F*FW)/DX
0047            CC3 = BETA*RL*(F*F +DDLW*W*W)
0048            DXX = DX/2.
0049            DO 120 M = 1,MM
0050            I = (NN-1)*MM +M
0051            IM = I - MM
0052            XPIM = BX(I)*(2.0*CC1-CC3)/(CC1+CC2)
0053            XPIM = XPIM + BX(IM)*(CC2-CC1)/(CC1+CC2)
0054    120     XL(M) =-(FL*BX(I) + W*0.5*(XPIM - BX(IM))/DX)/FW
0055    90      DO 95 M = 1,MM
0056            DZZ = DZ
0057            IF (M.EQ.1) GO TO 92
0058            IF (M.NE.MM) GO TO 95
0059    92      DZZ = DZ/2.
0060    95      XINT = XINT + RHO*XL(M)*XL(M)*DXX*DZZ
0061            WRITE(6,901) X,RH(N),DZ
0062            WRITE(6,902) (XL(M),M=1,MM)
0063    100     CONTINUE
0064            WRITE(6,904) XINT
0065    901     FORMAT(/'  X,D,DZ =',3E15.5)
0066    902     FORMAT(2X,10E12.5)
0067    903     FORMAT(///'   U'/)
0068    904     FORMAT(//'  U CONTRIBUTION TO KE =',E15.5/)
0069            RETURN
0070            END
```

# 1 Program CROSS Listing

```
0001          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0002          EXTERNAL FNA
0003          COMMON S(101),P(101),G(101),T(101),X(101),VS(101),NV,DZ,SURF
0004          DIMENSION RH(25)
0005    C
0006    C             MAIN PROGRAM TO CALL THE OTHER PIECES
0007    C
0008    C                 REVISED APRIL,1985
0009    C
0010          READ(5,*)NV,DD1
0011          READ(5,*)F,XMAX
0012          XMAX=XMAX*1.E5
0013          F=F*1.E-5
0014          IF(DD1.EQ.0.)WRITE(6,908)
0015          IF(DD1.EQ.1.)WRITE(6,909)
0016          WRITE(6,907)F,XMAX
0017          NN=25
0018          DX=XMAX/(NN-1)
0019          CALL DEP(DX,RH,NN)
0020          DZ=RH(NN)/(NV-1)
0021          K1=1
0022          X(1)=0.0
0023          X1=0.0
0024          X2=0.0
0025          H1=0.0
0026          H2=RH(1)
0027          DO 100 I=2,NV
0028          Z=FLOAT(I-1)*DZ
0029          IF(Z.GE.RH(NN))Z=RH(NN)
0030      160 IF(Z.LE.H2)GO TO 150
0031          K1=K1+1
0032          H1=RH(K1-1)
0033          H2=RH(K1)
0034          X1=DX*FLOAT(K1-2)
0035          X2=DX*FLOAT(K1-1)
0036          GO TO 160
0037      150 DH=H2-H1
0038          SX=(X2-X1)/DH
0039          BX=-(X2*H1-X1*H2)/DH
0040          X(I)=SX*Z+BX
0041      100 CONTINUE
0042          CALL NSQ(RH(NN),F,DZ,VS,NV)
0043          SURF=DD1*VS(1)*F*F/981.
```

```
0044          READ(5,*)NRS
0045          WRITE(6,901)DZ
0046          WRITE(6,902)(X(I),I=1,NV)
0047          WRITE(6,903)
0048          WRITE(6,902)(VS(I),I=1,NV)
0049          DO 200 IR=1,NRS
0050          READ(5,*)XMIN,XMAX,XINC
0051          XMIN=XMIN*1.E-7
0052          XMAX=XMAX*1.E-7
0053          XINC=XINC*1.E-7
0054          CALL ROOT(XMIN,XMAX,XINC,FNA,X3,IROOT)
0055          IF(IROOT.EQ.0)GO TO 200
0056          DO 300 I=2,NV
0057      300 P(I)=P(I)*DEXP(-X3*(X(I)+X(I-1))*.5)
0058          PSUM=0.0
0059          DO 350 I=2,NV
0060      350 PSUM=PSUM+P(I)**2
0061          PSUM=DSQRT(DZ*PSUM)
0062          DO 360 I=2,NV
0063      360 P(I)=P(I)/PSUM
0064          CF=F/X3
0065          WRITE(6,904)X3,CF
0066          WRITE(6,905)
0067          WRITE(6,902)(P(I),I=2,NV)
0068      200 CONTINUE
0069      901 FORMAT(//'  INVERSE TOPOGRAPHY, DZ=',D15.5/)
0070      902 FORMAT(1X,10D12.5)
0071      903 FORMAT(//'  NSQUARED/FSQUARED'/)
0072      904 FORMAT(////'  DISPERSION CURVE CROSSES W=F AT L=',D15.5,5X,
0073         1  'PHASE SPEED=',D15.5/)
0074      905 FORMAT(/'  PRESSURE FIELD (SURFACE TO BOTTOM)'/)
0075      907 FORMAT(//'  F,XMAX=',2D15.5)
0076      908 FORMAT(//'   RIGID LID'/)
0077      909 FORMAT(//'   FREE SURFACE'/)
0078          STOP
0079          END
```

```
0001          SUBROUTINE DEP(DX,RH,NN)
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          DIMENSION XL(101),BX(101),RH(NN)
0004      C
0005      C          SUBROUTINE TO READ AND INTERPOLATE DEPTH PROFILE
0006      C                  RH= DEPTH
0007      C
```

```
0008          READ(5,*)NRX
0009          DO 5 I=1,NRX
0010          READ(5,*)XL(I),BX(I)
0011          BX(I)=BX(I)*100.
0012        5 XL(I)=XL(I)*1.E5
0013          IF(NRX.EQ.1)GO TO 7
0014          DO 6 I=2,NRX
0015        6 IF(BX(I).LE.BX(I-1))GO TO 100
0016        7 RHMA=BX(NRX)
0017          XMA=XL(NRX)
0018          RH(1)=BX(1)
0019          DO 20 N=2,25
0020          X=DX*FLOAT(N-1)
0021          IF(X.GT.XMA)GO TO 15
0022          IC=0
0023          DO 8 J=2,NRX
0024          IF(IC.NE.0)GO TO 8
0025          I=J
0026          IF(X.GT.XL(I))GO TO 8
0027          IC=I
0028        8 CONTINUE
0029          IM=I-1
0030          RHX=(BX(I)-BX(IM))/(XL(I)-XL(IM))
0031          XX=X-XL(IM)
0032          RH(N)=BX(IM)+RHX*XX
0033          GO TO 20
0034       15 RH(N)=RHMA
0035       20 CONTINUE
0036          GO TO 200
0037      100 WRITE(6,901)
0038      901 FORMAT(/'  TOPOGRAPHY IS NOT MONOTONIC'//)
0039      200 RETURN
0040          END
```

```
0001          SUBROUTINE NSQ(RHMA,F,DZ,VS,NV)
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          DIMENSION XL(1000),VS(NV)
0004    C
0005    C           SUBROUTINE TO READ AND INTERPOLATE BUOYANCY FREQUENCY
0006    C
0007          FSQ=F*F
0008          READ(5,*)NR,DZR,ALPH
0009          ALPH=ALPH*1.E5
0010          DZR=DZR*100.
```

```
0011          READ(5,*)CMLT
0012          DO 20 I=1,NR
0013          READ(5,*)XL(I)
0014       20 XL(I)=XL(I)*CMLT
0015          TF=XL(NR)
0016          ZZZ=DZR*FLOAT(NR-1)
0017          I=NR+1
0018       50 Z=DZR*FLOAT(I-1)
0019          XL(I)=TF*DEXP((ZZZ-Z)/ALPH)
0020          IF(Z.GT.RHMA)GO TO 100
0021          I=I+1
0022          GO TO 50
0023      100 K1=1
0024          VS(1)=XL(1)/FSQ
0025          H1=0.0
0026          H2=DZR
0027          DO 200 I=2,NV
0028          Z=FLOAT(I-1)*DZ
0029          IF(Z.GE.RHMA)Z=RHMA
0030      160 IF(Z.LE.H2)GO TO 150
0031          K1=K1+1
0032          H1=FLOAT(K1-1)*DZR
0033          H2=FLOAT(K1)*DZR
0034          GO TO 160
0035      150 DH=H2-H1
0036          SN=(XL(K1+1)-XL(K1))/DH
0037          BN=-(XL(K1+1)*H1-XL(K1)*H2)/DH
0038          VS(I)=(SN*Z+BN)/FSQ
0039      200 CONTINUE
0040          RETURN
0041          END
```

```
0001          REAL*8 FUNCTION FNA(W)
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          COMMON S(101),P(101),G(101),T(101),X(101),VS(101),NV,DZ,SURF
0004     C
0005     C          SUBFUNCTION TO INTEGRATE FROM Z=O TO Z=-H
0006     C
0007          NVM=NV-2
0008          W2=W*W
0009          TW=-2.*W
0010          DO 100 I=1,NV
0011          EX=DEXP(TW*X(I))
```

```
0012          T(I)=W2*EX
0013        100 S(I)=EX/VS(I)
0014          P(NV)=1.0
0015          G(NV)=0.0
0016          DO 200 I=1,NVM
0017          J=NV-I
0018          K=J+1
0019          G(J)=G(K)-.5*DZ*(T(J)+T(K))*P(K)
0020        200 P(J)=P(K)+DZ*G(J)/S(J)
0021          G(1)=G(2)-.5*DZ*(T(1)+T(2))*P(2)
0022          FNA=G(1)+S(1)*SURF*.5*(3.*P(2)-P(3))
0023          RETURN
0024          END



0001          SUBROUTINE ROOT(X5,X6,X7,FNA,X3,IROOT)
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003      C
0004      C          SUBROUTINE TO FIND ROOT OF FNA BETWEEN X5 AND X6
0005      C
0006          IROOT=1
0007        220 IQ=INT((X6-X5)/X7)-1
0008          I1=0
0009        240 X1=X5+I1*X7
0010          X2=X1+X7
0011          Y1=FNA(X1)
0012          Y2=FNA(X2)
0013          IF(Y1*Y2.LT.0.0)GO TO 340
0014          I1=I1+1
0015          IF(I1.NE.IQ)GO TO 240
0016          WRITE(6,13)
0017          IROOT=0
0018         13 FORMAT(//'  NO ROOT FOUND'/)
0019          GO TO 400
0020        340 IF(X1.NE.X5)GO TO 370
0021          X7=X7/2.
0022          GO TO 220
0023        370 X2=X1
0024          X1=X1-0.1*X7
0025          Y1=FNA(X1)
0026          Y2=FNA(X2)
0027        420 X3=(Y2*X1-Y1*X2)/(Y2-Y1)
0028          IF(X3.GT.X5.AND.X3.LT.X6)GO TO 200
0029          WRITE(6,11)X3
0030         11 FORMAT(/'  PROJECTION OUT OF INTERVAL:',D15.5//)
```

```
0031      200 Y3=FNA(X3)
0032          IF(DABS((X3-X2)/X2).LT.1.E-7)GO TO 400
0033          X1=X2
0034          X2=X3
0035          Y1=Y2
0036          Y2=Y3
0037          GO TO 420
0038      400 RETURN
0039          END
```

# 1 Program BIGDRV2 Listing

```
0001          PROGRAM WDSTF
0002          COMMON F,DX,DT,NN,MM,NM,NMX
0003          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0004          COMMON XL(600)
0005          DOUBLE COMPLEX AX(425,53),BX(425),BXB(425),DCMPLX
0006          DIMENSION BBB(50),TX(25),TY(25),TYX(25),TXX(25)
0007          DIMENSION R(25),RX(25)
0008          DOUBLE PRECISION DREAL,DIMAG,DATAN2,A,B,AMT,THT,DSQRT
0009          DOUBLE COMPLEX FFC(25)
0010     C
0011     C              MAIN PROGRAM TO CALL ALL OF THE OTHER PIECES
0012     C
0013     C              DIM(AX) = NM,2NN+3
0014          READ(5,*) ICCM
0015          ICCC =1
0016          READ(5,*) F,XMAX
0017          FF = 180.0/3.14149
0018          XMAX = XMAX*1.0E+05
0019          F =F*1.0E-05
0020          WRITE(6,907) F,XMAX
0021          READ(5,*) ILW,IRL,IXY
0022          IF (ILW.EQ.1) GO TO 5
0023          ILW = 0
0024          WRITE(6,910)
0025          GO TO 10
0026     5    WRITE(6,911)
0027     10   DD2 = FLOAT(ILW)
0028          IF (IRL.EQ.1) GO TO 15
0029          IRL = 0
0030          WRITE(6,912)
0031          GO TO 25
0032     15   WRITE(6,913)
0033     25   DD1 = FLOAT(IRL)
0034          IF (IXY.EQ.1) GO TO 27
0035          IXY = 0
0036          WRITE(6,915)
0037          GO TO 29
0038     27   IF (ILW.EQ.0) GO TO 28
0039          WRITE(6,916)
0040          GO TO 29
0041     28   WRITE(6,917)
0042          GO TO 200
0043     29   MM = 17
0044          NN = 25
```

```
0045          NM = NN*MM
0046          NMX = 2*NN +3
0047          DX = XMAX/FLOAT(NN-1)
0048          DT = 1./FLOAT(MM-1)
0049          CALL DEP(BBB,RQPP)
0050          CALL NSQ
0051          CALL FRIC(R,RX)
0052          CALL WRD(TX,TY,TXX,TYX,IXY)
0053    3     READ(5,*) W,RL
0054          CALL MATS(RL,W,AX,BX,R,RX,DD1,DD2,TX,TY,TXX,TYX,FFC)
0055          DO 20 M = 1,MM
0056          DO 20 N = 1,NN
0057          IN = N + NN*(M-1)
0058          II = M + MM*(N-1)
0059    20    BXB(II) = BX(IN)
0060          DO 30 I = 1,NM
0061    30    BX(I) = BXB(I)
0062          WRITE(6,903) W,RL
0063          CALL VCAL(W,RL,BX,BXB,R,RX,VKE,DD2,TX,TY)
0064          CALL UCAL(W,RL,BX,BXB,R,UKE,DD2,TX,TY)
0065          CALL RHOC(BX,BXB,RPE,DD1,DD2,FFC)
0066          CALL PEC(BX,SPE,IRL)
0067          RRRR = (UKE + VKE)/(RPE + SPE)
0068          TE = UKE + VKE + RPE + SPE
0069          WRITE(6,914) TE
0070          WRITE(6,909) RRRR
0071    125   WRITE(6,908)
0072    126   DO 130 N = 1,NN
0073          X = DX*FLOAT(N-1)
0074          DZ = RH(N)*DT
0075          WRITE(6,904) X,RH(N),DZ
0076          ML = 1 +MM*(N-1)
0077          MH = MM*N
0078          DO 128 M = ML,MH
0079          A = DREAL(BX(M))
0080          B = DIMAG(BX(M))
0081          AMT = DSQRT(A*A + B*B)
0082          THT = FF*DATAN2(B,A)
0083    128   BX(M) = DCMPLX(AMT,THT)
0084          WRITE(6,901) (BX(M),M=ML,MH)
0085    130   CONTINUE
0086    140   ICCC = ICCC +1
0087          IF (ICCC.LE.ICCM) GO TO 3
0088    901   FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
0089    903   FORMAT(////' W,L =',2E15.5)
0090    904   FORMAT(/'  X,H,DZ =',3E15.5)
```

```
0091    907   FORMAT(//'  F,XMAX =',2E15.5//)
0092    908   FORMAT(//'  PRESSURE'/)
0093    909   FORMAT(/' KE/PE =',E15.5)
0094    910   FORMAT(' LONG WAVE')
0095    911   FORMAT(' GENERAL FREQUENCY AND WAVELENGTH')
0096    912   FORMAT(' RIGID LID')
0097    913   FORMAT(' FREE SURFACE')
0098    914   FORMAT(' TOTAL ENERGY/LENGTH =',E15.5)
0099    915   FORMAT(' TAU Y DRIVING')
0100    916   FORMAT(' TAU X DRIVING')
0101    917   FORMAT(' ERROR: TAU X DRIVING IN THE LONG WAVE LIMIT')
0102    200   STOP
0103          END
```

```
0001
0002          SUBROUTINE NSQ
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0005          COMMON XL(600)
0006    C
0007    C
0008    C             PROGRAM TO READ AND INTERPOLATE BUOYANCY FREQUENCY SQUARED
0009    C                   BV = BUOYANCY FREQUENCY SQUARED
0010    C                   BVZ = Z DERIVATIVE OF BV
0011    C
0012          READ(5,*) NR,DZR,ALPH
0013          ALPH = ALPH*1.0E+05
0014          DZR = DZR*100.
0015          MX = MM -1
0016          READ(5,*) CMLT
0017          DO 20 I = 1,NR
0018          READ(5,*) XL(I)
0019    20    XL(I) = XL(I)*CMLT
0020          TF = XL(NR)
0021          ZZZ = DZR*FLOAT(NR-1)
0022          DZI = RH(NN) - ZZZ
0023          NL = NR +1
0024          NH = 600
0025          DO 30 I = NL,NH
0026          Z = DZR*FLOAT(I-1)
0027    30    XL(I) = TF*EXP((ZZZ -Z)/ALPH)
0028          DO 200 N = 1,NN
0029          DD = RH(N)
0030          DZ = DT*DD
```

```
0031          IF (DZ.GT.DZR) GO TO 110
0032          BV(N,MM) = XL(1)
0033          DO 50 M = 1,MX
0034          Z = DD - DZ*FLOAT(M-1)
0035          IBXL = 1 + IFIX(Z/DZR)
0036          IBXH = IBXL +1
0037          ZS = DZR*FLOAT(IBXL -1)
0038     50   BV(N,M) = XL(IBXL) +(Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
0039          Z = DD - DZ
0040          IBXL = 1 + IFIX(Z/DZR)
0041          IBXH = IBXL + 1
0042          ZS = DZR*FLOAT(IBXL-1)
0043          AQ = XL(IBXL) +(Z-ZS)*(XL(IBXH)-XL(IBXL))/DZR
0044          GO TO 145
0045    110   ZD = DZ/2.
0046          XBB = 0.
0047          NAVG = 0
0048          DO 120 I = 1,NR
0049          ZC = DZR*FLOAT(I-1)
0050          IF (ZC.GT.ZD) GO TO 120
0051          XBB = XBB + XL(I)
0052          NAVG = NAVG +1
0053    120   CONTINUE
0054          BV(N,MM) = XBB/FLOAT(NAVG)
0055          DO 140 MQ = 1,MM
0056          M = MQ -1
0057          Z = DD - DZ*FLOAT(M-1)
0058          ZS = Z - DZ/2.
0059          ZD = ZS + DZ
0060    125   XBB = 0.
0061          NAVG = 0
0062          DO 130 I = 1,NH
0063          ZC = DZR*FLOAT(I-1)
0064          IF ( ZC.LT.ZS) GO TO 130
0065          IF (ZC.GT.ZD) GO TO 130
0066          XBB = XBB + XL(I)
0067          NAVG = NAVG + 1
0068    130   CONTINUE
0069          IF (NAVG.NE.0) GO TO 135
0070          IBXL = 1 + IFIX(Z/DZR)
0071          IBXH = IBXL + 1
0072          ZSS = DZR*FLOAT(IBXL-1)
0073          IF (M.EQ.0) GO TO 133
0074          BV(N,M) = XL(IBXL) +(Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
0075          GO TO 140
0076    133   AQ = XL(IBXL) +(Z-ZSS)*(XL(IBXH)-XL(IBXL))/DZR
```

```
0077          GO TO 140
0078     135  IF (M.EQ.0) GO TO 138
0079          BV(N,M) = XBB/FLOAT(NAVG)
0080          GO TO 140
0081     138  AQ = XBB/FLOAT(NAVG)
0082     140  CONTINUE
0083     145  DO 150 M = 2,MX
0084          IP = M +1
0085          IM = M -1
0086     150  BVZ(N,M) = (BV(N,IP) - BV(N,IM))/(2.*DZ)
0087          BVZ(N,1) = (BV(N,2) -AQ)/(2.0*DZ)
0088          BVZ(N,MM) = (BV(N,MM) - BV(N,MX))/DZ
0089     200  CONTINUE
0090          WRITE(6,901)
0091          WRITE(6,902) (BV(NN,J),J=1,MM)
0092     901  FORMAT(//'   NSQUARED AT XMAX '/)
0093     902  FORMAT(2X,10E12.5)
0094     250  RETURN
0095          END


0001
0002          SUBROUTINE MATS(RL,W,AX,BX,R,RX,DD1,DD2,TX,TY,TXX,TYX,FFC)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0005          COMMON XL(600)
0006          DOUBLE COMPLEX AX(425,53),BX(425),AB,DCMPLX,B3,B5
0007          DOUBLE COMPLEX AA3,AA4,AA5,DQ,AA1,AA2,FFC(25)
0008          DOUBLE COMPLEX GAM,B4,ABB
0009          DIMENSION R(25),RX(25),TX(25),TY(25),TYX(25),TXX(25)
0010     C
0011     C
0012     C            SUBROUTINE TO SET AND SOLVE THE P EQUATION
0013     C
0014          GG = 1./980.
0015          CSS = GG*2.0*DT*DD1
0016          CALL FFCCAL(TX,TY,TXX,TYX,FFC,W,RL,DD2)
0017          DXDT = DX/(2.*DT)
0018          DXX = DX*DX
0019          DDD = DXX/(DT*DT)
0020          RLL = DD2*RL*RL
0021          DDX = DX*DXDT
0022          RQP = 2.0*DX
0023          DQ = DCMPLX(RQP,0.0)
0024          FLW = F*RL/W
0025     3    DO 5 I = 1,NM
```

```
0026    5       BX(I) = (0.0,0.0)
0027            DO 10 J = 1,NMX
0028            DO 10 I = 1,NM
0029    10      AX(I,J) =(0.0,0.0)
0030            J = NN + 2
0031            IP1 = J + 1
0032            IM1 = J - 1
0033            IJJP = J + 2
0034            IJJM = J - 2
0035            IPM1 = 2*NN +1
0036            IPM = 2*NN + 2
0037            IPMM = 2*NN + 3
0038            DO 100 M = 1,MM
0039            DO 90 N = 1,NN
0040            I = N + NN*(M-1)
0041            CALL AS(A1,A2,A3,W,N,M,C1,C2,C3,DD2)
0042            RQP = A2*DXDT
0043            AB = DCMPLX(RQP,0.0)
0044            AX(I,1) = AB
0045            AX(I,IPM1) = -AB
0046            AX(I,3) = -AB
0047            AX(I,IPMM) = AB
0048            AX(I,IM1) = (1.0,0.0)
0049            AX(I,IP1) = (1.0,0.0)
0050            RQP = A1*DDD - A3*DDX
0051            AX(I,2) = DCMPLX(RQP,0.0)
0052            RQP = A1*DDD + A3*DDX
0053            AX(I,IPM) = DCMPLX(RQP,0.0)
0054            RQP = -2.0 -2.0*A1*DDD -RLL*DXX
0055            AX(I,J) = DCMPLX(RQP,0.0)
0056            IF ( N.EQ.1) GO TO 20
0057            IF (N.EQ.NN) GO TO 30
0058            IF (M.EQ.1) GO TO 40
0059            IF (M.EQ.MM) GO TO 50
0060            GO TO 90
0061    20      RQP = 2.0*DX
0062            FW = F*F - DD2*W*W
0063            CHH = RL*F*RH(N)
0064            CH = R(N)*DD2*2.0*F*W*RL/FW
0065            AA1 = DCMPLX(CH,CHH)
0066            CH = R(N)*(F*F + DD2*W*W)/FW
0067            CHH = W*RH(N)
0068            AA2 = DCMPLX(CH,CHH)
0069            CH = F*TY(N)
0070            CHH = W*TX(N)*DD2
0071            ABB = DCMPLX(CH,CHH)
```

```
0072          ABB = 2.0*DX*ABB/AA2
0073          BX(I) = BX(I) + AX(I,IM1)*ABB
0074          BX(I) = BX(I) + AX(I,IPM1)*ABB
0075          BX(I) = BX(I) + AX(I,1)*ABB
0076          AX(I,IP1) = AX(I,IP1) + AX(I,IM1)
0077          AX(I,J) = AX(I,J) + RQP*AA1*AX(I,IM1)/AA2
0078          AX(I,IPM) = AX(I,IPM) + AX(I,IM1)*A2*DX/DT
0079          AX(I,2) = AX(I,2) - AX(I,IM1)*A2*DX/DT
0080          AX(I,IPMM) = AX(I,IPMM) + AX(I,IPM1)
0081          AX(I,IPM) = AX(I,IPM) + AX(I,IPM1)*RQP*(AA1/AA2 + A2/DT)
0082          AX(I,J) = AX(I,J) - AX(I,IPM1)*A2*RQP/DT
0083          AX(I,3) = AX(I,3) + AX(I,1)
0084          AX(I,2) = AX(I,2) + AX(I,1)*RQP*(AA1/AA2 - A2/DT)
0085          AX(I,J) = AX(I,J) + AX(I,1)*A2*RQP/DT
0086          AX(I,1) = (0.0,0.0)
0087          AX(I,IPM1) = (0.0,0.0)
0088          AX(I,IM1) =(0.0,0.0)
0089    24    IF (M.NE.MM) GO TO 25
0090          AX(I,2) = AX(I,2) + AX(I,IPM)
0091          AX(I,3) = AX(I,3) + AX(I,IPMM)
0092          CZQ = RH(N)*BV(N,M)*CSS
0093          AX(I,IP1) = AX(I,IP1) - AX(I,IPMM)*CZQ
0094          AX(I,J) = AX(I,J) - AX(I,IPM)*CZQ
0095          BX(I) = BX(I) - AX(I,IPM)*2.0*RH(1)*DT*FFC(1)
0096          BX(I) = BX(I) - AX(I,IPMM)*2.0*RH(1)*DT*FFC(1)
0097          AX(I,IPM) = (0.0,0.0)
0098          AX(I,IPMM) = (0.0,0.0)
0099          GO TO 90
0100    25    IF (M.NE.1) GO TO 90
0101          CH = R(N)*DD2*2.0*F*W*RL/FW
0102          CHH = F*RL*RH(N)
0103          B3 = DCMPLX(CH,CHH)
0104          CH = R(N)*(F*F + DD2*W*W)/FW
0105          CHH = W*RH(N)
0106          B4 = DCMPLX(CH,CHH)
0107          CH = F*TY(N)
0108          CHH = W*DD2*TX(N)
0109          B5 = DCMPLX(CH,CHH)
0110          CH = -2.0*W*RL*DD2*F*RX(N)/FW
0111          CHH = -RL*F*RHX(N)
0112          AA1 = DCMPLX(CH,CHH)
0113          CH = (F*F + DD2*W*W)*RX(N)/FW
0114          CHH = W*RHX(N)
0115          AA2 = DCMPLX(CH,CHH)
0116          AA1 = AA1 + AA2*B3/B4
0117          CH = -R(N)*(F*F +DD2*W*W)*BVZ(N,M)/(BV(N,M)*BV(N,M))
```

```
0118          CH = CH + 2.0*W*RL*DD2*F*R(N)*RHX(N)/FW
0119          CHH = -W*FW/BV(N,M)
0120          AA2 = DCMPLX(CH,CHH)
0121          CH = (F*F + DD2*W*W)*(-R(N)*RHX(N))/FW
0122          AA2 = AA2 + CH*B3/B4
0123          CH = R(N)*(F*F +DD2*W*W)/BV(N,M)
0124          AA3 = DCMPLX(CH,0.0)
0125          CH = -(F*F + DD2*W*W)*RX(N)/FW
0126          CHH = -W*RHX(N)
0127          AA4 = DCMPLX(CH,CHH)
0128          AA4 = AA4/B4
0129          DZ = DT*RH(N)
0130          ABB = (0.5*AA2 - AA3/DZ)/DZ
0131          AX(I,J) = AX(I,J)+ AX(I,2)*(AA1 -2.0*AA3/(DZ*DZ))/ABB
0132          AX(I,IPM) = AX(I,IPM) +AX(I,2)*(0.5*AA2 +AA3/DZ)/(DZ*ABB)
0133          BX(I) = BX(I) - AX(I,2)*AA4*B5/ABB
0134          RQP = 2.0*DX
0135          CALL AACAL(W,RL,R,RX,N,M,DD2,AA1,AA2,AA3,AA4,AA5)
0136          ABB = AA1*0.5/DT -AA4/(DT*DT)
0137          AX(I,J) = AX(I,J) + AX(I,3)*(-AA3/RQP + AA5/(DX*DT))/ABB
0138          B3 = AX(I,3)*(AA2 -2.0*AA4/(DT*DT) -AA5/(DX*DT))/ABB
0139          AX(I,IP1) = AX(I,IP1) + B3
0140          AX(I,IJJP) = AX(I,IJJP) + AX(I,3)*AA3/(RQP*ABB)
0141          AX(I,IPM) = AX(I,IPM) -AX(I,3)*AA5/(DX*DT*ABB)
0142          B3 = AX(I,3)*(AA1*0.5/DT + AA4/(DT*DT) + AA5/(DX*DT))/ABB
0143          AX(I,IPMM) = AX(I,IPMM) +B3
0144          AX(I,3) = (0.0,0.0)
0145          AX(I,2) = (0.0,0.0)
0146          GO TO 90
0147    30    DZ = DT*RH(N)
0148          AX(I,IPM1) = (0.0,0.0)
0149          AX(I,3) = (0.0,0.0)
0150          AX(I,IPMM) = (0.0,0.0)
0151          AX(I,1) = (0.0,0.0)
0152          C1 = 1. + FLW*DX/2.
0153          C2 = 1. - FLW*DX/2.
0154          AX(I,J) = AX(I,J) +2.0*AX(I,IP1)/C1
0155          AX(I,IM1) = AX(I,IM1) -C2*AX(I,IP1)/C1
0156          AX(I,IP1) = (0.0,0.0)
0157          IF (M.NE.MM) GO TO 35
0158          AX(I,2) = AX(I,2) +AX(I,IPM)
0159          CZQ = RH(N)*BV(N,M)*CSS
0160          AX(I,J) = AX(I,J) - AX(I,IPM)*CZQ
0161          BX(I) = BX(I) -AX(I,IPM)*2.0*RH(N)*DT*FFC(N)
0162          AX(I,IPM) = (0.0,0.0)
0163          GO TO 90
```

```
0164    35    IF (M.NE.1) GO TO 90
0165          CA = R(N)*(F*F + DD2*W*W)/(F*F -DD2*W*W)
0166          CH = -CA*BVZ(N,M)/BV(N,M)
0167          CHH = -W
0168          ABB = DCMPLX(CH,CHH)
0169          ABB = ABB/CA
0170          AA1 = 1.0 -ABB*RH(N)*DT/2.0
0171          AA2 = 1.0 + ABB*DT*RH(N)/2.0
0172          AX(I,J) = AX(I,J) +AX(I,2)*2.0/AA1
0173          AX(I,IPM) = AX(I,IPM) -AX(I,2)*AA2/AA1
0174          AX(I,2) = (0.0,0.0)
0175          GO TO 90
0176    40    CALL AACAL(W,RL,R,RX,N,M,DD2,AA1,AA2,AA3,AA4,AA5)
0177          DTQ = 2.0*DT*DX
0178          RQP = 2.0*DX
0179          ABB = AA1*0.5/DT - AA4/(DT*DT)
0180          AX(I,IM1) = AX(I,IM1) + AX(I,2)*(-AA3/RQP + AA5/DTQ)/ABB
0181          AX(I,J) = AX(I,J) + AX(I,2)*(AA2 - AA4*2.0/(DT*DT))/ABB
0182          AX(I,IP1) = AX(I,IP1) + AX(I,2)*(AA3/RQP - AA5/DTQ)/ABB
0183          AX(I,IPM1) = AX(I,IPM1) + AX(I,2)*(-AA5/DTQ)/ABB
0184          AX(I,IPM) = AX(I,IPM) + AX(I,2)*(AA1*0.5/DT + AA4/(DT*DT))/ABB
0185          AX(I,IPMM) = AX(I,IPMM) + AX(I,2)*AA5/(DTQ*ABB)
0186          IF (N.EQ.2) GO TO 42
0187          AX(I,IJJM) = AX(I,IJJM) - AX(I,1)*AA3/(RQP*ABB)
0188          B3 = AX(I,1)*(AA2 - 2.0*AA4/(DT*DT) + AA5*2.0/DTQ)/ABB
0189          AX(I,IM1) = AX(I,IM1) + B3
0190          AX(I,J) = AX(I,J) + AX(I,1)*(AA3/RQP - AA5*2.0/DTQ)/ABB
0191          B3 = AX(I,1)*(AA1*0.5/DT + AA4/(DT*DT) - AA5*2.0/DTQ)/ABB
0192          AX(I,IPM1) = AX(I,IPM1) + B3
0193          AX(I,IPM) = AX(I,IPM) + AX(I,1)*AA5*2.0/(DTQ*ABB)
0194    41    AX(I,J) = AX(I,J) + AX(I,3)*(-AA3/RQP + AA5*2.0/DTQ)/ABB
0195          B3 = AX(I,3)*(AA2 - AA4*2.0/(DT*DT) -AA5*2.0/DTQ)/ABB
0196          AX(I,IP1) = AX(I,IP1) + B3
0197          AX(I,IJJP) = AX(I,IJJP) + AX(I,3)*AA3/(RQP*ABB)
0198          AX(I,IPM) = AX(I,IPM) - AX(I,3)*AA5*2.0/(DTQ*ABB)
0199          B3 = AX(I,3)*(AA1*0.5/DT +AA4/(DT*DT) + AA5*2.0/DTQ)/ABB
0200          AX(I,IPMM) = AX(I,IPMM) + B3
0201          AX(I,3) = (0.0,0.0)
0202          AX(I,1) = (0.0,0.0)
0203          AX(I,2) = (0.0,0.0)
0204          GO TO 90
0205    42    B3 = AX(I,1)*(AA2 -AA3/DX -2.0*AA4/(DT*DT) + AA5*2.0/DTQ)/ABB
0206          AX(I,IM1) = AX(I,IM1) + B3
0207          AX(I,J) = AX(I,J) + AX(I,1)*(AA3/DX - AA5*2.0/DTQ)/ABB
0208          B3 = AX(I,1)*(0.5*AA1/DT + AA4/(DT*DT) -AA5*2.0/DTQ)/ABB
0209          AX(I,IPM1) = AX(I,IPM1) + B3
```

```
0210          AX(I,IPM) = AX(I,IPM) + AX(I,1)*AA5*2.0/(DTQ*ABB)
0211          GO TO 41
0212    50    AX(I,1) = (0.0,0.0)
0213          AX(I,3) = (0.0,0.0)
0214          AX(I,IPM1) = (0.0,0.0)
0215          AX(I,IPMM) =(0.0,0.0)
0216          AX(I,2) = AX(I,2) + AX(I,IPM)
0217          CZQ = RH(N)*BV(N,M)*CSS
0218          AX(I,J) = AX(I,J) -AX(I,IPM)*CZQ
0219          BX(I) = BX(I) - AX(I,IPM)*2.0*RH(N)*DT*FFC(N)
0220          AX(I,IPM) = (0.0,0.0)
0221    90    CONTINUE
0222    100   CONTINUE
0223          CALL BANDG(AX,BX)
0224          RETURN
0225          END
```

```
0001
0002          SUBROUTINE AS(A1,A2,A3,W,N,M,C1,C2,C3,DD2)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0005          COMMON XX(600)
0006    C
0007    C             SUBROUTINE TO CALCULATE COEFFICIENTS FOR MATS
0008    C
0009          F2 = F*F
0010          W2 = DD2*W*W
0011          TZ = 1./RH(N)
0012          TH = -1. +DT*FLOAT(M-1)
0013          TX = -RHX(N)*TH*TZ
0014          AA = (RHX(N)/RH(N))**2
0015          TXX = (2.*AA -RHXX(N)/RH(N))*TH
0016          BW = BV(N,M)
0017          BWW = BW*BW
0018          A1 = TX*TX + TZ*TZ*(F2-W2)/BW
0019          A2 = TX
0020          A3 = TXX -TZ*BVZ(N,M)*(F2-W2)/BWW
0021          C1 = TZ
0022          C2 = RHX(N)*BV(N,1)/(F2-W2)
0023          C3 = C1*RHX(N)
0024          RETURN
0025          END
```

```
0001
0002          SUBROUTINE DEP(BX,RQPP)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0005          COMMON XL(600)
0006          DIMENSION BX(50)
0007     C
0008     C              SUBROUTINE TO READ AND INTERPOLATE DEPTH PROFILE
0009     C                    RH = DEPTH
0010     C                    RHX = X DERIVATIVE OF RH
0011     C                    RHXX = SECOND X DERIVATIVE OF RH
0012     C
0013          READ(5,*) NRX
0014          DO 5 I =1,NRX
0015          READ(5,*) XL(I),BX(I)
0016          BX(I) = BX(I)*100.
0017     5    XL(I) = XL(I)*1.0E+05
0018          RHMA = BX(NRX)
0019          RQPP = XL(NRX)
0020          RH(1) = BX(1)
0021          DO 20 N = 2,NN
0022          X = DX*FLOAT(N-1)
0023          IF (X.GT.XL(NRX)) GO TO 15
0024          IC = 0
0025          DO 8 J = 2,NRX
0026          IF (IC.NE.0) GO TO 8
0027          I = J
0028          IF (X.GT.XL(I)) GO TO 8
0029          IC = I
0030     8    CONTINUE
0031          IM = I-1
0032          RHX(N) = (BX(I) -BX(IM))/(XL(I)-XL(IM))
0033          XX = X - XL(IM)
0034          RH(N) = BX(IM) + RHX(N)*XX
0035          GO TO 20
0036     15   RH(N) = RHMA
0037     20   CONTINUE
0038          RHX(1) = (RH(2) - RH(1))/DX
0039          RHXX(1) = 0.
0040          D2 = 2.*DX
0041          DXX = DX*DX
0042          NX = NN -1
0043          DO 30 N = 2,NX
0044          IP = N +1
```

```
0045          IM = N -1
0046          RHX(N) =(RH(IP) - RH(IM))/D2
0047    30    RHXX(N) = (RH(IP) -2.*RH(N) + RH(IM))/DXX
0048          RHX(NN) = 0.
0049          RHXX(NN) = 0.
0050          RETURN
0051          END


0001
0002          SUBROUTINE VCAL(W,RL,BX,XL,R,RX,XINT,DD2,TX,TY)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0005          COMMON XB(600)
0006          DOUBLE COMPLEX BX(425),XL(425),XPX,XPT,DCMPLX,RMU,T3,B3,B5,AA1,AA2
0007          DOUBLE COMPLEX GAM,B4,AA3,AA4,AA5
0008          DOUBLE PRECISION DREAL,DIMAG,DATAN2,DSQRT,A,B
0009          DIMENSION R(25),RX(25),TX(25),TY(25)
0010          DOUBLE PRECISION XINT
0011    C
0012    C             SUBROUTINE TO CALCULATE V FROM P
0013    C
0014          RHO = 0.5015
0015          FF = 180.0/3.14159
0016          XINT = 0.
0017          WRITE(6,905)
0018          FW = F*F - DD2*W*W
0019          WL= DD2*W*RL
0020          RLW = RL/W
0021          FLW = F*RL/W
0022          MX = MM -1
0023          DXX = 2.*DX
0024          W2 = DD2*W*W
0025          DO 50 N = 1,NN
0026          XX = DX
0027          X = DX*FLOAT(N-1)
0028          D = RH(N)
0029          DD = RHX(N)/D
0030          DZ = DT*D
0031          RHZ = RHX(N)**2
0032          IF (N.EQ.1) GO TO 2
0033          IF (N.NE.NN) GO TO 20
0034          GO TO 18
0035     2    CH = R(N)*(F*F + W2)/FW
0036          CHH = W*RH(N)
0037          GAM = DCMPLX(CH,CHH)
```

```
0038           CH = F*TY(N)
0039           CHH = W*TX(N)*DD2
0040           XPX = DCMPLX(CH,CHH)
0041           XPX = XPX/GAM
0042           XPT = XPX
0043           CH = R(N)*DD2*2.0*W*F*RL/FW
0044           CHH = RL*F*RH(N)
0045           B4 = DCMPLX(CH,CHH)
0046           DO 5 I = 1,MM
0047           XPX = XPT -B4*BX(I)/GAM
0048     5     XL(I) = (F*XPX + WL*BX(I))/FW
0049           XX = DX/2.0
0050           GO TO 40
0051    18     C1 =  1. + FLW*DX/2.
0052           C2 = 1. - FLW*DX/2.
0053           DQ = 1./(F*DX)
0054           DO 19 M = 1,MM
0055           I = (N-1)*MM +M
0056           IM = I -MM
0057           XPX = (BX(I)*2.0/C1 - BX(IM)*(1. + C2/C1))/DXX
0058    19     XL(I) = (F*XPX + WL*BX(I))/FW
0059           XX = DX/2.
0060           GO TO 40
0061    20     I = (N-1)*MM +1
0062           M = 1
0063           IP = I + MM
0064           IM = I - MM
0065           I2 = I + 1
0066           I1 = I - 1
0067           IQQ = I -MM + 1
0068           IQR = I + MM + 1
0069           CALL AACAL(W,RL,R,RX,N,M,DD2,AA1,AA2,AA3,AA4,AA5)
0070           DTT = 2.0*DT
0071           GAM = AA1/DTT - AA4/(DT*DT)
0072           B3 = BX(IM)*(-AA3/DXX +AA5/(DXX*DT))
0073           B3 = B3 +BX(I)*(AA2-2.0*AA4/(DT*DT))
0074           B3 = B3 + BX(IP)*(AA3/DXX -AA5/(DXX*DT))
0075           B3 = B3 + BX(IQQ)*(-AA5/(DXX*DT))
0076           B3 = B3 + BX(I2)*(AA1/DTT + AA4/(DT*DT))
0077           B3 = B3 + BX(IQR)*(AA5/(DTT*DX))
0078           B3 = B3/GAM
0079           XPX = (BX(IP) - BX(IM))/DXX
0080           T = -1.0
0081           XPT = (BX(I2)-B3)/DTT
0082           XPX = XPX - T*DD*XPT
0083           XL(I) = (F*XPX + WL*BX(I))/FW
```

```
0084            I = N*MM
0085            IP = I + MM
0086            IM = I -MM
0087            XPX = (BX(IP) - BX(IM))/DXX
0088            XL(I) =(WL*BX(I) + F*XPX)/FW
0089            DO 30 M = 2,MX
0090            I = (N-1)*MM +M
0091            IP = I + MM
0092            IM = I -MM
0093            I2 = I +1
0094            I1 = I -1
0095            XPX = (BX(IP) - BX(IM))/DXX
0096            T = -1. +DT*FLOAT(M-1)
0097            XPT = (BX(I2) -BX(I1))/(2.*DT)
0098            XPX = XPX - T*DD*XPT
0099      30    XL(I) = (WL*BX(I) + F*XPX)/FW
0100      40    DO 45 M = 1,MM
0101            I = (N-1)*MM + M
0102            ZZ = DZ
0103            A = DREAL(XL(I))
0104            B = DIMAG(XL(I))
0105            CQ = A*A + B*B
0106            IF (M.EQ.1) GO TO 42
0107            IF (M.NE.MM) GO TO 45
0108      42    ZZ = DZ/2.
0109      45    XINT = XINT + RHO*XX*ZZ*CQ
0110            WRITE(6,904) X,D,DZ
0111            IL = (N-1)*MM +1
0112            IH = N*MM
0113            DO 46 IJ = IL,IH
0114            A = DREAL(XL(IJ))
0115            B = DIMAG(XL(IJ))
0116            AMT = DSQRT(A*A + B*B)
0117            THT = FF*DATAN2(B,A)
0118      46    XL(IJ) = DCMPLX(AMT,THT)
0119            WRITE(6,901) (XL(IJ),IJ=IL,IH)
0120      50    CONTINUE
0121            WRITE(6,902) XINT
0122            DO 110 N = 1,NN
0123            IJ = 1 + MM*(N-1)
0124            A = DREAL(XL(IJ))
0125            B = DIMAG(XL(IJ))
0126            AMT = R(N)*A
0127            THT = B
0128     110    XL(N) = DCMPLX(AMT,THT)
0129            WRITE(6,906)
```

```
0130          WRITE(6,901)(XL(N),N=1,NN)
0131    901   FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
0132    902   FORMAT(//'  V CONTRIBUTION TO KE =',E15.5/)
0133    905   FORMAT(////'  V'//)
0134    904   FORMAT(/'    X, H, DZ =',3E15.5)
0135    906   FORMAT(//'  Y BOTTOM STRESS (DYNES/CM2)='/)
0136          RETURN
0137          END



0001
0002          SUBROUTINE RHOC(BX,XL,XINT,DD1,DD2,FFC)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0005          COMMON XQ(600)
0006          DOUBLE COMPLEX BX(425),XL(425),DCMPLX,FFC(25)
0007          DOUBLE PRECISION DREAL,DIMAG,DATAN2,DSQRT,A,B,CDABS
0008          DOUBLE PRECISION XINT
0009    C
0010    C             SUBROUTINE TO CALCULATE DENSITY FROM P
0011    C
0012          G2 = 980.*980./2.06
0013          GRQ = (1.0/980.0)**2
0014          FF = 180.0/3.14159
0015          XINT = 0.
0016          WRITE(6,903)
0017          G = 980.
0018          DO 50 N = 1,NN
0019          DXX = DX
0020          IF (N.EQ.1) GO TO 2
0021          IF (N.NE.NN) GO TO 5
0022    2     DXX = DX/2.
0023    5     X = DX*FLOAT(N-1)
0024          DZ = DT*RH(N)
0025          GDZ = G*DZ
0026          DO 40 M = 1,MM
0027          DZZ = DZ
0028          I = (N-1)*MM +M
0029          IF (M.EQ.1) GO TO 10
0030          IF (M.NE.MM) GO TO 20
0031          DZZ = DZ/2.
0032          XL(M) = DD1*BV(N,M)*GRQ*BX(I) -FFC(N)/980.
0033          GO TO 35
0034    10    IP = I + 1
0035          DZZ = DZ/2.
```

```
0036          XL(M) = -(BX(IP) - BX(I))/GDZ
0037          GO TO 35
0038    20    IP = I +1
0039          IM = I -1
0040          XL(M) = -(BX(IP) -BX(IM))/(2.*GDZ)
0041    35    XX = G2*CDABS(XL(M))*CDABS(XL(M))/BV(N,M)
0042          XINT = XINT + XX*DZZ*DXX
0043    40    CONTINUE
0044          WRITE(6,901) X,RH(N),DZ
0045          DO 45 M = 1,MM
0046          A = DREAL(XL(M))
0047          B = DIMAG(XL(M))
0048          AMT = 1000.0*DSQRT(A*A + B*B)
0049          IF (AMT.NE.0.0) GO TO 42
0050    41    THT = 0.0
0051          GO TO 45
0052    42    THT = FF*DATAN2(B,A)
0053    45    XL(M) = DCMPLX(AMT,THT)
0054    50    WRITE(6,902) (XL(M),M=1,MM)
0055          WRITE(6,904) XINT
0056    901   FORMAT(/'  X,D,DZ =',3E15.5)
0057    902   FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
0058    903   FORMAT(///'  RHO (SIGMA-T UNITS)'/)
0059    904   FORMAT(//'  RHO CONTRIBUTION TO PE =',E15.5/)
0060          RETURN
0061          END
```

```
0001
0002          SUBROUTINE UCAL(W,RL,BX,XL,R,XINT,DD2,TX,TY)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17)
0005          COMMON XX(600)
0006          DIMENSION R(25),TX(25),TY(25)
0007          DOUBLE COMPLEX BX(425),XL(425),DCMPLX,XPT,XPX,XPIM,GAM,B4
0008          DOUBLE PRECISION XINT
0009          DOUBLE PRECISION CDABS,DREAL,DIMAG,DATAN2,DSQRT,A,B
0010    C
0011    C           SUBROUTINE TO CALCULATE U FROM P
0012    C
0013          RHO = 0.5015
0014          FF = 180.0/3.14159
0015          XINT = 0.
0016          FW = F*F - DD2*W*W
0017          FL = F*RL
```

```
0018          WRITE(6,903)
0019          DO 100 N = 1,NN
0020          DXX = DX
0021          X = DX*FLOAT(N-1)
0022          DZ = DT*RH(N)
0023          IF (N.EQ.1) GO TO 86
0024          IF (N.EQ.NN) GO TO 110
0025          DD = RHX(N)/RH(N)
0026          DO 85 M = 1,MM
0027          I = (N-1)*MM +M
0028          IP = I + MM
0029          IM = I -MM
0030          T = -1. + DT*FLOAT(M-1)
0031          IF (M.EQ.1) GO TO 10
0032          IF (M.EQ.MM) GO TO 15
0033          GO TO 20
0034    10    I1 = I +1
0035          XPT = (BX(I1) - BX(I))/DT
0036          GO TO 25
0037    15    XPT = (0.0,0.0)
0038          GO TO 25
0039    20    I2 = I+1
0040          I1 = I-1
0041          XPT = (BX(I2) - BX(I1))/(2.*DT)
0042    25    XPX = (BX(IP) - BX(IM))/(2.*DX)
0043          XPX = XPX -T*DD*XPT
0044    85    XL(M) = -(0.0,1.0)*(W*XPX + FL*BX(I))/FW
0045          GO TO 90
0046    86    CH = R(N)*(F*F + DD2*W*W)/FW
0047          CHH = W*RH(N)
0048          GAM = DCMPLX(CH,CHH)
0049          CH = F*TY(N)
0050          CHH = W*TX(N)*DD2
0051          XPX = DCMPLX(CH,CHH)
0052          XPX = XPX/GAM
0053          XPT = XPX
0054          CH = R(N)*DD2*2.0*W*F*RL/FW
0055          CHH = RL*F*RH(N)
0056          B4 = DCMPLX(CH,CHH)
0057          DO 87 M = 1,MM
0058          XPX = XPT - B4*BX(M)/GAM
0059    87    XL(M) = -(0.0,1.0)*(FL*BX(M) + W*XPX)/FW
0060          DXX = DX/2.0
0061          GO TO 90
0062    110   FLW = 0.5*FL/W
0063          DXX = DX/2.
```

```
0064           DO 120 M = 1,MM
0065           I = (NN-1)*MM +M
0066           IM = I - MM
0067           XPIM = 2.*BX(I)/DX +BX(IM)*(FLW -1./DX)
0068           XPIM = XPIM/(FLW + 1./DX)
0069     120   XL(M) = -(0.0,1.0)*(FL*BX(I) +W*0.5*(XPIM-BX(IM))/DX)/FW
0070     90    DO 95 M = 1,MM
0071           DZZ = DZ
0072           IF (M.EQ.1) GO TO 92
0073           IF (M.NE.MM) GO TO 95
0074     92    DZZ = DZ/2.
0075     95    XINT = XINT + RHO*CDABS(XL(M))*CDABS(XL(M))*DXX*DZZ
0076           DO 98 M = 1,MM
0077           A = DREAL(XL(M))
0078           B = DIMAG(XL(M))
0079           AMT = CDABS(XL(M))
0080           THT = FF*DATAN2(B,A)
0081     98    XL(M) = DCMPLX(AMT,THT)
0082           WRITE(6,901) X,RH(N),DZ
0083           WRITE(6,902) (XL(M),M=1,MM)
0084     100   CONTINUE
0085           WRITE(6,904) XINT
0086     901   FORMAT(/'  X,D,DZ =',3E15.5)
0087     902   FORMAT(2X,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2,E12.5,F9.2)
0088     903   FORMAT(///'   U'/)
0089     904   FORMAT(//'  U CONTRIBUTION TO KE =',E15.5/)
0090           RETURN
0091           END


0001
0002           SUBROUTINE BANDG(A,BB)
0003           COMMON F,DX,DT,NN,MM,NM,NMX
0004           COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
0005           DOUBLE COMPLEX A(425,53),BB(425)
0006           DOUBLE COMPLEX R
0007     C
0008     C              SUBROUTINE TO DO BAND GAUSSIAN ELIMINATION
0009     C
0010           MBW = NN + 1
0011           NH = NM -1
0012           MDIAG = MBW + 1
0013           DO 50 N = 1,NH
0014           IP = N + 1
0015           MH = N + MBW
0016           IF (MH.LE.NM) GO TO 5
```

```
0017          MH = NM
0018     5    DO 50 IR = IP,MH
0019          ICD = MBW + N + 1 -IR
0020          R = A(IR,ICD)/A(N,MDIAG)
0021          BB(IR) = BB(IR) -R*BB(N)
0022          DO 50 IC = IP,MH
0023          ICD = MBW + IC + 1 - IR
0024          ICB = MBW + IC + 1 -N
0025          A(IR,ICD) = A(IR,ICD) - R*A(N,ICB)
0026     50   CONTINUE
0027          DO 100 I = 1,NH
0028          N = NM - I + 1
0029          BB(N) = BB(N)/A(N,MDIAG)
0030          IL = N - MBW
0031          IH= N - 1
0032          IF (IL.GE.1) GO TO 60
0033          IL = 1
0034     60   DO 100 IR = IL,IH
0035          ICD = MBW + N + 1 - IR
0036          BB(IR) = BB(IR) -A(IR,ICD)*BB(N)
0037     100  CONTINUE
0038          BB(1) = BB(1)/A(1,MDIAG)
0039          RETURN
0040          END
```

```
0001
0002          SUBROUTINE FRIC(R,RX)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
0005          DIMENSION R(25),RX(25),A(25)
0006     C
0007     C              SUBROUTINE TO READ AND INTERPOLATE BOTTOM RESISTANCE
0008     C                   COEFFICIENT R
0009     C
0010     C                   RX = X DERIVATIVE OF R
0011     C
0012          READ(5,*) NF
0013          DO 5 I = 1,NF
0014          READ(5,*) A(I),XL(I)
0015     5    A(I) = A(I)*1.0E+05
0016          RMA = XL(NF)
0017          R(1) = XL(1)
0018          DO 20 N = 2,NN
0019          X = DX*FLOAT(N-1)
```

```
0020          IF (X.GT.A(NF)) GO TO 15
0021          IC = 0
0022          DO 8 J = 2,NF
0023          IF (IC.NE.0) GO TO 8
0024          I = J
0025          IF (X.GT.A(I)) GO TO 8
0026          IC = I
0027     8    CONTINUE
0028          IM = I - 1
0029          RQ = (XL(I) - XL(IM))/(A(I) -A(IM))
0030          XX = X - A(IM)
0031          R(N) = XL(IM) + XX*RQ
0032          GO TO 20
0033    15    R(N) = RMA
0034    20    CONTINUE
0035          RX(1) = (R(2) - R(1))/DX
0036          D2 = 2.0*DX
0037          NX = NN - 1
0038          DO 30 N = 2,NX
0039          IP = N + 1
0040          IM = N - 1
0041    30    RX(N) = (R(IP) - R(IM))/D2
0042          RX(NN) = 0.
0043          WRITE(6,901)
0044          WRITE(6,902) (R(I),I = 1,NN)
0045          WRITE(6,903)
0046    901   FORMAT(//'  R(CM/SEC)'/)
0047    902   FORMAT(10E12.5)
0048    903   FORMAT(///)
0049          RETURN
0050          END


0001
0002          SUBROUTINE FFCCAL(TX,TY,TXX,TYX,FFC,W,RL,DD2)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
0005          DIMENSION TX(25),TY(25),TXX(25),TYX(25)
0006          DOUBLE COMPLEX FFC(25),DCMPLX
0007    C
0008    C             SUBROUTINE TO CALCULATE WIND FORCING TERMS
0009    C
0010          FW = W*(F*F - DD2*W*W)
0011          FW = 1.0/FW
0012          M = MM
0013          DO 20 N = 1,NN
```

```
0014          CC = BV(N,M)*FW
0015          CR = DD2*(-W*TXX(N) + RL*F*TX(N))
0016          CI = F*TYX(N) -DD2*RL*W*TY(N)
0017          FFC(N) = DCMPLX(CR,CI)
0018    20    FFC(N) = CC*FFC(N)
0019          RETURN
0020          END


0001
0002          SUBROUTINE PEC(BX,SPE,IRL)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
0005          DOUBLE COMPLEX BX(425)
0006          DOUBLE PRECISION XINT,CDABS
0007    C
0008    C             SUBROUTINE TO CALCULATE THE POTENTIAL ENERGY ASSOCIATED
0009    C                   WITH FREE SURFACE ELEVATION
0010    C
0011          IF (IRL.EQ.0) GO TO 50
0012          GG = 1.0/(2.0*980.*1.03)
0013          XINT = 0.
0014          DO 20 N = 1,NN
0015          I = N*MM
0016          XX = DX
0017          IF (N.EQ.1) GO TO 5
0018          IF (N.EQ.NN) GO TO 5
0019          GO TO 20
0020    5     XX = DX/2.
0021    20    XINT = XINT + CDABS(BX(I))*CDABS(BX(I))*XX
0022          SPE = GG*XINT
0023          GO TO 60
0024    50    SPE = 0.
0025    60    WRITE(6,901) SPE
0026    901   FORMAT(/' FREE SURFACE POTENTIAL ENERGY =',E15.5)
0027          RETURN
0028          END


0001
0002          SUBROUTINE WRD(TX,TY,TXX,TYX,IXY)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
0005          DIMENSION TX(25),TY(25),TXX(25),TYX(25)
0006    C
```

```
0007    C            SUBROUTINE TO READ AND INTERPOLATE WIND STRESS PROFILES
0008    C
0009    C                  TX = X WIND STRESS
0010    C                  TXX = X DERIVATIVE OF TX
0011    C                  TY = Y WIND STRESS
0012    C                  TYX = X DERIVATIVE OF TY
0013    C
0014          READ(5,*) NRD
0015          IF (NRD.EQ.0) GO TO 100
0016          DO 5 I = 1,NRD
0017          READ(5,*) TXX(I),XL(I)
0018    5     TXX(I) = TXX(I)*1.0E+05
0019          RMA = XL(NRD)
0020          TX(1) = XL(1)
0021          DO 20 N = 2,NN
0022          X = DX*FLOAT(N-1)
0023          IF (X.GT.TXX(NRD)) GO TO 15
0024          IC = 0
0025          DO 8 J = 2,NRD
0026          IF (IC.NE.0) GO TO 8
0027          I = J
0028          IF (X.GT.TXX(I)) GO TO 8
0029          IC = I
0030    8     CONTINUE
0031          IM = I -1
0032          RQ = (XL(I) -XL(IM))/(TXX(I) -TXX(IM))
0033          XX = X - TXX(IM)
0034          TX(N) = XL(IM) + XX*RQ
0035          GO TO 20
0036    15    TX(N) = RMA
0037    20    CONTINUE
0038          GO TO 110
0039    100   DO 105 I = 1,NN
0040    105   TX(I) = 1.0
0041    110   TXX(1) = (TX(2) - TX(1))/DX
0042          D2 = 2.0*DX
0043          NX = NN - 1
0044          DO 30 N = 2,NX
0045          IP = N + 1
0046          IM = N - 1
0047    30    TXX(N) = (TX(IP) - TX(IM))/D2
0048          TXX(NN) = 0.
0049          IF (IXY.EQ.0) GO TO 140
0050          WRITE(6,901)
0051          GO TO 150
0052    140   WRITE(6,902)
```

```
0053          DO 145 I = 1,NN
0054          TY(I) = TX(I)
0055          TYX(I) = TXX(I)
0056          TX(I) = 0.0
0057    145   TXX(I) = 0.
0058          WRITE(6,903) (TY(I),I = 1,NN)
0059          GO TO 160
0060    150   DO 155 I = 1,NN
0061          TY(I) = 0.
0062    155   TYX(I) = 0.
0063          WRITE(6,903) (TX(I),I = 1,NN)
0064    901   FORMAT(' TAUX (DYNE/CM2)')
0065    902   FORMAT(' TAUY (DYNE/CM2)')
0066    903   FORMAT(10E12.5)
0067    160   RETURN
0068          END
```

```
0001
0002          SUBROUTINE AACAL(W,RL,R,RX,N,M,DD2,AA1,AA2,AA3,AA4,AA5)
0003          COMMON F,DX,DT,NN,MM,NM,NMX
0004          COMMON RH(25),RHX(25),RHXX(25),BV(25,17),BVZ(25,17),XL(600)
0005          DIMENSION R(25),RX(25)
0006          DOUBLE COMPLEX AA1,AA2,AA3,AA4,AA5,DCMPLX
0007    C
0008    C           SUBROUTINE TO CALCULATE COEFFICIENTS FOR MATS
0009    C
0010          CALL AS(A1,A2,A3,W,N,M,C1,C2,C3,DD2)
0011          CB = F*F + DD2*W*W
0012          CC = F*F - DD2*W*W
0013          CA = BV(N,M)/(CC**2)
0014          CH = (-RX(N)*C3-R(N)*C3*C3)*CB-R(N)*CC*CB*BVZ(N,M)*C1/(BV(N,M)**2)
0015          CH = -CA*(CH + 2.0*DD2*W*RL*F*R(N)*C3)
0016          CHH = W*(C1 + C2*C3)
0017          AA1 = DCMPLX(CH,CHH)
0018          CH = CA*2.0*DD2*W*RL*F*RX(N)
0019          CHH = C2*F*RL
0020          AA2 = DCMPLX(CH,CHH)
0021          CH = CA*CB*RX(N)
0022          CHH = W*C2
0023          AA3 = DCMPLX(CH,CHH)
0024          CH = CB*R(N)*C3*C3 + R(N)*CB*CC*C1*C1/BV(N,M)
0025          CH = -CA*CH
0026          CHH = 0.
0027          AA4 = DCMPLX(CH,CHH)
0028          CH = -CA*CB*R(N)*C3
```

```
0029            AA5 = DCMPLX(CH,CHH)
0030            RETURN
0031            END
```

# DOCUMENT LIBRARY

November 21, 1986

## *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
Colombia 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

Maury Oceanographic Library
Naval Oceanographic Office
Bay St. Louis
NSTL, MS 39522-5001
ATTN: Code 4601

50272-101

| REPORT DOCUMENTATION PAGE | 1. REPORT NO. WHOI-87-24 | 2. | 3. Recipient's Accession No. |
|---|---|---|---|

**4. Title and Subtitle**
**Programs for Computing Properties of Coastal-Trapped Waves and Wind-Driven Motions Over the Continental Shelf and Slope**
**-Second Edition-**

**5. Report Date** June 1987

**6.**

**7. Author(s)**
Kenneth H. Brink, and David C. Chapman

**8. Performing Organization Rept. No.** WHOI-87-24

**9. Performing Organization Name and Address**

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

**10. Project/Task/Work Unit No.**

**11. Contract(C) or Grant(G) No.**
(C)
(G) OCE 84-08563

**12. Sponsoring Organization Name and Address**

National Science Foundation

**13. Type of Report & Period Covered**
Technical

**14.**

**15. Supplementary Notes**

This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-87-24.

**16. Abstract (Limit: 200 words)**

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. This second edition includes several minor revisions and corrections in the computer code and the documentation.

**17. Document Analysis a. Descriptors**

1. coastal-trapped waves
2. wind-driven shelf currents
3. continental shelf

**b. Identifiers/Open-Ended Terms**

**c. COSATI Field/Group**

| 18. Availability Statement | 19. Security Class (This Report) | 21. No. of Pages |
|---|---|---|
| Approved for publication; distribution unlimited. | UNCLASSIFIED | 119 |
| | 20. Security Class (This Page) | 22. Price |

(See ANSI-Z39.18) *See Instructions on Reverse*

OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce