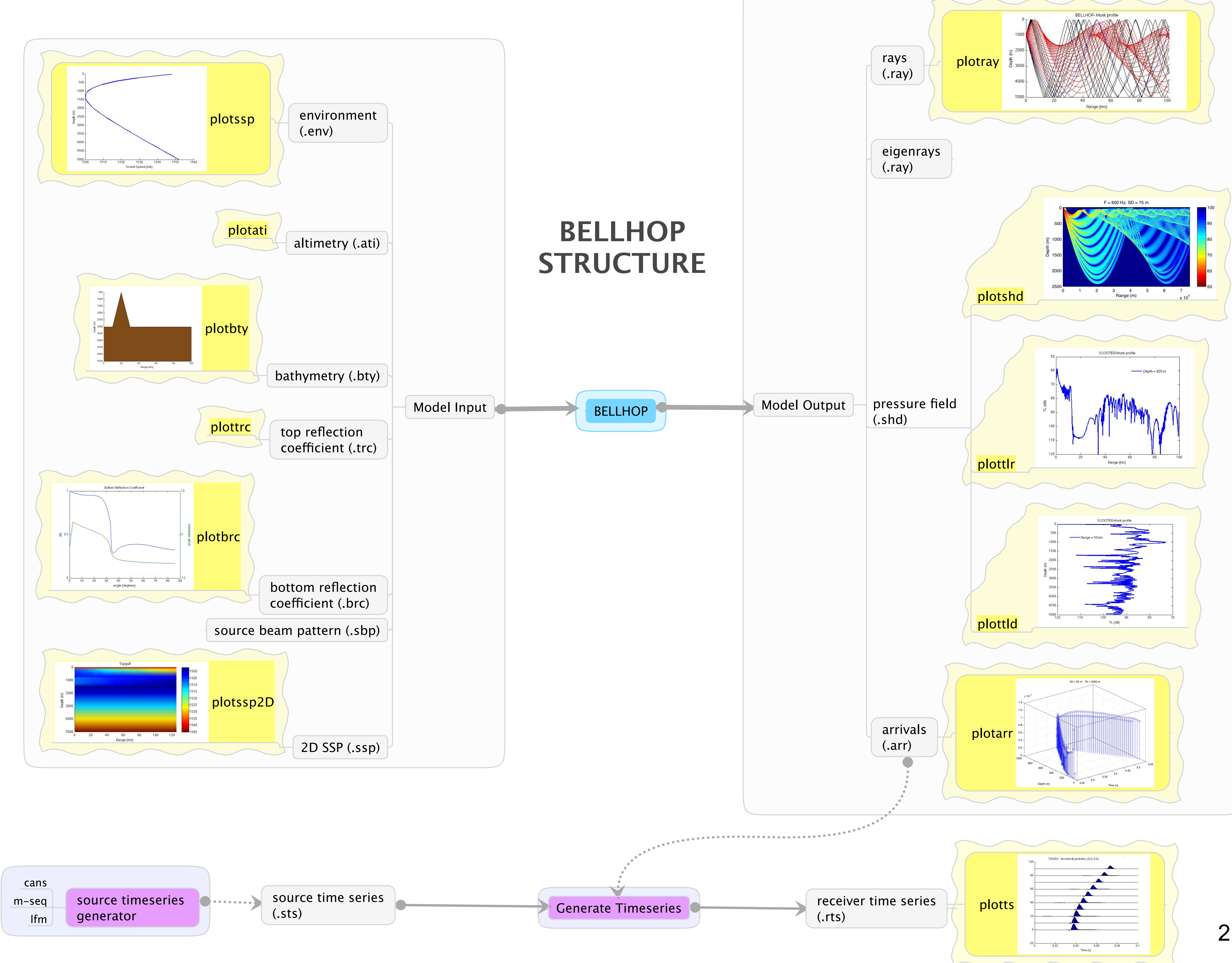
Introduction to BELLHOP

Acoustical Oceanography

Chen-Fen Huang

Institute of Oceanography
National Taiwan University

chenfen@ntu.edu.tw


BELLHOP STRUCTURE
plotssp
environment (.env)
plotati
altimetry (.ati)
plotbty
bathymetry (.bty)
plottrc
top reflection coefficient (.trc)
plotbrc
bottom reflection coefficient (.brc)
source beam pattern (.sbp)
plotssp2D
2D SSP (.ssp)
Model Input
BELLHOP
Model Output
rays (.ray)
plotray
eigenrays (.ray)
pressure field (.shd)
plotshd
plottlr
plottld
arrivals (.arr)
plotarr
cans
m-seq
lfm
source timeseries generator
source time series (.sts)
Generate Timeseries
receiver time series (.rts)
plotts

Overview

BELLHOP computes acoustic fields in oceanic environments via beam tracing. The treated environment consists of an acoustic medium with a sound speed that may depend on range and depth.

A theoretical description may be found in:

- M. B. Porter and Y.-C. Liu, “Finite-element ray tracing,” in *Theoretical and computational acoustics* (D. Lee and M. H. Schultz, eds.), 2, 947–956, 1994.
- M. B. Porter and H. P. Bucker, “Gaussian beam tracing for computing ocean acoustic fields,” *J. Acoust. Soc. Am.*, 82(4), 1349–1359, 1987.

The following programs are used with BELLHOP :

BELLHOP	Main program for doing Gaussian beam tracing
PLOTRAY	Produces plots of central rays of beams
ANGLES	Given the source and receiver sound speeds, computes the angle of the limiting ray.
PLOTSSP	Plots the sound speed profile
DELAYANDSUM	Convolve a source timeseries with the channel impulse response to get the received waveform.

Steps in running BELLHOP

1. Set up your environmental file and run PLOTSSP to make sure the SSP looks reasonable.
2. Do a ray trace:
   a. Run BELLHOP with the ray trace option to calculate about 50 rays.
   b. Run PLOTRAY to make sure you have the angular coverage you expect. Do the rays behave irregularly? If so reduce the step-size and try again.
3. Re-run BELLHOP using the coherent, incoherent or semicoherent option for transmission loss. (Use the default number of beams)
   - Run plotshd.m to plot a full range-depth field plot.
4. Double the number of beams and check convergence.

```
Files:
        Name            Unit          Description
Input
        *.ENV             1        ENVironmental data
Output
        *.PRT             6        PRinT file
        *.RAY            21        RAY   file
        *.SHD            25        SHaDe file
```

Sample Input (Environmental) File:

```
'Munk profile'          ! TITLE
   50.0                    ! FREQ (Hz) (affects the ray step size)
   1                       ! NMEDIA     (always sets to 1)
   'SVN'                   ! SSPOPT (Analytic or C-linear interpolation) OPTIONS 1
   51   0.0   5000.0       ! NMESH, SIGMA, DEPTH of bottom (m)
        0.0   1548.52  /    depth-sound speed pairs
      200.0   1530.29  /
      250.0   1526.69  /
      400.0   1517.78  /
      600.0   1509.49  /
      800.0   1504.30  /
     1000.0   1501.38  /
     1200.0   1500.14  /
     1400.0   1500.12  /
     1600.0   1501.02  /
     1800.0   1502.57  /
     2000.0   1504.62  /
     2200.0   1507.02  /
     2400.0   1509.69  /
     2600.0   1512.55  /
     2800.0   1515.56  /
     3000.0   1518.67  /
     3200.0   1521.85  /
     3400.0   1525.10  /
     3600.0   1528.38  /
     3800.0   1531.70  /
     4000.0   1535.04  /
     4200.0   1538.39  /
     4400.0   1541.76  /
     4600.0   1545.14  /
     4800.0   1548.52  /
     5000.0   1551.91  /
'V'   0.0                  bottom boundary OPTIONS 2
```

BLOCKs 1~6

BLOCK7: SOURCE/RECEIVER DEPTHS AND RANGES

```
1                  ! NSD The number of source depths
1000.0 /           ! SD(1:NSD) (m) The source depths (m)
2                  ! NRD The number of receiver depths
0.0 5000.0 /       ! RD(1:NRD) (m) The receiver depths (m)
501                ! NRR The number of receiver ranges
0.0  100.0 /       ! RR(1:NR ) (km) The receiver ranges (km)
```

BLOCK 8~10

```
'R'                       ! Run-type: 'R/C/I/S' OPTIONS 3
51                        ! NBEAMS
-11.0 11.0 /              ! ALPHA(1:NBEAMS) (degrees)
200.0  5500.0  101.0      ! STEP (m)  ZBOX (m)  RBOX (km)
```

+: ray launched towards the bottom

Options1 (Block 4)

Syntax:

OPTIONS1

Description:

OPTIONS1(1): interpolation method to calculate ss and its derivatives along the ray

'S': cubic spline interpolation;
'C': C-linear interpolation;
'N': N2-linear interpolation;
'A': analytic interpolation;
'Q': quadratic approximation to the sound speed field (requires the creation of a *.ssp file containing the field), mainly for the *range-dependent* environment.

OPTIONS1(2): type of surface

'V': vacuum above surface (SURFACE-LINE is not required);
'R': perfectly rigid media above surface (SURFACE-LINE is not required);
'A': acoustic half-space; SURFACE-LINE should be written as
`z-surface cp-surface cs-surface density-surface alpha-surface/`
'F': read a list of reflection coefficients from a *.trc file (requires running first BOUNCE program).

OPTIONS1(3): attenuation in the bottom

'F': attenuation units correspond to (dB/m)kHz;
'L': attenuation units correspond to the parameter loss;
'M': attenuation units correspond to dB/m;
'N': attenuation units correspond to Nepers/m;
'Q': attenuation units correspond to Q-factor;
'W': attenuation units correspond to dB/wavelength.

Options1 (Block 4)

OPTIONS1(4): Thorpe volume attenuation in the water column using 'T'.

OPTIONS1(5): surface shape

' ': flat surface;

'*': sea surface coordinates described in a *.ati file, with the following structure:

```
interpolation type
npoints
r(1) z(1)
r(2) z(2)
. .
. .
. .
r(npoints) z(npoints)
```

surface ranges r() in km,
surface depths z() in m

interpolation type

- 'L': for a linear interpolation of the surface;
- 'C': for curvilinear interpolation;

Options2 (Block 6)

Syntax:

OPTIONS2 a two-character string

Description:

OPTIONS2(1): type of media below the water column

'V': vacuum below water column (the BOTTOM-LINE is not required);

'R': rigid below water column (the BOTTOM-LINE is not required);

'A': acoustic half-space; the BOTTOM-LINE should be written as

`z-bottom cp-bottom cs-bottom density-bottom alpha-bottom/`

for obvious reasons z-bottom should be equal to z(nssp); cs(bottom) is ignored, density(bottom) should be specified in g/cm^3 and the units of alpha(bottom) depend on OPTIONS1(3);

'F': a list of reflection coefficients from a *.brc file.

OPTION2(2): shape of the bottom

‘ ’ : a flat bottom;

'*': use a *.bty file to describe the bottom coordinates (similar to the *.ati file).

Options3: Run Type (Block 8)

Syntax:

OPTIONS3

Description:

OPTIONS3(1): output type

- 'R': generates a ray file;
- 'E': generates an eigenray file;
- 'A': generates an amplitude-delay file (ascii);
- 'a': generate an amplitude-delay file (binary);
- 'C': Coherent TL calculation;
- 'I’: Incoherent TL calculation;
- 'S': Semicoherent TL calculation (Lloyd mirror source pattern).

OPTIONS3(2): the approximation used to calculate acoustic pressure

- 'G': Geometric beams (Hat-shaped beam; default);
- 'B': Gaussian beams;
- 'C': Cartesian beams;
- 'R': Ray-centered beams.

OPTIONS3(3): inclusion of the beam shift effect

- ‘ ': do not include beam shift effect (default);
- 'S': include beam shift effect;
- '*': use a source beam pattern file (requires a *.sbp file, similar to the *.ati and *.bty files, with angles in degrees and amplitudes, instead of ranges and depths).

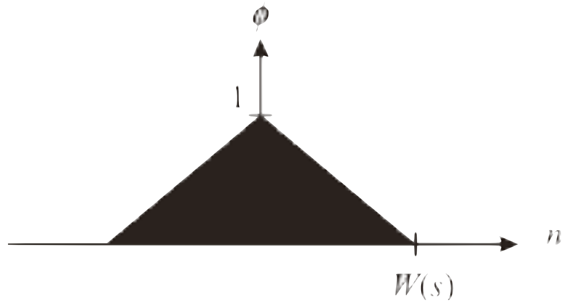
Options3: Run Type (Block 8)

OPTIONS3(4): source type
'R': point source (cylindrical coordinates) (default);
'X': line source (cartesian coordinates).

OPTIONS3(5): array type
'R': rectilinear receiver grid, receivers at rr(:) X rd(:) (default);
'I': irregular grid, receivers at rr(:) , rd(:).

The ray file and eigenray files have the same simple ascii format and can be plotted using the Matlab script PLOTRAY.M.

Beam Type



- An additional consideration is the role of the beam type. The way BELLHOP does such a calculation is to trace a fan of rays and write an arrival for every beam that comes within a beamwidth of the receiver.
  - **Hat-shaped beam (default):** you may see that arrivals come in pairs, corresponding to a ray tube that encloses the receiver. However, such rays have nearly identical travel times, and there is also logic inside BELLHOP to try and combine such pairs into a single arrival.
  - **Gaussian beam:** you will get many arrivals corresponding to a collection of rays with nearly the same take-off angle, but which pass within the zone of influence of the Gaussian beam. Technically, the Gaussian has an influence at any distance from the central ray; however, BELLHOP cuts it off at a point where the beam influence is negligible.
- **Which choice of beam to use:** The considerations are really the same as for a TL calculation. The Gaussian beam option is usually more accurate; however, it will lead to many more arrivals. Therefore the arrivals file is also much larger and takes longer to post-process.

Run Type

E: The eigenray option seems to generate a lot of questions. The way this works is that BELLHOP simply writes the trajectories for all the beams that contribute at a given receiver location. To get a useful picture you normally want to use a very fine fan, only one receiver location, and the Geometric beam option. See the examples in at/tests.

A: The amplitude-delay file can be used with the Matlab script DELAYANDSUM.M to 'combine echoes of the source waveform based on their amplitude and arrival time', i.e. to convolve them with the source timeseries and plot the channel response.

C/I/S: For TL calculations, the output is in the shdfil format used by all the codes in the Acoustics Toolbox and can be plotted using the Matlab script, PLOTSHD.M. The pressure field is normally calculated on a rectilinear grid formed by the receiver ranges and depths. If an irregular grid is selected, then the receiver ranges and depths are interpreted as a coordinate pair for the receivers. This option is useful for reverberation calculations where the receivers need to follow the bottom terrain.

There are actually several different types of beam options [OPTIONS3(2)] implemented in the code. Only the **Hat-shaped beam** & **Gaussian beam** are fully maintained.

Beam Fan (Block 9)

The SOURCE BEAM PATTERN file has the format

```
NSBPPts
angle1  power1
angle2  power2
 ...
```

with angle in degrees and power in dB. To match a standard point source calculation one would use anisotropic source with 0 dB for all angles. (See at/tests/BeamPattern for an example.)

Syntax:

```
NBEAMS ISINGLE
ALPHA(1:NBEAMS)
```

Description:

NBEAMS: Number of beams (use 0 to have the program calculate a value automatically, but conservatively)

ISINGLE: If the option to compute a single beam in the fan is selected (top option), then this selects the index of the beam that is traced

ALPHA(): Beam angles (negative angles toward surface)

For a ray trace you can type in a sequence of angles or you can type the first and last angles followed by a '/'. For a TL calculation, the rays must be equally spaced otherwise the results will be incorrect.

Numerical Integrator Info (Block 10)

Syntax:

STEP ZBOX RBOX

Description:

STEP: The step size used for tracing the rays (m).
ZBOX: The maximum depth to trace a ray (m).
RBOX: The maximum range to trace a ray (km).

The required step size depends on many factors including frequency, size of features in the SSP (such as surface ducts), range of rcvrs, and whether a coherent or incoherent TL calculation is performed.

If you use STEP=0.0 BELLHOP will use a default step-size and tell you what it picked.

You should then halve the step size until the results are convergent to your required accuracy.

To obtain a smooth ray trace you should use

- *spline SSP interpolation*
- *a step-size less than the smallest distance between SSP data points*

Rays are traced until they exit the box (ZBOX, RBOX). By setting ZBOX less than the water depth you can eliminate bottom reflections. Make ZBOX, RBOX a bit (say 1%) roomy to make sure rays are not killed the moment they hit the bottom or are just reaching your furthest receiver.

Running BELLHOP (1)

- The main issue: ray tracing is very sensitive to environmental interpolation (both boundary and volume).
  - The Gaussian beam option reduces that sensitivity significantly; however, one should still be attentive to this issue.
  - The spline interpolation option to the SSP should be used with particular caution. In some cases, the spline fit is very smooth as desired; in other cases, the spline introduces large wiggles between ssp points, in its effort to produce a smooth curve. Use PLOTSSP to see how your fit looks.
- BELLHOP numerically integrates the ray equations to trace a ray through the ocean. To avoid artifacts at discontinuties in the SSP, the step size is dynamically adjusted to make sure a step always lands on an SSP point, rather than stepping over it. (The beam curvature needs to be adjusted at each such point.) Use **fewer** points to describe the SSP than necessary to capture the physics because BELLHOP will end up using lots of small steps to have each ray land on the SSP points. Similarly, BELLHOP uses the altimetry and bathymetry points to define segments in range, and adjusts the step size so that the rays land on each segment boundary.

Running BELLHOP (2)

- You can have BELLHOP use a **range-dependent SSP** by creating a separate SSPFIL containing that SSP data in a matrix form. The range-dependent SSPFIL is read if you select 'Q' (quadrilateral) for the SSP interpolation. The depths for the SSP points are read from the ENVFIL; the ranges are specified in the SSPFIL. See the example in at/ tests/Gulf.
  - To get your best calculation with a 2D SSP, you should make sure the rays step on and not over the profile ranges. To do this, include a bathymetry or altimetry file that has those profile ranges amongst its samples. Those files will then dictate the range stepping during the ray trace.
- **Range-dependent bathymetry** is included by a separate BTYFIL containing the bathymetry. The file is read if you include an additional bottom option '*' as the flag to tell BELLHOP to read the supplemental bathymetry file '*.bty.'
- BELLHOP will produce some artifacts for receivers very close the surface or bottom, because a beam is essentially folded onto itself upon reflection. The zone of overlap (which depends on the fatness of the beam) is not treated with a lot of care. You can minimize such artifacts by making the beams narrow, which in turn can often be done by using lots of rays. If you want to explore some behavior for a receiver on the bottom, you generally should offset it a little bit.
- BELLHOP has no direct capability for modeling elastic wave propagation; however, elastic boundaries can often be treated using BOUNCE to generate an equivalent reflection coefficient.