The BELLHOP Manual and User's Guide: PRELIMINARY DRAFT

Michael B. Porter
Heat, Light, and Sound Research, Inc.
La Jolla, CA, USA

January 31, 2011

Abstract

BELLHOP is a beam tracing model for predicting acoustic pressure fields in ocean environments. The beam tracing structure leads to a particularly simple algorithm. Several types of beams are implemented including Gaussian and hat-shaped beams, with both geometric and physics-based spreading laws. BELLHOP can produce a variety of useful outputs including transmission loss, eigenrays, arrivals, and received time-series. It allows for range-dependence in the top and bottom boundaries (altimetry and bathymetry), as well as in the sound speed profile. Additional input files allow the specification of directional sources as well as geoacoustic properties for the bounding media. Top and bottom reflection coefficients may also be provided. BELLHOP is implemented in Fortran, Matlab, and Python and used on multiple platforms (Mac, Windows, and Linux). This report describes the code and illustrates its use.

Contents

1 Map of the BELLHOP program

1.1 Input

The overall structure of BELLHOP is shown in Fig. (1). Various files must be provided to describe the environment and the geometry of sources and receivers. In the simplest case, which is also typical, there is only one such file. It is referred to as an environmental file and includes the sound speed profile, as well as information about the ocean bottom. However, if there is a range-dependent bottom, then one must add a bathymetry file with range-depth pairs defining the water depth. Similarly, if there is a range-dependent ocean sound speed, the one must an an SSP file with the sound speed tabulated on a regular grid. Further, if one wants to specify an arbitrary bottom reflection coefficient to characterize the bottom, then one must provide a bottom reflection coefficient file with angle-reflection coefficient pairs defining the reflectivity. Similar capabilities are implemented for the surface. Thus there is the option of providing a top reflection coefficient and a top shape (called an altimetry file).

Usually one assumes the acoustic source is omni-directional; however, if there is a source beampattern, then one must provide a source beam pattern file with angle-amplitude pairs defining it.

BELLHOP reads these files depending on options selected within the main environmental file.

Plot programs (plotssp, plotbty, plotbrc, etc.) are provided to display each of the input files.

1.2 Output

BELLHOP produces different output files depending on the options selected within the main environmental file.

Usually one starts with a ray tracing option, which produces a file containing a fan of rays emanating from the source. If the eigenray option is selected, then the fan is winnowed to include only the rays that bracket a specified receiver location. The file format is identical to that used in the standard ray-tracing option. Ray files are usually used to get a sense of how energy is propagating in the channel. The program plotray is used to display these files.

Usually one is interested in calculating the transmission loss for a tonal source (or for a single tone of interest in a broadband waveform). The transmission loss is essentially the sound intensity due to a source of unit strength. The transmission loss information is written to a shade file which

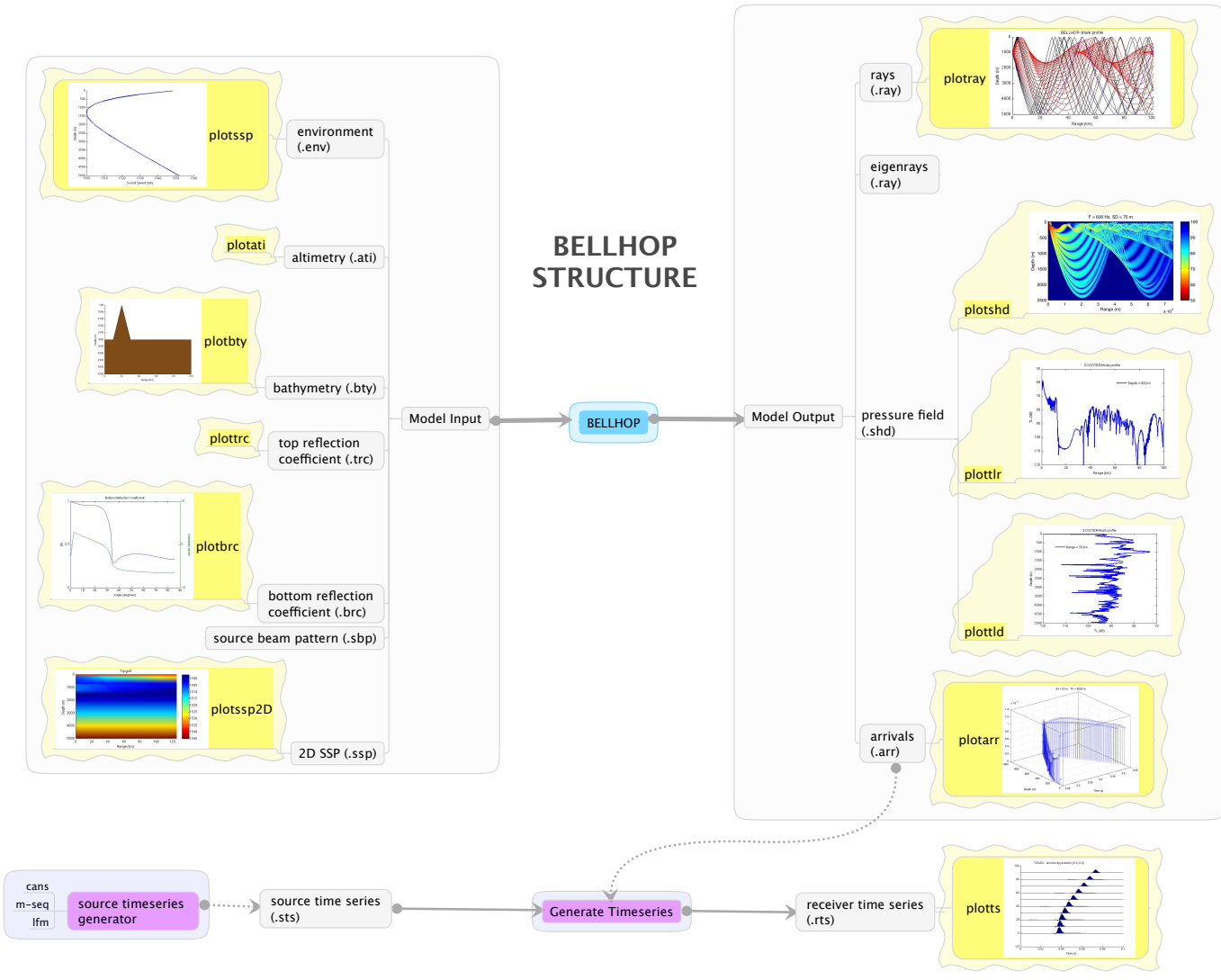


Figure 1: BELLHOP structure.

can be displayed as a 2D surface using plotshd, or in range and depth slices, using plottlr and plottld respectively.

If one wants to get not just the intensity due to a tonal source, but the entire timeseries then one selects an arrivals calculation. The resulting arrivals file contains amplitude-delay pairs defining the loudness and delay for every echo in the channel. This information can be plotted using plotarr to show the echo pattern. Alternatively it can be passed to a convolver, which sums up the echoes of a particular source timeseries to produce a receiver timeseries. The program plotts can be used to plot either the source or receiver timeseries.

2 Sound speed profile and ray trace

As a first example, we consider a deep water case with the Munk sound speed profile. Usually one should start by plotting the sound speed profile and doing a ray trace. The input file (also called an environmental file) is a simple text file created using any standard text editor and must have a '.env' extension. It is usually easiest to start from one of the example files. Here we consider at/tests/Munk/MunkB_ray.env:

MunkB_ray.env

```
'Munk profile'          ! TITLE
50.0                    ! FREQ (Hz)
1                       ! NMEDIA
'SVF'                   ! SSPOPT (Analytic or C-linear interpolation)
51  0.0  5000.0        ! DEPTH of bottom (m)
    0.0  1548.52  /
  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  /
'A' 0.0
 5000.0  1600.00 0.0 1.0 /
1                               ! NSD
1000.0 /                        ! SD(1:NSD) (m)
51                              ! NRD
0.0 5000.0 /                    ! RD(1:NRD) (m)
```

```
1001                                ! NR
0.0  100.0 /                        ! R(1:NR ) (km)
'R'                                 ! 'R/C/I/S'
41                                  ! NBeams
-20.0 20.0 /                        ! ALPHA1,2 (degrees)
0.0  5500.0  101.0                  ! STEP (m), ZBOX (m), RBOX (km)
```

The input file is read using list-directed i/o, so the data does not need to be precisely positioned on each line. As a convenience we also append comments, preceded by '!'. These are optional and are not read by the program.

The source frequency (line 2) is not terribly important for the basic ray trace. The rays are frequency independent; however, the frequency can have an impact on the ray step size, since the code assumes more accurate ray trajectories will be needed at higher frequencies.

NMedia (line 3) is always set to one in BELLHOP . This parameter is included for compatibility with other models in the Acoustics Toolbox, which are capable of handling multi-layered problems.

The top option (line 4) is next specifed as 'SVF' indicating that a spline fit should be used to interpolate the sound speed profile; that the ocean surface is modeled as a vacuum; and that all attenuation values are specified in dB/mkHz. We chose the spline fit here knowing that the profile is smoothly varying. In such cases, the spline fit produces smoother looking ray trace plots.

The only important parameter in the next line (5) is the bottom depth (5000 m), which indicates the last line that needs to be read in the sound speed profile. The first two parameters are not used by BELLHOP .

Next we see a sequence of depth-soundspeed pairs defining the ocean soundspeed profile. The last value in the soundspeed profile must start with the previously specified value for the bottom depth. To ensure compatibility with the other models in the Acoustics Toolbox, we normally terminate each line with a '/'. The other models are expecting attenuations, shear speeds, and a density as additional parameters and the '/' tells them to stop reading the line and use default values.

Next we have two lines specifying the bottom boundary. The option letter 'A' indicates that the bottom is to be modeled as an Acousto-Elastic halfspace. The lines following specify that halfspace as having a sound speed of 1500 m/s and unit density (which is not very realistic).

The next 6 lines specify the source depths, receiver depths, and receiver ranges. Depths are always specified in meters and ranges in kilometers.

For our first run, we are producing a ray trace so the receiver locations are irrelevant; however, they do need to be provided. Note also that 51 receiver depths have been specified. Often the user simply wants a uniform distribution of receiver depths to display the acoustic field. To avoid forcing the user to type in all those numbers one has the option of simply putting in the first and last values and terminating the line with a '/'. The code detects the premature termination and then produces a full set of receivers by interpolation. The sources and receivers must lie within the interior of the waveguide.

The choice of units is motivated by typical ocean acoustic scenarios. However, fundamentally the code is simply solving the wave equation so any self-consistent set of units could be used.

Next is the RunType (line 41). For a raytrace run, we select option 'R'. The following lines then specify the fan that will be used, given as a number of rays, together with the angular limits in degrees. We follow a convention that the angles are specified in declination, i.e. zero degrees is a horizontally launched ray, and a positive angle is a ray launched towards the bottom.

For a ray trace run, the plot usually becomes too cluttered if we use more than about 50 rays. This is a matter of taste. Likewise, the angular limits are determined by what part of the field the user is interested in seeing.

The last line (44) specifies the step size in meters used to trace a ray, along with the depth and range of a box beyond which no rays are traced. Usually, a step size of 0 should be selected and then BELLHOP will make an automatic selection of about a tenth of the water depth. Regardless of what step size is selected, BELLHOP dynamically adjusts the step size as the ray is traced, to ensure that each ray lands precisely on all depths where a sound speed is given. Thus, the sound speed profile itself usually controls the ray step size. If you provide more sound speed points than are necessary, BELLHOP will similarly run slower. On the other hand, for a given sampling of the sound speed profile, you may be able to obtain a more accurate ray trace by specifying a step size that is smaller than the default value.

Now that the input file has been created, we can start by plotting the soundspeed profile, using the Matlab routine plotssp.m. The syntax of the Matlab command to run this is:

```
plotssp 'MunkB_ray'
```

where 'MunkB_ray.env' is the name of the BELLHOP input file. This produces the plot in Fig. (2).

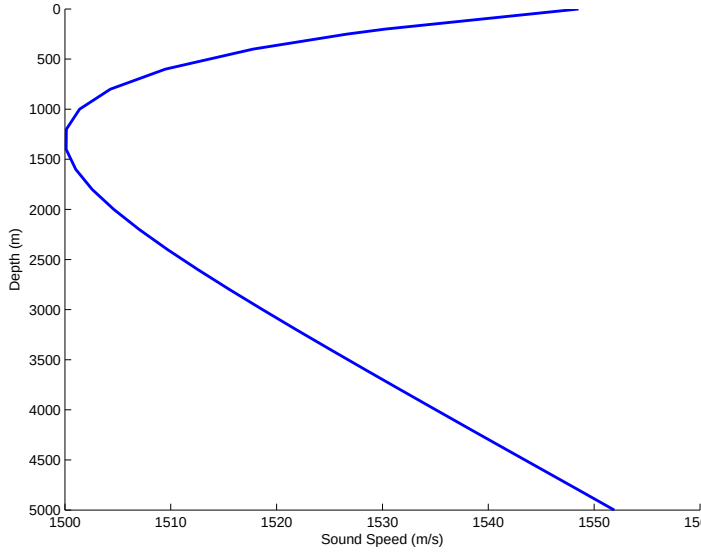


Figure 2: The Munk sound speed profile.

We started first with the sound speed profile plot, to introduce the scenario in a logical fashion. However, in practice it is recommended to do a trial BELLHOP run on the input file first. BELLHOP will produce a print file as show below, which echoes the input data in a clear format. In addition, it will stop at the first place it encounters something unintelligible. Thus, by examining the print file one can usually see clearly any formatting errors.

MunkB_ray.prt

```
BELLHOP- Munk profile
frequency =    50.00     Hz

Dummy parameter NMedia =            1

    SPLINE approximation to SSP
    Attenuation units: dB/mkHz
    VACUUM

Depth =    5000.0000000000000      m

Spline SSP option

Sound speed profile:
     0.00      1548.52
   200.00      1530.29
   250.00      1526.69
   400.00      1517.78
```

```
   600.00      1509.49
   800.00      1504.30
  1000.00      1501.38
  1200.00      1500.14
  1400.00      1500.12
  1600.00      1501.02
  1800.00      1502.57
  2000.00      1504.62
  2200.00      1507.02
  2400.00      1509.69
  2600.00      1512.55
  2800.00      1515.56
  3000.00      1518.67
  3200.00      1521.85
  3400.00      1525.10
  3600.00      1528.38
  3800.00      1531.70
  4000.00      1535.04
  4200.00      1538.39
  4400.00      1541.76
  4600.00      1545.14
  4800.00      1548.52
  5000.00      1551.91

                                ( RMS roughness =   0.00     )
    ACOUSTO-ELASTIC half-space
  5000.00      1600.00      0.00     1.00       0.0000    0.0000

Number of sources   =            1
Source depths (m)
  1000.00

Number of receivers =           51
Receiver depths (m)
  0.00000       100.000       200.000       300.000       400.000
  500.000       600.000       700.000       800.000       900.000
  1000.00       1100.00       1200.00       1300.00       1400.00
  1500.00       1600.00       1700.00       1800.00       1900.00
  2000.00       2100.00       2200.00       2300.00       2400.00
  2500.00       2600.00       2700.00       2800.00       2900.00
  3000.00       3100.00       3200.00       3300.00       3400.00
  3500.00       3600.00       3700.00       3800.00       3900.00
  4000.00       4100.00       4200.00       4300.00       4400.00
  4500.00       4600.00       4700.00       4800.00       4900.00
  5000.00

Number of ranges   =         1001
Receiver ranges (km)
  0.00000      0.100000      0.200000      0.300000      0.400000
```

```
 0.500000      0.600000      0.700000      0.800000      0.900000
  1.00000       1.10000       1.20000       1.30000       1.40000
  1.50000       1.60000       1.70000       1.80000       1.90000
  2.00000       2.10000       2.20000       2.30000       2.40000
  2.50000       2.60000       2.70000       2.80000       2.90000
  3.00000       3.10000       3.20000       3.30000       3.40000
  3.50000       3.60000       3.70000       3.80000       3.90000
  4.00000       4.10000       4.20000       4.30000       4.40000
  4.50000       4.60000       4.70000       4.80000       4.90000
  5.00000
 ...   100.000000

Ray trace run
Geometric beams
Point source (cylindrical coordinates)
Rectilinear receiver grid: Receivers at rr( : ) x rd( : )

Number of beams   =           41
Beam take-off angles (degrees)
 -20.0000      -19.0000      -18.0000      -17.0000      -16.0000
 -15.0000      -14.0000      -13.0000      -12.0000      -11.0000
 -10.0000      -9.00000      -8.00000      -7.00000      -6.00000
 -5.00000      -4.00000      -3.00000      -2.00000      -1.00000
  0.00000       1.00000       2.00000       3.00000       4.00000
  5.00000       6.00000       7.00000       8.00000       9.00000
  10.0000       11.0000       12.0000       13.0000       14.0000
  15.0000       16.0000       17.0000       18.0000       19.0000
  20.0000

Step length,     deltas =    500.00000000000000      m

Maximum ray Depth, zBox =    5500.0000000000000      m
Maximum ray range, rBox =    101000.00000000000      m
No beam shift in effect



CPU Time =       0.781E-01s
```

The Matlab command to run BELLHOP is:

bellhop 'MunkB_ray'

where 'MunkB_ray.env' is the name of the input file. Assuming a successful completion, BELLHOP produces a print-file called 'MunkB_ray.prt' and a ray file called 'MunkB_ray.ray'. One should carefully examine the print file to verify that the problem was set up as intended and that BELLHOP ran to

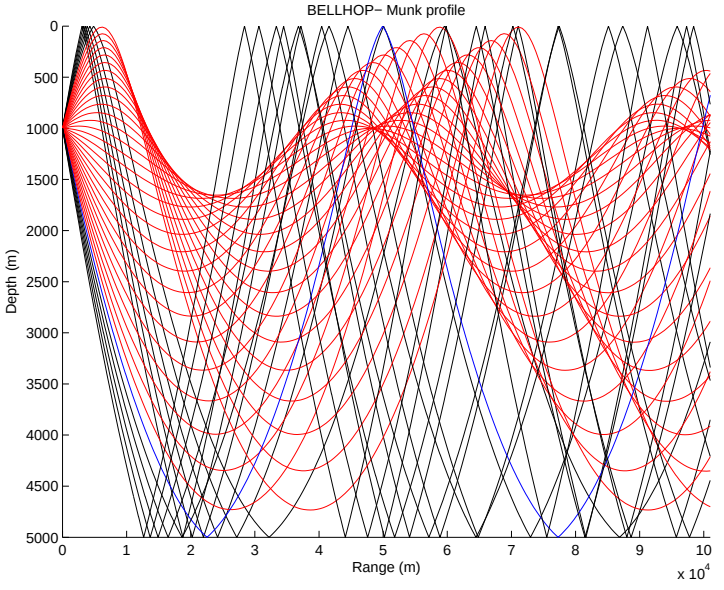


Figure 3: Ray trace for the Munk sound speed profile.

completion. The latter can be verified by checking that there are no error messages in the print file, and that the last line of the print file shows the CPU time used.

The next step is to plot the rays using the Matlab command:

```
plotray 'MunkB_ray'
```

This produces the plot in Fig. (3). Notice that the range axis is in meters. If kilometers are preferred, then one simply sets the global Matlab variable:

```
units = 'km'
```

The rays have are plotted using different colors depending on whether the ray hits one or both boundaries. The number of surface and bottom bounces are written to the ray file so it is simple to modify plotray to color code the rays in whatever way best illustrates the propagation physics.

3 Eigenray plots

BELLHOP can also produce eigenray plots showing just the rays that connect the source to a receiver. To do this, one simply changes the RunType to 'E'. However, to run this reliably one should understand the way this is implemented. The code does exactly the same computation as is done for a regular ray trace; however, it only saves the rays to the ray file, whose associated beams makes a contribution to the specified receiver points. There are many implications in this statement. First, one should be aware of which beam type is used. For a true eigenray calculation one should use the default beam, which has a beamwidth defined by the ray tube formed by adjacent rays. We call that a geometric beam. The default beam also has a hat-shape in the traditional finite element style, so that it vanishes outside the neighboring rays of the central ray of the beam. Other beam types, such as the Cerveny, Popov, Psencik beams are generally much broader beams, and so one would get lots of additional rays that pass at greater distances from the receiver. When we use the default beam type, the rays that are written will be only the bracketing rays for the receiver location.

Second, one typically needs to use a much finer fan. For instance, if one used 41 rays as we did in the previous example, then the rays are quite spread out at long ranges. Then when we save the bracketing rays, they may still miss the receiver location by a wide margin. For this example, we therefore increase the number of rays to 5001. The more rays used, the more precise the eigenray calculation will be. However, the run time will increase accordingly.

Finally, one should generally do an eigenray calculation with just a single source and receiver. Otherwise, the resulting ray plot would be too cluttered. The input file MunkB_eigenray.env with these changes is shown below. The eigenrays are plotted using the usual plotray command, yielding the plot in Fig. (4).

MunkB_eigenray.env

```
'Munk profile'          ! TITLE
50.0                    ! FREQ (Hz)
1                       ! NMEDIA
'CVF'                   ! SSPOPT (Analytic or C-linear interpolation)
51  0.0  5000.0        ! DEPTH of bottom (m)
    0.0  1548.52  /
  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  /
'A' 0.0
 5000.0  1600.00 0.0 1.0 /
1                                ! NSD
1000.0 /                         ! SD(1:NSD) (m)
1                                ! NRD
800.0 /                          ! RD(1:NRD) (m)
1                                ! NR
100.0 /                          ! R(1:NR ) (km)
'E'                              ! 'R/C/I/S'
5001                             ! NBeams
-25.0 25.0 /                     ! ALPHA1,2 (degrees)
0.0  5500.0  101.0               ! STEP (m), ZBOX (m), RBOX (km)
```

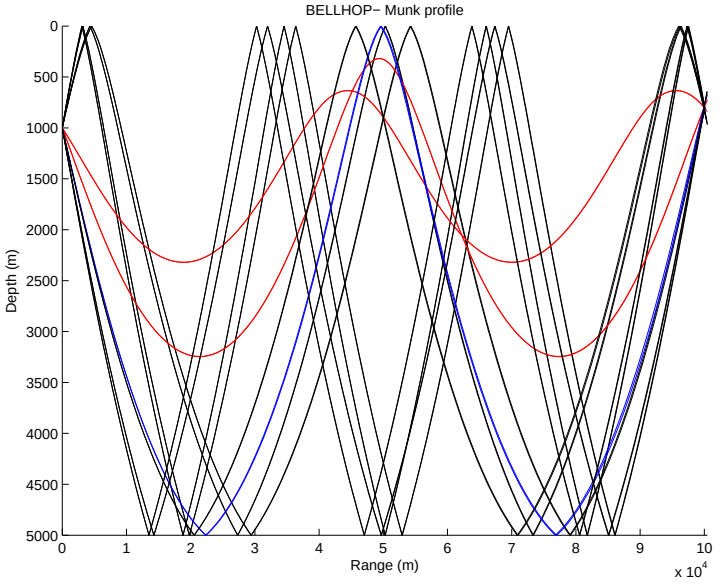


Figure 4: Eigenrays for the Munk sound speed profile with the source at 1000 m and the receiver at 800 m.

4 Transmission Loss

We can calculate transmission loss by selecting RunType = 'C' as shown in the listing below. As we will discuss in more detail shortly, 'C' stands for Coherent pressure calculations. The pressure field, p, is then calculated for the specified grid of receivers, with a scaling such that $20\log_{10}(|p|)$ is the transmission loss in dB. One may also select multiple source depths, in which case BELLHOP does a run in sequence for each source depth. The frequency (here 50 Hz) is now a very important parameter, since the interference pattern is directly related to the wavelength. The frequency also affects the attenuation, when present.

The number of beams, NBeams, should normally be set to 0, allowing BELLHOP to automatically select the appropriate value. The number needed increases with frequency and the maximum range to a receiver. To understand this one may imagine a point source in free space. The beam fan expands as we go away from the source. Meanwhile, the field at a given point is essentially an interpolation between adjacent beams. To accurately interpolate we need the wavefronts of adjacent beams to be sufficiently close.

MunkB_coh.env

```
'Munk profile, coherent'        ! TITLE
50.0                            ! FREQ (Hz)
1                               ! NMEDIA
'CVW'                           ! SSPOPT (Analytic or C-linear interpolation)
51  0.0  5000.0                ! DEPTH of bottom (m)
    0.0  1548.52  /
  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  /
'A' 0.0
 5000.0  1600.00 0.0 1.8 0.8 /
1                       ! NSD
1000.0 /                ! SD(1:NSD) (m)
201                     ! NRD
0.0 5000.0 /            ! RD(1:NRD) (m)
501                     ! NR
0.0  100.0 /            ! R(1:NR ) (km)
'C'                     ! 'R/C/I/S'
0                       ! NBEAMS
-20.3 20.3 /            ! ALPHA1, 2 (degrees)
0.0  5500.0  101.0      ! STEP (m), ZBOX (m), RBOX (km)
```

BELLHOP tends to be conservative in selecting the number of beams, assuming that the details of the interference pattern in the acoustic field are important. That makes sense for benchmarking applications, or often for lower frequencies. However, as we go to higher frequencies, e.g. 10 kHz, the environmental uncertainty in a real world case, makes it impossible to replicate these patterns (as they would be observed in a sea test) in detail. You may then wish to experiment with reducing the number of beams to save run time.

The next step is to plot the field using the Matlab command:

```
plotshd 'MunkB_Coh'
```

However, we can also do a multipanel plot using:

```
plotshd( 'MunkB_Coh', 2, 2, 1 )
```

where '2, 2, 1' tells Matlab that we wish to use the first panel in a 2x2 set. We repeat for several other cases discussed shortly, to produce the plot in Fig. (5). The lower two TL plots are reference solutions calculated by KRAKEN and SCOOTER respectively, which are other models in the Acoustics Toolbox. They may be considered exact. Most people would consider the agreement between BELLHOP and the other models to be excellent. People are also usually surprised to see that this occurs at 50 Hz, which is considered a low frequency in a certain taxonomy. Ray/beam methods are

based on high-frequency asymptotics so the incorrect prejudice is that they are not suitable for this sort of problem. We observe that fine details of the interference pattern are reproduced correctly. That interference pattern results from the ensemble of propagating beams and they must all add up in the right place and with the right phase. Thus, this case provides a rigorous test of the numerics.

Nevertheless, we can identify some well-known artifacts of classical ray theory, namely the perfect shadows and the caustics. These are places where the field goes to zero of infinity, respectively. For better accuracy, we invoke the Gaussian beam option.

The specific type of beam used is selected using the second letter in RunType. If, as in the above, we omit the second letter, the code uses option 'G' for geometric beams. Selecting RunType = 'CB' (B for beam), yields the result in the upper right panel. This produces some leakage energy in the shadow zones and smooths out the caustics. In general, we find this option produces more accurate TL plots. However, the geometric beam option is left as the default since people are usually most familiar with that approach.

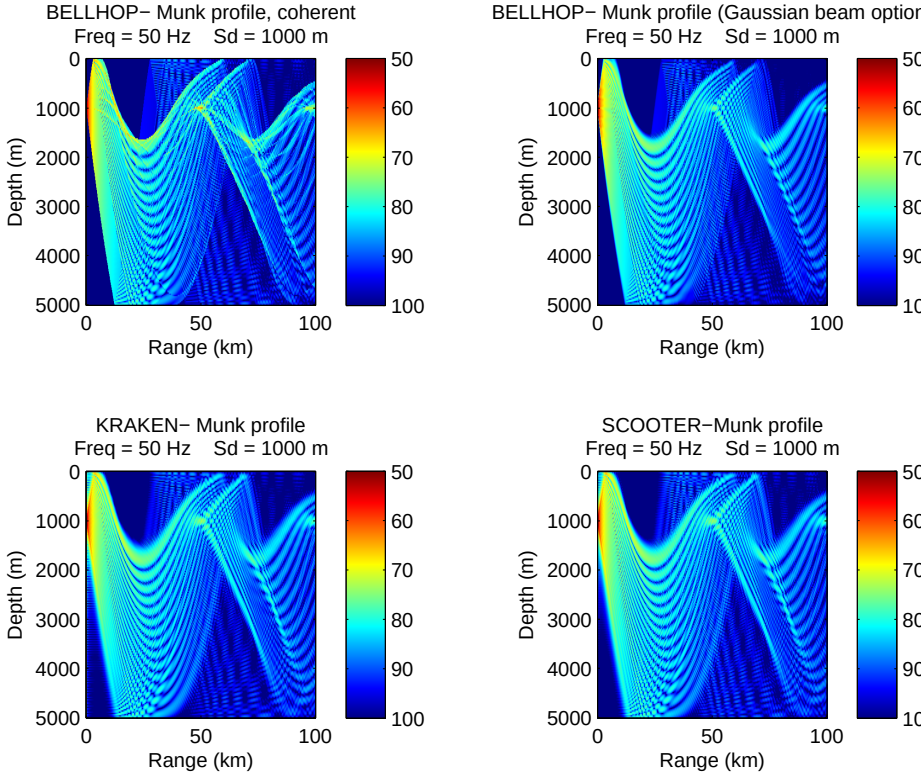


Figure 5: Transmission loss for the Munk sound speed profile using a) geometric beams, b) Gaussian beams, c) KRAKEN normal modes, d) SCOOTER wavenumber integration.

4.1 Coherent, Semicoherent, and Incoherent TL

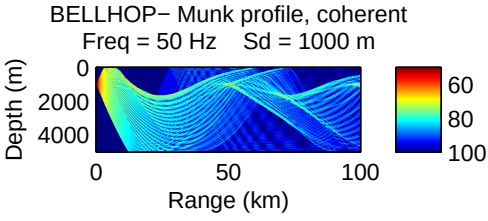
As discussed above, RunType = 'C' produces a so-called coherent TL calculation. By simply changing that first option letter to 'S', or 'I' we produce *semi-coherent* and *incoherent* TL calculations, respectively. For each of these options one may use the second letter to select the type of beam (geometric or Gaussian). The full set of combinations is shown in Fig. (6).

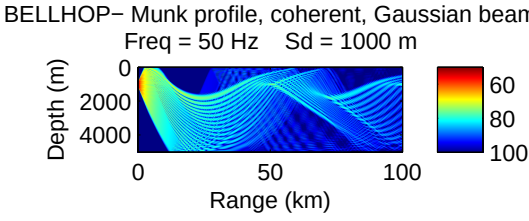
The motivation for the incoherent TL option is that sometimes the details of the interference pattern are not meaningful. For instance, if we consider an acoustic modem operating in the 10-15 kHz band, then the interference pattern will vary widely across the band. It will also be very sensitive to details of the sound speed profile that are not measurable. The individual TL plots are representative samples of what might be seen at a given frequency, but cannot be considered as deterministic forecasts. With that in mind, one may be just as happy to get a TL averaged (loosely speaking) across all the frequencies.

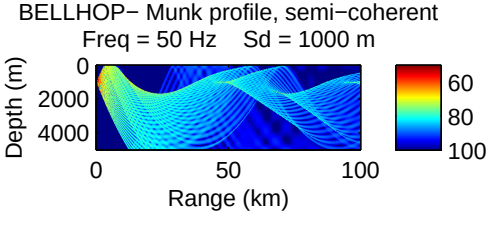
There are many ways to get such averaged TL surfaces. In ray models, we typically throw away the phase of each path, while in mode models we throw away the phase of the individual modes. The effects are not the same. However, these are both reasonable approaches to capturing a smoothed energy level. Hence, the very smoothed figures seen at the bottom of Fig. (6). The user must decide for herself which display captures the relevant information for her particular application.

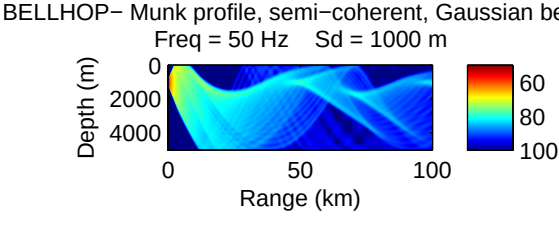
The middle panel shows a semi-coherent calculation which preserves some, but not all of the interference effects. The motivation for this option is that one may have a mid-frequency source near the surface. The frequency is sufficiently high that many of the interference effects are not significant or reliably predictable. However, a core feature is the interference with the surface image, i.e. the reflection of the source in the mirror formed by the ocean surface. Because the source is close to the surface this basic radiation pattern is a stable feature even at higher frequencies. The semi-coherent option captures this effect by putting a Lloyd mirror pattern in for the source beam pattern, but throwing away the phase of the rays. In practice, we have rarely used this option.

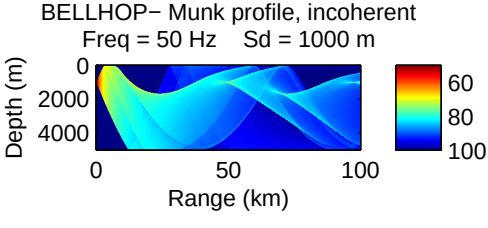
The incoherent and semi-coherent TL options attempt to capture less of the detail of the acoustic field. As a result, BELLHOP can be run with less stringent accuracy requirements (fewer beams and larger step sizes). This in turn can save on the run time.











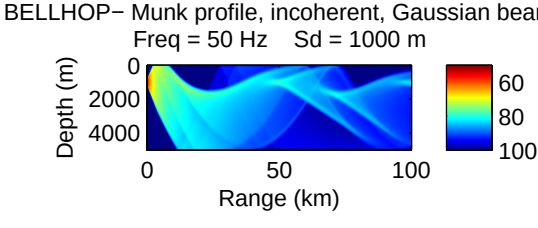


Figure 6: Transmission loss for the Munk sound speed profile using (top to bottom) coherent, semi-coherent, and incoherent TL calculations; and left to right geometric and Gaussian beams.

5 Directional Sources

We consider first a point source in free space (in a homogeneous medium). The BELLHOP environmental file uses a sound speed profile going from ±10000 m in depth. The depth interval is selected there to cover the range over which we are interested in seeing the field, since BELLHOP will only calculate the field within the waveguide.

To make sure there are no boundary reflections, we use the Acousto-Elastic halfspace option with a sound speed and density precisely matching that within the water column. We set NBeams and Step to zero, letting the code automatically select appropriate values. The environmental file for this case is shown below and the resulting TL plot is shown in the upper panel of Fig. (7).

omni.env

```
'Point source in free space'
100.0                  ! Frequency (Hz)
1
'CAF'
-10000.0 1500.0 0.0 1.0 /
1500 0.0 10000.0
-10000.0 1500.0 0.0 1.0 /
 10000.0 /
'A' 0.0
/
1                                ! NSD
0.0 /                            ! SD(1:NSD)
501                              ! NRD
-5000.0  5000.0 /                ! RD(1:NRD)
501                              ! NR
-10.0  10.0 /                    ! R(1:NR ) (km)
'C '                             ! Run type: 'Ray/Coh/Inc/Sem'
0                                ! NBEAMS
-180 180 /                       ! ALPHA1,2 (degrees)
0.0  10001.0  10.0               ! STEP (m), ZBOX (m), RBOX (km)
```

Directional sources are commonly used in underwater acoustics. Often, the directional patterns are generated by adjusting the phase and amplitude of a discrete set of omni-directional projectors. One may model such sources using BELLHOP or any of the other models in the Acoustics Toolbox, by simply running the model for each source position and then summing up the resulting acoustic fields using that same phase and amplitude weighting. This is often the most faithful approach; however, it has the disadvantage of

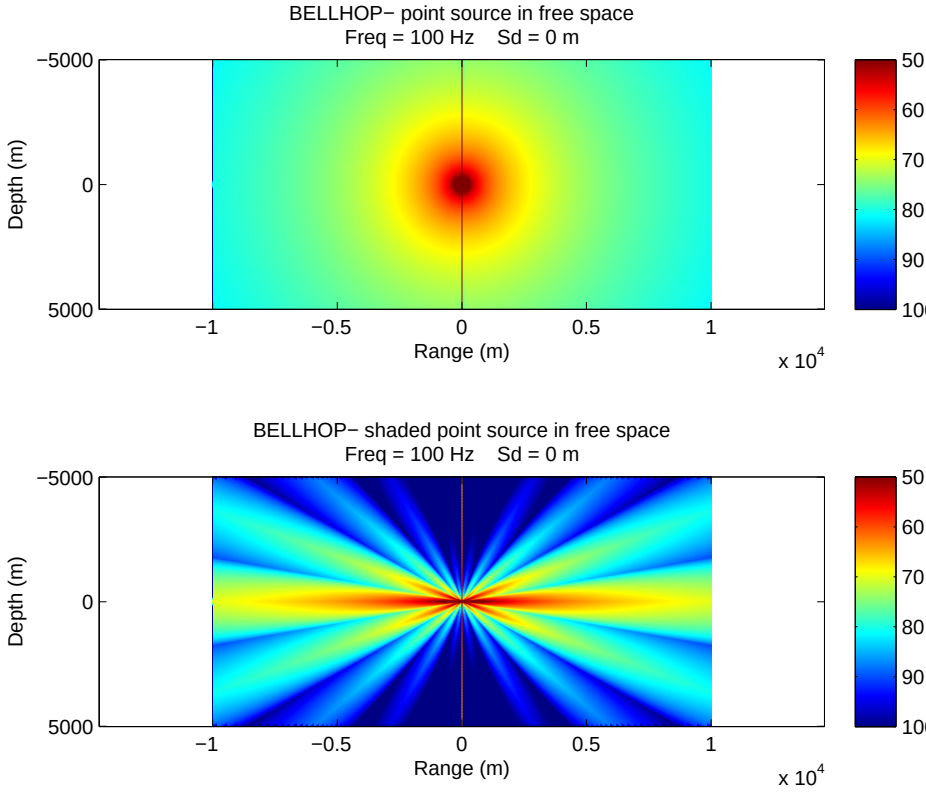


Figure 7: Upper panel: transmission loss for a point source in free space. Lower panel: same as upper panel but with a source beam pattern applied.

requiring field calculations for multiple sources, as well as the post-processing to combine the fields.

An alternative is to use the option of providing a source beam pattern to BELLHOP . To do this we simply provide an additional file with the '.sbp' extension, which contains angle-level pairs defining the beam pattern. The first line specifies the number of such pairs that are being provided. Ideally, the code should be able to figure this out itself; however, requiring the user to provide it at the beginning facilitates the dynamic memory allocation. On the following lines, the angles are given in degrees and the levels are specified in dB.

shaded.sbp

```
37
-180  10
-170 -10
-160   0
-150 -20
-140 -10
-130 -30
-120 -20
-110 -40
-100 -30
-90  -50
-80  -30
-70  -40
-60  -20
-50  -30
-40  -10
-30  -20
-20    0
-10  -10
  0   10
 10  -10
 20    0
 30  -20
 40  -10
 50  -30
 60  -20
 70  -40
 80  -30
 90  -50
100  -30
110  -40
120  -20
130  -30
140  -10
150  -20
160    0
```

```
170  -10
180   10
```

To instruct BELLHOP to read such a source beam pattern file, we simply set the third letter of the RunType to '*' as shown below. The roots of the filenames (in this case 'shaded') for the environmental and source beam pattern should match. The resulting TL plot is shown in the lower panel of Fig. (7).

shaded.env

```
'Shaded point source in free space'
100.0         ! Frequency (Hz)
1
'CAF'
0.0 1500.0 0.0 1.0 /
1500 0.0 5000.0
-5000.0 1500.0 0.0 1.0 /
 5000.0 /
'A', 0.0
/
1                               ! NSD
0.0 /                           ! SD(1:NSD)
501                             ! NRD
-5000.0  5000.0 /               ! RD(1:NRD)
501                             ! NR
-10.0  10.0 /                   ! R(1:NR ) (km)
'C *'                           ! Run type: 'Ray/Coh/Inc/Sem'
361                             ! NBEAMS
-180 180 /                      ! ALPHA1,2 (degrees)
0.0  5001.0  10.0               ! STEP (m), ZBOX (m), RBOX (km)
```

6 Range-dependent Boundaries

6.1 Piecewise-Linear Boundaries: Dickins seamount

As a first example, we consider the Dickins seamount case. The bottom bathymetry is idealized as being flat at 3000 m in depth, apart from where the seamount rises to a depth 500 m at a range of 20 km. The seamount is modeled as a triangle with a base of 20 km. The bathymetry is defined in a separate file with a '.bty' extension as shown below. The first line specifies 'L' for a piecewise linear fit or 'C' for a curvilinear fit. Since we want a triangular seamount, we select the piecewise linear fit. Line 2 is the number of bathymetry points, and the lines following that are the range-depth pairs defining the bathymetry. Following the usual BELLHOP convention the ranges are in kilometers and the depths in meters. The bathymetry must be defined starting from zero range (or a negative 'range' if we are interested in the backscattered field) to the maximum range of interest for the field calculation.

DickinsB.bty

```
'L'
5
  0 3000
 10 3000
 20  500
 30 3000
100 3000
```

The environmental file for this case is shown below. On line 4 we see the top option 'CVW'. The 'C' indicates that we wish to interpolate the sound speed, c, in a piecewise linear fashion. The 'V' is the standard vacuum option for the ocean surface. The 'W' indicates the attenuation will be specified in dB/wavelength, since those were the units used in providing the environmental description.

The water depth for the environmental file is specified as 3000 m. For such cases with a range-dependent bathymetry, this depth is not directly used. However, it is part of the definition of the ocean sound speed profile and therefore it must be deep enough to cover the deepest depth in the range-dependent bathymetry file.

On line 36 we see the bottom option is set to 'A*'. The 'A' indicates an acousto-elastic halfspace for the bottom, which is the typical choice. The '*' is introduced here as the flag to tell BELLHOP to read the supplemental bathymetry file 'Dickins.bty' for the range-dependent bathymetry.

DickinsB.env

```
'Dickins seamount'      ! TITLE
230.0                   ! FREQ (Hz)
1                       ! NMEDIA
'CVW'                   ! SSPOPT (Analytic or C-linear interpolation)
525 0.0 3000.0          ! DEPTH of bottom (m)
   0 1476.7 /
   5 1476.7 /
  10 1476.7 /
  15 1476.7 /
  20 1476.7 /
  25 1476.7 /
  30 1476.7 /
  35 1476.7 /
  38 1476.7 /
  50 1472.6 /
  70 1468.8 /
 100 1467.2 /
 140 1471.6 /
 160 1473.6 /
 170 1473.6 /
 200 1472.7 /
 215 1472.2 /
 250 1471.6 /
 300 1471.6 /
 370 1472.0 /
 450 1472.7 /
 500 1473.1 /
 700 1474.9 /
 900 1477.0 /
1000 1478.1 /
1250 1480.7 /
1500 1483.8 /
2000 1490.5 /
2500 1498.3 /
3000 1506.5 /
'A*' 0.0
3000.0  1550.0    0.0  1.5 0.5 /
1                               ! NSD
18.0 /                          ! SD(1:NSD) (m)
201                             ! NRD
    0.0 3000.0 /                ! RD(1:NRD) (m)
1001                            ! NR
  0.0  100.0 /                  ! R(1:NR) (km)
'CB'                            ! 'R/C/I/S'
0                               ! NBEAMS
-89.0 89.0 /                    ! ALPHA1,2 (degrees)
0.0  3100.0  101.0              ! STEP (m), ZBOX (m), RBOX (km)
```

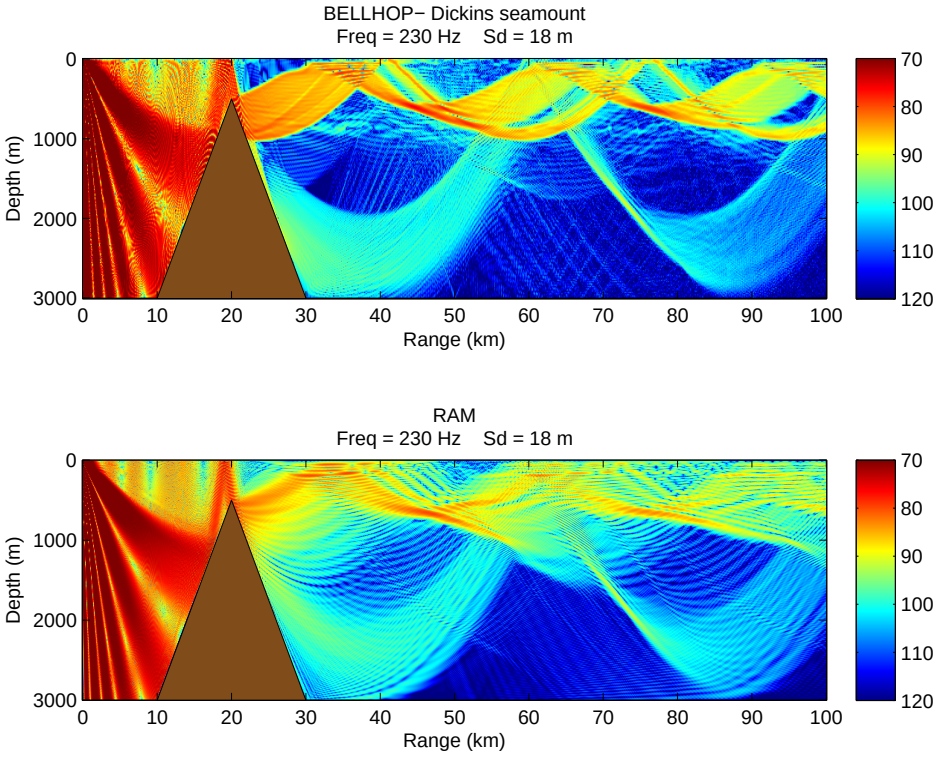


Figure 8: Transmission loss for the Dickins seamount using a) BELLHOP b) RAM (parabolic equation model).

The resulting TL plot is shown in Fig. (8) with the upper panel being the BELLHOP result and the lower panel being the RAM PE result, which we consider the reference solution. Note the bathymetry has also been plotted using the Matlab command:

plotbty 'Dickins'

The agreement is satisfactory; however, the artificially sharp point of the seamount causes a lot of diffracted energy. This can be further improved by inserting additional bathymetry points to round the discontinuity.

6.2 Plotting a single beam

BELLHOP computes the field by summing up the contributions of a fan of beams. It is sometimes useful to be able to plot an individual beam in the sum. We invoke this option by setting the fifth letter in the top option to 'I' as shown below (line 4). Then we must also add a second integer after NBeams (see line 45) to indicate the specific beam we are interested in displaying. In this case we have selected beam 12 out of a fan of 21 beams. We have specifically selected a small number of beams, so that each beam would be wider and show up better on the plot. The resulting TL plot is shown in Fig. (9).

DickinsB_oneBeam.env

```
1  'Dickins seamount'      ! TITLE
2  230.0                   ! FREQ (Hz)
3  1                       ! NMEDIA
4  'CVW  I'                ! SSPOPT (Analytic or C-linear interpolation)
5  525 0.0 3000.0          ! DEPTH of bottom (m)
6     0 1476.7 /
7     5 1476.7 /
8    10 1476.7 /
9    15 1476.7 /
10   20 1476.7 /
11   25 1476.7 /
12   30 1476.7 /
13   35 1476.7 /
14   38 1476.7 /
15   50 1472.6 /
16   70 1468.8 /
17  100 1467.2 /
18  140 1471.6 /
19  160 1473.6 /
20  170 1473.6 /
21  200 1472.7 /
22  215 1472.2 /
23  250 1471.6 /
24  300 1471.6 /
25  370 1472.0 /
26  450 1472.7 /
27  500 1473.1 /
28  700 1474.9 /
29  900 1477.0 /
30 1000 1478.1 /
31 1250 1480.7 /
32 1500 1483.8 /
33 2000 1490.5 /
34 2500 1498.3 /
35 3000 1506.5 /
```

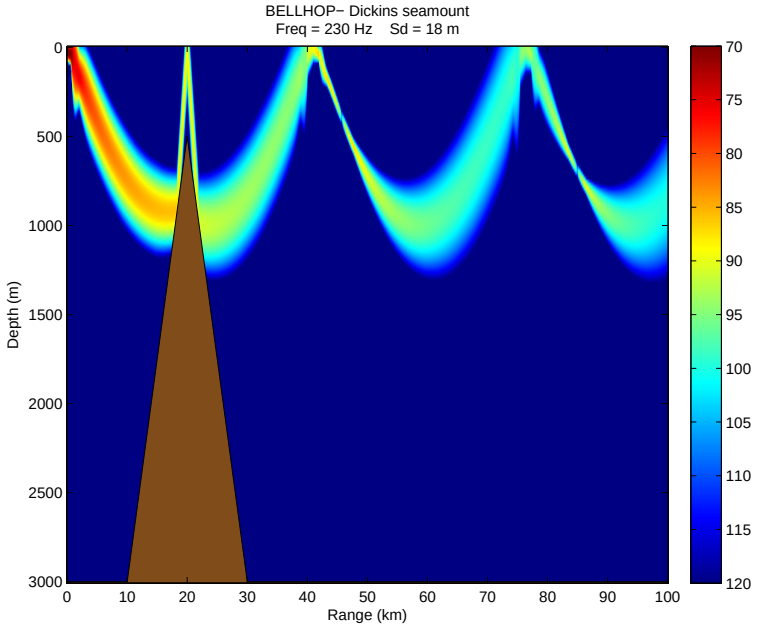


Figure 9: Transmission loss with the option selected to plot a single beam.

```
'A*' 0.0
3000.0  1550.0    0.0  1.5 0.5 /
1                                ! NSD
18.0 /                           ! SD(1:NSD) (m)
201                              ! NRD
    0.0 3000.0 /                 ! RD(1:NRD) (m)
1001                             ! NR
  0.0  100.0 /                   ! R(1:NR) (km)
'CB'                             ! 'R/C/I/S'
21 12                            ! NBeams, IBeam
-15.0 15.0 /                     ! ALPHA1,2 (degrees)
0.0  3100.0  101.0,              ! STEP (m), ZBOX (m), RBOX (km)
'MS' 1.0 100.0                   ! BeamType,  epmult, Rloop (km)
1 5 'P'                          ! NImage IBWin
```

6.3 Curvilinear Boundaries: Parabolic Bottom

In the previous example, the bathymetry was defined in a piece-wise linear fashion. However, in some cases a smoother shape is desired. As an example, we consider the parabolic bottom profile described by McGirr *et al.* [**?**]. The

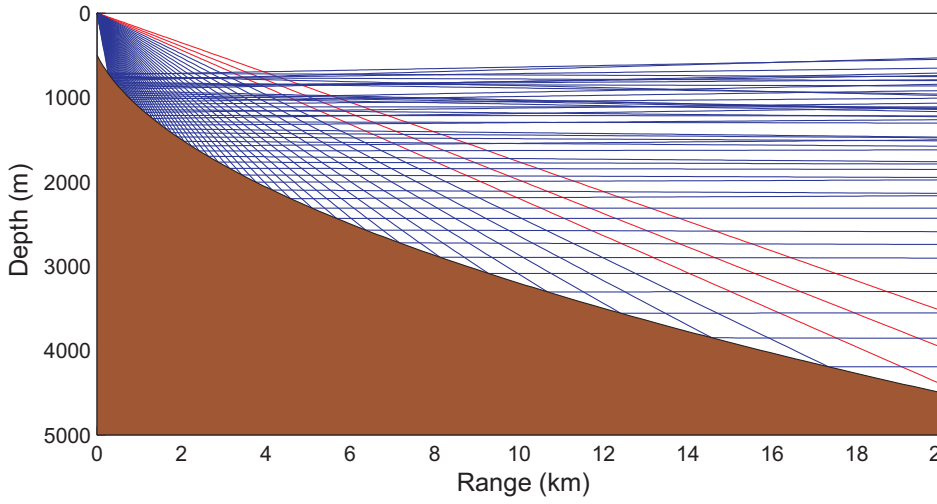


Figure 10: Ray reflecting from a piecewise linear boundary.

depth is given by $D(r) = 500\sqrt{1+4r}$ where ranges r are given in kilometers and depths D are given in meters. A source at the origin is at the focal point, so the bottom-reflected rays should, ideally, emerge parallel to the surface, much as the reflector in a flashlight produces a uniform beam. As we go out to 20 km in range, the position of the rays is extremely sensitive to the tilt of the bottom facets, and therefore an interesting metric on how well the bathymetry interpolation works. With the bathymetry sampled every 25 m in depth, and using the piecewise-linear option, we derive the ray trace shown in Fig. 10. The flaws are obviously revealed in the irregular ray structure.

To get a smoother ray trace, we invoke the curvilinear option by simply change the first letter in the bathymetry file to 'C'. As shown in the upper panel of Fig. 11, this improved boundary interpolation provides a set of perfectly (as far as the eye can see) parallel rays, as expected. As a point of interest for the reader, we also present the corresponding TL for a 10 Hz source frequency.

Given the improved results, one may wonder why we do not simply standardize on the curvilinear option. An answer is obvious from the previous Dickins seamount case. There the bottom was specified specifically to be piecewise linear, so a curvilinear interpolation would yield erroneous results. For bathymetry profiles provided by standard databases we do not typically see a great sensitivity to the type of interpolation used. However, passing to rough ocean surfaces generated by standard models or measurements of

the surface wave spectra we have generally found more satisfactory results using the curvilinear option.

Rough ocean surfaces may be incorporated in exactly the same way. BELLHOP looks for a *.ati file containing the range-depth pairs describing the surface. Note that depths in the altimetry file are positive-downward, following the standard convention.

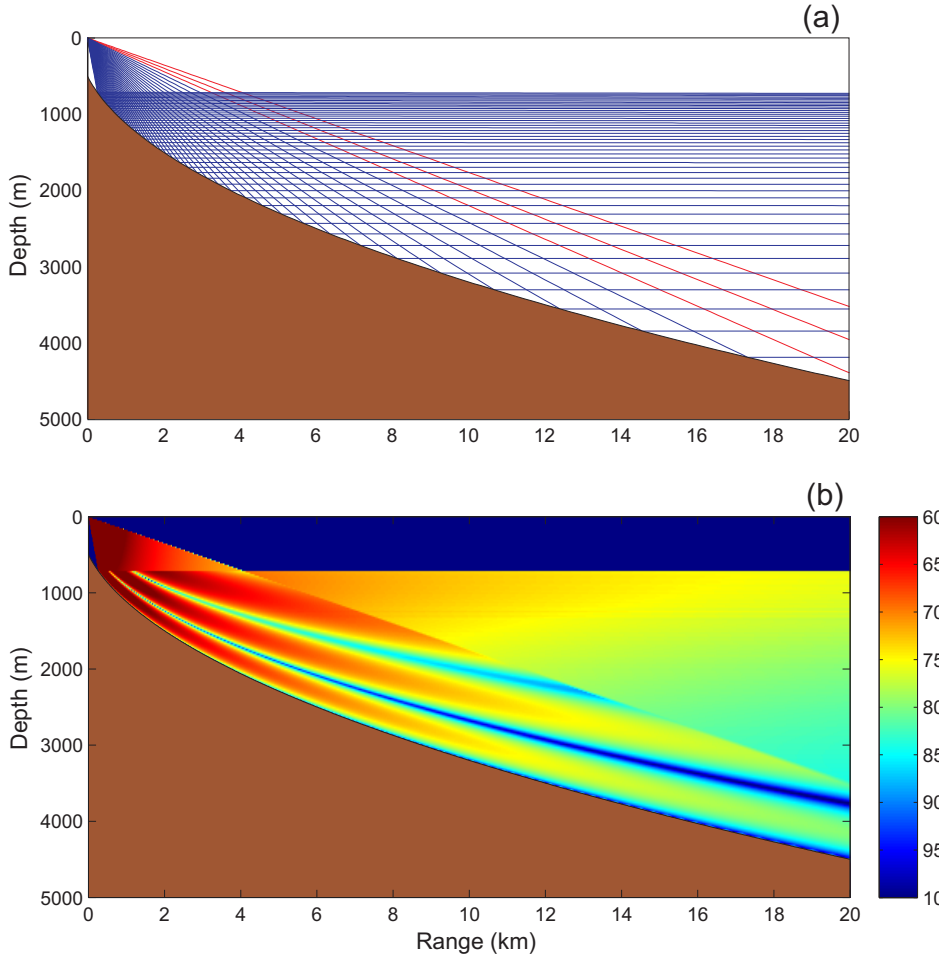


Figure 11: Curvilinear boundary interpolation for the parabolic bathymetry profile. **(a)** Ray trace and **(b)** transmission loss.

7 Tabulated Reflection Coefficients

Surface and bottom reflection coefficients are often provided in a tabular form so BELLHOP provides an option to read a file containing this data. The extensions for these files are '.trc' and '.brc' for a top and bottom reflection coefficient respectively. It is important to note that this also allows BELLHOP to incorporate a more complicated bottom model, since an arbitrary stratified sub-bottom is perfectly represented by an angle-dependent reflection coefficient. In fact, mixtures of elastic and acoustic layers with gradients, attenuation, and density are all exactly represented by a complex reflection coefficient depending on the angle of incidence. Thus, in a full-wave model such as KRAKEN or SCOOTER one may calculate an equivalent reflection coefficient for the sub-bottom and read that into the model in tabular form with essentially perfect accuracy. The only degradation is due to the sampling of the reflection coefficient in angle.

Unfortunately, BELLHOP is not able to exploit such a file with the same level of fidelity, partly due to its ray-theoretic underpinnings. This is too complicated to get into here, but BELLHOP does not implement ray shifts and ray splittings that would be necessary to provide a better treatment of the reflection coefficient. Nevertheless, one can still do a reasonable job of treating complicated layered bottoms through the tabulated reflection coefficient.

An example of the use of such a coefficient is given in the test case tests/TabRefCoef. We consider a sub-bottom consisting of a homogeneous layer of material, overlying ...

8 Range-dependent Sound Speed Profiles

Range-dependent sound speed profiles are easy to treat in BELLHOP . As an example we consider the test case from at/tests/Gulf, which is based on a Gulf Stream scenario with a mesoscale eddy. The basic input file is identical in form to one for a range-independent SSP, except that we use option letter 'Q' for the first letter. ('Q' is for 'quadrilateral' SSP interpolation.) The input file for this case is shown below.

Gulf_ray_rd.env

```
'Topgulf'
50.0
1
'QVW'
0 0.0 5000.0
    0.0  1536.00 /
  200.0  1506.00 /
  700.0  1503.00 /
  800.0  1508.00 /
 1200.0  1508.00 /
 1500.0  1497.00 /
 2000.0  1500.00 /
 3000.0  1512.00 /
 4000.0  1528.00 /
 5000.0  1545.00 /
'A*' 0.0
 5000.00   1800.0  0.0  2.0  0.1 0.0
1                       ! NSD
300.0 /                 ! SD(1:NSD) (m)
101                     ! NRD
0.0 5000.0 /            ! RD(1:NRD) (m)
1001                    ! NR
0.0  125.0 /            ! R(1:NR ) (km)
'R'                     ! 'R/C/I/S'
29                      ! NBeams
-14.0 14.0 /            ! ALPHA1,2 (degrees)
1000.0  5500.0 126.0    ! STEP (m), ZBOX (m), RBOX (km)
```

The required SSP must be provided on a rectangular grid in a file with the extension '.ssp':

Gulf_ray_rd.ssp

```
8
0.0     12.5    25.0    37.5    50.0    75.0    100.0   125.0
1536    1536    1536    1536    1536    1536    1536    1536
1506    1508.75 1511.5  1514.25 1517    1520    1524    1528
1503    1503    1503    1502.75 1502.5  1502    1502    1502
1508    1507    1506    1505    1504    1503    1501.5  1500
```

```
1508    1506.6  1505    1503.75 1502.5  1500.5  1499    1497
1497    1497    1497    1497    1497    1497    1497    1497
1500    1500    1500    1500    1500    1500    1500    1500
1512    1512    1512    1512    1512    1512    1512    1512
1528    1528    1528    1528    1528    1528    1528    1528
1545    1545    1545    1545    1545    1545    1545    1545
```

The first line indicates the number of profiles, i.e. the number of columns in the SSP matrix. The second line gives the actual ranges in km of those profiles. Subsequent lines provide the actual sound speed values in m/s, so that each row corresponds to a fixed depth. The actual depths used are taken out of the original environmental file, in this case, lines 6 – 15 of the envfil.

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. In this case, we have included the following bathymetry file:

Gulf_ray_rd.bty

```
'L'
8
  0.0 5000
 12.5 5000
 25.0 5000
 37.5 5000
 50.0 5000
 75.0 5000
100.0 5000
125.0 5000
```

Note that we have also used the '*' option in line 16 of the envfil to tell BELLHOP that there is bathymetry file to be read.

Now that the input file has been created, we can start by plotting the soundspeed profile, using the Matlab routine plotssp2d.m. The syntax of the Matlab command to run this is:

```
plotssp2d 'Gulf_ray_rd'
```

where 'MunkB_ray_rd.env' is the name of the BELLHOP input file. This produces the plots in Fig. (12) and Fig. (13) rendering the SSP both in slices at fixed ranges and as a color surface. The effect of the range dependence may be seen by comparing Figs. (14) and (15). The former shows the ray trace with a range-independent SSP, using the profile at zero range. In

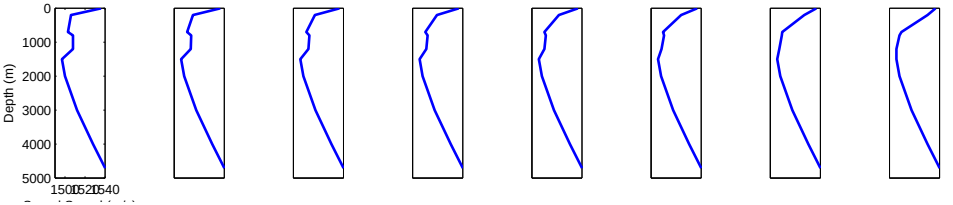


Figure 12: Range-dependent sound speed profile for the Gulf Stream scenario.

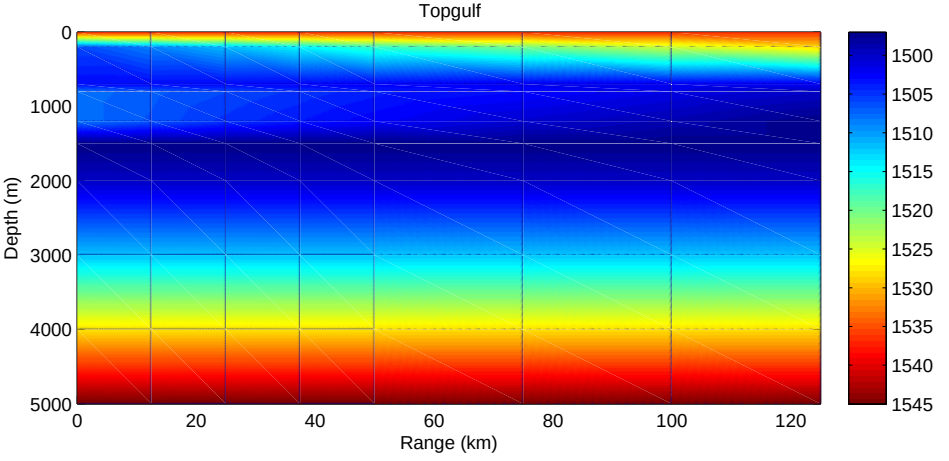


Figure 13: Range-dependent sound speed profile for the Gulf Stream scenario.

the second, the range-dependence has been included, causing rays trapped in the upper duct to transition into the main sofar duct.

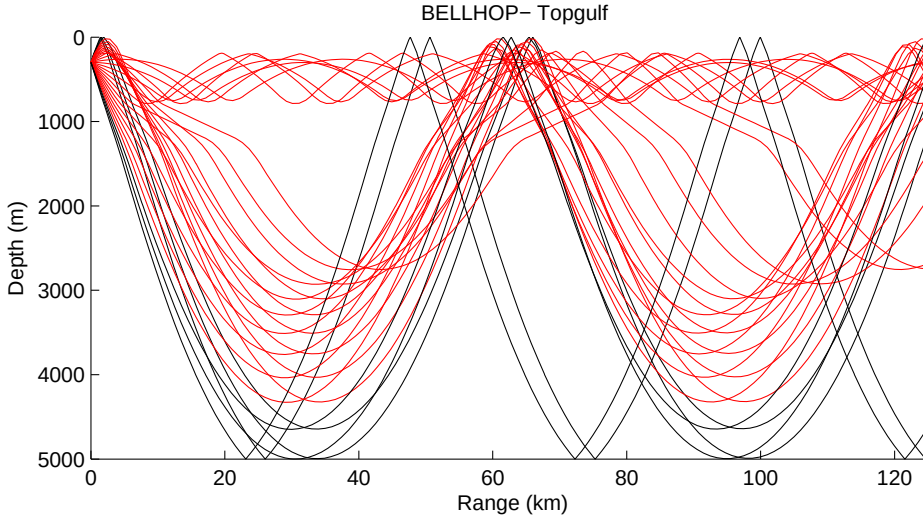


Figure 14: Range-dependent sound speed profile for the Gulf Stream scenario.

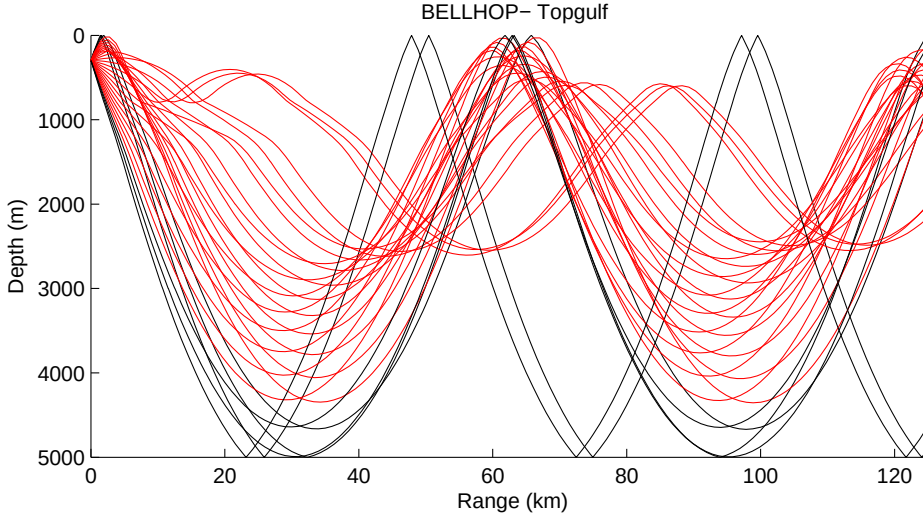


Figure 15: Range-dependent sound speed profile for the Gulf Stream scenario.

9 Arrivals calculations and broadband results

One of the strengths of ray/beam tracing models is the possibility of producing a full timeseries for a broadband source with essentially no additional effort relative to a tonal calculation. In contrast, most of the alternatives handle a broadband source by representing it as a sum of tonals, i.e. by working with the Fourier transform of the source. Then each tonal is modeled separately, and the final received wave form is assembled by summing up the received field on a tone-by-tone base (the inverse Fourier transform).

The ray/beam process calculates the amplitudes and travel-times of all the echoes and can therefore calculate the received timeseries by simply summing up the echoes. That is the high-level picture. A more detailed explanation would address the fact that phase changes due to caustics and boundary interactions require that we consider perfect echoes, as well as Hilbert transforms of the source waveform. Further, the ray/beam process uses attenuation values at the center frequency of the calculation. For sources with a great deal of bandwidth on should divide the frequency band into sub-bands so that the appropriate attenuation is used in each sub-band. Finally, certain BELLHOP parameters are automatically selected by the code based on the center frequency. Most of the time the user will not need to be concerned with these subtleties.

The arrivals option in BELLHOP is designed to produce an output file with the critical information for broadband calculations. As mentioned above we need only the amplitudes and travel times associated with each echo. This defines the impulse response of the channel, which is then convolved with the source waveform to produce the received timeseries.

As an example, we consider an acoustic modem simulation for a site off Kauai, Hawaii. The sound speed profile was measured by an XBT or a CTD and is plotted in Fig. (16). We suspect the bottom in the area is fairly reflective and will therefore choose a ray fan that includes bottom-reflected rays. The ray trace for a source at 50 m in depth is shown in Fig. (17).

9.1 Coherent and Incoherent TL

Acoustic modems may be designed for different frequencies depending on how one wishes to trade-off the maximum useable range against the bandwidth or data rate. We consider here a 10 kHz central frequency. The resulting coherent transmission loss is shown in Fig. (18). At 10 kHz, the wavelength is about 15 cm, so it is not surprising that the interference pattern shows variation on about that scale. (The interference pattern actually

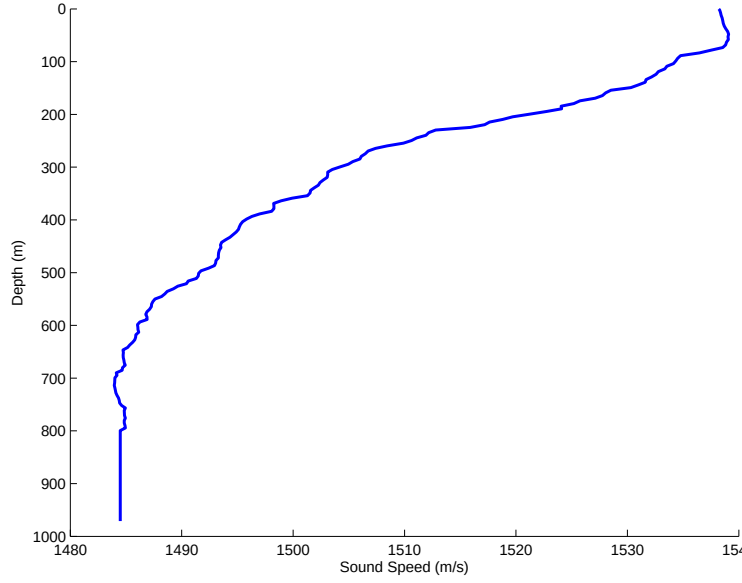


Figure 16: Sound speed profile for the Kauai environment.

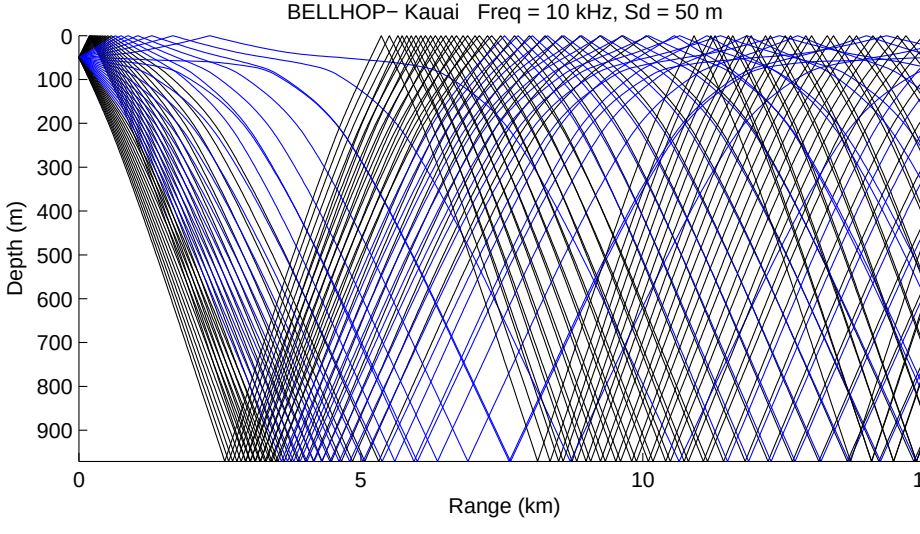


Figure 17: Ray trace for the Kauai environment.

varies on a somewhat longer scale because of the narrow fan of rays involved.)

The input file used to produce this run is shown below.

Kauai.env

```
'Kauai   Freq = 10 kHz, Sd = 50 m'      ! TITLE
10000.000                               ! FREQ (Hz) [acomms: 3.5 kHz, 10 kHz]
1                                       ! NMEDIA
'CVWT'                                  ! SSPOPT
0 0.0 971               ! NMESH SIGMA DEPTH_of_BOTTOM (m)
   0.0 1538.20 /  ! added point
  1.90 1538.28 /  ! xbt
  7.00 1538.36 /
 12.10 1538.41 /
 17.20 1538.51 /
 22.30 1538.56 /
 27.40 1538.61 /
 32.50 1538.70 /
 37.60 1538.83 /
 42.70 1539.00 /
 47.80 1539.09 /
 52.90 1539.03 /
 58.00 1539.06 /
 63.10 1538.90 /
 68.20 1538.82 /
 73.20 1538.56 /
 78.30 1537.52 /
 83.40 1536.47 /
 88.50 1534.77 /
 93.50 1534.52 /
 98.60 1534.35 /
103.70 1534.14 /
108.70 1533.54 /
113.80 1533.36 /
118.90 1532.78 /
123.90 1532.58 /
129.00 1532.16 /
134.00 1531.67 /
139.10 1531.60 /
144.10 1531.00 /
149.20 1530.31 /
154.20 1528.52 /
159.20 1528.05 /
164.30 1527.76 /
169.30 1527.11 /
174.30 1525.74 /
179.40 1525.20 /
184.40 1524.05 /
189.40 1524.10 /
194.50 1522.77 /
199.50 1521.22 /
```

```
204.50 1519.71 /
209.50 1518.81 /
214.50 1517.63 /
219.50 1517.20 /
224.60 1515.90 /
229.60 1512.79 /
234.60 1512.17 /
239.60 1511.92 /
244.60 1511.09 /
249.60 1510.63 /
254.60 1509.91 /
259.60 1508.44 /
264.50 1507.40 /
269.50 1506.74 /
274.50 1506.48 /
279.50 1506.12 /
284.50 1506.00 /
289.50 1505.36 /
294.40 1504.95 /
299.40 1504.23 /
304.40 1503.52 /
309.40 1503.11 /
314.30 1503.11 /
319.30 1503.06 /
324.30 1502.71 /
329.20 1502.41 /
334.20 1502.27 /
339.10 1501.89 /
344.10 1501.56 /
349.00 1501.52 /
354.00 1501.29 /
358.90 1499.91 /
363.90 1498.93 /
368.80 1498.25 /
373.80 1498.28 /
378.70 1498.28 /
383.60 1498.07 /
388.60 1497.00 /
393.50 1496.30 /
398.40 1495.84 /
403.40 1495.48 /
408.30 1495.29 /
413.20 1495.17 /
418.10 1495.10 /
423.00 1494.89 /
427.90 1494.61 /
432.90 1494.33 /
437.80 1493.93 /
442.70 1493.58 /
```

```
447.60 1493.48 /
452.50 1493.53 /
457.40 1493.38 /
462.30 1493.31 /
467.20 1493.29 /
472.10 1493.29 /
477.00 1493.08 /
481.90 1493.04 /
486.70 1492.94 /
491.60 1492.41 /
496.50 1491.74 /
501.40 1491.52 /
506.30 1491.49 /
511.10 1491.32 /
516.00 1490.58 /
520.90 1490.44 /
525.70 1489.65 /
530.60 1489.26 /
535.50 1488.70 /
540.30 1488.49 /
545.20 1488.20 /
550.00 1487.60 /
554.90 1487.41 /
559.80 1487.30 /
564.60 1487.28 /
569.40 1487.12 /
574.30 1486.88 /
579.10 1486.77 /
584.00 1486.88 /
588.80 1486.90 /
593.60 1486.25 /
598.50 1486.03 /
603.30 1486.07 /
608.10 1486.08 /
613.00 1486.15 /
617.80 1485.89 /
622.60 1485.86 /
627.40 1485.79 /
632.20 1485.59 /
637.10 1485.34 /
641.90 1485.15 /
646.70 1484.73 /
651.50 1484.77 /
656.30 1484.74 /
661.10 1484.76 /
665.90 1484.81 /
670.70 1484.84 /
675.50 1484.94 /
680.30 1484.69 /
```

```
685.10 1484.65 /
689.90 1484.13 /
694.60 1484.19 /
699.40 1484.01 /
704.20 1483.99 /
709.00 1483.98 /
713.80 1483.95 /
718.50 1484.00 /
723.30 1484.05 /
728.10 1484.09 /
732.80 1484.20 /
737.60 1484.33 /
742.40 1484.40 /
747.10 1484.43 /
751.90 1484.61 /
756.70 1484.95 /
761.40 1484.84 /
766.20 1484.86 /
770.90 1484.87 /
775.70 1484.95 /
780.40 1484.84 /
785.10 1484.84 /
789.90 1484.91 /
794.60 1484.95 /
799.40 1484.49 /
971.00 1484.49 /  ! added point, depth from Google Earth
'A' 0.0
971  1600.000  0.000  1.8  0.5  0.000 /  ! fit to data from KauaiEx
1                                ! NSD
50                               ! SD(1:NSD) (m)
300                              ! NRD  [1 m increments]
1    300 /              ! RD(1:NRD) (m)
1000                             ! NRR  [5 m increments]
0.015       15.0 /               ! RR(1:NR ) (km)
'CB'                             ! Run-type: 'R/C/I/S'
0                                ! NBEAMS
-15  15 /                       ! ALPHA(1:NBEAMS) (degrees)
0  1400.0   15.1                 ! STEP (m)  ZBOX (m)  RBOX (km)
```

We comment here on the sampling of the sound speed profile. As usual, the XBT or CTD provided a sampling with a fine resolution, roughly every meter in depth. We find, by comparison to full-wave models such as SCOOTER and KRAKEN that more accurate BELLHOP results are obtained by sub-sampling these profiles. For this particular frequency, a sound speed sampled every 5-10 m seemed to work best. A coarser spacing would have caused a poor resolution of the important surface duct feature. With a much finer sampling we find that the rays respond to small features, including tiny

ducts, creating artifacts in the TL field. We could probably obtain still better results by using a variable spacing with much coarser sampling below the surface duct.

Further, a little smoothing of the SSP also helps to eliminate these small features, which in reality are not adequately represented in a 1-D slide of a 4-D oceanographic field. There has been some research on how to optimally smooth such profiles for use in ray/beam codes, but there is no simple answer. Therefore, in working with large numbers of XBTs or CTDs we suggest one simply sub-sample.

Another consideration in the sampling is the run-time. BELLHOP adjusts the ray step to land on every interface in the sound speed profile, that is, on every depth where the SSP is sampled. This provides both a more accurate ray trace and TL calculation, since beam curvature corrections are required at first order interfaces. Thus, the SSP normally controls the stepsize used in integrating the ray equations, and a fine sampling can drive up the run time significantly.

Given the variation of the TL field on a short scale, these color plots are really not well enough sampled to capture the variation. In addition, the sound speed profile was measured at one time and one place with an instrument with limited accuracy. Considering that, we cannot take the fine details as serious, deterministic simulations. However, the TL plot provides a useful reminder of the fading over range that the modem sees (hears) for a particular tonal.

Nevertheless, the communications engineer is likely more interested in the overall energy, integrated across the bandwidth of the modem, that might be received in this scenario. This is a good place to consider the incoherent TL option in BELLHOP resulting in the plot in Fig.'19. (The term 'noncoherent' is used in the communications world ...)

9.2 Plotting the impulse response

To calculate the channel impulse response, we select the 'A' option for the RunType. ('A' for Arrivals.) When BELLHOP completes, it produces a file 'Kauai.arr' containing a table of the arrivals information (the impulse response). For each source depth, receiver depth, and receiver range it contains the number of echoes or arrivals. Then for each arrival the amplitude, phase, and travel time of the arrival is provided. Supplemental information is also provided on the ray take-off angle at the source and at the receiver, as well as the number of top and bottom bounces. The data in this arrivals file may be plotted using the command

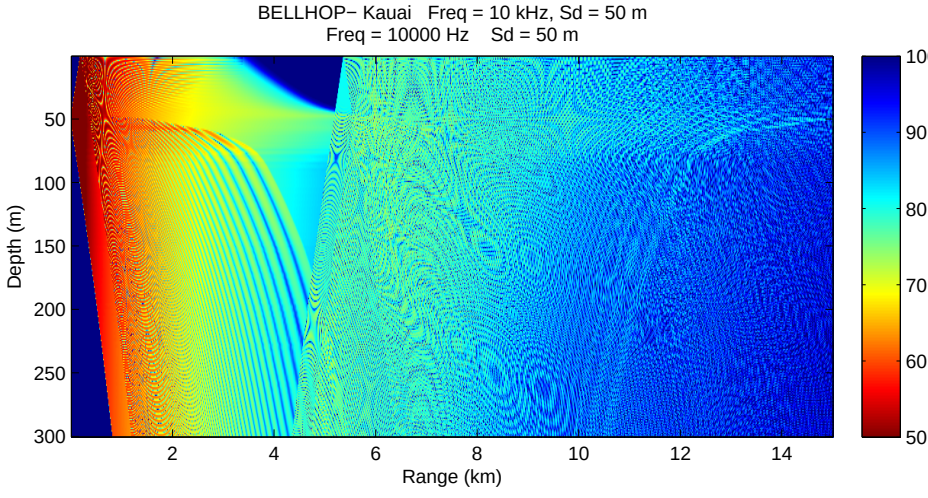


Figure 18: Coherent transmission loss for the Kauai environment.

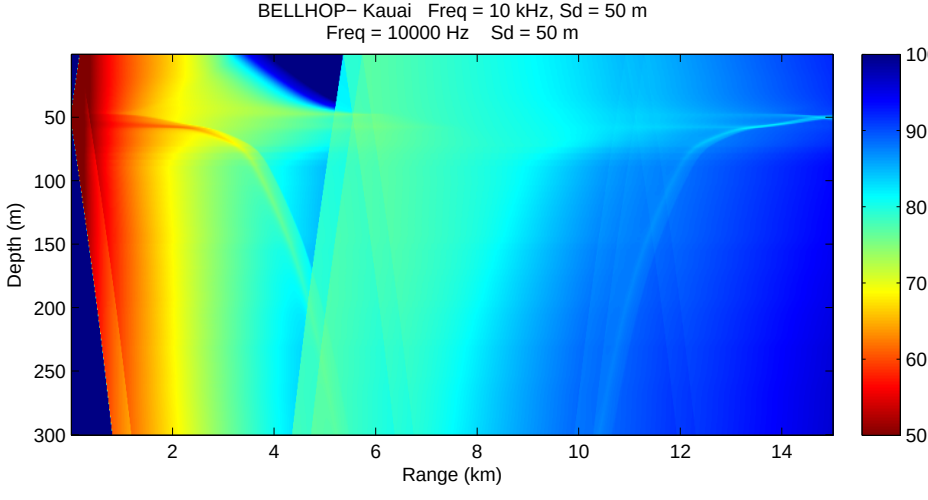


Figure 19: Inoherent transmission loss for the Kauai environment.

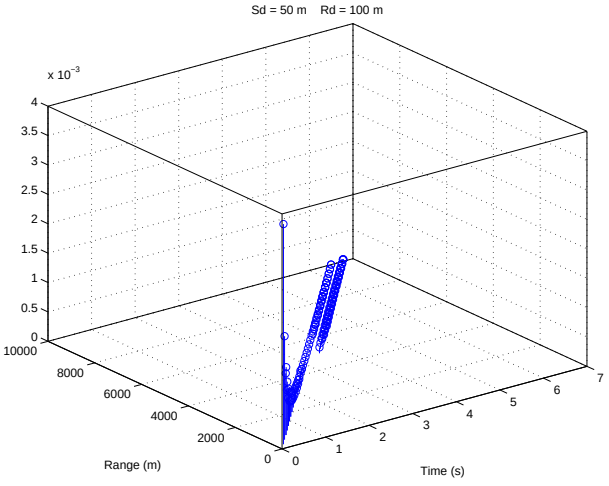


Figure 20: Impulse response as a function of range.

plotarr('Kauai', 20, 5, 1)

where the first parameter is the root of the filename, and the three integers specify the indices of the receiver range, receiver depth, and source depth. This produces three plots as shown in Figs. **??**) showing the impulse response as stem plots and in slices over fixed receiver depth, or fixed range.

The actual levels in this sort of impulse response plot are not directly useful. The spikes show discrete arrivals; however, in practice the true response is a convolution of these impulses with the source waveform.

9.3 Generating a receiver timeseries

Once we have the impulse response function, as represented by the arrivals file, we can produce a receiver timeseries by convolving the source timeseries with that impulse response function. The Matlab script that does this is called 'delayandsum'. In practice, we have found that there are usually various customizations that we want to make, so the user will generally need to make small changes to this script for their particular application.

As an example of its use, we consider a common scenario the channel response is probed with an LFM chirp.

In the real-world one receives the chirp in the waveguide and then correlates it with a replica of the source waveform. The autocorrelation of the

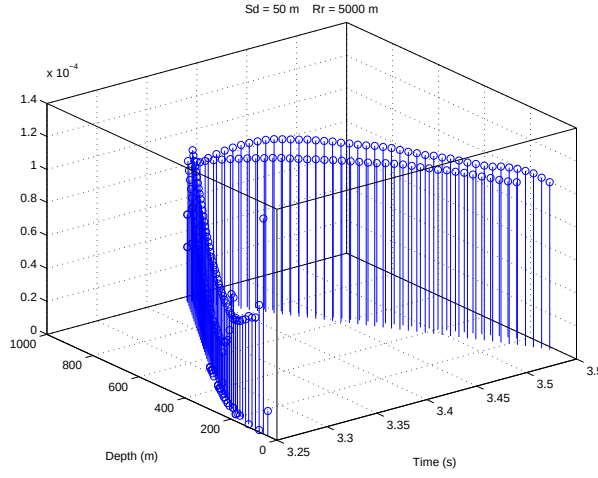


Figure 21: Impulse response as a function of depth.

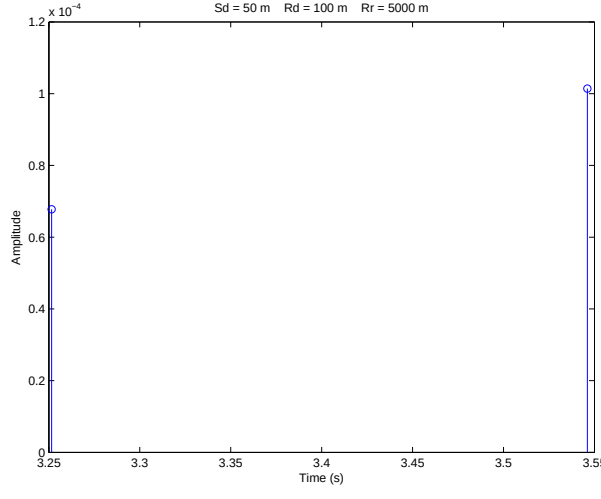


Figure 22: Impulse response at range of 5000 m and a depth of 100 m.

chirp with itself produces a waveform that is impulsive, with the length of the impulse approximately equal to the reciprocal of the bandwidth used in the chirp. (We remind that the autocorrelation of a chirp is a sinc-pulse.)

The above process may be emulated by convolving the impulse response with the chirp waveform, and then applying the correlation process to the output. However, the convolution and correlation processes commute, so one may equivalently propagate the autocorrelation of the chirp through the waveguide.

Now we consider the effect of two adjacent spikes in the impulse response function (the arrivals in the file). If these spikes are separated in time by a distance greater than the duration of the source waveform, then we get two distinct copies of the source waveform in the received timeseries. Conversely, if the impulses are close together, then the individual pulses in the source waveform are not resolved. The key point of this discussion is that we must interpret the impulse response function in sort of a clustered fashion, collecting the energy of groups of arrivals that are not resolved because of the duration of the channel probe. A way to do this is simply to convolve the impulse response with a pulse of appropriate duration.

An additional consideration is the role of the beam type. As discussed in the section on eigenrays, 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. If you use the geometric beam option, with hat-shaped beams (the default), then 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.

On the other hand, if you invoke geometric beams, with the Gaussian beam option, then 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.

One may wonder then about 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.

how to read the file

10 Acknowledgments

The BELLHOP model was originally developed at the Naval Ocean Systems Center (now SPAWAR), but continued to develop at the Naval Research Laboratory, the SACLANT Undersea Research Centre (now the NATO Undersea Research Centre), the New Jersey Institute of Technology, Science Applications International Corp., and Heat, Light, and Sound Research. It has been supported by various fundamental research programs, especially the Office of Naval Research (ONR), Ocean Acoustics program. Their support is gratefully acknowledged. This manual has been prepared under the support of the ONR Marine Mammals and Biological Oceanography Program as part of the Effects of Sound on the Marine Environment (ESME) project.