# bwavesp.m

## Stable barotropic coastal trapped wave modes: edge, shelf and Kelvin waves

**K.H. Brink**

March, 2020,
August 1, 2020

This set of Matlab mfiles (all with names beginning with "bwavesp") can be used to calculate barotropic coastal wave properties in the absence of density stratification. The wave frequency is taken to be entirely real (hence stable). You are allowed to have a mean alongshore flow, if desired, and you can apply the rigid lid and/or coastal long wave approximations. The model can be run in the non-rotating limit if desired. Once a wave's frequency is found, the modal structure is displayed, and a perturbation (weak friction) imaginary correction to the wave frequency is found. The code can use an exact open boundary condition or a closed condition at either side of the domain.

Although there are several mfiles in this package, there are really only three that the user is likely ever to use directly:

**bwavespsetup.m** This file asks the user a sequence of questions, and creates an array which summarizes the answers and is then used to drive the main code.

It is called as:

```
>> arrayin = bwavespsetup;
```

where arrayin is the resulting driver array.

**bwavespfinch.m** This file is used to change the input array without having to re-enter all the material. When you call it, it will first give a menu asking what sort of thing needs to be changed, and will then ask more specific questions.

It is called as:

```
>> newarrayin = bwavespfinch(arrayin);
```

where "newarrayin" is the revised input array, and "arrayin" is the original input array.

**bwavesp.m** This is the central file called in order to carry out the wave calculations. It calls several other mfiles for specific tasks.

It is called as:

>>bwavesp(arrayin,name);

where “arrayin” is the input array from bwavespsetup.m or bwavespfinch.m. The second input, name, is a string variable that would be used to name an output file. Outputs are displayed on your screen, and you are given the option of saving your results to a mat file “name”.

A good deal more detail, including examples, is provided at the end of this document.

## The Problem:

We seek coastal-trapped wave solutions (both sub-and superinertial) for an ocean without stratification. Specifically, we consider a straight boundary at $x = 0$, where the depth $h(x)$ varies offshore (toward larger $x$). All isobaths are parallel to the shoreline. Either open or closed boundary conditions are possible at either boundary. The effects of friction are confined to infinitesmally thin boundary layers. Frictional effects are considered only as small perturbations to inviscid solutions, or in conjunction with the first order wave equation coefficients. Further, there is a steady mean alongshore flow, $v_0(x)$.

The depth-integrated equations of motion with no bottom friction are

$$\varepsilon U_t + \varepsilon v_0 U_y \qquad - fV = -\frac{1}{\rho_0} p_x \tag{1a}$$

$$V_t + v_0 V_y + V v_{0x} \qquad + fU = -\frac{1}{\rho_0} p_y \tag{1b}$$

$$\delta \frac{1}{g\rho_0}\left(p_t + v_0 p_y\right) + U_x + V_y = 0 \quad . \tag{1c}$$

where ($U$, $V$) is the vector of depth-integrated (offshore, alongshore) velocity and $p$ is the pressure. Constants $f$, $g$ and $\rho_0$ are the Coriolis parameter, the acceleration due to gravity and the uniform water density. Subscripts with regard to independent variables ($x$, $y$, $t$) represent partial differentiation.

When $\varepsilon$ = 0, the coastal long wave approximation (e.g., Gill and Schumann, 1974) is applied so that the alongshore flow is in geostrophic balance. This approximation is valid when

$$|lL| << 1 \quad , |\omega/f| \; << 1 \text{ and } \left|\frac{r}{fh}\right| \ll 1 \tag{2}$$

where $l$ is an alongshore wavenumber, $\omega$ is a typical frequency, $r$ is a frictional coefficient (see below) and $L$ is a representative cross-shelf scale (like the width of the region where the bottom is sloping). Obviously, this assumption cannot be used when $f = 0$. Otherwise, $\varepsilon = 1$ (“general frequency and wavelength”).

When $\delta = 0$, the rigid lid approximation is enforced. This assumption is valid when

$$\frac{gH}{f^2 L^2} \gg 1 \quad , \tag{3}$$

where $H$ is a representative depth. Otherwise, $\delta = 1$ (free surface boundary condition).

In order to proceed, all dependent variables are taken to have a form like

$$p(x,y,t) = P(x)\exp[i(\omega t + ly)] \quad . \tag{4}$$

Using this simplification, a single equation for pressure is obtained:

$$0 = \left(\frac{h}{\gamma} P_x\right)_x + P\left[-\frac{\delta}{g} - \varepsilon\frac{l^2}{\gamma} h + \frac{fl}{\omega''}\left(\frac{h}{\gamma}\right)_x\right] \tag{5}$$

where

$$\gamma = ff'' - \varepsilon\omega''^2 \tag{6a}$$

$$f'' = f + v_{0x} \tag{6b}$$

$$\omega'' = \omega + lv_{ox} \quad . \tag{6c}$$

This problem can be solved with either open or closed boundary conditions at $x = 0$ and $x = x_{Max}$. The closed condition is

$$U = 0 \tag{7a}$$

so that

$$0 = \omega'' P_x + flP \quad . \tag{7b}$$

The open boundary condition is exact, but it requires that, at the boundary,

$$\begin{aligned} v_{0x} &= 0 \\ v_{0xx} &= 0 \\ h_x &= 0 \quad . \end{aligned} \tag{8}$$

When this is true, equation (5) has constant coefficients, and the solution $P_0$ for outside the boundaries of the grid is

$$P_0(x) = A\ \exp[-\beta_B |\ x - x_B|] \tag{9a}$$

where

$$\beta_B = \sqrt{\frac{\delta\gamma}{g h_B} + \varepsilon l^2} \tag{9b}$$

and $h_B$ is the water depth at the open boundary, $x = x_B$ ($x_B = 0$ and /or $x_{Max}$), and $A$ is a constant. Thus, the open boundary condition becomes

$$Px = \beta_B P(0) \quad , h_B = h(0) \tag{9c}$$

when applied at $x = 0$ and

$$Px = -\beta_B P(x_{Max}) \quad , h_B = h(x_{Max}) \tag{9d}$$

when applied at $x = x_{Max}$.

One caution is required. When

$$\omega^2 > f^2 + c_0^2 l^2 \quad , \tag{10}$$

(where $c_0^2 = g h_M$ and $h_M$ is the maximum water depth), there is a continuum of onshore-offshore propagating solutions, and no trapped modes exist.

The sorts of waves that can be treated with this software are summarized in a schematic diagram (Figure 1). See Huthnance (1975) for a thorough discussion. For $\omega < f$, there can only be trapped waves, and these fall into two categories. First, there is a rapidly propagating (speed often of order $c_0$) Kelvin wave, which has dynamics similar to a long gravity wave and yet has alongshore flow in essentially geostrophic balance. Further, there is an infinite (as long as topography is smooth) set of trapped topographic Rossby waves which propagate more slowly and are dispersive for larger wavenumbers. These are called barotropic continental shelf waves, or simply shelf waves. Higher shelf wave modes propagate more slowly and have an increasing number of zero crossings in the pressure modal structure $P(x)$. All of these waves propagate phase in the $-y$ direction when $f > 0$ (i.e., to the south off the east coast of the United States, or to the North off the west coast). The sense of phase propagation reverses in the southern hemisphere.

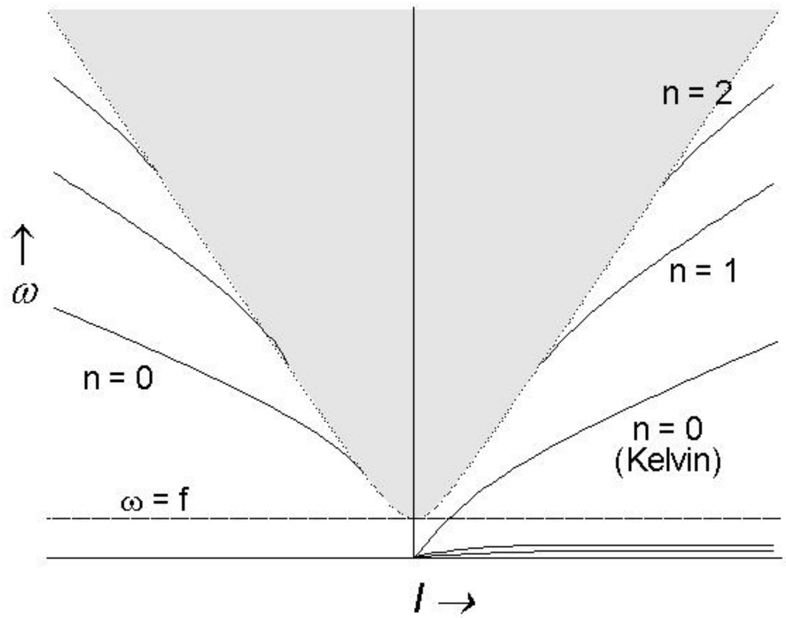



Figure 1: Schematic dispersion curves for the sorts of waves that can be resolved with this software. This plot is for $f > 0$, a closed boundary at the coast and an open boundary offshore where the water is deepest. The shaded area represents the continuum where there are no trapped waves. The first two shelf waves are plotted in the region $\omega < f$, $l > 0$.

For $\omega > f$, there are infinite sets of edge waves, which propagate in both the positive and negative $y$ directions. These waves are essentially long gravity waves trapped in shallow water because, for a given $l$, the wave cannot propagate once the water becomes deep enough. Higher order edge waves have higher frequencies and an increasing number of zero crossings in their modal structure. One interesting aspect of this collection involves the near-Kelvin wave (the only dispersion curve that crosses $\omega = f$ in Figure 1). Regardless of the frequency, the modal structure has no zero crossings in $x$, but its nature does change with increasing frequency or wavenumber. Specifically, for smaller $\omega$, including for $\omega = O(f)$, the wave has very large offshore scales as is the case for a traditional barotropic Kelvin wave in a flat-bottom ocean. However, for larger $\omega$ or $l$, the cross-shelf scale decreases and the wave becomes an $n = 0$ edge wave. I take $n$ to be the number of zero crossings in the pressure modal function.

**Bottom friction:** All of the free wave calculations done by this code use the inviscid equations described above. However, bottom friction is accounted for by including frictional effects for calculations in the coastal long wave limit (e.g., Brink 1989). Further, an estimate for frictional decay times is made for all calculations (not just long wave) by following the free-wave frictional damping perturbation analysis of Brink (1990). These estimates are only valid for weak frictional effects. Other calculations (Power et al., 1989, Brink, 2006) demonstrate that finite-amplitude dissipation changes wave modal structures so as to make small-amplitude estimates inaccurate.

To summarize, the Brink (1990) analysis begins by assuming that the bottom stress is given by $(\tau_B^x, \tau_B^y) = \rho_0\, r\, (u, v)$ where $(\tau_B^x, \tau_B^y)$ is the bottom stress vector, $r$ is a bottom resistance coefficient and $(u, v)$ is the interior velocity near the bottom. It is assumed that $(u, v) = (U, V)/h$. Thus, the governing equations are

$$\varepsilon U_t + \varepsilon v_0 U_y \qquad - fV = -\frac{1}{\rho_0} p_x - \varepsilon \frac{r}{h} U \tag{11a}$$

$$V_t + v_0 V_y + V v_{0x} \qquad + fU = -\frac{1}{\rho_0} p_y - \frac{r}{h} V \tag{11b}$$

$$\delta \frac{1}{g\rho_0}\left(p_t + v_0 p_y\right) + U_x + V_y = 0 \quad . \tag{11c}$$

These are then reduced to

$$0 = \left(\frac{h}{\gamma} P\right)_x - i\left(\frac{r}{h}\right)_x \frac{h}{\gamma\omega''} P_x + P\left[-\frac{\delta\omega''}{g\omega'} - \varepsilon \frac{l^2}{\gamma} h + \frac{fl}{\omega''}\left(\frac{h}{\gamma}\right)_x\right] \tag{12}$$

subject to boundary conditions analogous to (7) and (9) and where

$$\omega' = \omega + l v_0 - i\frac{r}{h} \quad . \tag{13}$$

At this point, it is assumed

$$P = P_0 + P_1 \quad , \omega = \omega_0 + \omega_1 \tag{14}$$

where variables subscripted “1” are small relative to the inviscid solutions subscripted “0”. It then follows that

$$I_1 \omega_1 = i I_2 \tag{15a}$$

where, for example (for the case with a closed boundary condition at $x = 0$ and open boundary condition at $x = x_{Max}$),

$$I_1 = P_0^2 \frac{hfl}{\gamma_0}\left(\frac{1}{{\omega_0}''^2} - \frac{2\varepsilon}{\gamma_0}\right)\Big|_{x=0}$$

$$+2\varepsilon \int \frac{{\omega_0}'' h}{\gamma_0^2}\left(P_{0x}^2 + \varepsilon l^2 P_0^2\right) dx + fl \int \frac{1}{{\omega_0}''^2}\left(\frac{h}{\gamma_0}\right)_x P_0^2\, dx \tag{15b}$$

$$-2\varepsilon \int \frac{1}{{\omega_0}''}\left(\frac{{\omega_0}'' h}{\gamma_0^2}\right)_x P_0^2\, dx$$

and

$$I_2 = \int \left(\frac{P_0}{\omega_0''}\right)_x \frac{r}{\gamma_0} P_{ox}\, dx \quad + \quad \varepsilon l^2 \int \frac{r}{\omega_0'' \gamma_0^2} (ff' + \varepsilon \omega_0''^2) P_0^2 dx$$

$$+2\varepsilon f l \int \frac{\omega_0'' r}{\gamma_0^2} \left(\frac{P_0^2}{\omega_0''}\right)_x dx \quad + \quad 2\varepsilon \int \frac{\omega_0'' r}{\gamma_0^2} P_0^2 dx \qquad (15c)$$

where all of the integrals range from $x_L$ to $x_H$ where $x_L$ is either -∞ or 0 and $x_H$ is either $x_{Max}$ or ∞, depending on whether boundary conditions are open or closed. The version of this in the mfile accounts for all the different possibilities for boundary conditions. Also,

$$\omega_0'' = \omega_0 + l v_0 \quad , \qquad (15d)$$

$$\gamma_0 = ff' - \varepsilon \omega_0''^2 \quad . \qquad (15e)$$

Note that (15c) differs from (25c) in Brink (1990) in two ways. One is that this formulation allows for $r$ to vary in $x$ (which is why it appears inside the integrals). Second, this statement corrects a typo (missing factor of $\omega_0''$ ) in the first term in Brink's equation (25c). Finally, since $\omega_1$ is imaginary, it represents damping on a time scale of $T_f = i/\omega_1$. In the model outputs, the results of this calculation are denoted by $1/T_f$ so as to avoid any confusion about the factor of $i$.

## How the software works:

For given $f$, $h$, $v_0$, alongshore wavenumber $l$ and boundary conditions, equation (5) is solved, and $\omega_0$ are found via resonance iteration. Specifically, an arbitrary forcing is applied to the left hand side of (5), and the frequency is varied in search of a maximum for $P$ response. The process is complete once $\omega$ has converged to a fractional accuracy set by "acc" in the input array. The responses is measured by

$$R = \int h P^2 dx \quad . \qquad (16)$$

The actual search is carried out using the Matlab function "fminsearch", searching for a minimum of $1/R$.

Initially, the program plots out $v_0$, $r$ and $h$ and outputs statements describing boundary conditions, $f$, number of grid points, etc. The code also checks the fields of $h(x)$ and $v(x)$ to make sure that they are consistent with the chosen boundary conditions (eqns. 8). Checks are also done for necessary conditions for instability. Once the solution (optimal $\omega$ and $P$) is found, the pressure field is normalized (as in eqn, 16) and the frictional damping time $T_f$ is calculated. If the coastal long wave assumption has been made, the pressure field is re-normalized according to the energy-conserving long wave norm (Brink, 1989) and the wind coupling $b_n$ and bottom friction

$a_{nn}$ coefficients are also computed. When a dispersion curve is calculated, the curve is plotted as it develops, along with the evolving modal structure. When the program is done, it also plots out $T_f$ as a function of alongshore wavenumber.

Finally, when all calculations are completed, the user is given the option of saving results to a file with the name provided in the call statement.

## Using the software:

Examples of usage are attached as an appendix to this document. All of the mfiles used by this package have names beginning with "bwavesp….". In practice, one first calls "bwavepsetup" in order to create an input array.

```
>>test = bwavespsetup;
```

This setup code asks a number of questions to obtain needed information about model assumptions, grid size and resolution, topography, mean flow and bottom friction. It asks for a first guess frequency corresponding to the first wavenumber to be considered. In the end, an array ("test" in this example) is then generated.
Next, one calls the main routine.

```
>> bwavesp(test,'title');
```

where 'title' is a string that (if desired) is used to create an output file like "title.mat". At the beginning of the run, a plot is given of $v_0$, $r$ and $h$ and some information is output regarding assumptions, $f$ and grid. Also, if the mean flow is non-zero, the code checks for whether ($f + v_{0x}$) changes sign in the domain and/or whether the necessary condition for barotropic instability is met. Then the code begins to search for the resonant frequency corresponding to the first wavenumber. Information about the search is provided on the screen. Once a resonance is found, the results are plotted, and an estimate is made for the first-guess frequency for the next wavenumber. Final plots include a dispersion curve and a plot of $1/T_f$ vs wavenumber.

## Some notes on using the software:

A good deal of caution is required near the inertial frequency, where (when $v_0 = 0$) a spurious solution exists (section 3.9 of Pedlosky, 1979; Dale, Sherwin and Huthnance, 2001). The solution obeys equations 5, 7 and 9, but does not satisfy the original equations of motion (equations 1). Further, the presence of this spurious mode deflects nearby dispersion curves. The frequency range over which this distortion occurs was shown by Dale et al. to depend on the grid resolution: the finer the resolution, the less problem you will have. I have found that for reasonable $x$ grid resolution (say, 1 km for a realistic shelf), the zone of influence is less than about $0.01f$ wide in frequency. However, sometimes a computed dispersion curve will find the spurious mode, and then follow it rather than a physically correct mode. When this happens, the sudden turn of the dispersion curve to follow $\omega = f$ is very obvious, and the offending solution can be rejected. One can usually keep the computed dispersion curves from "jumping" by using finer resolution in alongshore wavenumber and/or demanding more accurate calculations.

If you are interested in waves having $|\omega/f| > 1$ (e.g., edge waves and higher frequency Kelvin waves), you really should use the free surface boundary condition. Also, the coastal long wave assumption cannot be used in this case.

If the $x = 0$ or $x = x_{Max}$ boundary is open, the bottom should be flat and $v_{0x} = 0$ at the boundary. In the case of an open boundary condition, the water does not have to be shallow.

Although the coastal long-wave limit is well defined when the mean alongshore flow is non-zero, the resulting wave modes are generally not orthogonal. This means that you cannot compute the first order wave equations coefficients ($a_{nn}$, $b_n$, $c_n$) when there is a mean flow. Thus, you are not allowed to use the coastal long-wave normalization and coefficient routine when a mean flow is present. If you make the coastal long-wave approximation, you will only be allowed to calculate one point on the dispersion curve because the long waves are dispersive (constant phase speed with wavenumber).

Note that, for depth, mean velocity and bottom friction, the requested input (in bwavespsetup or bwavespfinch) can call for an array of *x* locations. The array *x* array does not have to start with zero, and it can be as short as only one element. The software will fill in any gaps you may leave near the boundaries. However, the array of *x* locations must have monotonically increasing values, e.g. [10 20 30 40]. It cannot be decreasing (e.g. [40 30 20 10]), and it has to be monotonic (e.g., it cannot be [10 30 20 40]).

The search can be very inefficient if you give it 0 as the starting guess for frequency. Better to give it a nonzero value. By the same token, if you give the algorithm an initial estimate for frequency that has few places of accuracy (say 1e-5), the Matlab search code will look around at fairly large increments (perhaps ± 20%), and the search can miss nearby solutions. If you provide more places of accuracy (say 1.01e-5), the search will be initially confined to more nearby locations (perhaps ± 3%), and it becomes much more likely that nearby solutions will be found.

The search can sometime miss a frequency.

If the model obtains a good solution, the inverse resonance parameter in the output (rni) should be several orders of magnitude lower than neighboring values. If this is not true, you may either have a bad solution, or you may only be finding the real part of a complex wavenumber (which can happen when the true frequency is complex, i.e., when the true solution with a mean flow is unstable). Using a finer accuracy tolerance (acc) will tend to eliminate false solutions.

For an accuracy estimate ("e" option in bwavespfinch.m), I find that 0.0001 works out fairly well. This means frequency accuracy to 0.01% of the absolute value of the initial frequency guess.

The bottom stress is assumed to be proportional to the bottom velocity. For the errors associated with this assumption in the presence of a mean flow, see Brink (1997).

All frequencies are in radians/sec and all wavenumbers are in radians/cm. Because radians are nondimensional, this could also be written as "1/sec" etc.

If the code is searching for a trapped wave in the continuum frequency range (gray area in Figure 1), this will be obvious because the displayed resonance estimator rni is complex in the outputs. If no resonance with real rni is found, the best strategy is to increase the magnitude of the wavenumber (so as to escape the continuum: see Figure 1) and try again.

In some cases, the structure of the forcing (a spike at mid-grid) will be obvious in the modal structure. This is a failing that typically happens when the forcing does not occur anywhere near (in $x$) to where the peak amplitude of the wave occurs. Improving grid resolution does not seem to help. Increasing the required fractional accuracy for frequency does help some. Alternatively, one can go into the bwavespsol.m file, near line 36, and change nf so that the forcing occurs closer (in $x$) to where the wave's modal structure has large amplitude.

The water depth $h$ must always be > 0, including at the boundaries. The depth at a coastal wall can be made extremely small, but it cannot vanish.

When the mean alongshore flow is nonzero, it is possible to have critical layer (where $\omega''$ passes through zero) "solutions". These are singular solutions that the present code, inviscid at lowest order, cannot deal with. A warning is given when such solutions occur.

## Disclaimer:

Although considerable effort has been made to make sure this software is correct and easy to use, there is no guarantee of perfection. If errors are found, or if documentation could be improved, please contact kbrink@whoi.edu .

## References:

Brink, K.H., 1989: Energy conservation in coastal-trapped wave calculations. *J. Phys. Oceanogr.,* 19, 1011-1016.

Brink, K.H., 1990: On the damping of free coastal-trapped waves. *J. Phys. Oceanogr.*, 20, 1219-1225.

Brink, K. H., 1997. Time-dependent motions and the nonlinear bottom Ekman layer. *Journal of Marine Research*, **55**(4), 613–631.

Brink, K. H., 2006. Coastal-trapped waves with finite bottom friction. *Dynamics of Atmospheres and Oceans*, **41**, 172-190.

Dale, A.C., J.M. Huthnance, and T.J. Sherwin, 2001: Coastal-Trapped Waves and Tides at Near-Inertial Frequencies. , **31**, 2958–2970.

Gill, A.E. and E.H. Schumann, 1974: The generation of long shelf waves by the wind. J. Phys. Oceanogr., 4, 83-90.

Huthnance, J.M., 1975: On trapped waves over a continental shelf. J. Fluid Mech., 69. 689-704.

Pedlosky, J., 1979: Geophysical Fluid Dynamics, Springer-Verlag, New York, 624pp.

Power, S. B., J. H. Middleton, and R. H. J. Grimshaw, 1989: Frictionally modified continental shelf waves and the subinertial response to wind and deep-sea forcing. J. Phys. Oceanogr., 19, 1486–1506, https://doi.org/10.1175/1520-0485(1989)019,1486: FMCSWA.2.0.CO;2.

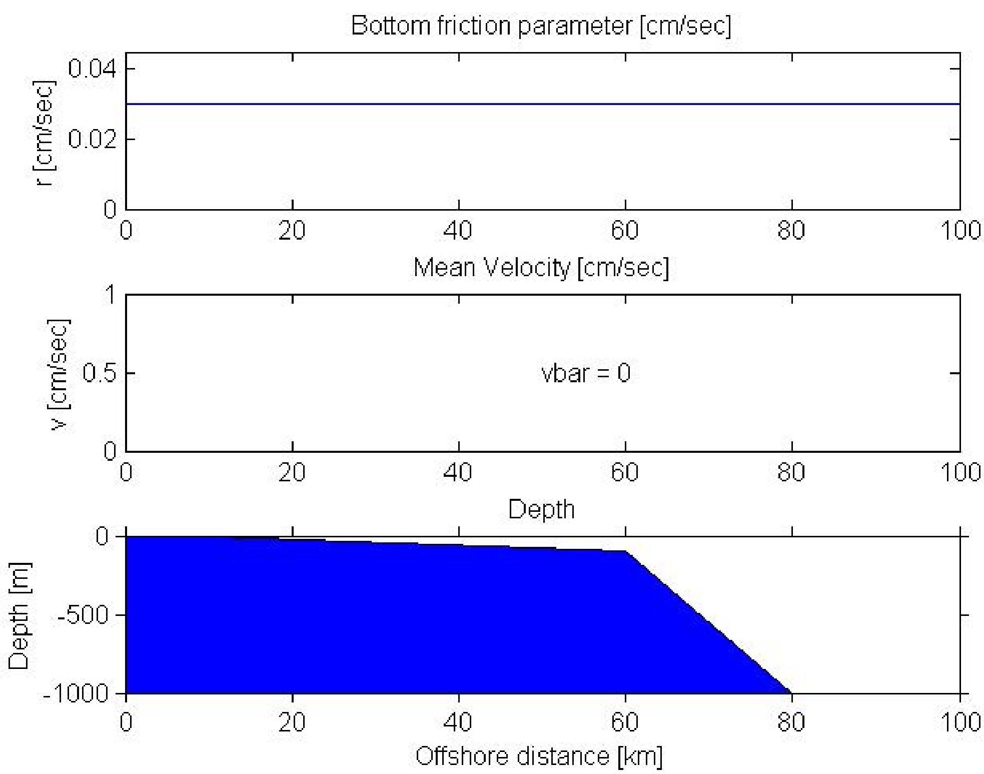
## Sample: Running bwavespsetup.m

*The following is a copy of what you see on the screen as you run bwavespsetup.m.*

```
>> ardemo = bwavespsetup;

   Barotropic coastal-trapped waves with real frequency

This mfile will ask you a sequence of questions
   that are used to build the input array.

How many total gridpoints do you want in the cross shelf direction? (nn) 100
Enter the domain width (W) (km) 100
Enter the nominal fractional accuracy for the solution (acc) 1e-4
Enter 0 for a rigid lid, 1 for a free surface (del) 1
Enter 0 for coastal long wave, 1 for general frequency and wavenumber (eps) 1
```

Enter 0 for closed boundary at x = 0, 1 for open boundary (icbc) 0
Enter 0 for closed boundary at x = x_max, 1 for open boundary (iobc) 1
Enter the Coriolis parameter (f) (1/sec) 1e-4
Enter the number of frequencies to be computed (nw) 5
Enter the first alongshore wavenumber to use (rlz) (1/cm) 1e-9
Enter the wavenumber increment to use after rlz (drl) (1/cm) 1e-8
First guess at real part of frequency (1/sec)? 1e-7
How many distance, depth pairs will you provide (ndep >=1) 3
Array of offshore distances for depth values (xdep in km) (dimension ndep) [10 60 80]
Array of depths corresponding to xdep (depr in m) [10 100 1000]
How many distance, mean v pairs will you provide (nv >= 0 ) 0
Number of distance, bottom friction pairs to read (nr) 1
Offshore distances for bottom friction values (xr in km) 0
Array of bottom friction values corresponding to xr (rr in cm/sec) 0.03
>>

*At this point, the following plot appears in a plot window:*




**Sample: running bwavesp.m**

*The following is a copy of what you see on the screen as you run bwavesp.m*

>> bwavesp(ardemo,'DemoFile')
f = 0.0001 1/sec
Free surface
General frequency and wavenumber
Closed boundary at x = 0
Open boundary at x = x_Max
100 grid points in x

rl, w, rni = 1e-09, 1e-07,  1.3665e-42
rl, w, rni = 1e-09, 1.05e-07,  1.1138e-43
rl, w, rni = 1e-09, 1.1e-07,  1.476e-43
rl, w, rni = 1e-09, 1.075e-07,  1.5323e-45
rl, w, rni = 1e-09, 1.1e-07,  1.476e-43
rl, w, rni = 1e-09, 1.0625e-07,  2.0651e-44
rl, w, rni = 1e-09, 1.0875e-07,  4.6238e-44
rl, w, rni = 1e-09, 1.0688e-07,  2.6414e-45
rl, w, rni = 1e-09, 1.0813e-07,  1.6369e-44
rl, w, rni = 1e-09, 1.0719e-07,  3.4872e-47
rl, w, rni = 1e-09, 1.0688e-07,  2.6414e-45
rl, w, rni = 1e-09, 1.0734e-07,  2.7804e-46
rl, w, rni = 1e-09, 1.0703e-07,  8.1766e-46
rl, w, rni = 1e-09, 1.0727e-07,  2.9142e-47
rl, w, rni = 1e-09, 1.0734e-07,  2.7804e-46
rl, w, rni = 1e-09, 1.0723e-07,  6.2514e-50
rl, w, rni = 1e-09, 1.0719e-07,  3.4872e-47
rl, w, rni = 1e-09, 1.0725e-07,  6.6307e-48
rl, w, rni = 1e-09, 1.0721e-07,  9.4666e-48
rl, w, rni = 1e-09, 1.0724e-07,  1.3519e-48

rl (cm^-1), w (sec^-1), 1/T_f (sec^-1) = 1e-09,  1.0723e-07,  1.8769e-05

rl, w, rni = 1.1e-08, 1.1795e-06,  6.9531e-47
rl, w, rni = 1.1e-08, 1.2385e-06,  5.235e-43
rl, w, rni = 1.1e-08, 1.1205e-06,  6.9819e-43
rl, w, rni = 1.1e-08, 1.209e-06,  1.4508e-43
rl, w, rni = 1.1e-08, 1.15e-06,  1.5722e-43
rl, w, rni = 1.1e-08, 1.1942e-06,  3.9407e-44
rl, w, rni = 1.1e-08, 1.1647e-06,  3.6067e-44
rl, w, rni = 1.1e-08, 1.1721e-06,  8.0568e-45
rl, w, rni = 1.1e-08, 1.1869e-06,  1.0903e-44
rl, w, rni = 1.1e-08, 1.1758e-06,  1.6367e-45
rl, w, rni = 1.1e-08, 1.1832e-06,  3.2067e-45
rl, w, rni = 1.1e-08, 1.1776e-06,  2.5585e-46

rl, w, rni = 1.1e-08, 1.1813e-06,  1.0593e-45
rl, w, rni = 1.1e-08, 1.1786e-06,  1.4536e-47
rl, w, rni = 1.1e-08, 1.1776e-06,  2.5585e-46
rl, w, rni = 1.1e-08, 1.179e-06,  5.1391e-48
rl, w, rni = 1.1e-08, 1.1795e-06,  6.9531e-47
rl, w, rni = 1.1e-08, 1.1788e-06,  5.957e-49
rl, w, rni = 1.1e-08, 1.1786e-06,  1.4536e-47
rl, w, rni = 1.1e-08, 1.1789e-06,  5.5922e-49

rl (cm^-1), w (sec^-1), 1/T_f (sec^-1) = 1.1e-08,  1.1789e-06,  1.8768e-05

rl, w, rni = 2.1e-08, 2.2506e-06,  3.3136e-46
rl, w, rni = 2.1e-08, 2.3631e-06,  5.4098e-43
rl, w, rni = 2.1e-08, 2.1381e-06,  6.8607e-43
rl, w, rni = 2.1e-08, 2.3069e-06,  1.5357e-43
rl, w, rni = 2.1e-08, 2.1943e-06,  1.5045e-43
rl, w, rni = 2.1e-08, 2.2225e-06,  3.2635e-44
rl, w, rni = 2.1e-08, 2.2787e-06,  4.366e-44
rl, w, rni = 2.1e-08, 2.2365e-06,  6.431e-45
rl, w, rni = 2.1e-08, 2.2647e-06,  1.3127e-44
rl, w, rni = 2.1e-08, 2.2436e-06,  9.4491e-46
rl, w, rni = 2.1e-08, 2.2576e-06,  4.4408e-45
rl, w, rni = 2.1e-08, 2.2471e-06,  3.8494e-47
rl, w, rni = 2.1e-08, 2.2436e-06,  9.4491e-46
rl, w, rni = 2.1e-08, 2.2488e-06,  3.6185e-47
rl, w, rni = 2.1e-08, 2.2506e-06,  3.3136e-46
rl, w, rni = 2.1e-08, 2.248e-06,  8.1912e-51
rl, w, rni = 2.1e-08, 2.2471e-06,  3.8494e-47
rl, w, rni = 2.1e-08, 2.2484e-06,  8.782e-48
rl, w, rni = 2.1e-08, 2.2475e-06,  9.9001e-48
rl, w, rni = 2.1e-08, 2.2482e-06,  2.0641e-48

rl (cm^-1), w (sec^-1), 1/T_f (sec^-1) = 2.1e-08,  2.248e-06,  1.8767e-05

rl, w, rni = 3.1e-08, 3.317e-06,  2.9719e-46
rl, w, rni = 3.1e-08, 3.4829e-06,  5.4514e-43
rl, w, rni = 3.1e-08, 3.1512e-06,  6.9505e-43
rl, w, rni = 3.1e-08, 3.3999e-06,  1.5437e-43
rl, w, rni = 3.1e-08, 3.2341e-06,  1.5288e-43
rl, w, rni = 3.1e-08, 3.2756e-06,  3.3369e-44
rl, w, rni = 3.1e-08, 3.3585e-06,  4.3682e-44
rl, w, rni = 3.1e-08, 3.2963e-06,  6.6718e-45
rl, w, rni = 3.1e-08, 3.3378e-06,  1.3024e-44
rl, w, rni = 3.1e-08, 3.3067e-06,  1.0217e-45

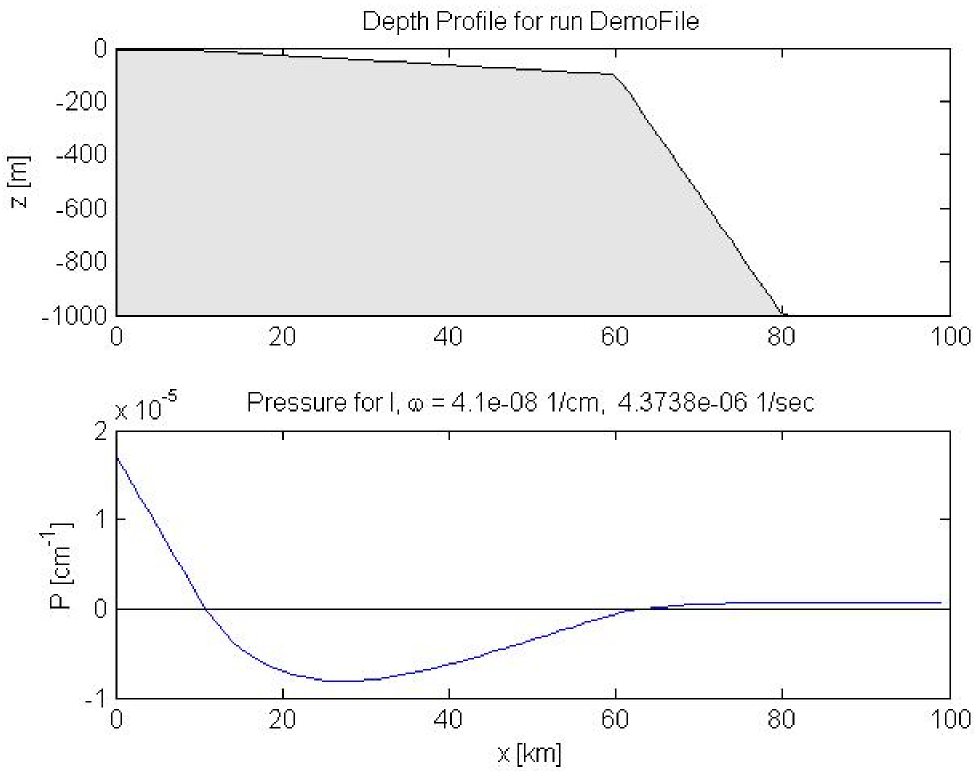
```
rl, w, rni = 3.1e-08, 3.3274e-06,  4.3471e-45
rl, w, rni = 3.1e-08, 3.3118e-06,  5.3268e-47
rl, w, rni = 3.1e-08, 3.3067e-06,  1.0217e-45
rl, w, rni = 3.1e-08, 3.3144e-06,  2.4863e-47
rl, w, rni = 3.1e-08, 3.317e-06,  2.9719e-46
rl, w, rni = 3.1e-08, 3.3131e-06,  1.3274e-48
rl, w, rni = 3.1e-08, 3.3118e-06,  5.3268e-47
rl, w, rni = 3.1e-08, 3.3138e-06,  3.679e-48
rl, w, rni = 3.1e-08, 3.3125e-06,  1.7845e-47
rl, w, rni = 3.1e-08, 3.3135e-06,  1.4687e-49

rl (cm^-1), w (sec^-1), 1/T_f (sec^-1) = 3.1e-08,  3.3135e-06,  1.8765e-05


rl, w, rni = 4.1e-08, 4.3789e-06,  3.1464e-46
rl, w, rni = 4.1e-08, 4.5979e-06,  5.516e-43
rl, w, rni = 4.1e-08, 4.16e-06,  7.0192e-43
rl, w, rni = 4.1e-08, 4.4884e-06,  1.5637e-43
rl, w, rni = 4.1e-08, 4.2695e-06,  1.5423e-43
rl, w, rni = 4.1e-08, 4.3242e-06,  3.3586e-44
rl, w, rni = 4.1e-08, 4.4337e-06,  4.4328e-44
rl, w, rni = 4.1e-08, 4.3516e-06,  6.6784e-45
rl, w, rni = 4.1e-08, 4.4063e-06,  1.3259e-44
rl, w, rni = 4.1e-08, 4.3653e-06,  1.007e-45
rl, w, rni = 4.1e-08, 4.3926e-06,  4.4485e-45
rl, w, rni = 4.1e-08, 4.3721e-06,  4.807e-47
rl, w, rni = 4.1e-08, 4.3653e-06,  1.007e-45
rl, w, rni = 4.1e-08, 4.3755e-06,  2.936e-47
rl, w, rni = 4.1e-08, 4.3789e-06,  3.1464e-46
rl, w, rni = 4.1e-08, 4.3738e-06,  5.6752e-49
rl, w, rni = 4.1e-08, 4.3721e-06,  4.807e-47
rl, w, rni = 4.1e-08, 4.3747e-06,  5.4456e-48
rl, w, rni = 4.1e-08, 4.373e-06,  1.4763e-47
rl, w, rni = 4.1e-08, 4.3742e-06,  6.2469e-49

rl (cm^-1), w (sec^-1), 1/T_f (sec^-1) = 4.1e-08,  4.3738e-06,  1.8764e-05

Do you want to save this curve? yes = 1, no = 0 1
Saved as DemoFile.mat
File includes dispersion curve, 1/T_f, x, h, vbar, r, pressure (last frequency only),
          f, del, eps, and long wave coefficients (if valid)
>>
```

*Note that the user is asked at the end whether she wants to save the information. Answered "1" here to save file.*

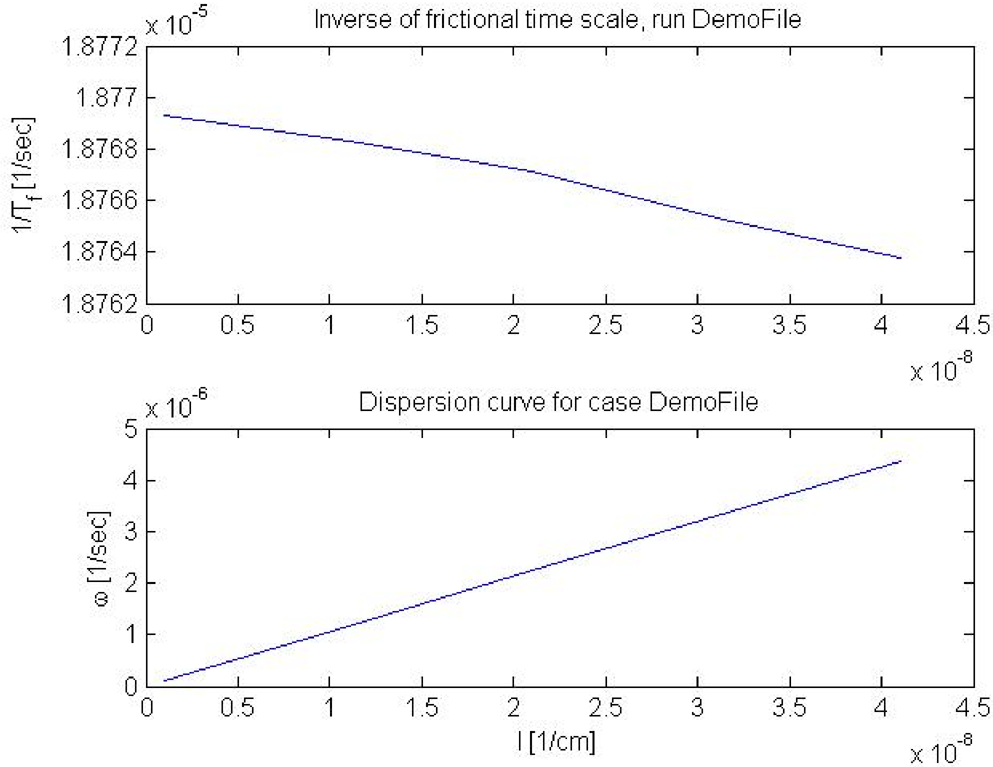
*Note, in the saved file, that "1/$T_f$" is saved as array "wcoms" within the "mat" file.*

*Three plots are produced. One is a repeat of the figure from* bwavespsetup.m. *The other two plots follow.*

*One shows modal structure:*




*and another shows the dispersion curve and frictional damping:*




## Sample: Running bwavesfinch.m

*The follow is what appears on the screen. Notice, for each question, that the present value of the parameter/condition is displayed before a query is made. This run changes the free surface condition to a rigid lid.*

```
>> ardemo = bwavespfinch(ardemo);
```

First you need to select what you want to change.

Options are:
Grid size: enter "g"
Initial Frequency guess: w
Coriolis parameter: f
Domain size:  x
Model assumptions:  a
Dispersion curve definition:   d
Nominal accuracy: e
Depth profile: h
Bottom friction:  r

Mean flow field: v

Any arrays are row arrays, not column arrays

Select an option a

Old model assumptions
Free surface
General Frequency and Wavenumber
Enter 0 for rigid lid, 1 for free surface 0
Enter 0 for coastal long wave assumption, 1 for general frequency and wavenumber 1

Closed boundary at x = 0
Open boundary at x = x_Max
Enter 0 for a closed boundary at x = 0, 1 for an open boundary 0
Enter 0 for a closed boundary at x = x_Max, 1 for an open boundary 1
>>

*Note that first the user selects a category ("a" in this case). Then more specific information is required.*

# Description of the relevant mfiles:

**bwavesp.m**

This is the main mfile that drives all of the others. It uses the input array to define the key variables, and calls routines to fill out depth, velocity and bottom friction arrays. It then steps through the required wavenumbers to obtain and store modal frequencies. Finally, it gives the user an option to save results to a file.

**bwavespcal.m**

This file is essentially the interface between the fminsearch function and the code that sets up and solves the governing equation. This is also the place where the response magnitude is calculated.

**bwavespdamp.m**

This file takes a given inviscid modal structure and frequency, and then computes the frictional perturbation correction to frequency (expressed as $1/T_f$). The expression used is somewhat more general than that in Brink (1990), but the analysis follows the same outline.

**bwavespdep.m**

This file takes the input information about water depth, interpolates it onto the model grid, and computes the depth gradient. If information has not been provided about depth near the boundaries (e.g., if xh(1) > 0), the topography is filled out by assuming that depth is constant over the gap. The file also tests to make sure the depth profile is consistent with assumptions.

**bwavespfinch.m**

This file (described above, and shown as a sample) allows the user to change aspects of the input array without having to create a new file from scratch. The user is first asked what category is to be changed, and then more specific questions are asked.

**bwavesplong.m**

This file is only called when the coastal long wave approximation has been made (eps = 0). The pressure is normalized according to the long wave norm, and the wind coupling (bn) and frictional damping (ann) coefficients are computed and displayed. If the coastal long wave assumption is not made, these coefficients are set to NaN. If r = 0 everywhere, ann will also be zero. See Brink (1989), but the present formulation accounts for the different boundary condition possibilities.

**bwavesppl.m**

This file plots out the modal structure each time a new modal frequency is calculated. For reference, the depth profile is also plotted.

**bwavespr.m**

This file carries out two functions. First, it takes input information about r and interpolates it onto the grid. If information is not provided for the whole grid, existing values are extended out to the boundaries. For example, if nr = 1, xrr = 0 and rrr = 0.01, the result will be to set r = 0.01 cm/sec over the whole domain.
Secondly, this file plots out the profiles of friction coefficient $r$, mean alongshore velocity $v_0$ and depth $h$.

**bwavespsetup.m**

This file (discussed above) is used to create an input array to drive wave calculations. It functions by asking a sequence of questions. It plots out the profiles of friction coefficient $r$, mean alongshore velocity $v_0$ and depth $h$. It also executes a few consistency checks to make sure nothing foolish gets done. The user will only see the results of these checks if there is a problem.

**bwavespsol.m**

This file creates a matrix equation representing equation (5) and applies the boundary conditions set by the user. Given a value for alongshore wavenumber and a guess at frequency, it then solves for pressure, and this array is returned to bwavespcal.m to evaluate for resonance. The arbitrary forcing is a spike at about the middle of the model grid.

**bwavespvel.m**

This file takes the input information about mean velocity and interpolates it onto the model grid. It also calculates the first and second derivatives of this quantity. It checks to see whether $(f+v_{0x})$ changes sign, and whether the necessary condition for barotropic instability is met, i.e., whether

$$Q_x = \left(\frac{f+v_{0x}}{h}\right)_x \tag{17}$$

changes sign. If there is a sign change in either quantity, a warning is given and a relevant plot is presented.

**bwavesp2c.m**

This file converts an input array for bwavesp.m (real frequency waves) into an array suitable for use with bwavesc.m (complex frequency solutions).