Copy No. _____

Defence Research and Development Canada | Recherche et développement pour la défense Canada

DEFENCE | DÉFENSE

# Users Guide to BellhopDRDC_V4

## *Active and Passive versions*


*Diana McCammon*
*McCammon Acoustical Consulting*

*Prepared by:*
*McCammon Acoustical Consulting*
*475 Baseline Road*
*Waterville, NS  B0P 1V0*



Contract Project Manager:  Diana McCammon, 902-538-3003
Contract number:  W7707-098272
Scientific Authority:  Dr. WA Roger, 902-426-3100 ext. 292





**Defence R&D Canada – Atlantic**

Contract Report
DRDC Atlantic CR 2010-134
October 2010


Canada

This page intentionally left blank.

# Users Guide to BellhopDRDC_V4

*Active and Passive versions*

Dr. Diana McCammon
McCammon Acoustical Consulting
Waterville, NS  B0P 1V0

Prepared By:
McCammon Acoustical Consulting
475 Baseline Road
Waterville  NS  B0P 1V0

Contract Project Manager: Dr. D. McCammon, 902-538-3003
Contract Number:  W7707-098272
CSA:  Dr. WA Roger, 902-426-3100 ext292



**Defence R&D Canada – Atlantic**

Contract Report
DRDC Atlantic CR 2010-134
October 2010

Project Scientific Authority

*Original signed by WA Roger*

---

WA Roger

Approved by

*Original signed by David Hazen*

---

David Hazen

H/TD

Approved for release by

*Original signed by Ron Kuwahara for*

---

C. Hyatt

DRP

# Abstract

The acoustic prediction model called Bellhop continues to be enhanced to more closely fit the requirements of DRDC Atlantic's Environment Modeling Manager (EMM). This version 4 contains both passive and active algorithms. In this version, linear range interpolation of the SSP and curvilinear interpolation of the bathymetry are added as input choices, and an additional output of the sampled SSP is provided. The major differences between BellhopDRDC and the web version dated May 2008 lie in the input data and file formats that have been altered to satisfy the requirements of the controlling programs within the Environment Modeling Manager. This document provides a users guide to the running of the active and passive versions of the BellhopDRDC_v4 program and the boundary loss program, and describes some plotting routines available for viewing the prediction results.

# Résumé

Les améliorations du modèle de prévision acoustique Bellhop se poursuivent afin de mieux l'adapter aux exigences du progiciel de gestion de la modélisation de l'environnement *Environment Modeling Manager* (EMM) de RDDC Atlantique. La version 4 contient des algorithmes passifs et actifs. Dans cette version, l'interpolation linéaire de distance du profil de vitesse du son (PVS) et l'interpolation curvilinéaire de la bathymétrie sont ajoutés aux choix d'intrants et une capacité de sortie supplémentaire sur le PVS échantillonné est offerte. Les principales différences entre la version RDDC du Bellhop et la version sur le Web en date de mai 2008 résident dans les formats des données d'entrée et des fichiers, qui ont été modifiés pour satisfaire aux exigences des programmes de commande de l'*Environment Modeling Manager*. Le présent document constitue un guide d'utilisation des versions active et passive de la version 4 RDDC du Bellhop ainsi que du programme de perte de transmission aux limites, et décrit certaines routines de traçage permettant de visualiser les résultats des prévisions.

This page intentionally left blank.

# Executive summary

## Users Guide to BellhopDRDC_V4: Active and Passive versions

**McCammon, D.F. ; DRDC Atlantic CR 2010-134; Defence R&D Canada – Atlantic; October 2010.**


## Introduction

Bellhop is a computer program created by Dr. Michael Porter that computes acoustic fields in oceanic environments via Gaussian beam tracing. The environment consists of an ocean that may have range variations in the sound speed profile, the bottom loss, and the bathymetry. Two programs were created for use with the Environment Modeling Manager (EMM), a tactical oceanography tool for naval planning and operations. The first is the passive acoustic version named BellhopDRDC_ray_TL_v4. The outputs from this program include transmission loss (coherent, semi-coherent, and incoherent) and ray traces. The second is the active version named BellhopDRDC_active_v4, which outputs the arrival tables, reverberation time series, target echo time series, and the signal excess versus range.

## Results

The major differences between BellhopDRDC and the web version dated May 2008 lie in the input data and file formats that have been altered to satisfy the requirements of the controlling programs within the Environment Modeling Manager. This document provides a User's Guide to the running of both the active and passive versions of the BellhopDRDC_v4 program, and describes some plotting routines available for viewing the prediction results.

## Significance

The Environment Modeling Manager is a sophisticated tactical oceanography system being developed to aid naval planning and operations. It provides tactical decision aids with accurate and consistent predictions of acoustic conditions and target detectability. The Bellhop software package is the heart of the system that provides the necessary acoustic predictions to client programs.

## Future plans

It is intended to continue enhancing the Bellhop program with more accurate models that are effective from an operational standpoint.

# Sommaire

## Users Guide to BellhopDRDC_V4: Active and Passive versions

**McCammon, D.F. ; DRDC Atlantic CR 2010-134; R & D pour la défense Canada – Atlantique; October 2010.**


## Introduction

Créé par M. Michael Porter, le programme informatique Bellhop permet de calculer les champs acoustiques en milieu océanique par traçage de faisceaux gaussiens. Le milieu est un océan qui peut présenter des variations de la portée acoustique dans le profil de vitesse du son, de la perte au fond et de la bathymétrie. Deux programmes ont été créés et seront utilisés avec l'*Environment Modeling Manager* (EMM), un outil d'océanographie tactique pour la planification et les opérations navales. Le premier est une version acoustique passive appelée BellhopDRDC_ray_TL_v4. Les résultats de ce programme comprennent l'affaiblissement acoustique (basé sur la sommation cohérente, semi-cohérente ou incohérente) et les tracés des rayons. Le deuxième est une version active appelée BellhopDRDC_active_v4, qui permet d'obtenir les tables d'arrivée, les séries de temps de réverbération, les séries de temps d'écho de cible et l'excès de signaux par rapport à la portée.

## Résultats

Les principales différences entre la version RDDC du Bellhop et la version sur le Web en date de mai 2008 résident dans les formats des données d'entrée et des fichiers, qui ont été modifiés pour satisfaire aux exigences des programmes de commande de l'*Environment Modeling Manager*. Le présent document constitue un guide d'utilisation des versions active et passive de la version 4 RDDC du Bellhop et décrit certaines routines de traçage permettant de visualiser les résultats des prévisions.

## Importance

L'*Environment Modeling Manager* est un système perfectionné dans le domaine de l'océanographie tactique pour le soutien à la planification et aux opérations navales. Il représente un outil d'aide à la décision tactique offrant des prévisions exactes et cohérentes sur les conditions acoustiques et la détectabilité des cibles. Le progiciel Bellhop est au cœur du système qui offre des prévisions acoustiques aux programmes clients.

## Perspectives

Il est prévu de poursuivre les améliorations du programme Bellhop en intégrant des modèles plus exacts qui seront plus efficaces d'un point de vue opérationnel.

# Table of contents

# List of figures

# List of tables

This page intentionally left blank.

# 1. Introduction

Bellhop is a computer program created by Dr. Michael Porter that computes acoustic fields in oceanic environments via Gaussian beam tracing. The environment consists of an ocean that may have range variations in the sound speed profile, the bottom loss, and the bathymetry. Two programs were created for use with the Environment Modeling Manager (EMM) of the Canadian Navy at the Defence R&D Canada – Atlantic (DRDC Atlantic) laboratory. The first is the passive version of Bellhop named BellhopDRDC_ray_TL_v4. The outputs from the passive program include transmission loss (coherent, semi-coherent, and incoherent) and ray traces. The second is the active version of Bellhop named BellhopDRDC_active_v4. The outputs from the active program include the arrival tables, reverberation time series, target echo time series, and the signal excess versus range.

Changes in version 4:

The changes between this fourth version and the previous version [1] include:

1. The addition of a range interpolation scheme for range dependent sound speed profiles.
2. The addition of curvilinear interpolation for bathymetry.
3. The incorporation of changes from Dr. Porter's Web version of Bellhop dated May 2008.

The Fortran coding approaches are similar to the previous versions. They consist of a frontend program that reads the input files and writes the output files, and a subroutine named BellhopDRDC_*_v4 (where * represents either 'active' or 'ray_TL'). This structure was used to enable repeated calls to the subroutine Bellhop from within the frontend, for looping calculations over source depth, frequency, or bearing, for example. It is anticipated that the user may rewrite or replace the frontend algorithm to suit his own needs.

Also included in this users guide is a simplified program to compute surface and bottom loss for separate analysis. Finally, also included are examples of simple IDL plot routines for: TL vs range, (full field or single depth), ray tracing, an SSP map, reverberation, signal excess, surface loss, surface scattering strength, bottom loss, and bottom scattering strength.

# 2. BellhopDRDC_ray_TL_v4

The BellhopDRDC_ray_TL_v4 model is intended for passive predictions of ray paths and transmission loss. This model consists of five Fortran source files and their subroutines:

1. datamod_ray_TL_v4.f90 – module with data array declarations
2. refcomod_ray_TL_v4.f90 – module with reflection coefficient array declarations and some loss models
    - CALCbotRC – compute bottom reflection coefficients using MGS NAVOCEANO routine
    - CALCtopRC – compute surface reflection coefficient using either Modified Eckart or Beckmann-Spezzichino
    - BOTT_NEW – MGS bottom loss function
    - SURF_NEW – Surface loss function using Bechmann-Spezzichino
    - LFSOPN – Low Frequency Open Ocean Surface loss using Modified Eckart
3. sspmod_ray_TL_v4.f90 – module with sound speed array declarations
4. frontend_ray_TL_v4.f90
    - frontend_ray_TL_v4 – main program and outputs to Bellhop.log and **TL.txt*.
    - clean_up – deallocate ray structure arrays
    - Raywrite – write out to *rays.txt* the ray trace path information
    - READIN_v4 – reads file *runinput_v4.inp* and allocates and initializes arrays for range and receiver depth
    - READBTY_v4 – reads file *bathy.inp* and allocates arrays for bathymetry
    - READSVP_v4 – reads file *speed.inp* and allocates arrays for sound speed
    - READBOTLOSS_v4 – reads file *bottomloss.inp* and allocates arrays for which ever bottom type was specified
    - READBPATTERNS_v4 – reads file *beampattern.inp* for sensor beam pattern, allocates arrays and converts loss to pressure coefficient
    - TMP_SPP – function to convert temperature to sound speed using Leroy's equation
    - dumpsspmap – write out to *SSPmap.txt* the SSP sampled over range and depth for contour plot
5. bellhopDRDC_ray_TL_v4.f90
    - BellhopDRDC_ray_TL_v4a – beginning of bellhop algorithm- initializes arrays, calls ray trace and calls Grab style TL computation
    - Trace – traces a ray for each launch angle
    - Step – takes a single step along the ray path
    - Reducestep – computes step size to land on key points
    - Reflect – changes ray direction and computes amplitude and phase at reflection. The two-layer geoacoustic bottom loss is embedded in this subroutine.
    - REFCO – interpolates for reflection coefficients from table if needed
    - INFLUGRB – Gaussian beam contribution to complex pressure for TL
    - Quad – chooses method of interpolation of sound speed with depth and range
    - Linear – preferred method of SSP range interpolation
    - Smoother – Savitsky-Golay smoothing filter for coherent TL
    - Thorpe – Thorpe attenuation
    - CRCI – converts real wave speed and attenuation to a single complex wave speed

ERROUT – outputs error messages

The three module files, datamod_ray_TL_v4.f90, refcomod_ray_TL_v4.f90, and sspmod_ray_TL_v4.f90 contain the data arrays and declarations, and must be compiled first. The executable is named BellhopDRDC_ray_TL_v4.exe

To run the program, place the executable BellhopDRDC_ray_TL_v4.exe in your working directory or on your path. Place the five input files listed below in your working directory. Then click on the .exe icon or use the windows start/run command. If programming in IDL, the spawn command can be used to run the executable. For example, the command to run this in IDL is: spawn, 'BellhopDRDC_ray_TL_v4.exe', result, /noshell.

## 2.1 Input files

There are five input files: runinput_v4.inp, speed.inp, bottomloss.inp, bathy.inp and beampattern.inp. The formats are free field, so the values on each row do not occupy specific column positions, but only need be separated by a space.

### 2.1.1 Runinput_v4.inp

This file contains scenario and runtime choices, as defined in Table 1. In this table, the following alphabetic choices are defined:

‘$X_1X_2X_3X_4$’ = the run choice options, consisting of 4 letters:

- $X_1$ = type of output
  - C = Coherent transmission loss in output file *CTL.txt*
  - S = Semi-coherent transmission loss in output file *STL.txt*
  - I = Incoherent transmission loss in output file *ITL.txt*
  - R= Ray trace path information in output file *rays.txt*
- $X_2$ = SSP range interpolation method
  - N = none, uses abrupt change
  - L = Linear, the preferred method and default
- $X_3$ = Bathymetry range interpolation method
  - L = piecewise linear, the preferred method and default
  - C = curvilinear interpolation
- $X_4$ = Flag for using Thorpe volume attenuation
  - T = use Thorpe attenuation, the preferred choice and default
  - N = use no attenuation

‘S’ = surface loss model choice

- ‘B’ = Beckmann Spezzichino surface loss
- ‘E’ = Modified Eckart low frequency open ocean surface loss, default model

(Note that the bottom loss model is chosen in the bottomloss.inp file)

*Table 1. runinput_v4.inp file structure*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1. | title | up to 70 characters enclosed in single quotes |
| 2. | frequency | Hz |
| 3. | source depth | Meters |
| 4. | number of receiver depths | # |
| 5. | top and bottom of receiver depth array | Meters- note: needs slash at end value to denote an array- a single value can also be used |
| 6. | Range step for output; longest range | Meters; Kilometres |
| 7. | wind speed; surface loss model choice ‘S’ | Knots; ‘S’ ={B,E} |
| 8. | Run Choice options ‘$X_1X_2X_3X_4$’ | ‘{C,S,I,R} {L,N} {L,C} {T,N}’ choosing one letter from each group to comprise a 4 letter sequence in single quotes |
| 9. | Internal step size; number of rays; start angle; stop angle; kill-after-bounce number | Default value = -1 Internal step size in m; angles in degrees; negative angles first. Default is -15 to 15 deg, and 100 bounces<br>For ray tracing, the number of rays and start and stop angle should be selected by the user. For transmission loss, these should be defaulted to -1. |
| 10 | Range smoothing flag, dumpSSP flag | Meters, Default = -1, no smoothing Smoothing only affects the ‘C’ coherent TL<br>Dumpssp flag = 1 to write out SSP in range and depth into the file SSPMap.txt |

```
‘Emerald basin toward Sambro Bank’    !title
1200.0                  !frequency (Hz)
70.                     !source depth, m
200                     !number of receiver depths
0. 250. /               !top and bottom of receiver depth array, or whole array, ** needs the slash
250. 50.                !range step (m) and maximum range (km)
10.0  'B'               !windspeed(kts), surface loss model {B,E, }
'ILLT'                  !run choice {I,S,C,R}; interpSSP {L,N}; interpBathy {L,C}; volatten {T,N}
-1 -1 -1 -1 -1          !defaults step size (m); number of rays; start angle; stop angle; Kill-after-bounce
-1  1                   !smoothing default (-1=off, 1=on); dumpSSP flag (-1=off, 1=on)
```

*Figure 1. Sample runinput_v4.inp file.*

### 2.1.2 Speed.inp

This file contains sound speed profiles in depth and range.

*Table 2. speed.inp file structure*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1 | Number of range dependent profiles | |
| 2 | Range to profile; number of points in that specific profile, n | km |
| 3 to n+3 | Depth; speed or temperature | M; m/sec or ºC |
| | Repeat from 2 for each profile | |

Note: there should always be a point at the surface and at or below the deepest bathymetry point.

```
  3                      !# range dependent profiles
 0.      18              !range(km), #points per profile
 0.    1498.0            !depth(m), speed(m/sec)
30.0   1499.2
35.0   1491.7
40.0   1483.8
45.0   1475.4
50.0   1466.5
75.0   1468.2
80.0   1470.0
90.0   1473.5
95.0   1475.3
100.0  1477.0
125.0  1479.6
150.0  1482.1
175.0  1484.6
200.0  1488.4
225.0  1489.5
250.0  1490.6
300.0  1490.5
10.      5               !range(km), #points per profile
0.    1499.0             !depth(m)  speed(m/sec)
50.0   1468.0
100.0   1470.0
200.0  1485.0
300.   1491.0
40.     9                !range(km), #points per profile
0.    1498.0             !depth(m)  speed(m/sec)
35.0   1491.7
45.0   1475.4
75.0   1468.2
90.0   1473.5
100.0  1477.0
150.0  1482.1
```

*(>> continued on next page)*

*(>>continued from previous page)*
250.0  1490.6

*Figure 2. Sample speed.inp file.*

The speed may be plotted on a single graph, spaced 10 m/sec apart using the plot routine read_plot_speed.pro. For this sample SSP file, the result is shown in Figure 3.

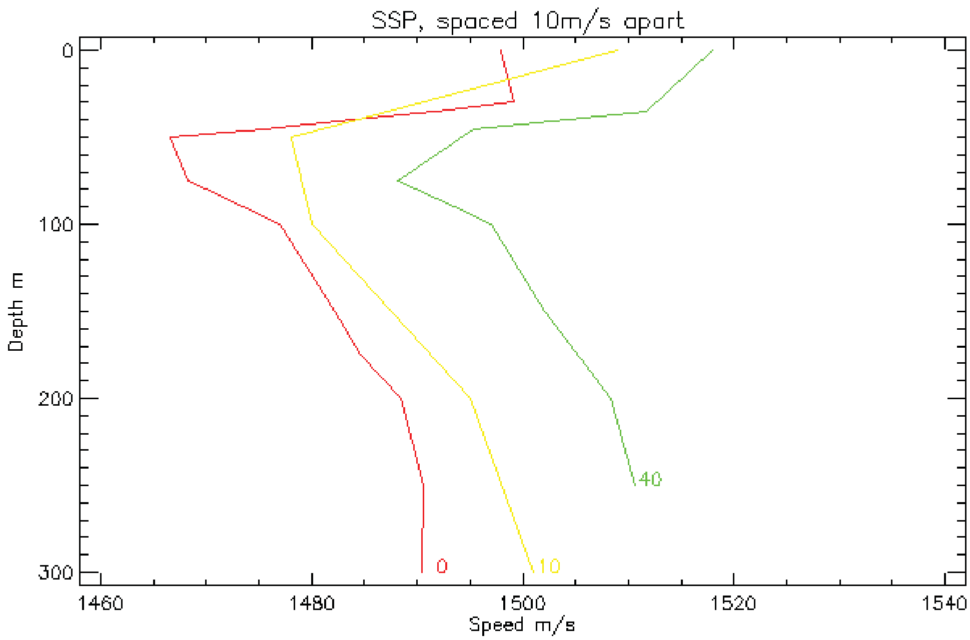


*Figure 3. SSP's plotted from speed.inp.*

### 2.1.3 Bottomloss.inp

This file contains the range dependent bottom loss descriptions.

*Table 3. bottomloss.inp file structure*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1 | Bottom treatment option; attenuation units 'XY' | 'XY': 'X'= {M,A,T} 'Y'= {F,M,W,N} |
| 2 | number of range dependent bottom sets, n | |
| 3 to n+3 | If 'X'='M': range; province number<br>If 'X'='A': range; c1; rho1; atten1; h1; c2; rho2; atten2<br>If 'X'='T': range; # of table rows<br>Angle; reflection coefficient; phase | Km; MGS province number<br>Km; m/sec; g/cc; units of 'Y'; m; m/sec; g/cc; units of 'Y'<br>Km; number of rows<br>Degrees; decimal fraction; degrees |

In this table, the following are defined:

‘X’ = the bottom treatment option
- 'M' = MGS or HFBL provinces
- 'A' = Two Geoacoustic fluid layers (no shear)
- 'T' = Read in table of pressure reflection coefficients and phases as a function of grazing angle

‘Y’ = The attenuation units which are used in the geoacoustic layers only, choices are
- 'F' = dB/(m kHz)
- 'M' = dB/m
- 'W' = dB/wavelength
- 'N' = nepers/m

```
  ‘M’                    ! Bottom option for MGS
  3                      ! number of range dependent bottom provinces
 0.    4                 ! range (km), province number
 10.  2
 50.  8
__________________________________________________________________
  'AF'                   !A=geoacoustic, F= dB/m kHz
 6                       !number of bottom regions
 0.0  1453.  1.41  0.038  10.  1557.  1.73  0.156
 7.4  1807.  2.175 0.131  30.  3500.  2.60  0.020
14.8  1630.  2.00  0.157   4.  1880.  2.175 0.085
16.7  1772.  2.11  0.142  10.  1880.  2.175 0.085
44.4  1630.  2.00  0.157   4.  1880.  2.175 0.085
48.1  1807.  2.175 0.131  30.  3500.  2.60  0.020
!range   c1    rho1   atten1  depth  c2   rho2    atten2
__________________________________________________________________
 ‘T’                     ! Bottom option for table of reflection coefficients vs angle
  1                      ! number of sets of tables
 0.  5                   ! range (km), number of entries in table
  0.  1.0  0.0           ! angle (deg), reflection coef fraction, phase (deg)
 10. 0.8  0.0
 30. 0.7  0.0
 50. 0.5  0.0
 90. 0.5  0.0
```

*Figure 4. Three samples of bottomloss.inp files.*

### 2.1.4 Bathy.inp

This file contains the bathymetry.

*Table 4. bathy.inp file structure*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1 | Number of bathymetry points, n | |
| 2 to n+2 | Range; depth | Km; m |

Note: needs a point at zero range

```
    12                          !number of bathymetry points
  0.0   238                     !range(km), bottom depth (m)
  5.5   256
 18.5   238
 22.2   219
 24.1   183
 25.9   165
 27.8   110
 42.6   128
 44.4   146
 46.3   165
 47.2   183
 55.5   219
```

*Figure 5. Sample bathy.inp file.*

### 2.1.5 Beampattern.inp

This file contains the receiver vertical beam pattern in dB.

*Table 5. beampattern.inp file structure*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1 | Number of vertical angles, n | |
| 2 to n+2 | Angle; loss | Deg; dB |

```
3                               !number of angles
-90.  0.0                       !angle(deg); loss(dB)
  0.  0.0
 90.  0.0
```

*Figure 6. Sample beampattern.inp for an omni-directional beam.*

## 2.2 Output files

There are six possible output files from BellhopDRDC_ray_TL_v4. The computed data is written to .txt files in ASCII, depending on the runtime choices made in the input file runinput_v4.inp.

*ITL.txt* created by run choice 'I'
*STL.txt* created by run choice 'S'
*CTL.txt* created by run choice 'C'
*rays.txt* created by run choice 'R'
*SSPmap.txt* created by dumpsspflag = 1
*bellhop.log*

### 2.2.1 CTL.txt, ITL.txt or STL.txt

This file contains the transmission loss (either coherent, semi-coherent or incoherent, depending on the choice made in runinput_v4.inp). At the top, it lists the run title, frequency and source depth. The next line contains the number of ranges and number of receiver depths. Following this are listed the range array in km, then the receiver depth array in m, then transmission loss in dB by range and receiver depth. An example listing is shown in Figure 7.

```
BELLHOP- Emerald basin toward Sambro Bank

 1200Hz   21.0.m source depth
       200         200
 0.2500000   0.5000000   0.7500000   1.000000    1.250000
  1.500000   1.750000    2.000000    2.250000    2.500000
  2.750000   3.000000    3.250000    3.500000    3.750000
  4.000000   4.250000    4.500000    4.750000    5.000000
    . . .
  49.00000   49.25000    49.50000    49.75000    50.00000
 0.000000    1.256281    2.512563    3.768844    5.025126
  6.281407   7.537688    8.793970    10.05025    11.30653
    . . .
  244.9749   246.2312    247.4874    248.7437    250.0000
  200.0000   57.63440    63.03972    68.00800    71.89261
  74.56378   78.60585    70.51897    71.89424    73.83934
      . . .
```

*Figure 7. Portion of an ITL.txt output.*

In the output sometimes the first several transmission loss values are 200dB, as shown above. This default loss occurs if the first depth point was high above the source and the trace angles were defaulted (runinput_v4.inp, line 9, start and stop angle) to be ±15º, therefore this point might not have been ensonified in a downward refracting profile. The same default loss can occur at a deep depth point below the source. To provide very short range loss values it is necessary to open up the angle fan to ±25º or more, at the cost of some runtime.

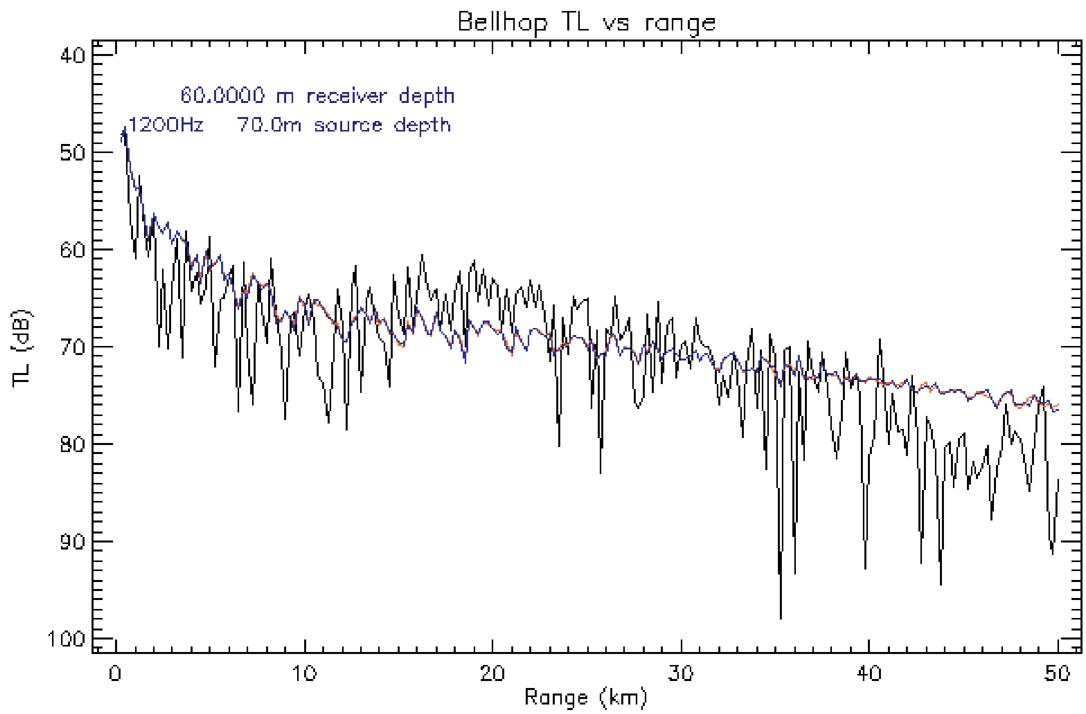


*Figure 8. Example transmission loss plot for 60m receiver. Black is coherent, CTL.txt. Red is semi-coherent, STL.txt and blue is incoherent, ITl.txt, using 200 range points.*

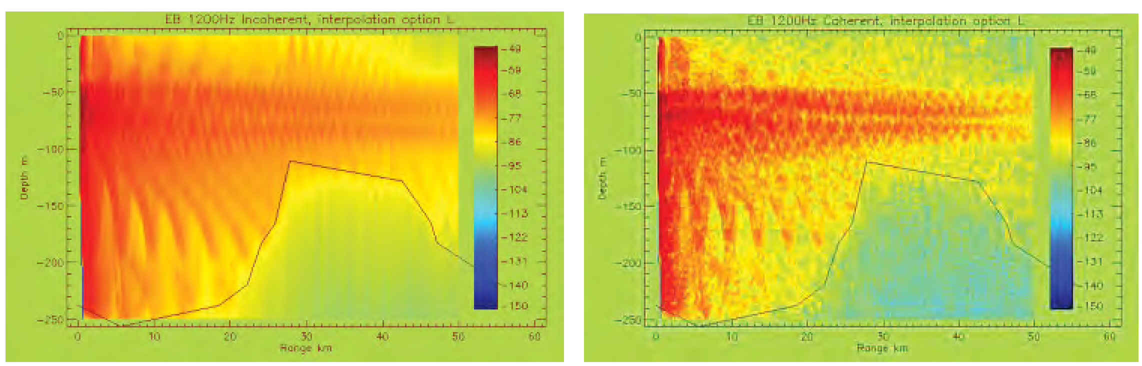


*Figure 9. Left: example of full field plot of ITL.txt (Incoherent calculation) which was computed using 200 receiver depths from 0 to 250m. The bathymetry is plotted as a line along the bottom. Right: full field plot of CTL.txt (coherent calculation). Both used the linear SSP range interpolation and linear bathymetry interpolation with a 70m source depth.*

Figure 9 displays a good example of a potential pitfall in using this range dependent model. Note that in the figure, there are places where the field extends below the bathymetry, since the receiver array was defined to 250m to cover the deeper part of the water, but the bathymetry then rises to 110m. The portion of the field below the bathymetry is not a true representation of the acoustic field there. The loss generated by Bellhop on reflection from the bottom into the water column is correct, however the field shown within the bottom does not have the right level. It is an artifact of the Gaussian beam representation in Bellhop. It should be ignored or blanked out in the figure, and in all other Bellhop applications, care should be exercised that the user is only working with transmission loss values from those receivers positioned above the bathymetry. When receivers are defined that extend below the bathymetry at some point, a warning is generated and written to bellhop.log.

### 2.2.2 Rays.txt

The output file named rays.txt contains ray tracing information. Its structure is to echo some of input choices in the first few lines. The number of rays being traced is listed (in the case shown below it is 6). Then in a loop over the number of rays, each ray is described by the launch angle (-10.0) and number of steps or points in the trace (4521). Finally, the [r,z] coordinates, ray angle, delay time, and number of surface and bottom bounces of each ray are listed for each step. Both r and z are given in m, angle is in degrees, and time is given in seconds.

Figure 10 shows a portion of the rays.txt listing for the 70m source, and it demonstrates an anomaly that always occurs in Bellhop ray traces. That is that there are often a number of repeated points ( see for example the line at 184.1597m) that result as Bellhop tries to place a ray exactly on a sound speed depth or a defined bathymetry range. The new subroutine in Bellhop called reducestep.f90 is responsible. It does not affect the result but it does enlarge the file sizes.

The rays.txt output can be plotted with the bathymetry, as shown in Figure 11. The case shown was computed using the default 20 rays from -10º to +10º, with a 70m source depth so that the figure would correspond directly to the full field transmission loss plot in Figure 9.

```
BELLHOP- Emerald basin toward Sambro Bank

   1200.0Hz      70.0m source depth
Kill Trace after   50 bottom bounces
        20
 -10.00000          4735
 0.0000000E+00   70.00000     -9.999999     0.0000000E+00     0     0
 1.4772117E-07   70.00000     -9.999999     1.0218958E-10     0     0
  14.77122      67.39022     -10.03921     1.0219570E-02     0     0
  29.54066      64.77034     -10.07839     2.0440368E-02     0     0
     . . .
  184.1567      40.00029     -5.499722     0.1269062        0     0
  184.1597      40.00000     -5.499527     0.1269083        0     0
  184.1597      40.00000     -5.499527     0.1269083        0     0
  199.1020      38.67991     -4.599111     0.1370111        0     0
     . . .
```

*Figure 10. Portion of a rays.txt output.*

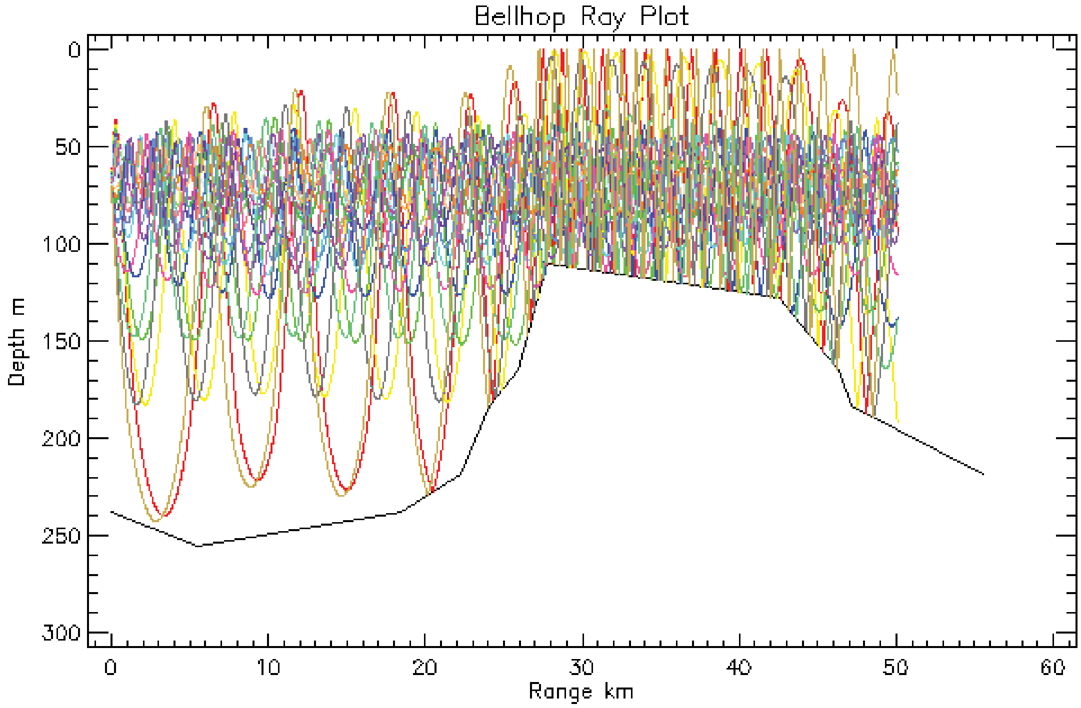


*Figure 11. Plot of rays.txt for a 70m source showing the reflections from the uneven bathymetry.*

### 2.2.3 Bellhop.log

This file contains a log of the runtime statements generated in any run. Some inputs are echoed, and any warnings or errors are listed here as generated by the Bellhop code.

```
BELLHOP- Emerald basin toward Sambro Bank

 Frequency=    1200.000
 Source depth=    70.00000
 range step(m) =    250.0000      Maximum range(km)=    50.00000
 Wind speed (kts)=    10.00000
 Beckman-Spezzichino surface loss
 Runchoice= Ray trace
 Thorpe volume attenuation used for frequency dependent water column absorption
 No range smoothing
 range,depth computed SSP matrix written to SSPmap.txt
 Number of receiver depths=          200
 Top and bottom of Receiver depths=   0.0000000E+00   250.0000
 Bathymetry interpolation is linear piecewize
 Number of bathymetry points=           12
         Range(km)  Depth(m)
          1  0.0000000E+00   238.0000
          2   5.500000       256.0000
          3   18.50000       238.0000
```
*(>> Continued on next page)*

```
(>> Continued from previous page)
                    . . .

 *** WARNING ***
 Generated by program or subroutine: Bathy.inp
 Receiver deeper than bathymetry

 Number of sound speed profiles=           3
 Linear range interpolation used on SSP
   Range(km)=  0.0000000E+00
         1  0.0000000E+00   1498.000
         2   30.00000       1499.200
         3   35.00000       1491.700
                    . . .
Bottom option= Acoustic parameters
 Atteunation unit choice= dB/(m kHz)
 Number of range dependent bottom properties=           6
  Range(km)=   0.0000000E+00
 c2,rho2,a2,h2,c3,rho3,a3
   1453.000       1.410000      3.7999999E-02   10.00000       1557.000
   1.730000      0.1560000
  Range(km)=    7.400000
 c2,rho2,a2,h2,c3,rho3,a3
   1807.000       2.175000      0.1310000       30.00000       3500.000
   2.600000      2.0000000E-02
                    . . .
 Sensor Beampattern
 angle(deg), bpat(dB)
  -90.00000      0.0000000E+00
  0.0000000E+00  0.0000000E+00
   90.00000      0.0000000E+00

 Successful input read

 BELLHOP- Emerald basin toward Sambro Bank

Number of rays =          20 from  -10.00000     deg to    10.00000     deg
 Kill-after-bounce          50
 Minimum Step size(m) =    15.0000000000000

 CPU Time =        1.07     seconds
```

*Figure 12. Sample portions of a bellhop.log.*

### 2.2.4 Sspmap.txt

This file contains a 200x200 sample map of the sound speed profile with range using the interpolation scheme selected in runinput_v4.inp, line 8. The first line of sspmap.txt lists the number of range and depth points and a letter indicating the type of range interpolation, L=linear and N=none. Then the range points are listed, followed by the depth points, followed by the sound speed in range and depth.

```
200        200  L
 0.0000000E+00  0.2500000      0.5000000      0.7500000       1.000000
  1.250000       1.500000       1.750000       2.000000       2.250000
  2.500000       2.750000       3.000000       3.250000       3.500000
   . . .
```

*Figure 13. Sample portion of the sspmap.txt file.*

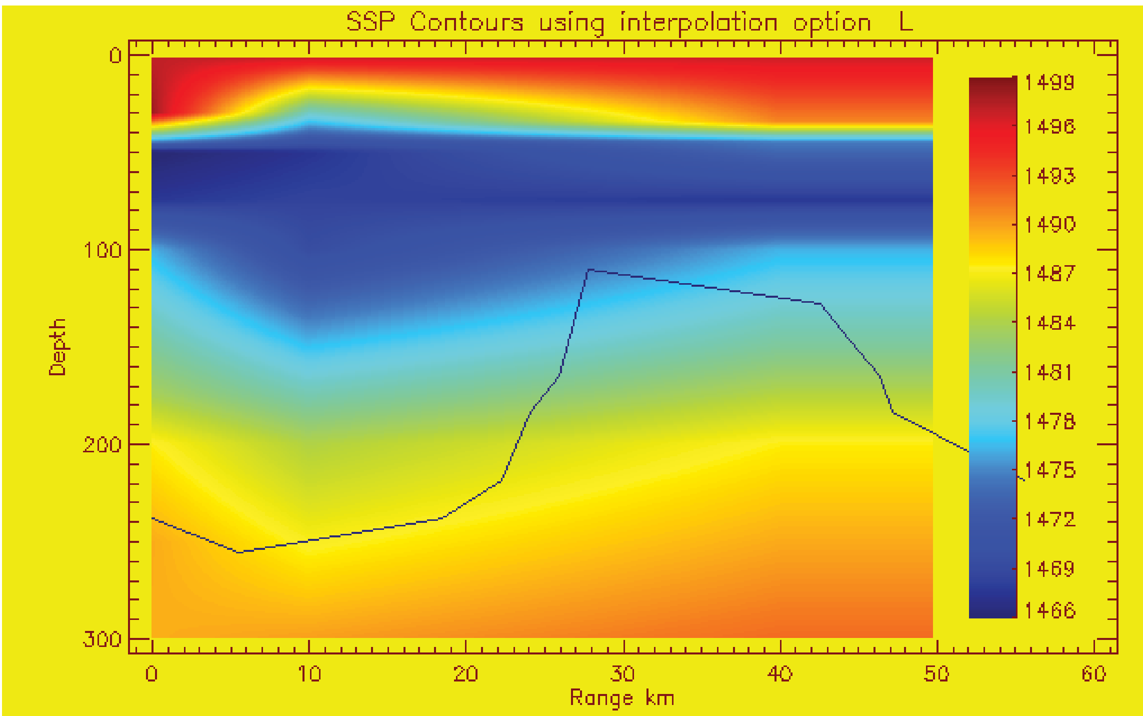


*Figure 14. Example of sspmap.txt, a contour plot of the SSP with range and depth.*

## 2.3 Plot routines

Several IDL plot routines have been prepared to provide a simple graphic representation of the output products from BellhopDRDC. These should be freely altered to suit the users' data and output requirements.

Read_tl_plot_loss.pro: Routine to read each of the xTL.txt output files, as in the example in Figure 8. The user will be asked to enter the receiver depth. If it does not exactly match one of the computed depths, the plot routine will choose the next closest depth. Presently the plot routine is set to open each of the three xTL.txt files and over plot them all in color.

Read_tl_plot_field.pro:  Routine to read the xTL.txt output and the bathy.inp file as in the examples in Figure 9.  Presently the plot routine is set to open each of the three xTL.txt files and plot each in a separate window.

Read_rays_plot_trace.pro: Routine to read the output in rays.txt and the bathymetry in bathy.inp and produce a ray trace figure as shown in Figure 11.

Read_plot_speed.pro:  Routine to read the input file of SSP, speed.inp, and overplot all the profiles spacing them 10m/s apart as shown in Figure 3.

Read_sspmap.pro:  Routine to read the sspmap.txt that was created if the *dumpsspmap* option was selected in runinput_v4.inp. The plot shows a contour map of the SSP in range and depth as shown in Figure 14 with the bathy.inp file overplotted.

# 3. BellhopDRDC_active_v4

The BellhopDRDC_active_v4 model is intended for active predictions of bistatic target echo time series, bistatic reverberation and active signal excess using SALT (Sound Angle, Level and Time) tables produced by the incoherent output from Bellhop. This model consists of fourteen Fortran source files and their subroutines:

1. datamod_active_v4.f90 – module of data array declarations and size limitations
2. refcomod_active_v4.f90 – module of reflection coefficient array declarations

    CALCbotRC – computes bottom reflection coefficients using MGS NAVOCEANO routine
    CALCtopRC – computes surface reflection coefficient using either Modified Eckart or Beckmann-Spezzichino
    BOTT_NEW – MGS bottom loss function
    SURF_NEW – Surface loss function using Bechmann-Spezzichino
    LFSOPN – Low Frequency Open Ocean Surface loss using Modified Eckart

3. saltmod_active.f90 – module with SALT table array allocation declarations
4. SEmod_active.f90 – module with SE input variable allocation declarations
5. SSPmod_active_v4.f90 – module with sound speed array allocation declarations
6. frontend_active_v4.f90

    frontend_active_v4 – main program
    Setdefaults – assigns default inputs for active applications

7. readinput_active.f90

    readinput_active – reads input files for SE, speed, bathy, beam patterns
    READBOTLOSS_S – reads bottom loss and allocates arrays for whichever bottom type was specified
    READreverb – read user input reverberation table
    CALCreverb - Rough estimate of reverb in dB using 40log(t) fall-off

8. bellhopDRDC_active_v4.f90

    BellhopDRDC_active_v4 – beginning of bellhop algorithm which: initializes arrays, calls ray trace and calls TL computation, and defines extra receiver points on surface and bottom (conforming to bathymetry) for reverberation
    Trace – traces a ray for each launch angle
    Step – takes a single step along the ray path
    Reducestep – refines the step length to land on points of interest
    Reflect – changes ray direction and computes amplitude and phase at reflection. The geoacoustic bottom loss is embedded in this subroutine
    REFCO – interpolates for reflection coefficients from table if needed
    INFLUGRB – computes Gaussian beam contribution to complex pressure. The point on the bottom is shifted at each step to conform to the bathymetry. Results are sent to AddArr

QUAD – finds sound speed and gradient using interpolation style *None* or *Linear*
Linear – Bilinear quadrilateral interpolation of SSP
TMP_SPP – function to convert temperature to sound speed using Leroy's equation
Smoother – Savitsky-Golay smoothing filter
AddArr – creates arrival SALT table for surface, bottom, and target depths from all sensors and transmitter along all bearings
Thorpe – Thorpe attenuation
CRCI – converts real wave speed and attenuation to a single complex wave speed
ERROUT – outputs error messages

9. envstore_v4.f90 – moves range dependent environments from input storage arrays into Bellhop runtime arrays for each bearing and sensor. Computes internal trace step size, deltas, based on the minimum depth of the bathymetry on that bearing
10. reverb.f90 – computes bistatic reverberation from surface and bottom using SALT tables for each sensor and target bearing. Formulas for various surface and bottom scattering strengths are embedded. Output is reverberation time series without source level for each sensor
11. scatstrength_v4 – contains all scattering strength models
    - OE – Ogden-Erskine surface scattering strength
    - CH – Chapman-Harris surface scattering strength
    - EC – Ellis-Crowe bottom scattering strength
    - LB – Lambert's rule bottom scattering strength
    - OM – Omni bottom scattering strength
12. salt_v4.f90 – stores SALT tables for each sensor and bearing
13. SE_active.f90 – computes signal excess from reverb, target echo and noise for each sensor and bearing. Source level and target strength are applied. The result is saved as a function of range, target depth, target bearing and sensor.
14. targetecho.f90 – computes bistatic signal intensity as a function of time, target range and depth along target bearing. Output is signal time series without source level or target strength.
15. writeoutput_active.f90

WriteArrival – writes SALT arrival tables for each sensor. Note this output file is only a portion of the SALT tables on the target bearing.
Writerevb – writes reverberation time series for each target bearing and sensor
WriteSE – writes SE for target bearing, target depth, range and sensor
Writesignal – writes target echo time series for each target bearing, target depth, range and sensor
WriteTL – writes TL from transmitter and from sensor to target vs range for target depth

The five module files, datamod_active_v4.f90 and refcomod_active_v4.f90, saltmod_active.f90, SEmod_active.f90, and SSPmod_active_v4.f90 contain the data arrays and declarations, and must be compiled first. The executable is named BellhopDRDC_active_v4.exe

To run the program, place the executable BellhopDRDC_active_v4.exe in your working directory or on your path. Place the five input files listed below in your working directory. Then click on the .exe icon or use the windows start/run command. If programming in IDL, the spawn command can be used to run the executable. For example, the command to run this in IDL is: spawn, 'BellhopDRDC_active_v4.exe', result, /noshell .

## 3.1 Input Files

There are five input files: active_general.inp, radial_ssp.inp, radial_bottomloss.inp, radial_bathy.inp and beampat_active.inp. The formats are free field, so the values on each row do not occupy specific column positions, but only need be separated by a space.

For active use, the following are defaulted in the file frontend_active_v4, subroutine setdefaults:

runchoice = 'I'; computes incoherent pressure
Thorpe = 'T'; uses Thorpe attenuation
numbotkill = 100; only allow up to 100 surface or bottom bounces
angle1, angle2 = ±25 deg; range of up and down angles to be traced
deltas0= -1; default to internally calculate the ray trace range step
Nbeams0= -1; default to internally calculate the number of rays to trace

The following are the current array size limitations that are set in datamod_active_v4.f90:

Nprofmax = 25; max # of different SSP's and/or bottom losses along any single bearing
Nsspmax = 200; max # of points in any SSP
NBathymax = 500; max # of points in any bathymetry track
Ntab = 181; max # of table points in bottomloss and beampattern table input
MxnArr = 100; max # of arrivals for each (depth,range) in SALT tables
Mxn = 200000; max # of steps in each ray trace

### 3.1.1 active_general.inp

This file contains the basic choices for the scenario, system parameters, scattering strength models, and surface loss models.

Table 6 lists the model choices and Figure 15 shows a sampling listing. The options are:

- 'M' = surface loss model choice
  - 'B' = Beckmann Spezzichino surface loss
  - 'E' = Modified Eckart low frequency open ocean surface loss = default model
  - (Note that the bottom loss model is chosen in the NUWbottomloss.inp file)
- 'SM' = surface scattering strength model choice
  - 'OE' = Ogden-Erskine surface scattering strength- a combination Chapman Harris with low wind speed algorithms
  - 'CH' = Chapman Harris surface scattering strength = default model
- 'BM' = bottom scattering strength model choice
  - 'EC' = Ellis and Crowe= Lambert's rule with a high angle facet scattering term= default model
  - 'LB' = Lambert's Rule with Mackenzie Coefficient
  - 'OM' = Omni-directional Rule with Mackenzie Coefficient

‘SB’ = two letters for the interpolation choices for ssp and bathymetry
First position = ssp range interpolation, N=none, L=linear
Second position = bathymetry range interpolation, L=linear, C=curvilinear
Default string is ‘LL’, that is, both interpolations are linear

*Table 6. active_general.inp file structure*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1 | Title | 80 characters enclosed in single quotes |
| 2 | Number of receiving sensors, nsensor | |
| 3 | Array of Noise level at each sensor | Array 1 to nsensor, values separated by spaces, units=dB |
| 4 | Source level; detection threshold; target strength; system loss; pulse length | dB; dB; dB; dB; seconds |
| 5 | Array of Blast arrival time at each sensor | Array 1 to nsensor, values separated by spaces, units= seconds<br>sign will position the sensor to the right or left of the transmitter as you face the target bearing. Negative=left, positive=right |
| 6 | Frequency | Hz |
| 7 | Array of Asset depths (sensors and transmitter) | Array 1 to nsensor +1, values separated by spaces, units=m, extra last point is the transmitter depth |
| 8 | Maximum range to target | Km; Note: This will be increased internally to include the max distance between sensor and transmitter plus a pulse length. |
| 9 | Number of target depths; Target depth minimum; Target depth maximum | Program will create an array of target depths, units= m, Note: must have both min and max depth, even if the number of target depths=1 |
| 10 | Wind speed; surface loss model choice ‘M’ | kts; ‘M’={B,E}, default=E |
| 11 | Surface scattering strength model choice ‘SM’ | ‘SM’={OE, CH}, default=CH |
| 12 | Bottom scattering strength model choice ‘BM’; Mackenzie coefficient; Normal incidence bottom loss; facet width or RMS slope | ‘BM’={EC, LB, OM}, default=’EC’;<br>dB, used with all choices;<br>dB, used with ‘EC’;<br>degrees, used with ‘EC’ |
| 13 | Range interpolation choices for SSP and Bathymetry ‘SB’ | ssp = {N,L}; Bathy = {L,C}; input both choices as two letter string; default is ‘LL’ |

```
‘Test run with 1 sensor, 4 bearings’
1                                    !Number of sensors
70.8                                 !Noise level by sensor, dB
230.0  0.0  0.0  0.0  1.0            !SL, DT, TS, Syslos, in dB, pulse length in sec
-12.34                               !blasttime to sensor, right=+  left=-
1200.                                !freq
18.3  21.                            !asset depth array, including transmitter
50.                                  !maximum range to target locations   km
2  21.0  80.0                        !number of target depths, dmin, dmax
10.       'E'                        !wind speed in knots,  surface loss model
'OE'                                 !Surface scattering strength model
'EC' 27. 5. 10.                      !Bottom scattering strength model and inputs to ‘EC’ model
‘LL’                                 !SSP interp = {N,L}, Bathy interp = {L,C}
```

*Figure 15. Example of active_general.inp file.*

### 3.1.2 radial_ssp.inp

This file contains the bearing and range dependent sound speed profiles for each asset. Currently the dimensions of the sound speed arrays are limited to at most 25 profiles along each bearing, each profile having at most 200 points. At least one of the profiles along a bearing track must have as SSP point deeper than the deepest point in the bathymetry along that track. Any other profiles on that track will be linearly interpolated to that depth. The format is shown in Table 7 and a sample listing is illustrated in Figure 16.

*Table 7. radial_ssp.inp file structure.*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| (no input required) | Loop over assets (transmitter last) | Ensure sensors match the order used in radial_bathy.inp. The number of recievers nsensor is on line 1 in active_general.inp. The number of assets is nsensor+1 to include the transmitter |
| 0 (no input required) | Loop over bearing for each asset | Note: there must be the same number of bearings for all assets. The number of bearings is on line 1 in radial_bathy.inp |
| 1 | Number of profiles along each bearing | Limited to 25 |
| 2 | Range to profile; number of svp points n | Km; # pts limited to 200 |
| 3 to 3+n | Depth; speed or temperature | m; m/sec or degrees C |
| | Repeat from line 1 for next bearing | |
| | Repeat from line 0 for next asset | |

```
3            ! first receiver,  first bearing,  number of ssp on this bearing
0.        18        ! range to ssp (km); number of points in svp
0.    1498.0
30.0  1499.2
35.0  1491.7
40.0  1483.8
45.0  1475.4
50.0  1466.5
75.0  1468.2
80.0  1470.0
90.0  1473.5
95.0  1475.3
100.0  1477.0
125.0  1479.6
150.0  1482.1
175.0  1484.6
200.0  1488.4
225.0  1489.5
250.0  1490.6
300.0  1490.5
10.   5
0.        1499.0
50.       1468.0
100.      1470.0
200.      1485.0
300.      1491.0
40. 9
0.    1498.0                !depth(m) speed(m/sec)
35.0  1491.7
45.0  1475.4
75.0  1468.2
90.0  1473.5
100.0  1477.0
150.0  1482.1
200.0  1488.4
250.0  1490.6
1                   !number of ssp on this bearing
0.        18          !90 deg bearing, 18 points
0.    1498.0
30.0  1499.2
35.0  1491.7
. . .                          Repeat for each bearing and asset, transmitter last
```

*Figure 16. Example of portion of radial_ssp.inp file.*

### 3.1.3 radial_bottomloss.inp

This file contains bottom loss information for each asset and bearing. The type of bottom selected (MGS, geoacoustic, or table) will apply to all assets and bearings. Currently the dimensions of the loss arrays are limited to 25 different regions along each bearing, and for table entries, the number of points is limited to 91. Table 8 shows the available options and a sample listing is illustrated in Figure 17. The options are :

'X' = the bottom treatment option

'M' = MGS or HFBL provinces
'A' = Geoacoustic fluid layers (no shear)
'T' = Read in table of pressure reflection coefficients and phases as a function of grazing angle

'Y' = the attenuation units that are used in the geoacoustic layers only, choices are:
'F' = dB/(m kHz)
'M' = dB/m
'W' = dB/wavelength
'N' = nepers/m

*Table 8. radial_bottomloss.inp file structure.*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1 | Bottom treatment option; attenuation units | 'XY': 'X'= {M,A,T} 'Y'= {F,M,W,N} |
| (no input required) | Loop over assets (transmitter last) | Ensure sensors match the order used in radial_bathy.inp. The number of receivers nsensor is on line 1 in active_general.inp. The number of assets is nsensor+1 to include the transmitter |
| 0<br>(no input required) | Loop over bearing for each asset | Note: there must be the same number of bearings for all assets. The number of bearings is on line 1 in radial_bathy.inp. |
| 2 | number of range dependent bottom sets, n | Currently limited to 25 |
| 3 to n+3 | If 'X'='M': range; province number<br>If 'X'='A': range; c1; rho1; atten1; h1; c2; rho2; atten2<br>If 'X'='T': range; # of table rows; then loop over # table rows with angle; reflection coefficient; phase | Km; MGS province number<br>Km; m/sec; g/cc; units of 'Y'; m; m/sec; g/cc; units of 'Y'<br>Km; number of rows; Degrees; decimal fraction; degrees |
| | Repeat from line 2 for next bearing | |
| | Repeat from line 0 for next asset | |

```
 'AF'               !attenuation units db/m khz
 4                                          !0 deg bearing  - receiver #1
 0.0  1453.  1.41   0.038  10.  1557.  1.73   0.156
26.8  1547.  1.72  0.158  20.  1880.  2.175  0.085
31.5  1630.  2.00  0.157    4.  1880.  2.175  0.085
40.7  1772.  2.11  0.142  10.  1880.  2.175  0.085
 4                                          !90 deg  receiver
 0.0  1453.  1.41    0.038  10.  1557.   1.73  0.156
24.0  1547.  1.72   0.158  20.  1880.   2.175 0.085
27.8  1807.  2.175  0.131  30.  3500.  2.60   0.020
46.3  1630.  2.00   0.157    4.  1880.  2.175  0.085
 6                                          !180 deg  receiver
 0.0   1453.  1.41   0.038  10.  1557.  1.73  0.156
14.8  1807.  2.175 0.131  30.  3500.  2.60  0.020
19.4  1453.  1.41   0.038  10.  1557.  1.73  0.156
33.3  1807.  2.175 0.131  30.  3500.  2.60  0.020
44.4  1630.  2.00   0.157   4.  1880.  2.175 0.085
50.0  1772.  2.11   0.142  10.  1880.  2.175 0.085
 6                                          !290 deg  receiver
 0.0  1453.  1.41  0.038  10.  1557.  1.73  0.156
      . . .
```

*Figure 17. Example of portion of radial_bottomloss.inp showing range dependent geoacoustic parameters for several bearings.*

### 3.1.4 radial_bathy.inp

This file contains the radials and bathymetry for all radials desired. The current maximum number of bathymetry points on any radial is 500. A flag is defined to mark the radial bearing which could contain the target. This flag will be used to trigger the computation of the reverberation, target echo and SE, and when outputting the arrival tables. Table 9 shows the available options and a sample listing is illustrated in Figure 18.

*Table 9. radial_bathy.inp file structure.*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| 1 | Number of radial bearings | Number applies to all sensors and transmitter |
| (no input required) | Loop over assets (transmitter last) | The number of receivers, nsensor, is on line 1 in active_general.inp. The number of assets is nsensor+1 to include the transmitter |
| 0 (no input required) | Loop over bearings for each asset | Note: there are the same number of bearings for each asset |
| 2 | Radial bearing; number of bathymetry points n; target bearing flag | Degrees measured from line between sensor and transmitter; # of points currently limited to 500 per radial; flag identifying expected target bearing |
| 3 to 3+n | Range; depth of bathymetry | Km; m |
| | Repeat from line 2 for next bearing | |
| | Repeat from line 0 for next asset | |

```
4                         !number of radials- same for all assets
0. 11  0                  !loop over assets, radial direction phi(deg);#pts; 0/1 for target bearing
  0.0   238               !range; depth of bathy for #pts
  4.6   256
 18.5   238
 22.2   219
 33.3   183
 36.1   165
 38.0   146
 40.7   128
 41.7   110
 51.9    91
 55.6    91
90. 7  0
  0.0   238
 22.2   219
 24.1   183
 27.8   165
 31.5   146
 46.3   128
```

*(>> Continued on next page)*

```
(>> Continued from previous page)
  55.5   146
180.  12  0
  0.0   238
  2.8   219
    . . .
                          !repeat for transmitter radials
```

*Figure 18. Example portion of radial_bathy.inp file*

### 3.1.5 beampat_active.inp

This file contains the beampatterns for sensors and transmitter. In the program, these beams will be assumed to be pointing along the target bearings, defined by the target bearing flag in radial_bathy.inp, line 1. Table 10 shows the available options and a sample listing is illustrated in Figure 19.

There is presently no ability to specify towed array beams.

*Table 10. beampat_active.inp file structure.*

| ***Line #, entry*** | | ***Notes*** |
|---|---|---|
| (no input required) | Loop over assets (transmitter last) | Ensure sensors match the order used in radial_bathy.inp. The number of recievers nsensor is on line 1 in active_general.inp. The number of assets is nsensor+1 to include the transmitter |
| 1 | Number of vertical D/E angles in pattern, n | |
| (no input required) | Loop over number of angles given in line 1 | |
| 2 to 2+n | D/E angle; loss | Degrees; dB |
| | Repeat from line 1 for next asset | |

```
3                    !number of sensor D/E angles
-90.0   0.0                   !angle(deg), loss (dB)
0.0     0.0
90.0    0.0
37                   !number of transmitter D/E angles
-90.0   51.4                          !angle(deg), loss (dB)
-80.0   21.2
-70.0   28.5
-60.0   19.85
-50.0   20.55
-45.0   25.3
-40.0   19.0
-35.0   14.0
-30.0   13.66
-25.0   20.0
-20.0   18.44
-17.5   11.69
-15.0   7.72
-12.5   5.00
-10.0   3.05
-7.5    1.66
-5.0    0.721
-2.5    0.178
0.0     0.0
2.5     0.178
 . . .
```

*Figure 19. Example of portion of beampat_active.inp.*

## 3.2 Output files

### 3.2.1 Arrival.txt

This file contains incoherent ray arrival structures also called the SALT tables. While the SALT tables are computed for all bearings and assets, this output is only triggered on the target bearing defined by the flag in the input file radial_bathy.inp. For a configuration of BellhopDRDC which outputs all the SALT tables, this trigger should be set to 1 on all bearings.

As illustrated in Figure 20, this file begins with the title as input from active_general.inp. Next, it lists the frequency and sensor depth; then the bearing angle and sensor number. Next it lists by column the target depth(m), range(km), acoustic intensity, phase(rad), delay time(sec), source angle(deg), target angle(deg), number of reflections from the surface, and the number of reflections from the bottom. A header with abbreviations of these outputs is given for the reader's convenience. The Fortran output format for these numbers is (f7.1,f7.2,2e12.4,f7.3,2f7.2,2i4). This listing is repeated for each bearing that was designated a target bearing in the input file radial_bathy.inp and for each sensor and the transmitter (last listings). The write statements are in subroutine Writearrival in the file writeoutput_active.f90. An example plot of the arrival angle vs range is plotted in Figure 21.

The tables include all the target depths that were specified in active_general.inp on line 9. They also include entries for the surface and bottom that are required to compute reverberation. The surface entries are listed first at 0 depth, then come the target depth entries, followed by the bottom entries. Note that the depths listed for the bottom entries change with range as the bottom contour changes.

The amplitude in this table is the incoherent acoustic intensity. The transmission loss from these entries is TL=10*log(sum of entries at the same range and depth). It is possible to change the default run choice setting to produce a coherent SALT table, however in that case, the amplitude in the table is an acoustic pressure and the phase and delay time must be used to produce a coherent intensity. In that case, the computer codes in the other active products of reverberation, target echo time series and signal excess that are programmed to use intensity inputs would also have to be changed to a coherent calculation.

```
'Test run with 1 sensor, 4 bearings'

1200Hz  18.30m source depth
 290. deg Bearing  1 sensor number
 Tdepth  Range Intensity     Phase-rad   Time  Sangle Rangle Ntop Nbot
    0.0   0.75  0.7292E-06  0.3142E+01  0.502  -0.89   1.96   1   0
    0.0   0.75  0.7174E-06  0.0000E+00  0.502  -0.20  -1.35   0   0
    0.0   0.75  0.1607E-07  0.0000E+00  0.502   0.10  -1.05   0   0
    0.0   0.75  0.7408E-09  0.0000E+00  0.502   0.30  -0.86   0   0
    0.0   0.75  0.7176E-09  0.0000E+00  0.498   1.58  10.12   0   0
    0.0   1.51  0.1182E-09  0.5528E+01  1.067 -16.60  16.99   2   1
    0.0   1.51  0.2881E-08  0.5756E+01  1.067 -16.40  16.81   2   1
                    . . .
```

*Figure 20. Portion of output file arrival.txt showing some of the surface entries using Figure 15 input.*

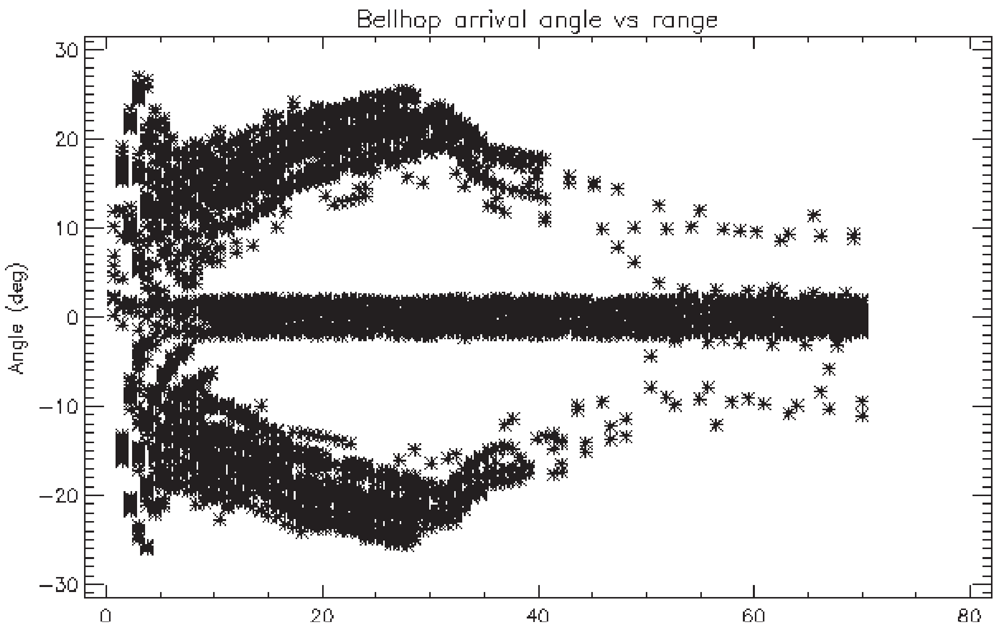


*Figure 21. Example of arrival angle vs range plotted using arrival.txt for the 21m target depth.*

### 3.2.2 Reverb.txt

This file contains the surface and bottom reverberation as a function of time, with the source level removed. The first line contains the title from active_general.inp. The next line gives the number of time points, the sensor number and depth, and the target bearing as specified in radial_bathy.inp. Next is a listing of the time array. Then follows the bottom reverberation time series in dB (without SL), and the surface reverberation time series in dB (without SL). This file structure is repeated for each target bearing and each receiving sensor. The write statements are in subroutine Writerevb in the file writeoutput_active.f90.

The -300.0000 value shown at the start of the reverberation section of Figure 22 is a default value

```
'Test run with 1 sensor, 4 bearings'

189          1   18.30000     21.00000   290.0000
  0.5000000     1.000000     1.500000     2.000000     2.500000
   3.000000     3.500000     4.000000     4.500000     5.000000
   5.500000     6.000000     6.500000     7.000000     7.500000
      . . .
-300.0000    -300.0000    -300.0000    -300.0000    -300.0000
-156.5099    -130.0203    -128.0659    -130.6896    -131.9055
  -132.8520    -136.2401    -137.7021    -140.5639    -145.7615
  -147.2074    -148.5177    -150.4951    -152.6780    -154.4710
     . . .
```

*Figure 22. Selected portions of Reverb.txt output file using Figure 15 loss models.*

designating no reception at that time. To obtain values at earlier times, the defaulted start and stop trace angles should be increased from ±25 deg to perhaps ±45 deg.

While the reverberation is an important computation in its own right, this file is primarily intended to be combined with the signal.txt file to produce the signal excess that may be computed at a later date or in another language. The source level is not included to make the signal excess computation more flexible.

In Figure 23, the top plot used the Beckmann Spezzichino ‘B’ surface loss, the Chapman Harris ‘CH’ (surface) and Omni ‘OM’ (bottom) scattering strengths as models chosen in active_general.inp. By way of contrast, the bottom plot used Modified Eckart ‘E’ surface loss, and the Ogden Erskine ‘OE’ (surface) and Ellis Crowe ‘EC’ (bottom) scattering strengths. This

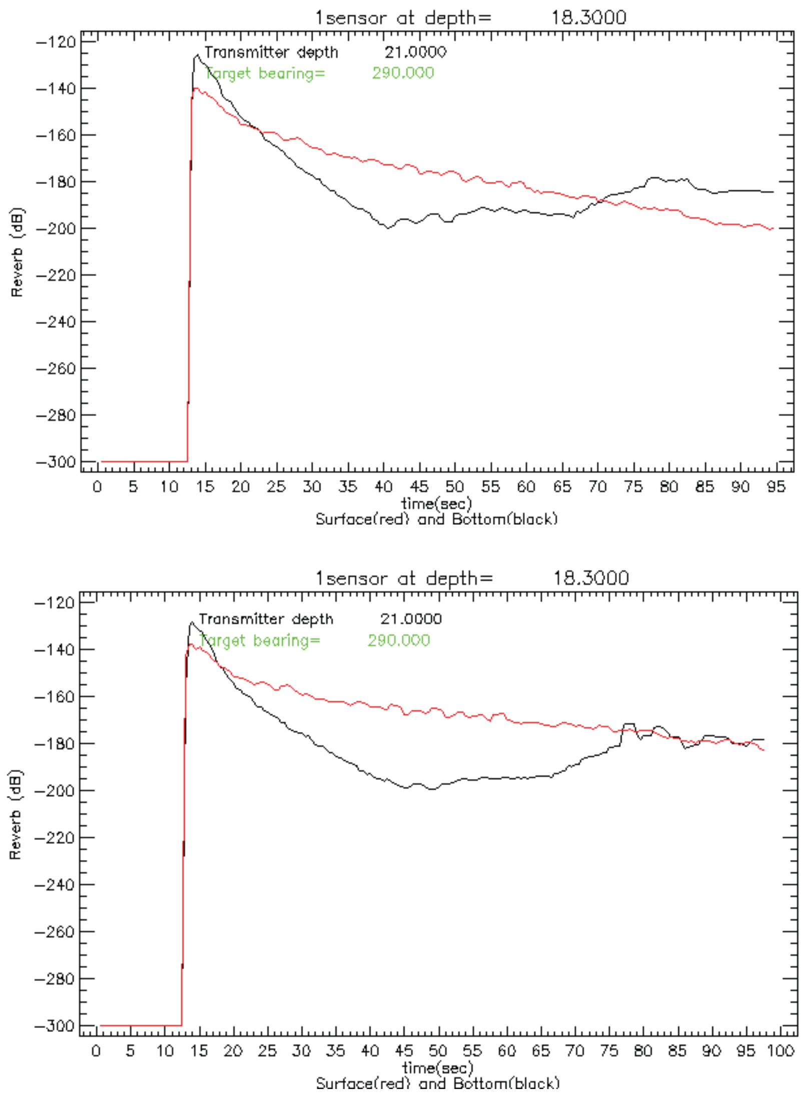


*Figure 23. Plots of surface and bottom reverberation from reverb.txt (note: source level is not applied). The top plot used model choices {B,CH,OM} while the bottom plot used {E,OE,EC .*

latter choice of models produces a similar contribution from the knoll bathymetry of the bottom reverberation, while the surface reverberation is uniformly higher.

### 3.2.3 Signal.txt

This file contains the target echo time series for each sensor's range, target depth and target bearing. The source level and target strength are not included in this output. At present there are no graphics that make use of this file's output. It is provided to be an input along with the reverb.txt file for signal excess calculations that may wish to be computed at a later date or in another language. The source level and target strength are not applied to this file to make the signal excess computation more flexible. The write statements are in subroutine Writesignal in the file writeoutput_active.f90.

The output begins with the run title from active_general.inp. Next, looping over each bearing of interest and asset, the number of time points, the number of ranges, the number of target depths, the sensor number and target bearing are listed, followed by the target echo time series in dB.

### 3.2.4 SE.txt

This file contains the signal excess computed using the Fortran file SE_active.f90 which is included in this program. The SE computation begins by working in intensity units. It forms the signal term by multiplying the target echo time series by the source level and target strength. It forms the interference level power sum by adding the surface and bottom reverberation intensity time series, multiplying by the source level and adding the noise intensity. From these two terms the signal-to-interference ratio time series is formed. Taking the maximum signal to interference ratio from the time series, the program converts this to dB and subtracts the dB values of detection threshold and system loss to form the signal excess.

The SE is listed in the file SE.txt. The first line contains the title. The second line lists the frequency and transmitter depth. The next line contains the number of ranges, the number of target depths, the number of target bearings (those that were identified in radial_bathy.inp), and the number of receiving sensors. The range array is listed next. Following this, the output loops over sensor number and writes the sensor number and sensor depth. Inside this sensor loop, the output loops over the target bearing and writes its value in degrees. Lastly, the output loops over target depth and writes each depth in meters, followed by the SE(dB) vs range array. The write statements are in subroutine WriteSE in the file writeoutput_active.f90.

```
'Test run with 1 sensor, 4 bearings'

Frequency=        1200  transmitter depth (m)=   21.00000
93          2          1          1
  0.7526418      1.505284      2.257925      3.010567      3.763209
   4.515851      5.268493      6.021134      6.773776      7.526418
```
*(>> Continued on next page)*

*(>> Continued from previous page)*

```
  8.279059      9.031701      9.784343      10.53699      11.28963
 12.04227      12.79491      13.54755      14.30019      15.05284
 15.80548      16.55812      17.31076      18.06340      18.81605
 . . .
        1   18.30000    sensor# and depth
 290.0000    target bearing (deg)
 21.00000    target depth (m)
-7.8558557E-02  -1.335800      2.1457110E-02  0.7401745      2.898602
 8.519861      6.849106      4.837643      9.012436      11.73837
 9.486148      10.76469      12.86479      7.269334      9.710820
 13.33085      11.35298      7.134454      9.388466      10.77402
 10.35002      10.18052      8.118537      9.292543      10.42265
 . . .
   80.00000    target depth (m)
-35.57817     -37.65511     -37.85839     -39.54664     -36.73313
-37.13431     -38.05925     -39.03345     -40.48875     -42.64893
-42.27207     -42.21347     -43.35333     -43.96018     -45.02712
-43.42149     -45.56827     -51.34261     -48.63745     -47.65379
-49.21247     -46.95230     -51.52979     -49.59116     -50.81988
   . . .
```

*Figure 24. Example of portion of SE.txt output.*

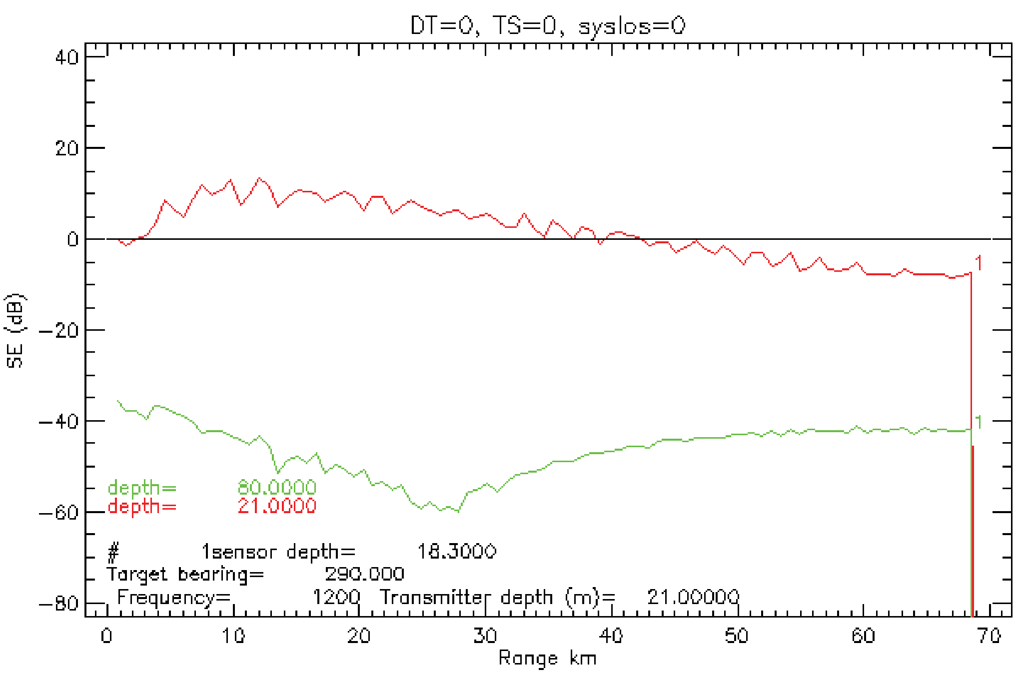


*Figure 25. Example of signal excess plot, with DT=TS=syslos=0. Two curves are for two different target depths.*

### 3.2.5 TL.txt

This file contains the transmission loss as a function of range for the transmitter location to the target depth, and for the sensor location to the target depth along the target bearing. The beam patterns of the transmitter and sensor are included in the calculation. Note that the maximum range is larger than the value specified in the active_general.inp file. Inside the frontend_active_v4.f90 file, the maximum range is automatically increased by the length of the baseline between sensor and transmitter plus a pulse length. The output file consists of the title of the run, followed by a loop over sensor number and target bearing in which the next two lines contain the number of ranges, the number of target depths, then the sensor number, sensor depth, transmitter depth and the target bearing. Following this is the range array. Then in a loop over target depth, the depth is listed, followed by the transmitter transmission loss array vs range, and lastly the sensor transmission loss array vs range. The write statements are in subroutine WriteTL in the file writeoutput_active.f90.

```
 ‘Test run with 1 sensor, 4 bearings’

          93          2
         1   18.30000       21.00000       290.0000
  0.7526418      1.505284      2.257925      3.010567      3.763209
   4.515851      5.268493      6.021134      6.773776      7.526418
   8.279059      9.031701      9.784343      10.53699      11.28963
      . . .
68.49041      69.24305      69.99569
   21.00000
   55.33667      59.86486      59.65900      60.15039      65.99412
   63.31894      63.20134      67.48380      68.84501      65.57792
   67.26232      71.43450      67.33181      67.46899      70.77728
          . . .
```

*Figure 26. Example of portion of TL.txt output*

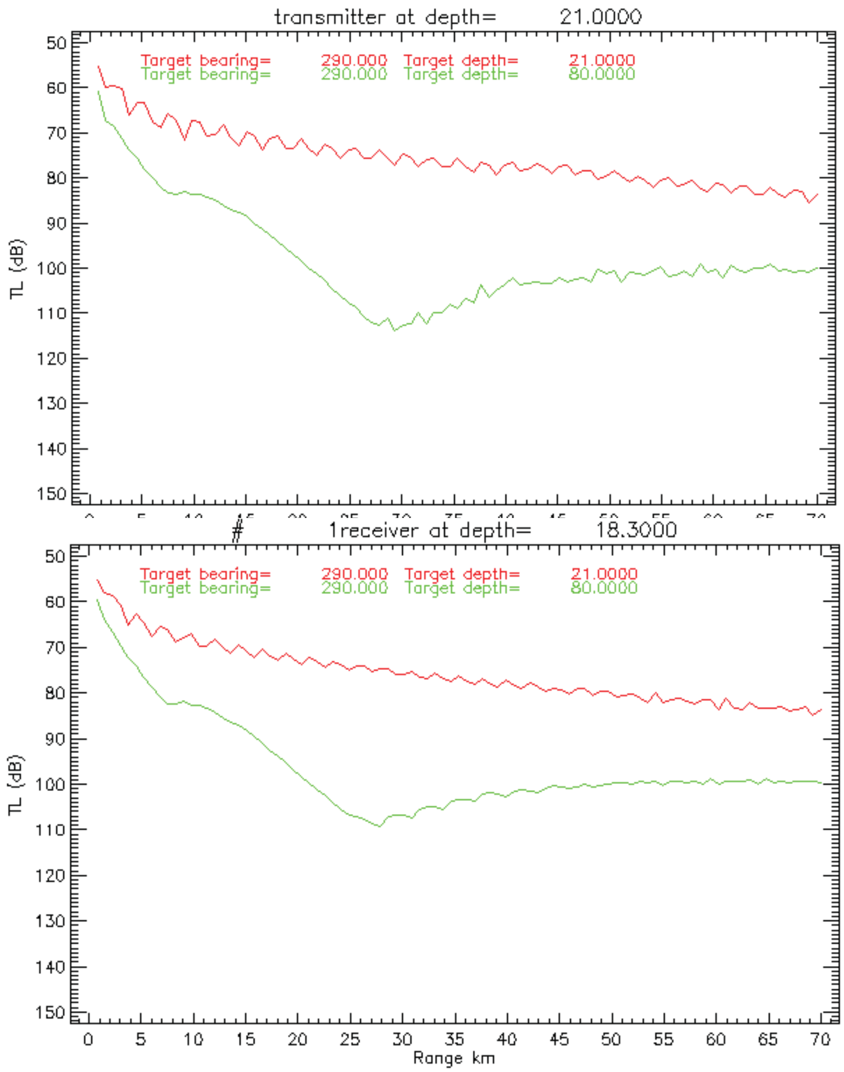


*Figure 27. Example of TL plot from the TL.txt file. Left: transmission loss from transmitter to target, including transmitter beam pattern for two target depths. Right: transmission loss from receiver to target, including receiver beam pattern for two target depths.*

### 3.2.6 Bellhop_active.log

This file contains a log of the runtime statements generated in any run. Some inputs are echoed, and any warnings or errors are listed here as generated by the Bellhop code.

## 3.3 Plot Routines

Some IDL plot routines have been prepared to provide simple graphic representations of the active program outputs. These should be freely altered to suit the users' data and output requirements.

Read_arrivals_plot_angleortimes.pro – reads SALT table output in arrival.txt and plots arrival angle vs range as shown in Figure 21.

Read_reverb_plot_reverb.pro – reads reverb.txt and plots surface and bottom reverberation as shown in Figure 23.

Read_SE_plot_SE.pro – reads SE.txt and plots signal excess vs range as shown in Figure 25.

Read_TL_plot_TL_active.pro – reads TL.txt from the active program output and plots receiver to target transmission loss and transmitter to target transmission loss in two windows as a function of range as shown in Figure 27.

# 4. Boundary loss

To help understand the transmission loss predictions a separate program called boundaryloss is provided that will compute the bottom and surface losses and the surface and bottom scattering strengths. There are two versions of this program for active and passive.

## 4.1 Boundaryloss_passive

Boundaryloss_Passive.f90 – main program
- READIN_V4- reads runinput.inp
- READSVP_v4 – reads speed.inp
- READBOTLOSS_v4 – reads bottomloss.inp
- BOTT_NEW – function to compute MGS bottom loss from province numbers
- SURF_NEW – function to compute surface reflection coefficients using Bechmann-Spezzichino formulas
- LFSOPN – computes surface loss per bounce for open ocean using Modified Echart
- TMP_SSP – function to convert temperature to speed using Leroy's equation
- twolayerRefl – computes reflection coefficient from two fluid layers of sediment
- CRCIS – function to convert real wave speed and attenuation to a single complex wave speed
- ERROUT – outputs error messages

Boundarylossmod_passive.f90 – module with array declarations

### 4.1.1 Input files

#### 4.1.1.1 Runinput_v4.inp

This is the same file that BellhopDRDC_ray_TL_v4 uses, described in section 2.1.1. The boundary loss algorithm uses the run title, frequency, and wind speed from this file. For this example, the frequency was increased to 3000Hz and the wind speed was set to 20 kts.

#### 4.1.1.2 Speed.inp

This is the same file that BellhopDRDC_ray_TL_v4 uses, described in section 2.1.2. The boundary loss algorithm uses the sound speed from the surface and the bottom of the first profile in this file.

#### 4.1.1.3 Bottomloss.inp

This is the same file that BellhopDRDC_ray_TL_v4 uses, described in section 2.1.3. The boundary loss algorithm uses all lines in this file. For this example, the acoustic bottom was described as a simple half-space with speed =1453 m/s, density = 1.41 g/cc and attenuation = 0.038 dB/mkHz.

### 4.1.2 Output files

#### 4.1.2.1 Botloss_passive.txt

This file contains the bottom loss and grazing angles for each bottom description in the input file. The structure of this file begins with the run title, the frequency, the number of range dependent bottom's being specified, and a string showing the type of bottom being computed, such as the two layer model, the MGS model or a user input table. Next, the number of grazing angles is listed, followed by an array of grazing angles then an array of dB losses. These last three are repeated for each bottom specified. An example of this file is shown in Figure 28 generated using a single half-space. An example of the plot of these data is shown in Figure 29.

```
BELLHOP- Emerald basin toward Sambro Bank

   3000.000
          1
 two-layer fluid reflection bottom loss
         91
          0        1        2        3        4        5
          6        7        8        9       10       11
         12       13       14       15       16       17
              . . .
         90
1.1498389E-02   1.870405     3.750788     5.653327     7.592482
   9.586963     11.66344     13.85982     16.23368     18.87825
   21.95951     25.81959     31.36469     42.31104     37.66504
   31.09075     27.73483     25.58814     24.06392     22.91468
   22.01334     21.28639     20.68758     20.18612     19.76052
   19.39529     19.07889     18.80259     18.55959     18.34455
              . . .
```

*Figure 28. Portion of a Botloss_passive.txt file.*

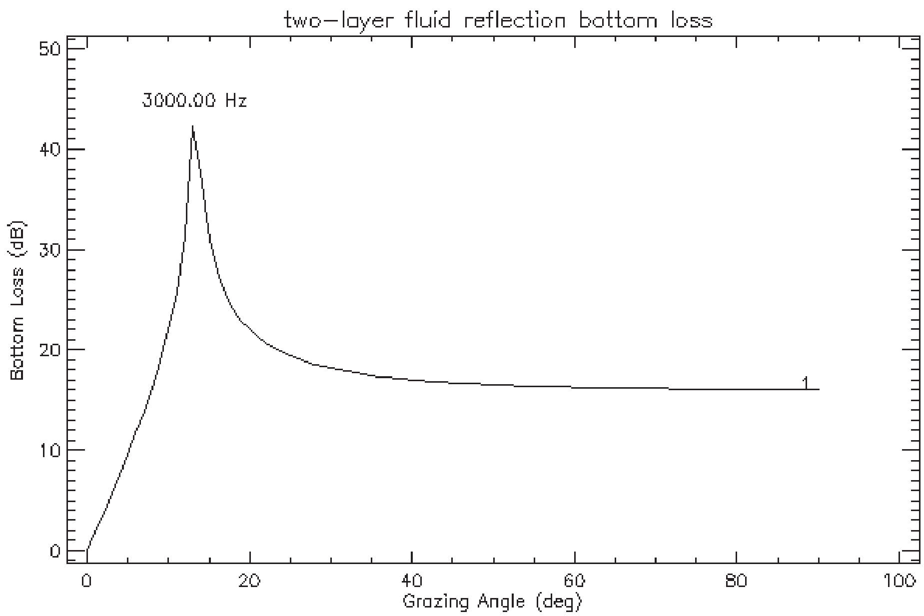


*Figure 29. Plot of botloss_passive.txt showing the single half-space region in bottomloss.inp for the acoustic bottom descriptions.*

#### 4.1.2.2 Surfloss_passive.txt

This file contains the surface loss and grazing angles from all available loss algorithms. The structure of this file begins with the run title, the frequency, a string showing the surface loss algorithm, the number of points in the table, then an array of grazing angles and an array of losses in dB. This format is repeated for the second surface loss algorithm. An example of this file is shown in figure Figure 30.  A plot of these data is shown in Figure 31.

```
BELLHOP- Emerald basin toward Sambro Bank

3000.000      20.00000
 modified Eckart- LFSOPN open ocean low frequency surface loss
         0          1          2          3          4          5
         6          7          8          9         10         11
        12         13         14         15         16         17
              . . .
        90
0.0000000E+00  0.2538207      1.032815      2.244289      3.774943
   5.455753      7.154128      8.735363      10.13659      11.00000
   11.00000      11.00000      11.00000      11.00000      11.00000
              . . .
  11.00000
 Beckman Spezzichino surface loss
        91          2
         0          1          2          3          4          5
         6          7          8          9         10         11
        12         13         14         15         16         17
              . . .
     90
   7.882047      7.920110      7.958499      7.997206      8.036225
   8.075550      8.115174      8.155088      8.195287      8.235761
   8.276503      8.317503      8.358752      8.400243      8.441965
              . . .
```

*Figure 30. Example portion of surfloss_passive.txt.*

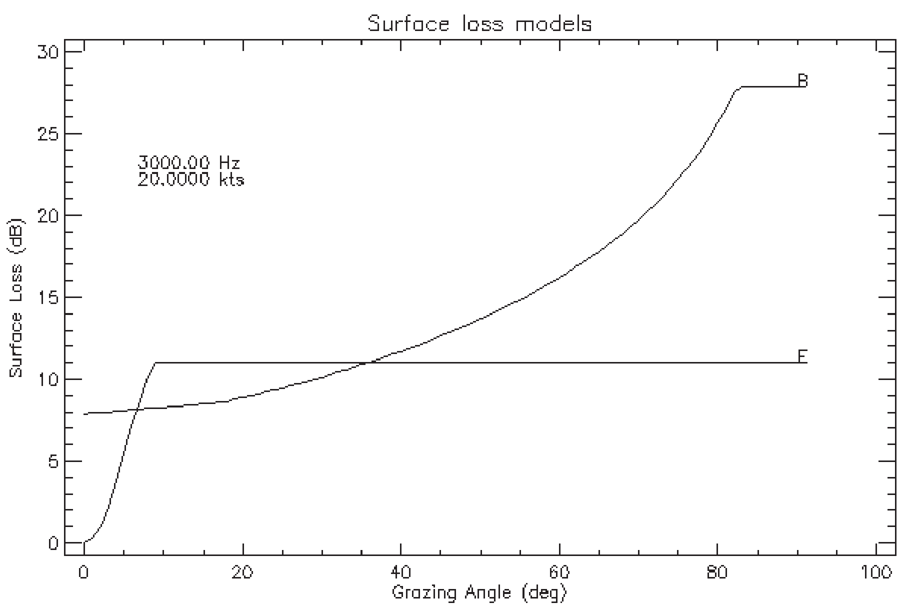


*Figure 31. Plot of surfloss_passive.txt for both Beckman Spezzichino (B) and modified Eckart (E) generated for 3 kHz and 20 kt wind speed.*

### 4.1.3 Plot routine

An IDL plot routine has been prepared to provide a simple graphic representation of the boundaryloss_passive outputs. This should be freely altered to suit the users' data and output requirements.

> Read_loss_plot_boundaryloss_passive.pro - Routine to read the output in Botloss_passive.txt and Surfloss_passive.txt and produce two loss figures.

## 4.2 Boundaryloss_active

Boundaryloss_active.f90 – main program
- READINPUT_ACTIVE - reads active input files
- READBOTLOSS_S – reads bottomloss.inp
- TMP_SSP – function to convert temperature to speed using Leroy's equation
- twolayerRefl – computes reflection coefficient from two fluid layers of sediment
- CRCIS – function to convert real wave speed and attenuation to a single complex wave speed
- ERROUT – outputs error messages

datamod_active_v4 .f90– same as file in BellhopDRDC_active_v4
refcomod_active_v4.f90 - same as file in BellhopDRDC_active_v4
scatstrength_v4.f90 - same as file in BellhopDRDC_active_v4
SEmod_active .f90- same as file in BellhopDRDC_active_v4
Sspmod_active_v4.f90 - same as file in BellhopDRDC_active_v4

### 4.2.1 Input files

#### 4.2.1.1 Active_general.inp

This is the same file that BellhopDRDC_active_v4 uses, described in section 3.1.1. The boundary loss algorithm uses the run title, number of sensors, frequency, wind speed, and 'EC' parameters μ, υ, and σ from this file.

#### 4.2.1.2 Radial_ssp.inp

This is the same file that BellhopDRDC_active_v4 uses, described in section 3.1.2. The boundary loss algorithm uses the sound speed at the surface in the first profile of the first sensor and radial for the Eckart surface loss calculation. It also uses the sound speed of the bottom in the first profile at each sensor and each radial for the acoustic bottom loss calculations.

#### 4.2.1.3 Radial_bottomloss.inp

This is the same file that BellhopDRDC_active_v4 uses, described in section 3.1.3. The boundary loss algorithm uses all of this file.

#### 4.2.1.4 Radial_bathy.inp

This is the same file that BellhopDRDC_active_v4 uses, described in section 3.1.4. The boundary loss algorithm uses the number of radials from this file.

### 4.2.2 Output files

#### 4.2.2.1 Botloss_active.txt

This file contains the bottom loss and grazing angles for each bottom description in the input file. The structure of this file begins with the run title, the frequency, the number of bearings and assets, a string showing the type of bottom being computed, such as the two layer model, the MGS model or a user input table, and the number of range dependent bottom's being specified. Next the number of grazing angles, index of range dependence, bearing and asset are listed, followed by an array of grazing angles then an array of dB losses. These last three are repeated for each bottom specified along the range, then the output loops over the number of bearings for each asset. An example of this file is shown in Figure 32. An example of the plot of these data using the IDL program read_loss_plot_boundaryloss_active.pro is shown in Figure 33 for bearing #4 and asset#1. The different curves are labelled according to their order in the radial_bottomloss.inp file.

```
Test run with 1 sensor, 4 bearings

  1200.000
       4         2
 two-layer fluid reflection bottom loss
       4
      91         1         1         1
       0         1         2         3         4         5
       6         7         8         9        10        11
      12        13        14        15        16        17
          . . .
      84        85        86        87        88        89
      90
5.8247424E-03  0.5140966     0.9780043     1.415306      1.952464
  2.857528      4.540282      6.287792      5.386145      3.989525
  3.439673      3.525668      3.814885      3.853908      3.888648
  4.145320      4.254133      4.699588      6.281219      8.995653
           . . .
```

*Figure 32. Example portion of Botloss_active.txt.*

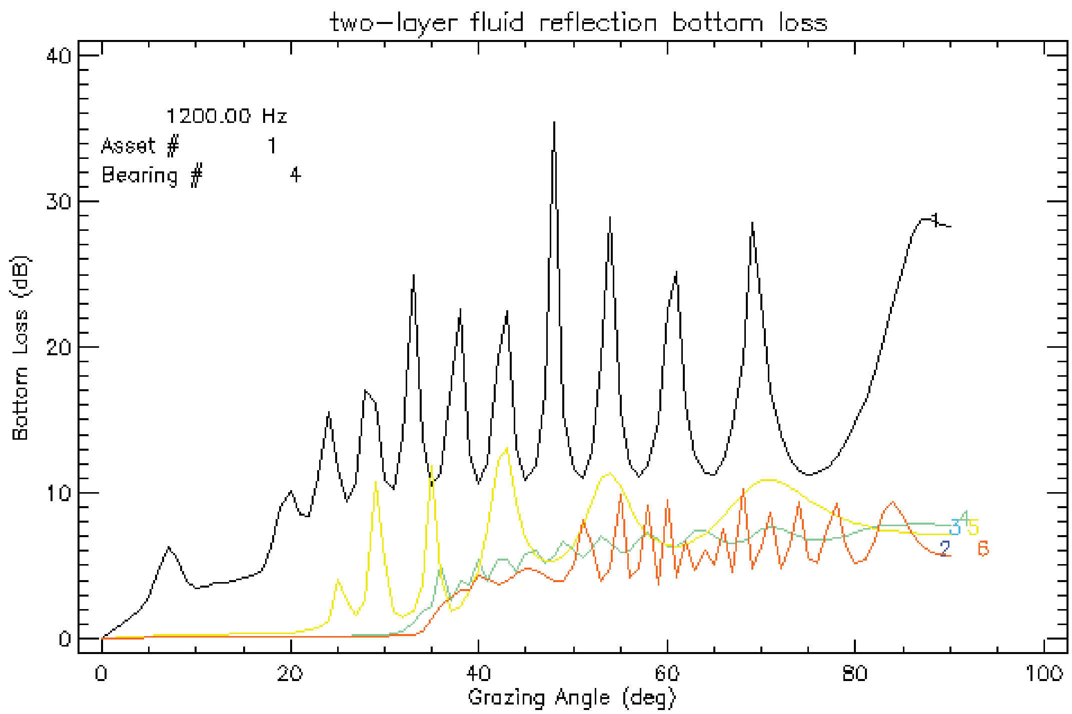


*Figure 33. Range dependent bottom loss along bearing 4.*

#### 4.2.2.2 Surfloss_active.txt

This file contains the surface loss and grazing angles from the two available loss algorithms Eckart ('E') and Beckman Spezzichino ('B'). The structure of this file begins with the run title, the frequency and wind speed, a string showing the surface loss algorithm, the number of points in the table, then an array of grazing angles and an array of losses in dB. This format is repeated for the second surface loss algorithm. An example of this file is shown in Figure 34. A plot of these data using the IDL program read_loss_plot_boundaryloss_active.pro is shown in Figure 35. This output plot can be compared with the passive case output in Figure 31, which used a frequency of 3 kHz and a wind speed of 20 kts.

```
Test run with 1 sensor, 4 bearings

  1200.000       10.00000
 modified Eckart- LFSOPN open ocean low frequency surface loss
        91          1
         0          1          2          3          4          5
         6          7          8          9         10         11
        12         13         14         15         16         17
           . . .
        84         85         86         87         88         89
        90
0.0000000E+00  2.5019804E-03  1.0008620E-02  2.2524085E-02  4.0054973E-02
 6.2609777E-02  9.0200447E-02  0.1228408      0.1605476      0.2033409
  0.2512421      0.3042762      0.3624716      0.4258584      0.4944697
  0.5683423      0.6475151      0.7320306      0.8219351      0.9075918
              . . .
```

*Figure 34. Example portion of Surfloss_active.txt.*

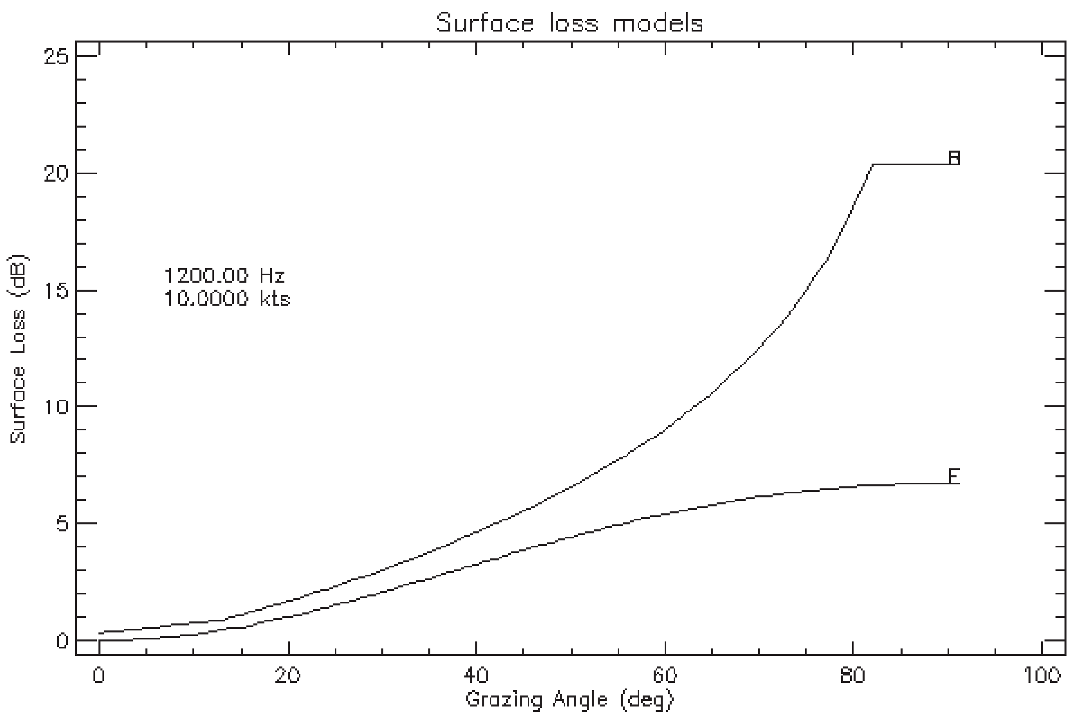


*Figure 35. Plot of surfloss._active.txt for both Beckman Spezzichino (B) and modified Eckart (E).*

#### 4.2.2.3 Surfscat.txt

This file contains the surface monostatic backscattering strength and grazing angles from the two available scattering strength algorithms, Ogden-Erskine ('OE') and Chapman-Harris ('CH'). The structure of this file begins with the run title, the frequency and wind speed, a string showing the surface scattering strength algorithm, the number of points in the table, then an array of grazing angles and an array of losses in dB. This format is repeated for the second surface loss algorithm. An example of this file is shown in Figure 36. A plot of these data using the IDL program read_loss_plot_boundaryloss_active.pro is shown in Figure 37.

```
Test run with 1 sensor, 4 bearings

  1200.000      10.00000
 Ogden-Erskine monostatic surface backscattering strength
        90        1
         1        2        3        4        5        6
         7        8        9       10       11       12
        13       14       15       16       17       18
             . . .
 85       86       87       88       89       90
  93.20633      82.62815      76.43996      72.04900      68.64273
  65.85920      63.50536      61.46596      59.66667      58.05673
  56.59995      55.26957      54.04532      52.91141      51.85532
  50.86697      49.93810      49.06189      48.23261      47.44540
             . . .
 25.76276      25.59860      25.43632      25.27587      25.11720
 Chapman-Harris monostatic surface backscattering strength
        90        2
         1        2        3        4        5        6
         7        8        9       10       11       12
        13       14       15       16       17       18
             . . .
```

*Figure 36. Example portion of Surfscat.txt.*

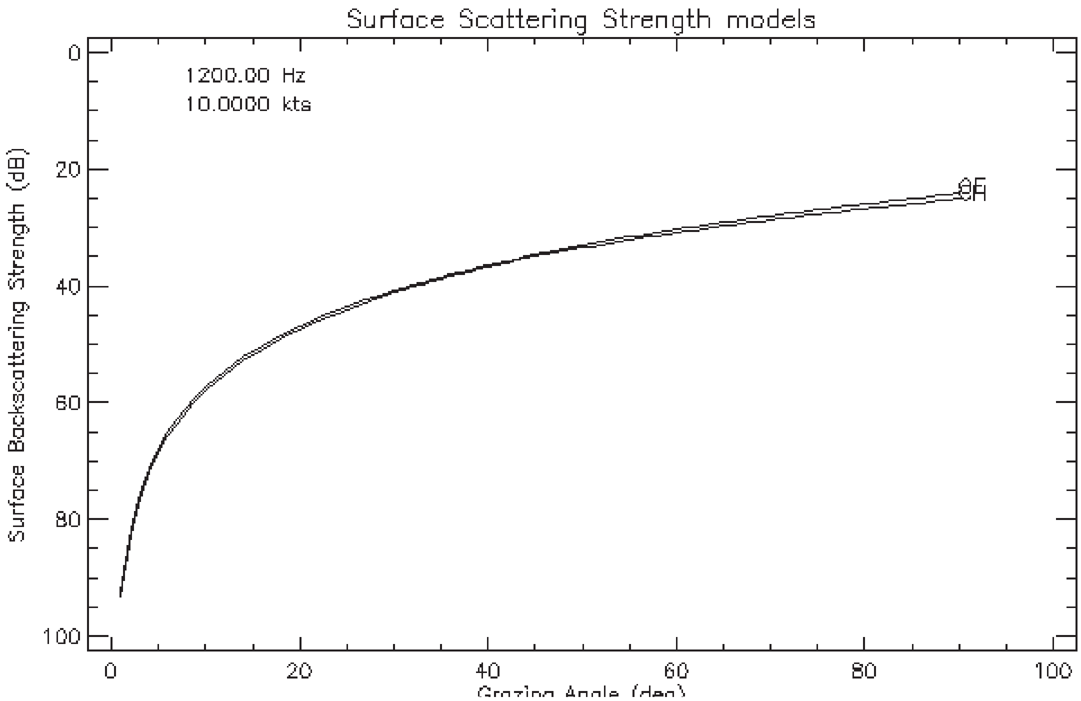


*Figure 37. Plot of surfscat.txt showing surface scattering strength models Ogden-Erskine (OE) and Chapman-Harris (CH).*

#### 4.2.2.4 Botscat.txt

This file contains the bottom monostatic backscattering strength and grazing angles from the three available loss algorithms, Ellis-Crowe ('EC'), Lambert's Rule ('LB'), and the Omni Rule ('OM'). The structure of this file begins with the run title, the frequency, a string showing the bottom scattering strength algorithm, the number of points in the table, then an array of grazing angles and an array of losses in dB. This format is repeated for the other two bottom scattering strength algorithms. An example of this file is shown in Figure 38. A plot of these data using the IDL program read_loss_plot_boundaryloss_active.pro is shown in Figure 39.

```
Test run with 1 sensor, 4 bearings

  1200.000
 Ellis Crowe monostatic bottom backscattering strength
        90         1
         1         2         3         4         5         6
         7         8         9        10        11        12
        13        14        15        16        17        18
             . . .
        85        86        87        88        89        90
62.16290    56.14362    52.62400    50.12831    48.19408
46.61531    45.28211    44.12889    43.11335    42.20660
41.38802    40.64242    39.95824    39.32650    38.74007
             . . .
5.048865    5.015474    4.991682    4.977431    4.972684
 Lambert Rule monostatic bottom backscattering strength
        90         2
         1         2         3         4         5         6
         7         8         9        10        11        12
        13        14        15        16        17        18
             . . .
        85        86        87        88        89        90
 62.16290    56.14362    52.62400    50.12831    48.19408
 46.61531    45.28211    44.12889    43.11335    42.20660
 41.38802    40.64242    39.95824    39.32650    38.74007
             . . .
  27.02118    27.01191    27.00529    27.00132    27.00000
 Omni monostatic bottom backscattering strength
        90         3
         1         2         3         4         5         6
         7         8         9        10        11        12
        13        14        15        16        17        18
             . . .
        85        86        87        88        89        90
  44.58145    41.57181    39.81200    38.56416    37.59704
  36.80766    36.14106    35.56445    35.05667    34.60330
  34.19401    33.82121    33.47912    33.16325    32.87004
  32.59662    32.34065    32.10018    31.87358    31.65948
             . . .
```

*Figure 38. Example portion of botscat.txt.*

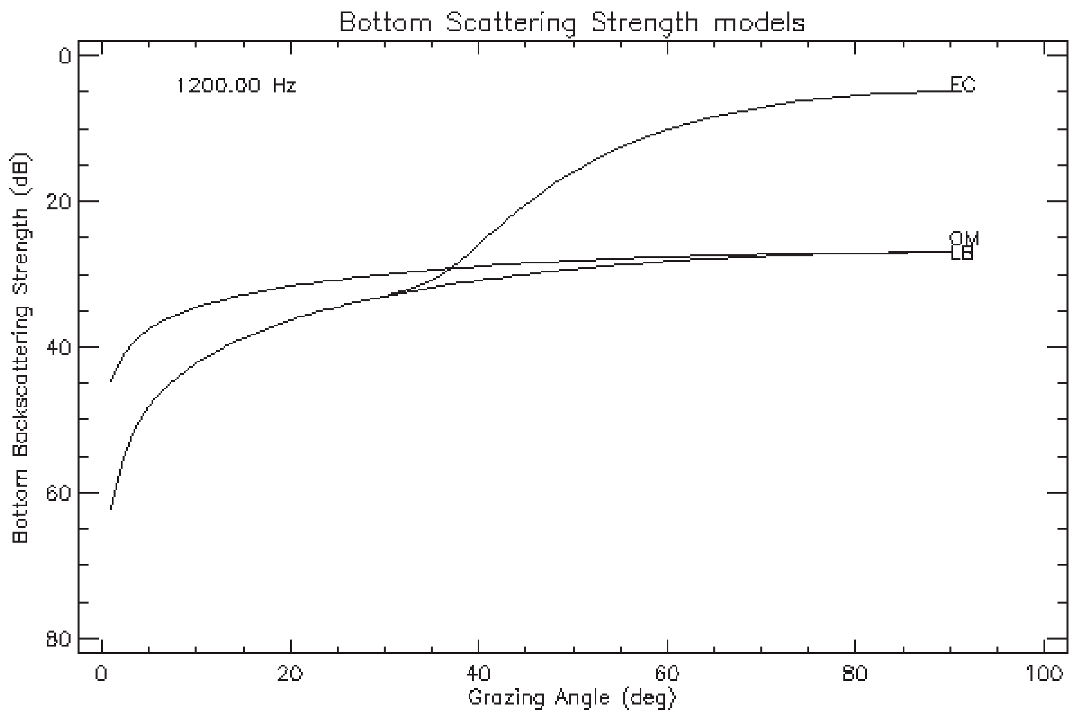


*Figure 39. Plot of botscat.txt showing Ellis-Crowe (EC), Lambert (LB) and omni (OM) bottom scattering strength models.*

### 4.2.3 Plot routine

An IDL plot routine has been prepared to provide a simple graphic representation of the boundaryloss_active outputs. This should be freely altered to suit the users' data and output requirements.

Read_loss_plot_boundaryloss_active.pro - Routine to read the output in Botloss_active.txt, Surfloss_active.txt, surfscat.txt and botscat.txt and produce four loss figures.

# Distribution list

Document No.: DRDC Atlantic CR 2010-134

<u>LIST PART 1: Internal Distribution by Centre:</u>

3 DRDC Atlantic Library (1 hard copy, 2 CDs)
1 Scientific Authority
1 Marcel Lefrançois
1 Jim Theriault
1 Garfield Mellema
1 John Osler
1 Paul Hines
1 Sean Pecknold
1 Dale Ellis
1 Mark Trevorrow
1 Brian Maranda
1 Dan Hutt, NL, TTCP MAR TP9

14 TOTAL LIST PART 1

<u>LIST PART 2: External Distribution by DRDKIM</u>

1 Library and Archives Canada, Attn: Military Archivist, Government Records Branch
1 (CD) NDHQ/DRDKIM
1 McCammon Acoustical Consulting, 475 Baseline Road, Waterville NS B0P 1V0
1 DMRS 5-3,
HDHQ
101 Colonel By Drive
Ottawa ON K1A 0K2
1 DSTM 7,
DRDC, 305 Rideau St.,
Ottawa, ON K1A 0K2
2 SSO METOC
MARLANT HQ
PO Box 99000, Stn Forces
Halifax, NS B3K 5X5
1 CFMWC / DUWB
PO Box 99000, Stn Forces
Halifax, NS B3K 5X5

8 TOTAL LIST PART 2

**<u>22</u> <u>TOTAL COPIES REQUIRED</u>**

This page intentionally left blank.

## DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

| | |
|---|---|
| 1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)<br><br>McCammon Acoustical Consulting<br>475 Baseline Road<br>Waterville NS B0P 1V0 | 2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.) |

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C, R or U) in parentheses after the title.)

Users Guide to BellhopDRDC_V4: Active and Passive versions

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

McCammon, D.F.

| | | |
|---|---|---|
| 5. DATE OF PUBLICATION (Month and year of publication of document.)<br><br>October 2010 | 6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)<br><br>62 | 6b. NO. OF REFS (Total cited in document.)<br><br>0 |

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Atlantic
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7

| | |
|---|---|
| 9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)<br><br>11ca | 9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)<br><br>W7707-098272 |
| 10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) | 10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)<br><br>DRDC Atlantic CR 2010-134 |

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

( X ) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The acoustic prediction model called Bellhop continues to be enhanced to more closely fit the requirements of DRDC Atlantic's Environment Modeling Manager (EMM). This version 4 contains both passive and active algorithms. In this version, linear range interpolation of the SSP and curvilinear interpolation of the bathymetry are added as input choices, and an additional output of the sampled SSP is provided. The major differences between BellhopDRDC and the web version dated May 2008 lie in the input data and file formats that have been altered to satisfy the requirements of the controlling programs within the Environment Modeling Manager. This document provides a users guide to the running of the active and passive versions of the BellhopDRDC_v4 program and the boundary loss program, and describes some plotting routines available for viewing the prediction results.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

acoustic prediction; active sonar; reverberation; clutter

This page intentionally left blank.

## Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

## R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca