Documentation for program BIGSMPD: properties of seamount-trapped waves in a stratified ocean (January 10, 1989)
Modified 9/1/97, 10/10/97

FUNCTION:

This program computes the free wave "dispersion curves", modal structures and frictional decay times for subinertial frequency seamount-trapped waves in a stratified ocean. The major assumptions in the theory are that the equations of motion are linearized, the seamount has perfect azimuthal symmetry, a rigid-lid ocean surface, and that the bottom boundary layer is infinitesimally thin. You have to tell the program what the Coriolis parameter (f) is, what the shape of the seamount is, what the stratification is, and what some initial guesses at the wave frequency are. See, for more details, Brink (1989) (The effect of stratification on seamount-trapped waves, Deep-Sea Research, 36, 825-844).

PROGRAM OPERATION:

The program first computes filled-in profiles of buoyancy frequency squared, and of bottom topography, given user-provided information. Following this, the program solves for an arbitrarily forced seamount wave problem for pressure, and iterates on the frequency until a peak response (resonance) is found. This resonant frequency is the free wave frequency. The technique is usually called resonance iteration. For each iteration, the program prints out the estimated frequency, and a measure of the response strength. Searching for the peak is done by a combination of bisection and Newton's method.

Once the the free wave frequency is found, the program prints out that fact, and the damping decay time due to bottom friction. (see Brink, 1989). You also have the option to have it print out the modal structures of the mode for u,v,rho and p. (radial and azimuthal velocity, density and pressure). You can tell the program how many points you want to have calculated along a given "dispersion curve". You get to chose the nominal accuracy of the frequency of the solutions (variable EPS).

The problem is solved on an NN (number of radial grid points) by MM (number of vertical grid points) vertically stretched (sigma coordinate) finite difference grid. The grid size is determined by a PARAMETER statement which appears in each subroutine. Because the grid is finite, the true boundedness condition as r (radius) gets very large is replaced by

$$\frac{d(ru)}{dr} = 0$$

at a large distance (XMAX) from the seamount center. All units in the program are cgs, although inputs are variously in m, km or other convenient units.

The program's inputs come from unit INPF (set to 7 in the PARAMETER statements), and outputs go to unit IOPF (currently set to 8). In addition, there is a unit IGPF to receive condensed output for graphics, if you want it. Unit IGPF will not be used if IGG=0. If you use IGG not 0, then you will be asked, for each variable, whether you want graphics. Graphics can be done with matlab m-file "cstwplt.m" (contours) or "pstwplt.m" (colors).

The newer version, bigsmpd.for, is modified to be suitable for DOS use. The main difference is that it prompts you for the names of input and output files, and the uses an "open" statement to access them. In this use, INPF = 7, IOPF = 8. Differences otherwise are minor. "Bigsmp.for" is no longer maintained.

PROGRAM INPUTS:

You must prepare an input file which contains the following information, in the following order.

line 1: EPS, EST

EPS is the desired fractional accuracy for the free wave frequency. Typically, it might be 0.005.

EST is a fractional search increment in case the original estimates of the wave frequency is poor enough that it has to search around. Typically, EPS = 0.05.

line 2: ICCM, NCALM, NITM, ISD

ICCM is the number of dispersion curves to be calculated. Usually, I use ICCM =1. All of the following input has to be repeated for each dispersion curve.

NCALM is the number of points on the dispersion curve to be calculated.

NITM is the maximum number of iterations allowed to find the wave frequency. If it goes to more iterations than this, it quits. Typically, I use NITM = 30.

ISD determines the direction of the frequency search:

ISD = 0: it looks for the nearest frequency

ISD = 1: it searches only toward lower frequencies

ISD = -1: it searches only towards higher fequencies

line 3: IWS, IGG

If IWS=0, only an abbreviated printout of results is given.

If IWS not 0, a full output is given.

IGG = 0: no condensed output is printed to IGPF for graphics.

IGG not 0: a condensed output is printed to unit IGPF. I suggest staying with IGG = 0.

line 4: F,XMAX

F is the Coriolis parameter (sec-1) divided by 10**-5. That is, if you want f=1*10**-04sec-1, use F=10.

XMAX is the outer radius of the grid in km. It should be at least twice the radius of the seamount, preferably more.

line 5: NCAL, WH(1)

If you are continuing a dispersion curve, where you already know some of the values, this will help you pick it up in the middle. In this case, NCAL is the azimuthal wavenumber (an integer) of the last frequency calculated, and WH(1) is the last frequency computed, divided by 10**-6. For example, if the last thing you did was n=5, frequency = 1.0*10**-6sec-1, use NCAL=5, WH(1)=1.0.

If you are just starting a dispersion curve, use NCAL = 1, WH(1)=0.

line 6: NRF,NRD

NRF is the first azimuthal wavenumber (integer) for which you want to compute a frequency.

NRD is the increment in wavenumber. Thus, the second wavenumber used will be NRF+NRD, and the third wavenumber will be NRF + 2*NRD, etc.

line 7: WW(1),WW(2),WW(3)

These are three first estimates of the wave frequency corresponding to n=NRF. These are entered as frequency in sec-1 divided by 1.0*10.**-6. That is, if you want WW(1) = 2.0*10.**-6sec-1, enter 2.0 for the first value.

line 8: NRD (read from subroutine DEP)

This is the number of (radius, depth) pairs to be read in. This does not count the depth at the center of the seamount. For radius greater than that of the last pair read in, the last depth read in is used. This should assure that the bottom is flat at great distance from the seamount, as it should be. The last value of RHR should be less than XMAX.

line 9: RHZ

This is the depth at the center of the seamount in meters.

line 10 and following: RHR(I), RHRR(I) (NRD times)

These are pairs of radius (in kilometers) and water depth (in meters). NRD pairs are read in , each on its own line. The first value of RHR should be >0.

next line (read from subroutine NSQ):

NR, DZR, ALPH

NR is the number of buoyancy frequency squared values to be read in.

DZR is the depth increment of the values, in meters.

ALPH (kilometers) is the length scale of an expontial tail tacked onto the buoyancy frequency squared values. Typically, you only have values for the upper 1000m or so of the ocean. Values are extended to the bottom by attenuating the last value read by this exponential length scale.

next line: CMLT

This is a factor which multiplies the user-provided values of buoyancy frequency squared to get it into units of rad**2/sec**2. This lets you enter stratification in whatever units are convenient to you.

next NR lines: BX(I)

These are the values of the buoyancy frequency squared, in the user's units. The first value is for the surface, and the next represents a depth of DZR meters, etc.

SUMMARY OF INPUT FILE STRUCTURE:

```
EPS, EST                main program
ICCM, NCALM, NITM, ISD
IWS, IGG
F, XMAX
NCAL, WH(1)
NRF, NRD
WW(1),WW(2),WW(3)
```

```
      NRD                      subroutine DEP
      RHZ
      RHR, RHRR (NRD times)

      NR, DZR, ALPH                  subroutine NSQ
      CMLT
      BX(I) (NR times)
```

Note: do not actually leave any blank lines in the input file.

INTERPRETING THE OUTPUT:

The output is fairly self-explanatory. There are a few comments:

1) All units of output are cgs.

2) Modal structures have arbitrary amplitudes, put are mutually consistent. For example, if you take pressure to be in dynes/cm2, then velocities are in cm/sec and density in gm/cm3.

3) Outputs of modal structures are given in "sigma coordinates". That is the delta(z) varies with the depth of the water. Thus, for each radius, a different delta(z) is given. Values are given from the bottom to the top of the water column. The first value will be at the bottom, and the last will be at the surface. The header for each radius gives the depth increment. You can actually get used to reading these outputs!

4) The frictional decay time is given as (delta*TF), where
    delta is the proportionality constant between bottom velocity and stress, e.g. stress = delta*velocity. It has units of cm/sec.
    TF is the frictional decay time (e-folding).

   If you want to get the actual frictional decay time in sec, divide delta*TF by your value of delta.

   Recall that the estimate of TF is only valid if (wave frequency*TF)>>1 .

DISCLAIMER:

I have tried hard to make this program as good and as flexible as possible. I can not guarantee that it is perfect, though. User beware!

Ken Brink