



Resolving Bellhop Issues: Negative Range Arrivals and Volume Attenuation

Chung T C Fang (1), Anthony C Zander (1), Eyad R Hassan (1), Stephen Franklin (2)
and William S P Robertson (1)

(1) *School of Electrical and Mechanical Engineering, The University of Adelaide, SA, Australia*

(2) *Saab Australia Pty. Ltd, SA, Australia*

ABSTRACT

Bellhop is a widely used underwater acoustic ray-tracing model, but its 1990s Fortran core imposes legacy constraints including half-space simulations and single-frequency absorption. We identify and correct a key numerical issue in the *ArrMod.f90* module that prevented negative-range propagation, enabling single-run full-space simulations without separate half space runs. In addition, we extended Bellhop's single frequency absorption to broadband application by post processing along individual ray trajectories at lower computational cost. Validation against analytical solution shows agreement in both positive and negative ranges and accurate frequency-dependent attenuation. These improvements simplify Bellhop workflows and provide a more generalised volume attenuation solution.

1 INTRODUCTION

The Bellhop ray tracing platform has seen wide application in the sphere of underwater acoustics. Its diverse use cases include event reconstruction for NATO anti-submarine exercises where daily event data is simulated to identify detection opportunities for crew debriefing and enhanced training (Strode et al., 2023). In wireless underwater communication, Casari et al. (2014) described Bellhop as a component in a pipeline of oceanographic and network simulation tools to predict performance, minimising deployment effort. In research, Bellhop calculated sound field was used to demonstrate enhanced probability of detection of submarines with multiple collaborative agents (Lan et al., 2024). References to the application of Bellhop are documented widely over several decades.

The facility to simulate underwater sound propagation using Bellhop has been conveniently brought into the Python environment through wrapper libraries such as ARLpy, expediently interfacing with popular open-source tools for machine learning and data analysis. Yet, for the most part, the libraries only act as data conduits to the core Bellhop legacy code. A focus of this work is the removal of Bellhop's long-standing half-space restriction to enable full-space ray arrival simulation. The standard Bellhop imposes positive receiver ranges only for some run types (e.g. Arrivals 'A') but not others (e.g transmission loss 'C'), this inconsistency causes confusion and effectively requires two separate runs to model propagation on both sides of a source. Perhaps developed with hardware considerations for its time, the limitation complicates workflows and introduces potential risk of errors. By examining the open-source Fortran code we identified the specific calculation that produced undefined amplitudes for negative ranges and implemented a simple but robust correction. This modification allows a single simulation to capture propagation in all directions around the source, eliminating redundant runs and ensuring consistent amplitude behaviour across the zero-range axis.

The second part of this paper highlights a potential issue with some simulations which predict geometric transmission loss through an ideal media but may not be accurately accounting for absorptive losses of specific media. Absorptive losses resulting from fine scale interactions between particles may be the dominant mechanism of

signal attenuation depending on frequency and distance. Media such as seawater have well documented absorption, including both measurements at specific locations as well as parameterised approximations. In a complex propagation environment where the resultant field is a combination of many refracted and reflected signal paths, the application of volume attenuation is well suited to raytracing because it explicitly calculates the geometric path of energy between source and receiver. However, the strict application of frequency dependent absorptive losses to a ray-traced result can be computationally expensive because of numerous rays and parameters governing the absorption such as temperature and pressure, continuously vary along each ray path. This paper describes an efficient process for combined ray-trace propagation with absorption across a two-dimensional acoustic field simulation of an ocean environment for an arbitrary input signal.

2 Bellhop Full Space Simulation

Bellhop's half-space limitation is documented in the User Manual (Porter, 2011) as well as by expert users such as M. Chitre, the developer of ARLpy Python interface who noted the behaviour in the library documentation. The trait is somewhat entrenched with modern iterations such as BellhopCuda (Pisha et al. 2023) replicating the limitation.

2.1 Error Reproduction

If we ignore the stated constraint and set up a simulation with an omnidirectional point source and receivers with both positive and negative range values, execute Bellhop with the 'A' run option for ray arrivals, then output data for all negative range receiver locations will report an "arrival_amplitude" value of NaN (Figure 1). This error is not introduced from interface libraries such as ARLpy but originates directly from the Bellhop output file.

tx_depth_ndx	rx_depth_ndx	rx_range_ndx	tx_depth	rx_depth	rx_range	arrival_number	arrival_amplitude	time_of_arrival	complex_time_of_arrival	angle_of_departure	angle_of_arrival	surface_bounces	bottom_bounces	
1068	0	0	249	50.0	25.0	-0.8	0	NaN+0.000000j	0.109115	0.109115-0.000001j	-67.425003	-157.425003	1	1
1069	0	0	249	50.0	25.0	-0.8	1	NaN+0.000000j	0.086938	0.086938-0.000001j	14.983334	104.983330	1	1
1070	0	0	249	50.0	25.0	-0.8	2	NaN+0.000000j	0.087614	0.087614-0.000001j	22.475000	112.474998	1	1
1071	0	0	249	50.0	25.0	-0.8	3	NaN+0.000000j	0.058668	0.058668-0.000000j	29.966667	-119.966667	0	1
1072	0	0	249	50.0	25.0	-0.8	4	NaN+0.000000j	0.058721	0.058721-0.000000j	37.458332	-127.458336	0	1
1073	0	0	251	50.0	25.0	0.8	0	0.018304-0.000295j	0.016160	0.016160-0.000000j	-82.408333	-82.408333	0	0
1074	0	0	251	50.0	25.0	0.8	1	-0.001101-0.000547j	0.108156	0.108156-0.000001j	-67.425003	-157.425003	1	1
1075	0	0	251	50.0	25.0	0.8	2	-0.000036-0.000589j	0.086669	0.086669-0.000001j	14.983334	104.983330	1	1
1076	0	0	251	50.0	25.0	0.8	3	0.004015+0.001191j	0.087217	0.087217-0.000001j	22.475000	112.474998	1	1
1077	0	0	251	50.0	25.0	0.8	4	-0.000170+0.003450j	0.058149	0.058149-0.000000j	29.966667	-119.966667	0	1

Figure 1: Arrival data output from Bellhop simulation with negative range receivers.

2.2 Error Source and Fix

Fortunately, the authors of Bellhop (HLS Research) have made the Fortran source code available with the Acoustics Toolbox download (HLS, 2024). Upon inspection, the error can be traced to a single statement in the "ArrMod.f90" module. Specifically, at line 109, a scaling factor is calculated for cylindrical spreading. The statement is written as:

```
factor = 1. / SQRT( r( ir ) ) ! line 109
```

where "r" is an array of type "real" (float64) for values of the receiver range. When range is negative, this causes the square root of a negative "real" type which is not supported and returns NaN.

Since cylindrical spreading is symmetrical down the zero axis where the source is located, "r" is simply the distance to the source and sign can be ignored. Thus, the error can be fixed simply by forcing the value of "r" to positive before the square root with the following modification.

```
factor = 1. / SQRT( ABS( r( ir ) ) ) ! line 109
```

Applying this fix eliminates the NaN value error from Bellhop's "arrival_amplitude" output.

2.3 Fix Validation

To confirm the new output for negative ranges is correct we take the analytical case of a Lloyd's Mirror (Figure 2). The exact solution for amplitude in this scenario with two interfering rays is:

$$p(r_h, z_h) = \frac{e^{ikR_1}}{R_1} - \frac{e^{ikR_2}}{R_2} \quad (1)$$

where

$$R_1 = \sqrt{r_h^2 + (z_h - z_s)^2} \quad (2)$$

$$R_2 = \sqrt{r_h^2 + (z_h + z_s)^2} \quad (3)$$

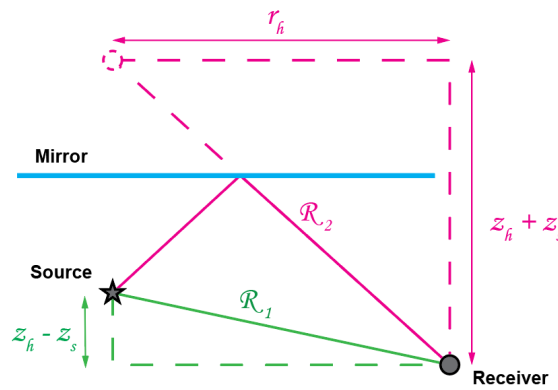


Figure 2: Llyod's Mirror where the amplitude at the receiver is the interference between a direct ray and a reflected ray.

A Llyod's Mirror scenario was constructed in Bellhop with the source at 15 m depth and a row of receivers spanning between range -2 km and 2 km at depth of 400 m. The results demonstrate an accurate agreement between the fixed Bellhop and the exact solution across negative and positive ranges.

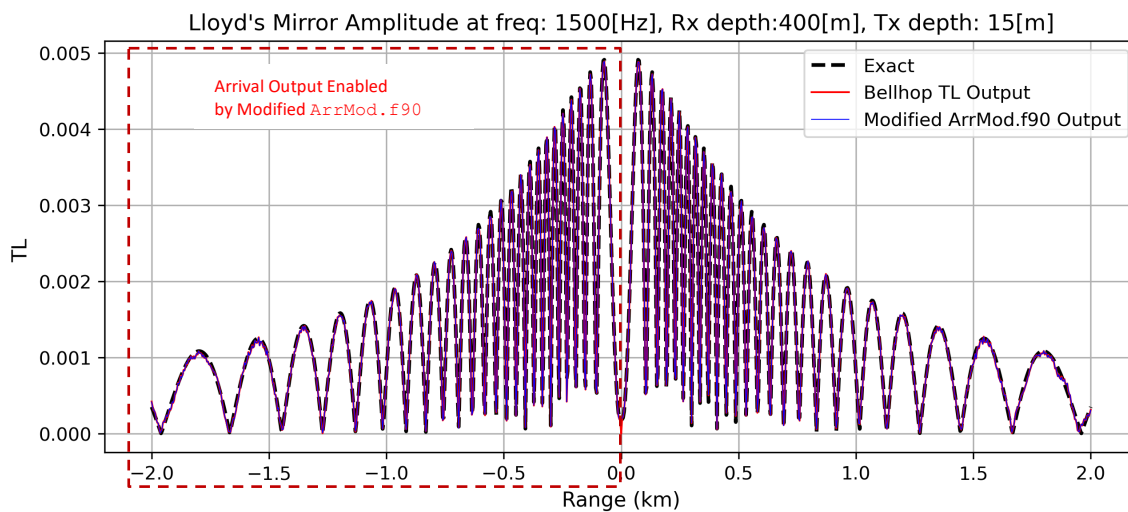


Figure 3: Bellhop calculated Lloyd's Mirror amplitude in negative and positive ranges (full space) compared with exact solution.

3 Absorption

Absorption of acoustic energy in seawater is a combination of three components: the pure water viscous contribution, boric acid reactions contribution and the magnesium sulphate reactions contribution (Finn et al., 2011, p.36). Pure water loss arises from viscous drag as a result of particle motion and is a function of temperature and pressure (depth). It becomes dominant at frequencies above 1 MHz.

Reaction mediated losses stem from molecules having more than one stable state that changes with pressure. Changes between states convert the fluctuating acoustic pressure into heat. These phase changes occur over a reaction time and are characterised by the 'relaxation frequency' of the reactions. High frequencies are too fast to affect chemical changes, hence these losses occur at lower frequencies (Figure 4). The main components in seawater which contribute to relaxation absorption are boric acid and magnesium sulphate.

Ainslie and McColm (1998) refined previous models based on measurements. They provide the relationship for absorption, α in [dB/km]

$$\alpha = 0.106 \frac{f_1 f^2}{f_1^2 + f^2} e^{(pH-8)/0.56} + 0.52 \left(1 + \frac{T}{43}\right) \left(\frac{S}{35}\right) \frac{f_2 f^2}{f_2^2 + f^2} e^{-D/6} + 0.00049 f^2 e^{-(T/27+D/17)} \quad (4)$$

where

$$f_1 = 0.78 \sqrt{(S/35)} e^{(T/26)} \quad (5)$$

is the boric acid relaxation frequency and

$$f_2 = 42 e^{(T/17)} \quad (6)$$

is the magnesium sulphate relaxation frequency. D is depth [m], f is acoustic frequency, S is salinity [parts per thousand or g/L] and T is temperature [°C].

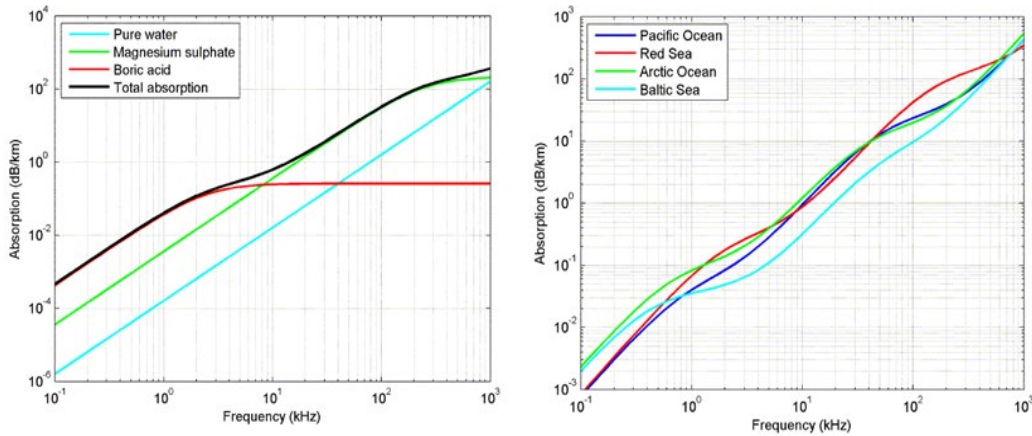


Figure 4: a) Relative contributions from sources of absorption as function of frequency. b) Variations in total absorption for different oceans (Ainslie & McColm, 1998).

Absorption is applied over the length of each ray however, the Ainslie & McColm model is a function of temperature, depth, salinity and pH hence the absorption value changes along the ray trajectory as it travels through the water column.

3.1 Bellhop Volume Attenuation

Bellhop implements several volume attenuation methods for user selection via the input “environment file”. Attenuation units can be explicitly specified with depth along the Speed of Sound Profile (SSP) block of data or a choice of two attenuation models can be selected via the SSP options string. The simplest option for including seawater attenuation is by selecting the Thorp model which requires no additional user input. In the Python framework using the ARLpy interface, this is the default and only run option.

We constructed an idealised scenario to examine Bellhop’s volume attenuation result and found some unexpected behaviour that may be relevant particularly when using ARLpy.

3.1.1 Test Scenario

A simple simulation environment was created to match an analytical case. The environment has constant SSP of 1500 m/s, no bottom reflections with bottom absorption coefficient set to 1, ray launch angles was set to only span the bottom facing hemisphere between 0° and 180°, the point source was set to a depth of 15 m and a line of receivers was placed at a depth of 400 m between ranges -2 km and 2 km.

The analytical solution for amplitude without attenuation is then just the previous case with a single direct ray

$$p(r_h, z_h) = \frac{e^{ikR_1}}{R_1} \quad (7)$$

where

$$R_1 = \sqrt{r_h^2 + (z_h - z_s)^2} \quad (8)$$

The simulation ray trajectories are visualised in Figure 5. While Bellhop reflects rays from the surface and bottom these reflections carry no energy due to bottom absorption. Without reflected energy, the single source is effectively in an infinite field and the magnitude of the exact solution without attenuation is frequency independent.

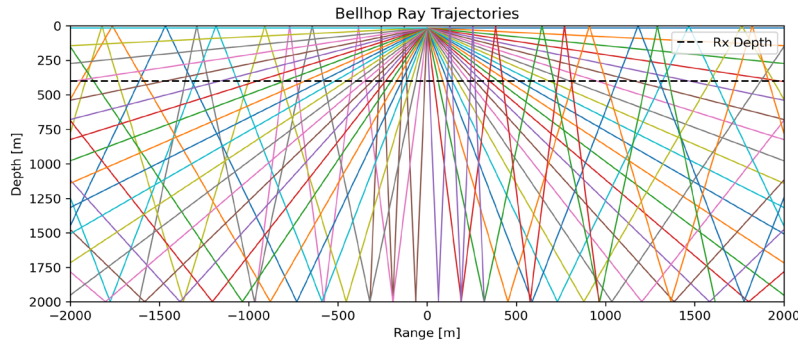


Figure 5: Ray paths from simulation, bottom and surface are fully absorbent.

Thorp attenuation as implemented in Bellhop uses the following equation for seawater absorption coefficient α :

$$\alpha' \simeq 3.3 \times 10^{-3} + \frac{0.11f^2}{1 + f^2} + \frac{44f^2}{4100 + f^2} + 3.0 \times 10^{-4} f^2 \quad (\text{dB/km}) \quad (9)$$

```
85 CASE ( 'T' )
86 ! Updated formula from JKPS Eq. 1.34
87 f2 = ( freq / 1000.0 ) ** 2
88 Thorp = 3.3d-3 + 0.11 * f2 / ( 1.0 + f2 ) + 44.0 * f2 / ( 4100.0 + f2 ) + 3d-4 * f2 ! dB/km
89 Thorp = Thorp / 8685.8896 ! Nepers / m
90 alphaT = alphaT + Thorp
```

Figure 6 Code snippet of Bellhop's Thorp implementation in AttenMod.f90.

3.1.2 Attenuation Output

Four sets of Bellhop outputs are produced. No attenuation case was produced by directly modifying the input 'environment file' and matches the exact solution. A 10 kHz and 20 kHz run were produced from the ARLpy interface using the 'Coherent' run type with 5000 geometric beams. The 10 kHz result was also produced by summing each ray amplitude using the 'Arrival' run type.

Examining Figure 7, the 10 kHz and 20 kHz results we see Bellhop results reflect frequency dependent attenuation with increased losses at higher frequency.

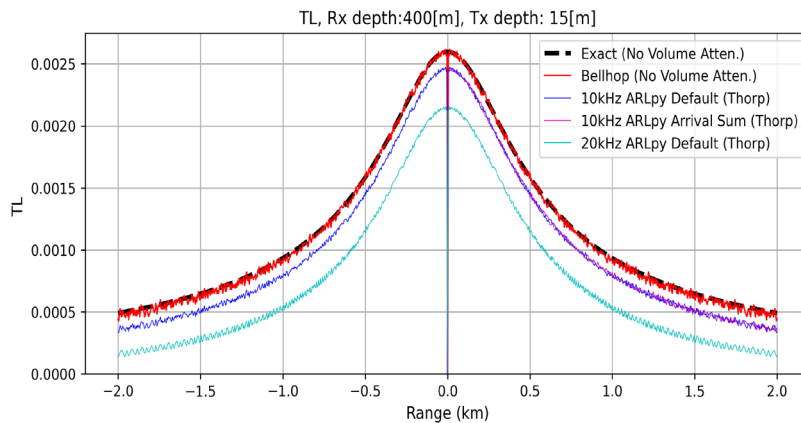


Figure 7 (left) Amplitude vs Range comparison between no attenuation, 10 kHz and 20 kHz Bellhop output.

A limitation of Bellhop's native attenuation is that each run is monochromatic (single frequency). A broadband analysis requires multiple independent runs which is time consuming when ray count is high and the tasks performed is mostly redundant since the speed of sound and ray paths are not frequency dependent.

3.2 Post Process Attenuation

To resolve this issue, Bellhop's attenuation model can be bypassed to obtain the unattenuated results. Bellhop's other outputs are sufficient to calculate the correct attenuation with post processing.

3.2.1 Disable Bellhop Attenuation

Disabling Bellhop's attenuation model is a simple matter of omitting the switch (i.e. 'T') from the SSP options on line 4 of the environment input file. However, ARLpy hardcodes this portion of the input file generation. To remove the switch, the ARLpy library module "uwapm.py" can be modified by replacing the 'T' with a space character from line 933 and line 935.

```

932     if env['surface'] is None:
933         self._print(fh, "%cV" % svp_interp)
934     else:
935         self._print(fh, "%cV" % svp_interp)
936         self._create_bty_ati_file(fname_base+'.ati', env['surface'], env['surface_interp'])

```

Figure 8 Code snippet from ARLpy's "uwapm.py" module. Replace 'T' with space to disable Thorp attenuation and obtain unattenuated results.

3.2.2 Ray Arrivals and Trajectories

Bellhop can output a list of ray trajectories to a receiver as series of points using the 'E' run type or `compute_eigenrays` function using ARLpy. Ray arrival amplitude and time delay for each receiver point can be generated using the 'A' run type or ARLpy function `compute_arrivals`.

Note that the 'C' (coherent) run type or `compute_transmission_loss` function only outputs the combined amplitude from all rays arriving at each receiver and is not appropriate for attenuation which needs to account for path length of each ray.

Run Type	Relevant Output Parameters		
'E' Eigenrays	ray angle of departure	ray trajectory points array of <code>[[range, depth]]</code>	
'A' Arrivals	ray angle of departure	ray arrival time	ray arrival amplitude

Independent results are produced from these separate runs; however, arrival entries can be attributed to the trajectories by matching with the closest launch angles.

The arrivals data lists all rays which contribute to the amplitude at the receiver however many of these rays only reach the receiver after many bounces and are extremely weak at less than 1% of the dominant ray amplitude. A threshold can be set to cull those that make little contribution in order to save computation time while making negligible difference to the final summed result (Figure 10).

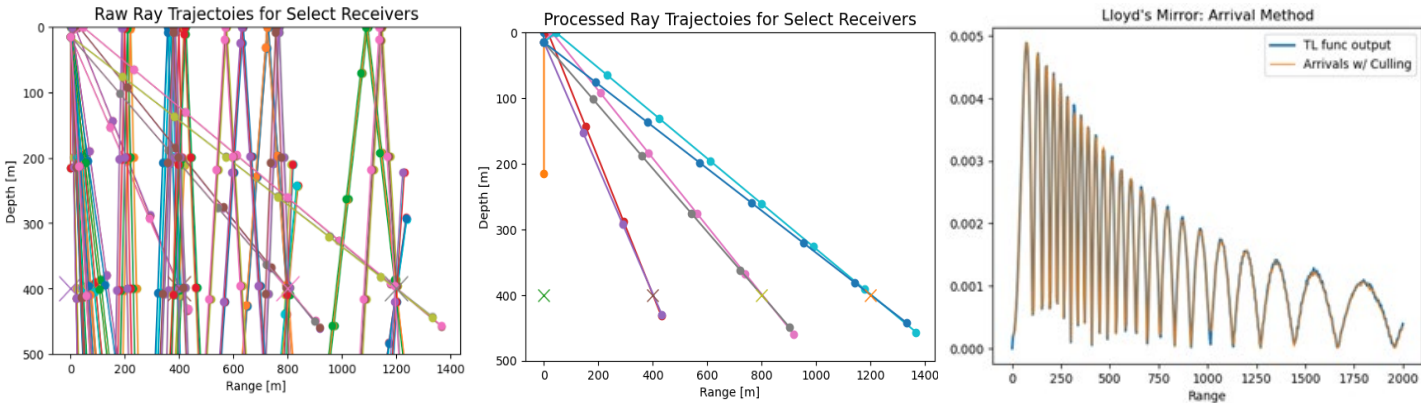


Figure 9 (Left) All rays attributable to arrivals at select receivers (at range 0 m, 400 m, 800 m, 1200 m). (Centre) same receivers after ray culling. (Right) Comparison of result for transmission loss by summing all rays versus culled arrivals.

3.2.3 Attenuation

Obtaining the trajectory points for each ray arrival provides locations to obtain input parameters for the absorption model. Assuming water properties are range independent. Absorption input parameters are provided as depth profiles similar to the speed of sound. Figure 11 shows examples of how the absorption spectrum changes with temperature (note: the profile is used as a lookup table and can consist of value pairs from any model).

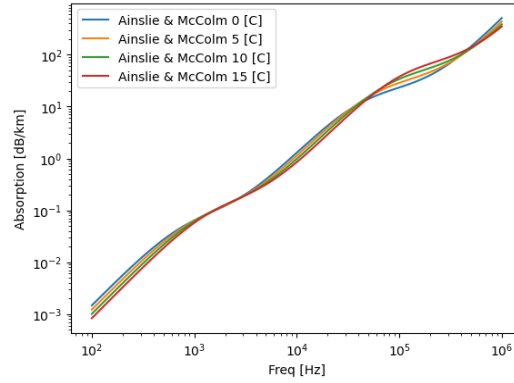


Figure 10 Change in absorption for temperature 0 °C to 15 °C using the Ainslie and McColm model (depth: 50 m, salinity: 35 ppt, pH: 8)

Variations in absorption along the ray path can be captured by calculating the local absorption spectrum along trajectory points and summing to find the total extra attenuation for application to the Bellhop amplitude.

The model lookup provides absorption α as dB/km, this should be converted to dB/m for use with ARLpy results in meters. Attenuation can be calculated per trajectory point, interpolated between adjacent values, scaled by the distance between points and sum for all segments along the ray path to find the total ray attenuation using

$$\alpha_{ray} = \sum_i \frac{1}{2}(\alpha_i + \alpha_{i+1})s_i \quad (10)$$

where s_i is the distance between trajectory points (x_i, z_i) and (x_{i+1}, z_{i+1})

$$s_i = \sqrt{(x_{i+1} - x_i)^2 + (z_{i+1} - z_i)^2}.$$

The total absorption along each ray is applied to the Bellhop output amplitude for each ray (14)(8). The final intensity at each receiver is the sum of all its eigenray amplitudes.

$$I_{ray,atten} = I_{ray,bellhop} 10^{-\frac{\alpha_{ray}}{10}} \quad (13)$$

An advantage of the post process attenuation approach is the ability to generalise Bellhop's single frequency output to broadband signals. A channel impulse response can be obtained at a receiver location using ray arrival time and amplitude data (Porter, 2010). Convolution of the impulse response with any arbitrary input signal produces the received signal time series $x[n]$. Absorption can then be applied to the received signal as a filter with the following.

Given $x[n]$ is a real signal, obtain the positive only RFFT frequency domain signal (`numpy.fft.rfft`)

$$X^+[k] = \mathcal{F}_N^+\{x[n]\} \quad (14)$$

Obtain bin frequencies f_k (`numpy.fft.rfftfreq`) for N samples at sampling rate F_s

$$f_k = \frac{k}{N}F_s \quad (15)$$

Using the absorption model to obtain attenuation values for all bin frequencies at each trajectory step $\alpha_i[k]$ as per (11) to find the total ray attenuation spectrum array $\alpha_{ray}(k)$. Convert the attenuation spectrum to linear scale if the model provides decibel units.

$$H[k] = 10^{-\frac{\alpha_{ray} dB[k]}{10}} \quad (16)$$

Use absorption spectrum ($H[k]$) to scale the signal spectrum ($X^+[k]$)

$$Y^+[k] = X^+[k] H[k] \quad (17)$$

Apply inverse RFFT (`numpy.fft.irfft`) to obtain attenuated intensity $y[n]$. Sum all rays to find total intensity at the receiver.

$$y[n] = \mathcal{F}_N^{-1,+}\{Y^+[k]\} \quad (18)$$

3.2.4 Results

The described method is applied to the unattenuated Bellhop output with a broadband white input signal. The results in Figure 12 (left) demonstrates the obtained frequency attenuation with the no absorption signal (horizontal lines) and the spectrum curves after absorption at select range values between 0 m and 2000 m. Figure 12 (right) illustrates the data against range.

Validation was performed using a three-tone input signal. Attenuation of each frequency component at the receiver with 1 km path distance from the source matches with the Ainslie & McColm absorption model curve Figure 13 (right).

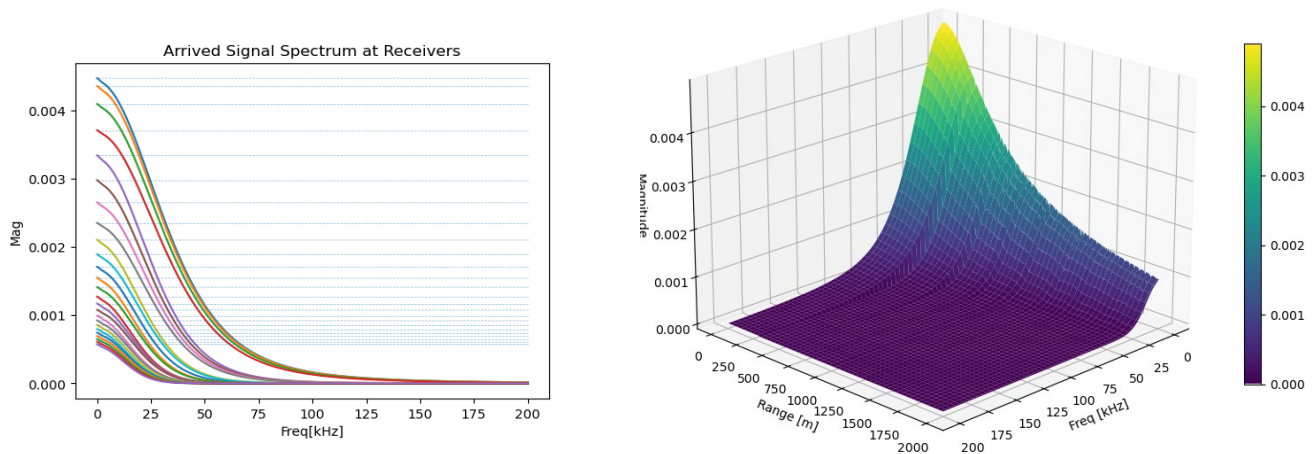


Figure 11. (left) A white signal spectrum after post processed absorption at various ranges from 0 km to 2 km. (right) the 3D contour illustrates both range and frequency dependence.

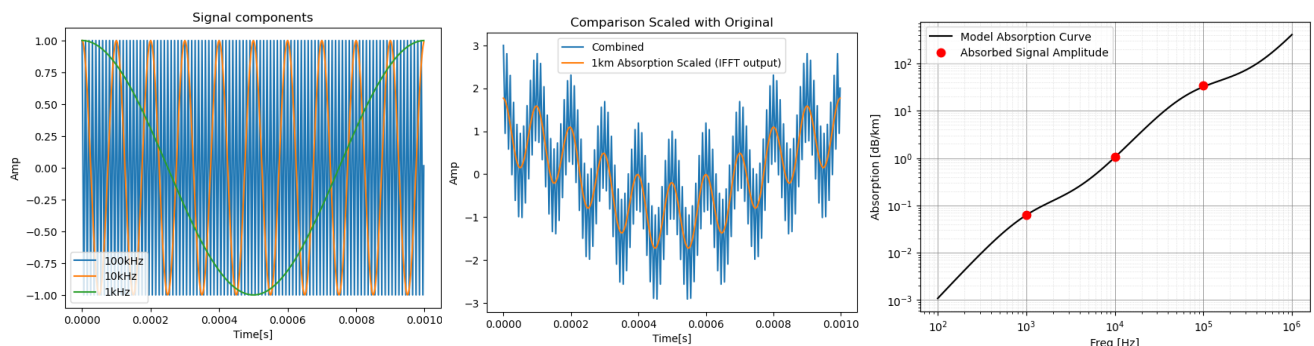


Figure 12 (Left) Three component mixed frequency input signal. (Centre) Received signal after 1km path length with and without attenuation. (Right) Output signal frequency component attenuation vs expected absorption curve.

4 Conclusion

This study demonstrated practical methods to overcome Bellhop's legacy limitations and to extend its usability and utility. By identifying and correcting the range-sign error in the `ArrMod.f90` module, we enabled full-space simulation without the need for dual half-space runs. The post process ray-path extraction and step-wise application of the absorption model provided an alternative way to include frequency-dependent attenuation within a ray-traced environment. Validation against the analytical cases confirmed the accuracy of the modified code and the absorption post-processing. The improvements streamline workflows for full space simulation and the post processed attenuation provide a pathway to obtain broadband, frequency-dependent volume loss with Python-based toolchains.

ACKNOWLEDGEMENTS

I would like to acknowledge this publication is supported by Saab Australia Pty Ltd, The University of Adelaide and the Australian Government Department of Education through the Trailblazer Universities Program.

REFERENCES

- Ainslie M. A., McColm J. G. (1998), "A simplified formula for viscous and chemical absorption in sea water", *Journal of the Acoustical Society of America*, 103(3), 1671-1672, 1998.
- Casari, P., Tapparello, C., Guerra, F., Favaro, F., Calabrese, I., Toso, G., Azad, S., Masiero, R., & Zorzi, M. (2014). *Open source suites for underwater networking: WOSS and DESERT underwater*. *IEEE Network*, 28(5), 38–46. <https://doi.org/10.1109/MNET.2014.6915438>
- Chitre, M. (2025). *Bellhop – UnderwaterAcoustics.jl*. Github.io; Github. <https://org-arl.github.io/UnderwaterAcoustics.jl/bellhop.html>
- HLS Research (2024). *Acoustics Toolbox*. [online] Hlsresearch.com. Available at: <http://oalib.hlsresearch.com/AcousticsToolbox/>.
- Jensen, F. B., Kuperman, W. A., Porter, M. B., & Schmidt, H. (2011). *Computational Ocean Acoustics* (2nd ed. 2011.). Springer New York. <https://doi.org/10.1007/978-1-4419-8678-8>
- Lan, C, Yu, Z, Chen, H, Zhang, L & Zhang, M. (2024), 'Research on Underwater Collaborative Detection Method Based on Complex Marine Environment', *IEEE Access*, vol. 12, pp. 1–1. <https://doi.org/10.1109/ACCESS.2023.3348782>
- National Physical Laboratory, *Sea Absorption*, Accessed: Jun 19 2025 [Online]. Available: <http://resource.npl.co.uk/acoustics/techguides/seaabsorption/physics.html>
- Pisha, L.A., Snider, J., Jackson, K. and Jaffe, J.S. (2023). *Introducing bellhopcxx/bellhopcuda: Modern, parallel BELLHOP(3D)*. *The Journal of the Acoustical Society of America*, [online] 153(3_supplement), pp.A218–A218. doi:<https://doi.org/10.1121/10.0018709>.
- Strode, C., Akbulut, B., Been, R., & Tesei, A. (2023). *Acoustic reconstruction of ASW serials during NATO exercises*. Underwater Acoustics Conference and Exhibition (UACE) 2023, Grecotel Filoxenia, Greece. https://www.uaconferences.org/docs/2023_papers/UACE2023_2074_Tesei.pdf