

## **Bigr\*.m**

### **Stable coastal-trapped waves with stratification, topography and mean flow in Matlab**

**K.H. Brink**

September, 2003

March, 2004

September, 2004

December, 2004

May, 2006

September, 2006

June, 2009

September, 2010

June, 2011

March, 2012

March, 2014

This set of Matlab mfiles (all having names that begin with “bigr” or “bigc”) can be used to calculate stable, inviscid coastal-trapped wave modal structures and dispersion curves under very general circumstances. Only a real frequency is allowed, so that instability and damping cannot be accounted for directly, but computations are more efficient than for the general case, long-wave parameters can be computed for first order wave equation calculations (see Brink, 1989), and a more general perturbation decay time (Brink, 1990) can also be obtained. Modal structures and energy diagnostics are provided. Generally speaking, the code is only useful for subinertial wave frequencies.

The present code does everything that the older “bigload3.for” does except that it allows for a non-zero mean alongshore flow, and it allows for a closed offshore boundary condition. Further, graphical outputs are embedded.

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

**bigrsetup.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 = bigrsetup;
```

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

>> bigromp(datnew)

**bigfinch.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 = bigromp(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, and are only included for calculating coefficients for use with the first order wave equation.

The equations of motion are:

$$\begin{aligned} v_t + u v_{0x} + v_0 v_y + w v_{0z} + f u &= -\rho_0^{-1} p_y + \rho_0^{-1} \tau_z^y \\ \delta u_t + \delta v_0 u_y - f v &= -\rho_0^{-1} p_x + \delta \rho_0^{-1} \tau_z^x \\ 0 &= -p_z - g \rho_2 \\ u_x + v_y + w_z &= 0 \\ \rho_{2t} + u \rho_{1x} + v_0 \rho_{2y} + w \rho_{1z} &= 0 \end{aligned}$$

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  $\rho_0 \gg \rho_1 \gg \rho_2$ . Using  $\delta = 1$  corresponds to a general frequency and wavenumber, while  $\delta = 0$  produce the long wave limit, and only allows one frequency to be computed.

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  $\tau^x, \tau^y$ , and the bottom stress ( $\tau_0^x, \tau_0^y$ ) 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. The turbulent stresses are not used for computing the modal structures, but they are used to compute the damping coefficients after the inviscid wave mode is known. 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

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

where the frequency  $\omega$  and the alongshore wavenumber  $l$  are real. Using this assumption, the problem can be reduced to a single partial differential equation for pressure

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

where

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

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 (1981) and 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:

$z = 0$ :

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

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

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

$z = -h(x)$ :

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

where the Ekman transport is given as

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

and the subscript “B” denotes a quantity evaluated at the bottom (along  $z = h(x)$ ). This complete form is used only in computing the long wave frictional pattern once the inviscid mode is known. For the computing the inviscid modal structure, the boundary condition is:

$$0 = w + h_x u$$

In terms of pressure,

$$\begin{aligned} 0 = & + [is\omega' - i\omega'h_x] p_{xB} \\ & + \{ -i\omega' N^2 (ff' - \delta\omega'^2) + is\omega'h_x \} p_{zB} \\ & + \{ islf - ilfh_x \} p. \end{aligned}$$

**$x = 0$ :**

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

**Closed boundary condition:**

This boundary condition is used when  $icbc = 0$ :

$$0 = u.$$

Since there is no bottom friction, this boundary condition is sensible regardless of the coastal wall height. Damping coefficient calculations do account for Ekman transport here.

Stated in terms of pressure, this inviscid condition is:

$$0 = -i\omega'h p_x - ilfh p + i\omega' sh p_z.$$

**“Open” boundary condition:**

To mimic an open boundary ( $icbc = 1$ ), we use

$$u_x = 0,$$

which does a reasonable job for linear coastal-trapped waves at an “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:

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

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 open boundary condition is the same as the coastal “open” position, and is called by setting  $iobc = 1$ . This 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:

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

In addition, using  $iobc = 0$  allows use of a closed offshore boundary condition (the same as for the coastal boundary).

At this point, the problem is converted into 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  $\omega$  is searched to find a real 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, “bigromp”, requires an input array that can be generated by “bigrsetup” and modified by “bigrfinch”. When you run “bigrsetup” or “bigrfinch”, 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 “bigromp”, 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 you are allowing pauses during the run, you are advised of pauses, and need to hit a key to continue. At this point, the alongshore wavenumber is displayed, and the program displays the frequency and the inverse resonance parameter ( $rrr = \text{inverse of the integral of } |p|^2 \text{ over } x \text{ and } z$ ) for each  $\omega$  estimate. When you are at a resonance (free mode),  $rrr$  becomes very small. You can thus watch the output converge 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  $u$ ,  $v$ ,  $w$ ,  $\rho$ , energy diagnostics, 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 dispersion curve once all pairs have been obtained.

## Some Notes on using the software:

The  $x = 0$  and  $x = x_{max}$  boundary conditions can either be closed (as at a coast), or open. Since there is no bottom friction, the coastal wall can have any depth.

If the  $x = 0$  or  $x = x_{max}$  boundary is open, the bottom should be flat at the boundary. 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  $\theta$  grid points). There is also a grid point outside each boundary. Thus, for example,  $nn = 5$  gives one interior grid point, 2 on the 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.

Results for pressure, velocity and density are plotted.

Although the 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 ( $a_n$ ,  $b_n$ ,  $c_n$ ) when there is a mean flow. Thus, you are not allowed to call the long-wave normalization and coefficient routine when a mean flow is present.

All outputs are of arbitrary amplitude, except when the long-wave routine has been called (which requires that there be no mean flow), in which case pressure and all other variables have been subjected to the long wave normalization (Brink, 1989), corrected for the possibility of a closed offshore boundary. Units are always consistent within c.g.s., i.e. cm/sec, dyne/cm<sup>2</sup>, gm/cm<sup>3</sup>.

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  $\pm 20\%$ ), 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 frequency. If it is a good solution, the inverse resonance parameter in the output (rrr) 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 means you should use bigcomp.m to see if the flow is stable). Using a smaller accuracy tolerance (e) will tend to eliminate false solutions.

For an accuracy estimate (“e” option in bigrfinch.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 frequencies).

In the absence of a mean flow, the perturbation bottom frictional decay time is computed for each frequency (Brink, 1990), allowing for the possibility of a closed offshore boundary and variable resistance coefficient. The  $e$ -folding time is only valid for weak bottom friction (small  $r$ ). Computing this, and the long wave coupling terms, are the only reasons for providing  $r$  for the program.

The format for the input files differs slightly from the general case mfiles (bigc\*.m) and these mfiles for use only with real frequencies (bigr\*.m). In order to make these input conversions conveniently, there are two translation functions, “c2rinpcon.m” and “r2cinpcon.m”.

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,  $|\omega| < |f|$  (except in the long wave limit, where  $\omega/l$  is what is actually important, and the idea of super- or sub-inertial does not make sense). If, in computing a dispersion curve, you start to obtain frequencies greater than the inertial, the superinertial results should be discarded.

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.

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  $x_{\max} = 70$  km, trying to find a flat-bottom internal Kelvin wave with internal Rossby radius = 3 km yields bad results. Doing the same calculation with  $x_{\max} \leq 20$  km yields reasonable results. This problem arises because the arbitrary forcing used to create the resonance is applied near  $x_{\max}/2$ . 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., 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.
- Dale, A.C., J.M. Huthnance, and T.J. Sherwin, 2001: Coastal-Trapped Waves and Tides at Near-Inertial Frequencies. , **31**, 2958–2970.
- 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 bigrsetup.m**

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

```
>> datnew= bigrsetup;
```

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

```
How many total gridpoints do you want in the cross shelf direction? (nm) 100  
How many total gridpoints do you want in the vertical? (mm) 30  
First guess at frequency (rad/sec)? 7.3e-7  
Enter 0 for a rigid lid, 1 for a free surface (del) 1  
Enter 0 for a closed x= 0 boundary, 1 for open (icbc) 0  
Enter 0 for a closed x =xmax boundary, 1 for an open (iobc)1  
Enter 1 for general frequency, wavenumber or 0 for long wave limit 0  
Enter the Coriolis parameter (f) (rad/sec) 1e-4  
Enter the domain width (xmax) (km) 200  
Enter the nominal fractional accuracy for the solution (eps) 0.0001  
Enter the number of frequencies to be computed (npts) 1  
Enter the first alongshore wavenumber to use (rlz) (rad/cm) 1e-8  
How many distance, depth pairs will you provide (ndep >=1) 2  
Array of offshore distances for depth values (xdep in km) (dimension ndep) [0 100]  
Array of depths corresponding to xdep (depr in m) [100 1000]  
Number of distance, bottom friction pairs to read (nr) 1
```

Offshore distances for bottom friction values ( $x_r$  in km) 0  
Array of bottom friction values corresponding to  $x_r$  ( $r_r$  in cm/sec) 0.05  
Number of Nsquared values to read? (nnsq) 2  
Depth increment for input of Nsquared values? ( $z_r$  in m) 2000  
Exponential tail length for Nsquared extrapolation (alph in km) 1  
Nsquared values starting at the surface (nsqr in  $\text{rad}^2/\text{sec}^2$ ) (nnsq values) [1e-7 1e-7]  
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.

**Comment:**

- 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.
- In setting up the dispersion curve in the input file, if you only want one point (one wavenumber: npts = 1), you will only be asked for that wavenumber. If you are setting up a curve with more than one point (npts > 1), you will also be asked for the wavenumber increment along that curve.

## Sample: Running bigromp.m

```
>> bigromp(datnew)
```

```
Long wave limit
```

```
Free surface
```

```
Closed BC at x = 0
```

```
Open BC at x = xmax
```

```
f = 1.0000e-004 rad sec^-1
```

```
nn, mm = 100 30
```

```
Max consistency ratio = 3.9194 at x = 2.0619 km
```

```
    This should be kept less than one, and definitely less than 10
```

```
Pause on: hit any key to continue
```

```
Wavenumber = 1e-008 rad cm^-1
```

```
wr, rrr = 7.3000e-007 1.3403e-029
```

```
wr, rrr = 7.6650e-007 2.1347e-028
```

```
wr, rrr = 6.9350e-007 5.3942e-029
```

```
wr, rrr = 7.1175e-007 3.2887e-030
```

```
wr, rrr = 6.9350e-007 5.3942e-029
```

```
wr, rrr = 7.2088e-007 8.6113e-031
```

```
wr, rrr = 7.3000e-007 1.3403e-029
```

```
wr, rrr = 7.1631e-007 1.9480e-031
```

```
wr, rrr = 7.1175e-007 3.2887e-030
```

```
wr, rrr = 7.1859e-007 5.9344e-032
```

```
wr, rrr = 7.2088e-007 8.6113e-031
```

```
wr, rrr = 7.1745e-007 9.7607e-033
```

```
wr, rrr = 7.1631e-007 1.9480e-031
```

```
wr, rrr = 7.1802e-007 5.2453e-033
```

```
wr, rrr = 7.1859e-007 5.9344e-032
```

```
wr, rrr = 7.1774e-007 1.7374e-034
```

```
wr, rrr = 7.1745e-007 9.7607e-033
```

```
wr, rrr = 7.1788e-007 8.7752e-034
wr, rrr = 7.1760e-007 3.1346e-033
wr, rrr = 7.1781e-007 6.7591e-035
```

Converged!

```
wr, rrr = 7.1781e-007 6.7591e-035
Perturbation frictional decay time (sec) = 5.3245e+005
```

```
Long wave phase speed, cn (cm/sec) = 71.781
Long wave coupling coefficient, bn ((cm sec)^-0.5)) = -0.44506
Long wave damping term, ann (cm^-1) = 2.6159e-008
```

Pause on: hit any key to continue

```
EKE = 6.7745e-003      EPE = 1.3940e-005
Surface EPE = 1.0882e-005
>>
```

## Sample: Running bigrfinch.m

```
>> datnewer = bigrfinch(datnew);
```

First you need to select what you want to change.

```
Options are:
  Grid size: enter "g"
  Initial Frequency guess: w
  Long wave limit: l
  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 n

```
Old # of N^2 points, depth increment (m), exponential tail (km) = 2 2000      1
Enter number of Nsquared values to read (>=1) 2
Enter depth increment for Nsquared values to read (m) 2000
Enter exponential tail scale for Nsquared (km) 1
Old N^2 array (rad^2 sec^-2) 1e-007      1e-007
Enter array of Nsquared values, starting at the surface (sec^-2) [1e-6 1e-6]
>>
```

## **Mfiles in this package:**

This collection of mfiles includes the following:

### ***Mfiles that you can call directly:***

#### **bigromp.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.

#### **bigrsetup.m**

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

#### **bigrfinch.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 bigromp.m (either directly or indirectly):***

#### **bigrdep.m**

Computes a gridded depth profile, along with its derivatives.

#### **bigconchk.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.

**bigrrhocal2.m**

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

**bigrdriver.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.

**bigrengdiag.m**

Does the energy diagnostics for each solution.

**bigruvwr.m**

Called through bigrengdiag.m. Calculates  $u, v, w, \rho$  fields and plots them.

**bigcztosig.m**

Converts z coordinates to sigma coordinates.

**bigrlong.m**

Computes coupling coefficients and normalizations for the long-wave limit.

**bigrtf.m**

Computes perturbation estimate of frictional damping time.

*The following are called by bigrdriver.m:*

**Bigrcc.m**

Computes coefficients for the coastal boundary condition.

**bigroff.m**

Computes coefficients for the offshore boundary condition.

**bigrbot.m**

Computes coefficients for the bottom boundary condition.

**bigrsbc.m**

Computes coefficients for the surface boundary condition.

**bigrmatco.m**

Computes coefficients for interior points on the finite difference grid.

**bigrint.m**

Computes the integral of the response magnitude.

Note: Most of these files were originally written for use with the ‘bigcomp’ system, and then modified for use with strictly real frequency. When they were changed, the names were changed to begin with ‘bigr’. If the code did not change at all, the name was not changed. Hence, not all of the files associated with ‘bigcomp’ begin with ‘bigr’.