

Bigc*.m

Coastal-trapped waves with stratification, topography, mean flow and bottom friction in Matlab

K.H. Brink

September, 2003
March, 2004
September, 2004
January, 2005
May, 2006
September, 2006
June, 2009
September, 2010
June, 2011
October, 2011
March, 2012

This set of Matlab mfiles (all having names that begin with “bigc”) can be used to calculate coastal-trapped wave modal structures and dispersion curves under very general circumstances. A complex frequency is allowed, so that instability and damping can be accounted for directly. Modal structures and energy diagnostics are provided. For most applications, the code is only useful for subinertial wave frequencies (i.e., the real part of wave frequency is smaller than the Coriolis parameter).

Although a number of mfiles are involved here, there are only three that you would actually call directly:

bigcsetup.m Helps you set up the input data array (which comes out as array “datnew” in the following example). Call the mfile and it will prompt you with questions. Call as:

```
>> datnew = bigcsetup;
```

bigcomp.m This is the file that actually calls all the others and so does the calculations. Call as:

```
>> bigcomp(datnew)
```

bigcfinch.m This is used to modify a pre-existing input array. It will give you a menu for what you want to change. Call as:

>> datnewer = bigcomp(datnew);

For examples of usage, see the screen displays copied at the end of this document.

The Problem:

We seek to find the free wave solutions for linearized coastal-trapped waves along a straight coast. The depth $h(x)$ is uniform in the alongshore (y) direction. The offshore direction is x ($x = 0$ is the coast) and the vertical direction is z ($z = -h(x)$ is the bottom), where the origin is at the surface at the coast. Dissipative effects are limited to the infinitesimally thin bottom boundary.

The equations of motion are:

$$v_t + u v_{0x} + v_0 v_y + w v_{0z} + f u = -\rho_0^{-1} p_y + \rho_0^{-1} \tau_z^y$$

$$u_t + v_0 u_y - f v = -\rho_0^{-1} p_x + \rho_0^{-1} \tau_z^x$$

$$0 = -p_z - g \rho_2$$

$$u_x + v_y + w_z = 0$$

$$\rho_{2t} + u \rho_{1x} + v_0 \rho_{2y} + w \rho_{1z} = 0$$

Where u, v, w are the perturbation velocity components in the x, y, z directions, respectively, p is perturbation pressure, f is the Coriolis parameter, density is broken up as: $\rho_0 + \rho_1(x, z) + \rho_2(x, y, z, t)$ and $\rho_0 \gg \rho_1 \gg \rho_2$.

In all cases, subscripted independent variables represent partial differentiation. The mean velocity $v_0(x, z)$ is in geostrophic balance with the mean pressure p_0 and density $\rho_0 + \rho_1$. Turbulent stresses in the x, y directions are given by τ^x, τ^y , and the bottom stress (τ_0^x, τ_0^y) is related to the interior bottom velocity by

$$\tau_0^x = \rho r u_B, \quad \tau_0^y = \rho r v_B,$$

where $r(x)$ is a bottom resistance coefficient having units of velocity, and a subscript "B" denotes a quantity evaluated at the bottom. We also define

$$N^2 = -g \rho_0^{-1} \rho_{1z} \quad \text{and}$$

$$M^2 = -g \rho_0^{-1} \rho_{1x}.$$

The problem is solved by assuming that

$$p = p(x, z) \exp[i(\omega t + l y)], \text{ etc.,}$$

where the frequency ω can be complex and the alongshore wavenumber l is real. Using this assumption, the problem can be reduced to a single partial differential equation for pressure

$$0 = \omega' p_{xx} - 2 \omega' s p_{xz} + \omega' N^{-2} (ff' - \omega'^2) p_{zz} - \omega' (Q + s_z) p_x \\ - [-\omega' s Q + (\omega' s)_x + lf^{-1} s (f^2 - \omega'^2) - (\omega' N^{-2})_z (ff' - \omega'^2) - (\omega' N^{-2}) (sM^2)_z] p_z \\ - [lfQ + \omega' l^2 + lfs_z] p,$$

where

$$\omega' = \omega + lv_0 \\ s = M^2/N^2, \\ f' = f + v_{0x} \\ Q = (ff^* - \omega'^2)^{-1} [(ff^* - \omega'^2)_x - s(ff^* - \omega'^2)_z] \\ f^* = f + v_{0x} - M^2 s/f.$$

This equation and the notation are similar to those of Mooers (1975 a,b), except that these are for hydrostatic conditions.

As noted by Brink, 1982 or Dale et al. (2001), this equation admits a spurious solution at $\omega' = f$ when $v_{0z} = v_{0x} = 0$. The presence of this mode also distorts modal structures and dispersion curves for nearby frequencies. Typically, this spurious pressure mode is vertically uniform and it decays only slowly with distance offshore.

The boundary conditions are as follows:

$z = 0$:

$$w = \delta (g\rho_0)^{-1} [p_t + v_0 p_y],$$

where $\delta = 1$ for a free surface condition and $\delta = 0$ for a rigid lid condition. In terms of pressure,

$$0 = (\omega' N^2) (ff' - \omega'^2) p_z - s \omega' p_x - [lfs - \omega' \delta (ff^* - \omega'^2)/g] p$$

$z = -h(x)$:

$$0 = w + h_x u + U_{Ex} + V_{Ey},$$

where the Ekman transport is given as

$$U_E = r (ff^* - \omega'^2)^{-1} (-i \omega' u - fv)_B, \\ V_E = r (ff^* - \omega'^2)^{-1} (f^* u - i \omega' v)_B,$$

and the subscript “B” denotes a quantity evaluated at the bottom (along $z = h(x)$). In terms of pressure,

$$\begin{aligned}
0 = & -r (\overline{ff}^* - \omega'^2)^{-1} (\overline{ff}^* + \omega'^2) p_{xBx} + r (\overline{ff}^* - \omega'^2)^{-1} (2s\omega'^2) p_{zBx} \\
& + [is\omega' - i\omega'h_x - (\overline{ff}^* + \omega'^2)F + r (\overline{ff}^* - \omega'^2)^{-1} (-2l v_{0Bx} \omega' - \overline{ff}^*_x + 2 \omega'lf^*)] p_{xB} \\
& - r (\overline{ff}^* - \omega'^2)^{-1} (2 \omega'lf^*) p_{Bx} \\
& + \{ -i \omega' N^{-2} (\overline{ff}^* - \omega'^2) + i s \omega' h_x + 2s \omega'^2 F + r (\overline{ff}^* - \omega'^2)^{-1} [2(s \omega'^2)_{Bx} \\
& - (\omega'sl/f) (\overline{ff}^* + \omega'^2)] \} p_{zB} \\
& + \{ islf - ilfh_x - 2 \omega'lfF + r (\overline{ff}^* - \omega'^2)^{-1} [-2fl^2 v_{0Bx} + l^2 (\overline{ff}^* + \omega'^2)] \} p,
\end{aligned}$$

where

$$F = r_x (\overline{ff}^* - \omega'^2)^{-1} + 2r [(\overline{ff}^* - \omega'^2)^{-1}]_{Bx}.$$

$x = 0$:

There is a choice of two boundary conditions at the coast.

Closed boundary condition:

$$0 = U_E + uh \approx U_E + \int_{-h}^0 u dz.$$

When there is bottom friction ($U_E \neq 0$), this condition really only makes sense if the vertical variations in u are negligible over the depth $h(0)$. With no bottom friction, this boundary condition is sensible regardless of the coastal wall height.

Stated in terms of pressure, this condition is:

$$\begin{aligned}
0 = & -i \omega'h p_x - ilfh p + i \omega' sh p_z \\
& + R r (\overline{ff}^* - \omega'^2)^{-1} [-(\overline{ff}^* + \omega'^2) p_x + 2 s (\omega')^2 p_z - 2 \omega'fl p]_B
\end{aligned}$$

where

$$R = (\overline{ff}^* - \omega'^2) / (\overline{ff}^* - \omega'^2)_B.$$

“Open” boundary condition:

To mimic an open boundary, we use

$$u_x = 0,$$

which does a reasonable job for linear coastal-trapped waves at a distant “open” offshore boundary, where variations should have decayed (see Brink, 1982). To help make sure this is used reasonably, use of this condition requires that the bottom be flat at $x = 0$.

Stated in terms of pressure, this condition is:

$$\begin{aligned} 0 = & -\omega' p_{xx} + s\omega' p_{xz} + [-lf' + \omega' (ff^* - \omega'^2)_x (ff^* - \omega'^2)^{-1}] p_x \\ & + [lsv_{0x} + \omega' s_x - s\omega' (ff^* - \omega'^2)_x (ff^* - \omega'^2)^{-1}] p_z \\ & + lf (ff^* - \omega'^2)_x (ff^* - \omega'^2)^{-1} p \end{aligned}$$

This boundary condition can only be expected to work well at subinertial frequencies. At higher frequencies, it can lead to internal wave reflection, rather than the free exit of internal waves.

$x = L$:

The offshore boundary condition is either the same as the coastal “open” condition or the coastal “closed condition”. The open condition seems to work well as long as the grid boundary ($x = L$) occurs far offshore of where the modal solutions have substantial structure. The condition, as used, requires that the bottom is flat at the offshore boundary. The condition is:

$$u_x = 0,$$

or, in terms of pressure:

$$\begin{aligned} 0 = & -\omega' p_{xx} + s\omega' p_{xz} + [-lf' + \omega' (ff^* - \omega'^2)_x (ff^* - \omega'^2)^{-1}] p_x \\ & + [lsv_{0x} + \omega' s_x - s\omega' (ff^* - \omega'^2)_x (ff^* - \omega'^2)^{-1}] p_z \\ & + lf (ff^* - \omega'^2)_x (ff^* - \omega'^2)^{-1} p. \end{aligned}$$

The closed condition is:

$$U_E + uh \approx 0.$$

The problem is now expressed in stretched coordinates that map the problem into a rectangular domain (e.g. Wang and Mooers 1976). Specifically,

$$\theta = z/h(x)$$

and

$$x' = x.$$

The problem is then solved numerically via resonance iteration. An arbitrary forcing is applied through the surface boundary condition (essentially, a localized wind stress curl near the grid center), and ω is searched to find a complex frequency corresponding to a given wavenumber l . This search makes use of the Matlab function “fminsearch.m”.

Using the software:

Examples of what your Matlab screen will look like as you run the different mfiles are shown at the end of this document. The main routine, “bigcomp”, requires an input array that can be generated by “bigcsetup” and modified by “bigcfinch”. When you run “bigcsetup” or “bigcfinch”, you simply need to answer the questions posed. Do not forget to use square brackets when entering more than one number on a line.

When running “bigcomp”, you will first be shown a few simple parameter values. You also get a section plot showing topography, stratification (minus a constant background value), and the mean flow. If symmetric instability is possible, f^* will also be plotted (in red) on the density plot. If you are allowing pauses during the run, you are advised of pauses, and need to hit a key each time to continue. During iteration toward a solution, the alongshore wavenumber is displayed, and the program displays complex frequency and the inverse resonance parameter ($rrr =$ inverse of the integral of $|p|^2$ over x and z) for each ω estimate. When you are at a resonance (free mode), rrr becomes very small. You can thus watch the convergence onto a complex free modal frequency. Convergence to your given tolerance is then announced, and the free mode pressure is plotted as a section. A pause is announced at this point. Hit any key to continue, and you will then also get section plots of u , v , w , ρ , energy diagnostics (pool sizes and transfers: the dissipation integral should be a negative quantity), and line plots of p , v , u at the surface. If you set the program to run more than one frequency, it will then display the next wavenumber and go on from there.

If you have set the program to compute more than one frequency, wavenumber pair, the program will also draw a complex dispersion curve once all pairs have been obtained.

Some Notes on using the software:

The $x = 0$ and $x = L$ boundary conditions can either be closed (as at a coast), or open. If it is closed, then the water there ought to be shallow if bottom friction is nonzero (with no dissipation, this does not matter). The reason for this is that model balances the bottom Ekman transport in a depth-uniform way at the coastal wall. This makes sense if the water is shallow enough that natural depth variations in u are negligible (hence the model is not imposing any physics that is not there already). This means that with bottom friction, the boundary water depth should be probably less than 50 m, but this depends considerably on conditions, such as the alongshore wavelength (see

also Mitchum and Clarke, 1986). If there is no bottom friction, then the coastal wall can have any depth.

If a lateral boundary is open, the bottom should be flat there. In the case of an open boundary condition, the water does not have to be shallow. Care should be taken to make sure that the onshore and offshore boundaries are located far from the main modal structure's location of maximum value.

The numerical grid extends one point outside the x,z domain at each end (nn is the number of x grid points, and mm the number of θ grid points). There is also a grid point beyond each boundary. Thus, for example, $nn = 5$ gives one interior grid point, 2 on boundaries and 2 outside the domain.

In choosing a grid size, caution is advised. One should experiment with grid sizes in order to be confident of a good answer. I find that $nn = 80-120$, $mm = 20-30$ seems to work well in many cases. The number of operations or amount of memory needed goes something like $nn*mm*mm$. There is a test for hydrostatic consistency (the ratio computed is $h_x \Delta x / (h \Delta \theta)$), and results seem best if the ratio given is less than around 1-5.

The mean alongshore flow is assumed to be Gaussian (with different onshore, offshore, up and down length scales, as well as a variable center position and depth). This provides a fair bit of flexibility. Once the velocity field is computed, the density structure is computed using the thermal wind equation. The density field can be left undisturbed at either the offshore or onshore end of the domain.

There is substantial documentation within the mfile "bcomp.m" for further detail.

Results for pressure, velocity and density are plotted. The smaller of the real or imaginary is suppressed if it is too small (maximum absolute value less than about 1000 times less than the other's maximum absolute value, as it is set now). This ratio is controlled by the value of "ratnplot" in "bigcuvwr.m" and "bigcomp.m".

All outputs are of arbitrary amplitude (with pressure made real at the top at the coast). Units are consistent within c.g.s., i.e. cm/sec, dyne/cm², gm/cm³.

The search can be very inefficient if you give it 0 as the starting guess for real or imaginary frequency. Better to give it a very small nonzero value. The code will virtually always converge to a complex frequency, even if the true result is a real frequency. You need to judge what imaginary value is small enough to be effectively zero, but this is usually pretty obvious. This means, also, that there can be very small energy exchanges in the energy diagnostics when the flow is stable. Again, you need to decide what is effectively zero.

By the same token, if you give the algorithm an initial estimate for real or complex frequency that has few places of accuracy (say $1e-5$), the Matlab search code will look around at fairly large increments (perhaps $\pm 20\%$), 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 $\pm 3\%$), and it becomes much more likely that nearby solutions will be found.

The search can sometime miss a complex frequency, for example if searching all takes place only near the real axis, but the true result has a substantial imaginary part. If it is a good solution, the inverse resonance parameter in the output (rrr) should be several orders of magnitude lower than neighboring values. Use a smaller accuracy tolerance (ϵ) or start the search again farther from the

real axis.

For an accuracy estimate (“e” option in bigcfinch.m), I find that 0.0001 works out fairly well. This means 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).

If more than one frequency is computed, there is an option to save the “dispersion curve” information (wavenumbers and complex frequencies).

The input file formats differ slightly between the “bigr*.m” system (for real frequencies only), and this more general “bigc*.m” set. To make the conversions back and forth conveniently, use “c2rinpcon.m” and “r2cinpcon.m”.

When the Ertel potential vorticity $q = -N^2 f f^*$ changes sign relative to f , symmetric instability is possible (Hoskins, 1974). So, the routine “bigcrhocal2.m” tests to see if f^*/f is negative anywhere. If this happens, f^* is contoured in red on the same plot as mean density, and a message appears on your Matlab screen. If f^*/f does not become negative, it is not contoured, and no message about it is given.

Given the presence of the near-inertial spurious mode, and given that the “open” boundary condition is only reasonable for subinertial frequencies, the model results should only be trusted for subinertial frequencies, $|\text{Re}(\omega)| < |f|$. If, in computing a dispersion curve, you start to obtain real frequencies greater than the inertial, the superinertial results should be discarded.

All frequencies are in radians per second, and all wavenumbers in radians per cm. Radians do not have units, so these units could be written as “1/sec” for example.

You can get spurious results if the domain is too large relative to the natural scale of the wave under consideration. For example, with a domain size $x_{\text{max}} = 70$ km, trying to find a flat-bottom internal Kelvin wave with internal Rossby radius = 3 km yields bad results. Doing the same calculation with $x_{\text{max}} \leq 40$ km yields reasonable results. This problem arises because the arbitrary forcing used to create the resonance is applied near $x_{\text{max}}/2$. Trouble can arise when the desired wave has negligible amplitude at this distance offshore.

References:

- Brink, K.H., 1982. A comparison of long coastal trapped wave theory with observations off Peru. *Journal of Physical Oceanography*, **12**(8), 897–913.
- Brink, K. H., 1997. Time-dependent motions and the nonlinear bottom Ekman layer. *Journal of Marine Research*, **55**(4), 613–631.
- Dale, A.C., J.M. Huthnance, and T.J. Sherwin, 2001: Coastal-Trapped Waves and Tides at Near-Inertial Frequencies. , **31**, 2958–2970.

Hoskins, B.J., 1974: The role of potential vorticity in symmetric stability and instability. [Quarterly Journal of the Royal Meteorological Society](#), 100(425), 480-482.

Mitchum G. and A.J. Clarke, 1986: The frictional nearshore response to forcing by synoptic scale winds. *J. Phys. Oceanogr.* 16: 934-946.

Mooers, CNK, 1975a: Several effects of a baroclinic current on the cross-stream propagation of inertial-internal waves. *Geophys. Fluid Dyn.*, 6, 245-275.

Mooers, CNK, 1975b: Several effects of a baroclinic current on the three-dimensional propagation of inertial-internal waves. *Geophys. Fluid Dyn.*, 6, 277-284.

Wang, D.-P. and CNK Mooers, 1976: Coastal-trapped waves in a continuously stratified ocean. *J. Phys. Oceanogr.*, 6: 853-863.

Sample: Running bigcsetup.m

This is what you see on your Matlab screen. After each question, you type in the numerical answer

```
>> finsamp = bigcsetup;
```

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
How many total gridpoints do you want in the vertical? (mm) 30
First guess at real part of frequency (rad/sec)? 2e-5
First guess at imaginary part of frequency (rad/sec)? 1e-12
Enter 0 for a rigid lid, 1 for a free surface (del) 0
Enter 0 for a closed x= 0 boundary, 1 for open (icbc) 0
Enter 0 for a closed offshore boundary, 1 for open (iobc) 1
Enter the Coriolis parameter (f) (rad/sec) 1e-4
Enter the domain width (xmax) (km) 160
Enter the nominal fractional accuracy for the solution (eps) .0001
Enter the number of frequencies to be computed (npts) 1
Enter the first alongshore wavenumber to use (rlz) (1/cm) 1e-7
How many distance, depth pairs will you provide (ndep >=1) 2
Array of offshore distances for depth values (xdep in km) (dimension ndep) [0
80]
Array of depths corresponding to xdep (depr in m) [30 2000]
Number of distance, bottom friction pairs to read (nr) 0
Number of Nsquared values to read? (nnsq) 2
Depth increment for input of Nsquared values? (zr in m) 3000
Exponential tail length for Nsquared extrapolation (alph in km) 1
Nsquared values starting at the surface (nsqr in rad^2/sec^2) (nnsq values) 1e-
5*[1 1]
Input peak value of mean alongshore flow (vzero: cm/sec) 0
Enter 0 to skip pauses to see graphics, 1 to see graphics during execution 1
>>
```

Comment: For the depth h and bottom resistance coefficient r , if the last value you enter is for an x that is less than the domain size, the last value you enter will be used to fill out all values farther offshore. That is, the value is constant beyond the last one entered.

Comments:

- For r , if you enter 0 values to read, it will not ask for any more information on r . If you enter one or more values to read, it will prompt you for the x , r pairs.
- Likewise, if you enter that the peak $v_0 = 0$, then it will ask for no further information about the mean flow. If you enter that the peak v_0 is nonzero, then you will be asked for the location (x , *depth*) of the maximum flow, and the horizontal (onshore, offshore) and vertical (up, down) Gaussian scales.
- Similarly, if you enter 1 for $npts$ while defining the dispersion curve, it will only ask for the one wavenumber you want to use. If you use $npts > 1$, it will additionally ask for the wavenumber increment you want to use along the dispersion curve.

Sample: Running bigcomp.m

```
>
>> bigcomp(finsamp)

Rigid lid
Closed BC at x = 0
Open offshore BC
f = 1.0000e-004 rad sec^-1
nn, mm = 100 30

Max consistency ratio = 13.656 at x = 0 km
This should be kept less than one, and definitely less than 10
```

Pause on: hit any key to continue

Wavenumber = 1e-007 rad cm^-1

wr, wi, rrr = 2.0000e-005	1.0000e-012	4.9603e-023
wr, wi, rrr = 2.1000e-005	1.0000e-012	3.1035e-023
wr, wi, rrr = 2.0000e-005	1.0500e-012	4.9603e-023
wr, wi, rrr = 2.1000e-005	1.0500e-012	3.1035e-023
wr, wi, rrr = 2.2000e-005	1.0000e-012	1.7766e-023
wr, wi, rrr = 2.3000e-005	9.7500e-013	8.7679e-024
wr, wi, rrr = 2.3000e-005	9.2500e-013	8.7679e-024
wr, wi, rrr = 2.5000e-005	9.0000e-013	4.6175e-025
wr, wi, rrr = 2.7000e-005	8.5000e-013	2.0097e-024
wr, wi, rrr = 2.5000e-005	9.5000e-013	4.6175e-025
wr, wi, rrr = 2.6000e-005	9.6250e-013	1.4797e-025
wr, wi, rrr = 2.8000e-005	8.8750e-013	5.9084e-024
wr, wi, rrr = 2.6750e-005	9.0937e-013	1.3494e-024
wr, wi, rrr = 2.4250e-005	9.5312e-013	2.2697e-024
wr, wi, rrr = 2.6125e-005	9.2031e-013	2.6548e-025
wr, wi, rrr = 2.7125e-005	9.8281e-013	2.3875e-024
wr, wi, rrr = 2.5531e-005	9.2070e-013	1.1987e-026
wr, wi, rrr = 2.5406e-005	9.6289e-013	5.8848e-026
wr, wi, rrr = 2.4938e-005	9.2109e-013	5.5858e-025
wr, wi, rrr = 2.5734e-005	9.5215e-013	1.1146e-026
wr, wi, rrr = 2.5859e-005	9.0996e-013	5.6265e-026
wr, wi, rrr = 2.5746e-005	9.2319e-013	1.3909e-026
wr, wi, rrr = 2.5520e-005	9.4966e-013	1.4869e-026
wr, wi, rrr = 2.5689e-005	9.2981e-013	3.3801e-027
wr, wi, rrr = 2.5893e-005	9.6125e-013	7.4027e-026
wr, wi, rrr = 2.5622e-005	9.3084e-013	1.8670e-028
wr, wi, rrr = 2.5577e-005	9.0850e-013	3.7551e-027
wr, wi, rrr = 2.5616e-005	9.1941e-013	3.7956e-028
wr, wi, rrr = 2.5548e-005	9.2045e-013	8.3657e-027
wr, wi, rrr = 2.5654e-005	9.2747e-013	4.3284e-028
wr, wi, rrr = 2.5584e-005	9.2279e-013	2.9155e-027
wr, wi, rrr = 2.5636e-005	9.2630e-013	4.5008e-030
wr, wi, rrr = 2.5642e-005	9.3773e-013	6.2980e-029

```
wr, wi, rrr = 2.5657e-005 9.3318e-013 5.6219e-028
wr, wi, rrr = 2.5630e-005 9.3143e-013 1.8628e-029
wr, wi, rrr = 2.5625e-005 9.2000e-013 1.0267e-028
wr, wi, rrr = 2.5638e-005 9.3329e-013 1.1695e-029
wr, wi, rrr = 2.5644e-005 9.2817e-013 9.7136e-029
wr, wi, rrr = 2.5634e-005 9.3061e-013 5.9677e-031
wr, wi, rrr = 2.5633e-005 9.2362e-013 4.2888e-030
wr, wi, rrr = 2.5630e-005 9.2793e-013 2.4655e-029
wr, wi, rrr = 2.5635e-005 9.2671e-013 1.2247e-031
wr, wi, rrr = 2.5636e-005 9.3370e-013 2.7168e-030
wr, wi, rrr = 2.5635e-005 9.3118e-013 5.1627e-031
```

Converged!

```
wr, wi, rrr = 2.5635e-005 9.2671e-013 1.2247e-031
```

Pause on: hit any key to continue

>>

Comment: Each of the pauses corresponds to graphics appearing in other windows.

Sample: Running bigcfinch.m

```
>> finsampnew = bigcfinch(finsamp);
```

First you need to select what you want to change.

Options are:

```
Grid size: enter "g"
Initial Frequency guess: w
Boundary conditions: b
Coriolis parameter: f
Domain size: x
Dispersion curve definition: d
Nominal accuracy: e
Depth profile: h
Bottom friction: r
Stratification: n
Mean flow field: v
Pauses during execution: p
```

Any arrays are row arrays, not column arrays

Select an option r

Old number of r values = 0

Enter number of (x,r) pairs (>=0) 2

Enter array of x locations where resistance coefficient is given (km) [0 50]

Enter array of resistance coefficient (cm/sec) [0.02 0.01]

>>

Mfiles in this package:

This collection of mfiles includes the following:

Mfiles that you can call directly:

bigcomp.m

The driver for the coastal trapped wave calculations. Carries out all input functions, calls the functions for computing mean density and velocity fields, and for searching for resonant frequencies. Also does some output and graphics functions.

bigcsetup.m

This mfile sets up and input array for “bigcomp”, based on user responses to questions. An example of its use is given above.

bigfinch.m

This mfile changes an existing “bigcomp” input array, without having to generate an all new array. An example of its use is given above.

c2rinpcon.m

The “bigcomp” (complex frequency) and “bigromp” (strictly real frequency) model systems call for slightly different input array structures. Use this mfile to convert a “bigcomp” input file into a “bigromp” file.

r2cinpcon.m

The “bigcomp” (complex frequency) and “bigromp” (strictly real frequency) model systems call for slightly different input array structures. Use this mfile to convert a “bigromp” input file into a “bigcomp” file.

Mfiles that are called through bigcomp.m (either directly or indirectly):

bigcdep.m

Computes a gridded depth profile, along with its derivatives.

bigcconchk.m

Grid consistency check to see if the bottom slope is large relative to vertical spacing.

bigcr.m

Computes gridded bottom resistance parameter, r .

bigcnsq.m

Computes an undisturbed (by mean flow) gridded $N^2(z)$ profile.

bigcvzcal2.m

Calculates the mean velocity field and its vertical derivative, from inputs.

bigcrhocal2.m

Calculates the modifications to the basic density field caused by the mean flow.

bigcdrver.m

Mainly called from `fminsearch`. This sets up the matrix equations that are equivalent to the field differential equation and its boundary conditions. Also solves the equation, and computes an integral of the response to evaluate convergence.

bigcsigtoz.m

Converts a field from sigma coordinates to z coordinates.

bigcengdiag.m

Does the energy diagnostics for each solution.

bigcuvwr.m

Called through `bigcengdiag.m`. Calculates u, v, w, ρ fields and plots them.

bigcztosig.m

Converts z coordinates to sigma coordinates.

The following are called by `bigcdrver.m`:

bigccc.m

Computes coefficients for the coastal boundary condition.

bigcoff.m

Computes coefficients for the offshore boundary condition.

bigcbot.m

Computes coefficients for the bottom boundary condition.

bigcsbc.m

Computes coefficients for the surface boundary condition.

bigmatco.m

Computes coefficients for interior points on the finite difference grid.

bigcint.m

Computes the integral of the response magnitude.