# Differentiable Ocean Acoustic Propagation Modeling

Mandar Chitre
ARL, National University of Singapore

***Abstract*—Physics-based ocean acoustic propagation models take in the location of an acoustic source and an environmental description, and yield estimates of acoustic field at points of interest. In many cases, acoustic field measurements are available but the environment is not completely known. Determining unknown environmental parameters given field measurements is of interest in *inverse problems*. Techniques for solving inverse problems are usually based on iterative optimization of a loss function, and greatly benefit from the availability of the gradient of the loss function with respect to the unknown parameters. Traditional acoustic propagation models are complicated programs that do not provide any easy way to obtain gradients. While gradients can be estimated using finite difference techniques, such estimates suffer from poor numerical accuracy and are computationally expensive to obtain. Recent advances in machine learning have led to the development of automatic differentiation techniques that yield gradients of complicated mathematical algorithms efficiently and accurately. We present two open-source ocean acoustic propagation models that are designed ground-up to work well with automatic differentiation. These models not only work well with gradient-based optimization techniques, but also can be used together with machine learning techniques such as neural networks to learn from data. To illustrate the value of such models, we present four application examples where the gradients are used to solve inverse problems.**

## I. Introduction

Since electromagnetic waves are rapidly absorbed by seawater, acoustics is very commonly used for remote sensing and communication applications underwater. Sound strongly interacts with the sea surface and seabed, typically resulting in complicated interference patterns due to multipath. It also refracts in the water column due to changes in sound speed caused by temperature, salinity and pressure changes. In order to understand or predict the underwater acoustic field due to a sound source, we use well-established ocean acoustic propagation models such as ray/beam tracing or method of normal modes [1]. These models typically take in an environmental description in terms of the source location, sound speed profile, bathymetry, seabed properties and sea surface properties, and estimate the acoustic field at various receiver locations.

In many cases, the environmental information may only be partially known, but some acoustic field measurements may be available. In such cases, the environmental information may be inferred from the available measurements. Such problems are known as *inverse problems*, and are of great interest in the field [2]. Techniques such as *matched field processing* (MFP) incorporate incorporate inference of environmental information into the processing of the acoustic signals [3]. Most such techniques rely on iterative evaluation of acoustic propagation models to test various hypothesized environmental parameters to reduce the error between the model predictions and field measurements. Mathematically, one can generalize this approach as an optimization problem:

$$\min_{\Theta} \mathcal{L}(\Theta),$$

where $\mathcal{L}$ is a loss function that measures the mismatch between predictions and measurements. A typical sum-square loss may be written as:

$$\mathcal{L}(\Theta) = \sum_i \left| \mathcal{M}(\mathbf{x}_i, \Theta) - y_i \right|^2, \tag{1}$$

where $y_i$ are field measurements made at locations $\mathbf{x}_i$, $\Theta$ contains environmental knowledge (including source location), and $\mathcal{M}(\mathbf{x}, \Theta)$ is the acoustic propagation model that predicts the acoustic field at location $\mathbf{x}$ given environmental knowledge $\Theta$.

While the non-linear global optimization problem in (1) is generally hard to solve, local optima may be readily found using gradient-based optimization techniques [4]. With partial environmental information to guide the initialization of the search, the resulting local optimum may often provide the desired result. Even in cases where global optimization methods are invoked, the availability of the gradient $\nabla\mathcal{L}$ can significantly speed up the optimization. Other approaches to solving inverse problems include Bayesian inference using Markov Chain Monte Carlo (MCMC) sampling and variational inference [5], [6]. Efficient MCMC techniques such as HMC [7] or NUTS [8] require gradient $\nabla\mathcal{L}$ to be available. Variational inference also requires $\nabla\mathcal{L}$.

The challenge, however, is that $\mathcal{M}$ is generally not available in closed-form, and therefore cannot be symbolically differentiated. In most cases, $\mathcal{M}$ is a complicated algorithm involving numerical differential equation solvers (e.g. to solve the Eikonel equation) and optimization techniques (e.g. to find eigenray launch angles). While numerical differentiation is possible, it requires multiple evaluations of the model per degree of freedom in $\Theta$ and becomes computationally expensive as the dimensionality of $\Theta$ increases. Moreover, the gradient estimates from numerical differentiation suffer from poor accuracy due to floating point errors.

Automatic differentiation (AD) is a powerful tool to automate the calculation of derivatives of complicated mathematical algorithms [9]. For AD to be applied to a computational propagation model, the model needs to be written in a way that is designed for AD. Well-established acoustic propagation models such as Bellhop [10] are written in Fortran and not easily amenable to AD techniques available today. We present two new open-source *differentiable* acoustic propagation models developed in the Julia programming language, and designed to be compatible with popular AD tools in the Julia ecosystem.

The acoustic propagation models are not only amenable to gradient-descent based optimization techniques, but also support uncertainty propagation [11] and can be used as part of probabilistic programs [12], [13]. Since modern machine learning tools rely on AD during training, these acoustic propagation models can be used together with machine learning models such as neural networks to learn from data. For example, if a seabed reflection model or a sound speed profile is unknown, it may be substituted by a neural network that is trained end-to-end using observed acoustic data.

In the next section, we present the two propagation models in detail. This is followed by four illustrative application examples to show how these models may be used together with machine learning techniques to solve inverse or MFP problems.

## II. Propagation Modeling Toolkit

The acoustic propagation modeling toolkit (`UnderwaterAcoustics.jl`) [14] is an open-source project that provides a common interface for various acoustic propagation models, including popular models such as Bellhop and Kraken [15]. We have two differentiable propagation models available with the toolkit:

### A. Pekeris Ray Model

The `PekerisRayModel` is a fast differentiable ray propagation model for 3D range-independent iso-velocity ocean environments, and is available out-of-the-box in `UnderwaterAcoustics.jl`. It implements the model described in [16].

The `PekerisRayModel` models high-frequency acoustic propagation in an environment with constant bathymetry and sound speed. Ray paths from a source to a receiver in such an environment can be computed analytically from knowledge of the channel geometry. This yields closed form expressions for path length, spreading loss, time of arrival, angle of arrival, number of seabed and surface interactions, and incidence angles at surface and seabed [16]. By employing appropriate volume absorption (e.g. Francois-Garrison [17]) and geo-acoustic (e.g. Rayleigh reflection [18] with absorption [19]) models, we compute each arrival intensity and phase at the receiver. We then sum the arrivals to compute the received acoustic field. The interface exposed by the `PekerisRayModel` permits other more comprehensive geo-acoustic models to be plugged-in, if desired.

The `PekerisRayModel` is fully differentiable with forward-mode and reverse-mode AD packages such as `ForwardDiff.jl` [20] and `Zygote.jl` [21]. It is also compatible with probabilistic programming tools such as `Turing.jl` [12] and measurement uncertainty propagation tools such as `Measurements.jl` [11].

### B. Ray Solver

In environments with a depth-dependent sound speed or range-dependent bathymetry, we need a full-featured ray/beam propagation model for high-frequency acoustic propagation modeling. The `RaySolver` is an open-source ray/beam propagation model for 2.5D ocean environments with complex bathymetry and sound speed profile. It implements a similar model as Bellhop, but is differentiable using AD tools such as `ForwardDiff.jl`. It is available by installing the Julia package `AcousticRayTracers.jl` [22].

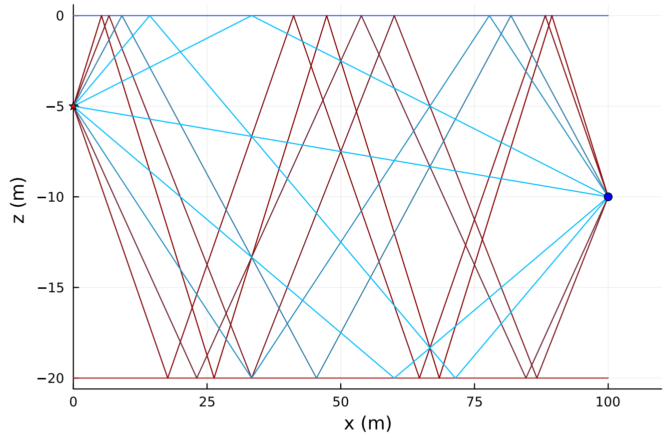
The `RaySolver` implements a ray model as described in [1, Ch. 3]. It traces ray paths between the transmitter and the receiver by solving a system of first-order non-linear partial differential equations (PDE) in cylindrical coordinates (range $r$, depth $z$):

$$\begin{aligned}\frac{dr}{ds} = c\xi(s), \quad \frac{d\xi}{ds} &= -\frac{1}{c^2}\frac{\partial c}{\partial r},\\ \frac{dz}{ds} = c\zeta(s), \quad \frac{d\zeta}{ds} &= -\frac{1}{c^2}\frac{\partial c}{\partial z},\end{aligned}$$

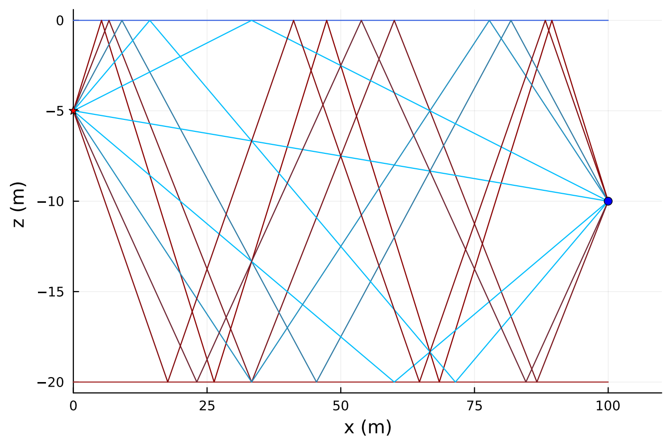
where $c$ is the sound speed, $s$ is the path length along the ray, and $\zeta(s)$, $\xi(s)$ are auxiliary variables to convert the PDE into a first-order form. The system of PDEs is solved using either Runge-Kutta [23] or Rosenbrock methods [24], based on user configuration.

The absorption along ray paths is computed using the Francois-Garrison volume absorption model [17], and each interaction with the seabed and sea surface is governed by a reflection model (e.g. Rayleigh reflection [18] with absorption [19]). Other geo-acoustic models can be used in place of Rayleigh reflection, if desired.

When computing the acoustic field at a small number of receivers, the `RaySolver` computes eigenrays between each source and receiver pair by iterative optimization of ray launch angles. This results in an accurate arrival structure and acoustic field estimate, but is computationally expensive. When computing the acoustic field over a large area, `RaySolver` follows a similar approach as Bellhop to use a rectangular grid of virtual receivers, each accumulating contributions from Gaussian beams fanning out from the

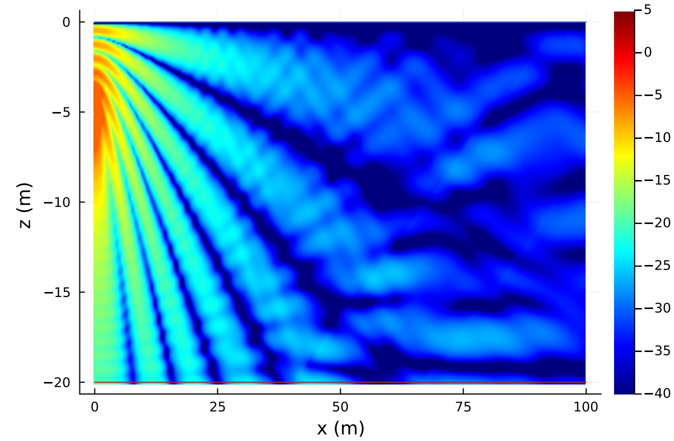



(a) `PekerisRayModel`

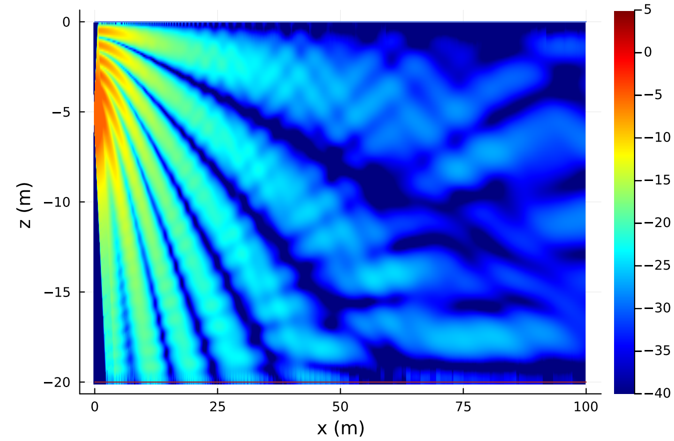



(b) `RaySolver`

Figure 1: Comparison of ray traces from the `PekerisRayModel` and the `RaySolver` for an environment with an iso-velocity sound speed profile. The transmitter is at a 5 m depth and the receiver is at a 10 m depth, in a water depth of 20 m. The range between the transmitter and receiver is 100 m.




(a) `PekerisRayModel`




(b) `RaySolver`

Figure 2: Comparison of transmission loss estimates from the `PekerisRayModel` and the `RaySolver` for an environment with an iso-velocity sound speed profile. The transmitter is at a 5 m depth in channel with 20 m water depth.

source [1, Sec. 3.3.5.5]. This is significantly faster, but less accurate.

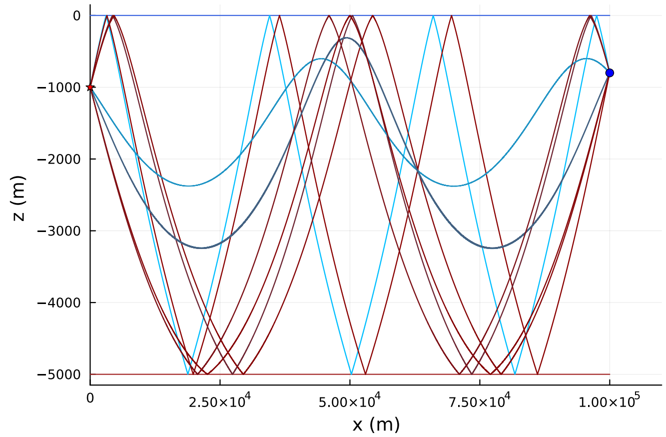
The `RaySolver` is compatible with forward-mode AD (using `ForwardDiff.jl`), but with one limitation: it is currently unable to compute gradients with respect to change in launch angle of eigenray. If gradients with respect to geometry changes are required, numerical differentiation using `FiniteDifferences.jl` [25] may be used instead. This limitation may be removed in future versions of `RaySolver`.

### C. *Model comparison*

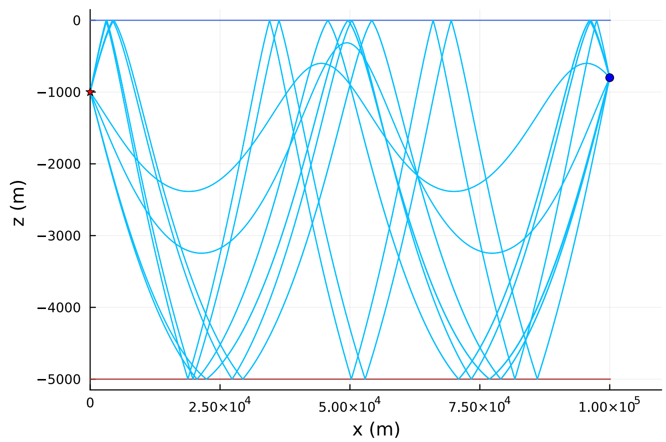
Figure 1 shows a comparison of the ray traces from `PekerisRayModel` and `RaySolver` for an environment with an iso-velocity sound speed profile in 20 m of water depth. The transmitter is placed at a 5 m depth and the receiver is placed at a 10 m depth. The range between the transmitter and receiver is 100 m. We see that the ray traces produced by both models are identical.

Figure 2 shows a comparison of transmission loss estimates from `PekerisRayModel` and `RaySolver` for the same environment. While the estimates are not identical, they are very similar. The small differences are due to the Gaussian beam approximation used by the `RaySolver`, as against exact eigenrays used by `PekerisRayModel`.

Figure 3 shows a comparison of ray traces from the `RaySolver` and Bellhop for an environment with Munk sound speed profile (Figure 4). The transmitter is placed at a depth of 1 km and the receiver is placed at a depth of 800 m. The water depth is 5 km, and the range between the transmitter and receiver is 100 km. Only the strongest 9 rays are shown to avoid cluttering the figure. We see that




(a) `RaySolver`




(b) Bellhop

Figure 3: Comparison of ray traces from the `RaySolver` and Bellhop for an environment with Munk sound speed profile (Figure 4). The transmitter is at a 1 km depth and the receiver is at a 800 m depth, in a water depth of 5 km. The range between the transmitter and receiver is 100 km. Only the strongest 9 rays are shown.

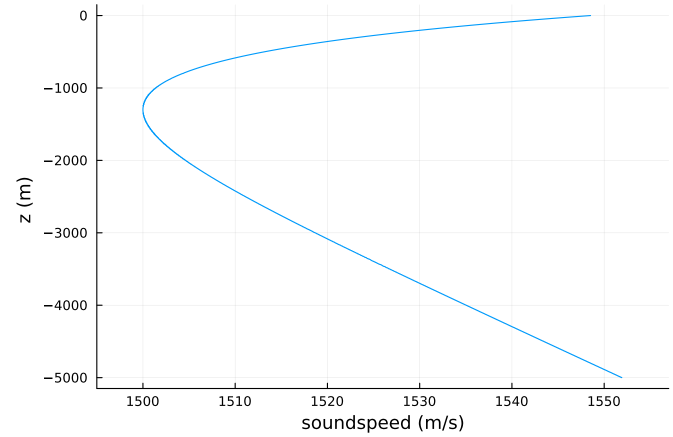



Figure 4: Munk sound speed profile.

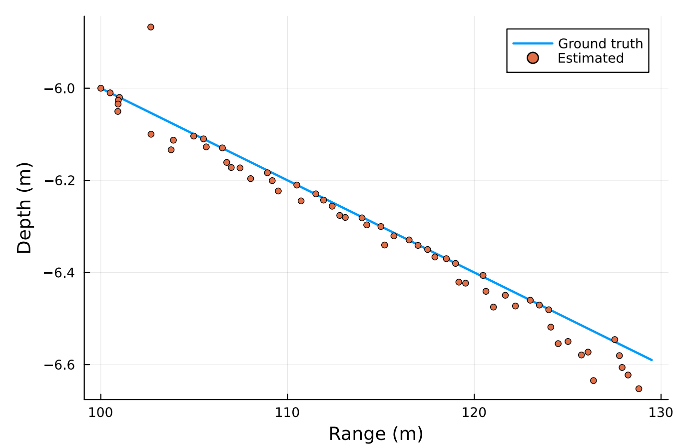



Figure 5: Estimated and actual drifter positions.

the ray traces from both models are essentially identical.

## III. Application Examples

To illustrate the usefulness of differentiable acoustic propagation models, we present four simulated application examples next. The examples have been intentionally kept simple to focus on key ideas, and not on practical issues that no doubt have to be dealt with in real experiments. A detailed tutorial and the source code for all four application examples is available online [14].

### A. Tracking a drifting transmitter

Consider a scenario where a drifting probe acoustically transmits its sensor data periodically to a static receiver. We assume that the initial position of the probe is perfectly known, and so is the environment, but the path of the probe as it drifts is not known. We wish to estimate the path from the received acoustic communication signals.

The simulated environment is an iso-velocity channel with a constant depth. The probe uses a 1–2 kHz band for data transmission, and includes $N = 101$ pilots at 10 Hz spacing to aid with channel estimation. The transmission loss $\gamma_i$ can be accurately measured at those pilot frequencies $f_i$, since the transmit source level is assumed to be known, but phase information is assumed to be unavailable at each pilot.

The location of the probe $\mathbf{x}_t$ at each transmission is estimated by using gradient descent to solve the optimization problem:

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \gamma_{1t}, \gamma_{2t}, \cdots, \gamma_{Nt}),$$

where

$$\mathcal{L}(\mathbf{x}; \gamma_1, \gamma_2, \cdots, \gamma_N) = \sum_{i=1}^{N} \left| \mathcal{M}(\mathbf{x}, f_i) - \gamma_i \right|^2 .$$

Here, $\mathcal{M}(\mathbf{x}, f_i)$ is the modeled transmission loss at location $\mathbf{x}$ and frequency $f_i$. For for each transmission $t$, $\gamma_{it}$ is the measured transmission loss at frequency $f_i$.

The modeled transmission loss is computed using the `PekerisRayModel`, and forward-mode AD is used to obtain the gradient $\nabla\mathcal{L}$ for gradient descent. At each transmission $t$, 10 iterations of gradient descent are used to estimate position $\mathbf{x}_t$ with initial guess $\mathbf{x}_{t-1}$.

The estimated and actual range of drifter from the receiver is shown in Figure 5, as the sensor slowly drifts over 30 m. We see good agreement between the two.

### *B. Geo-acoustic inversion with an acoustic profiler*

We next consider a geo-acoustic inversion problem where we have a static omnidirectional broadband acoustic source transmitting in a 5–7 kHz band, and a single omnidirectional receiver that records the signal at a fixed range. The receiver is able to profile the water column, and therefore makes transmission loss measurements at various depths. We wish to estimate seabed parameters (relative density $\rho$, relative sound speed $c$, and dimensionless absorption coefficient $\delta$)[1] from the transmission loss measurements. Do note that although we have acoustic measurements at various depths, they cannot be used for beamforming to separate out the bottom reflected arrival from other arrivals. We therefore only have transmission loss at each depth for our inversion.

In geo-acoustic inversion problems, we usually have some priors on the range of values that the geo-acoustic parameters might take. In this example, we assume $\rho \sim \mathcal{U}(1,3)$, $c \sim \mathcal{U}(0.5, 2.5)$, and $\delta \sim \mathcal{U}(0, 0.003)$, where $\mathcal{U}$ represents a uniform distribution. We also assume uncorrelated Gaussian transmission loss measurement noise with variance $\sigma^2 = 0.5$. The problem can then be written as:

$$\mathbf{y} \sim \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I}),$$

where $\mathbf{x}$ is a vector containing the modeled transmission loss at various depths, $\mathbf{y}$ is a vector with measured transmission loss at those depths, $\mathcal{N}$ represents the multivariate Normal distribution, and $\mathbf{I}$ is the identity matrix. The modeled transmission loss $x_i$ at depths $d_i$ can be computed using an acoustic propagation model $\mathcal{M}$:

$$x_i = \mathcal{M}(d_i; \Theta),$$

where $\Theta = (\rho, c, \delta)$ are the geo-acoustic parameters of interest.

Given model $\mathcal{M}$ and prior distributions over parameters $\Theta$, we can formulate the geo-acoustic inversion problem as a Bayesian inference problem:

$$\mathcal{P}(\Theta|\mathbf{y}) = \frac{\mathcal{P}(\mathbf{y}|\Theta)\mathcal{P}(\Theta)}{\int_\Theta \mathcal{P}(\mathbf{y}|\Theta)\mathcal{P}(\Theta)}. \quad (2)$$

[1]The relative sound speed is the ratio of sound speed in the seabed to the sound speed in water. Similarly, the relative density is the ratio of density of seabed to density of water.

Table I: Estimated geo-acoustic parameters.

| $\Theta$ | Ground truth | Estimated |
|---|---|---|
| $\rho$ | 1.300 | $1.2929 \pm 0.0043$ |
| $c$ | 1.100 | $1.1024 \pm 0.0002$ |
| $\delta$ | 0.001 | $0.0014 \pm 0.0002$ |

Bayesian inference problems are readily solved with probabilistic programming [26] using either MCMC sampling or variational inference. We use the Turing probabilistic programming language [12] with automatic differentiation variational inference (ADVI) [13] to solve problem (2).

The resulting estimated distributions of the seabed parameters are shown in Table I. We see that the estimated values of the parameters agree well with the simulated *ground truth* seabed parameters.

### *C. Estimating channel geometry from impulse response*

In the previous example, we had perfect knowledge of channel geometry and needed to estimate seabed parameters using probabilistic programming. We now consider an example where we estimate channel geometry using a similar technique.

Consider a scenario where a bottom-mounted sensor is deployed on a sub-surface mooring at an unknown altitude over the seabed. The exact location of the sensor is unknown, but we know the general area where it is deployed. The sensor is equipped with an acoustic transponder that we can query from a surface unit when we are within a 250 m range of it. We deploy the transducer of the surface unit from the boat on a 7 m rope with a weight attached. Due to currents, the transducer is not hanging perfectly vertically, and so it's exact depth $d_1$ is not known. The depth sounder of our boat tells us that the water depth $h = 20$ m, and the Captain of the boat assures us that the bathymetry is quite flat. We query the transponder and get a response. The broadband acoustic response from the transponder allows us to estimate the delays $\{y_i\}$ of 4 multipath arrivals. We wish to estimate the depth $d_2$ of the sensor and the range $r$ between the boat and the sensor using the multipath arrival delays $\{y_i\}$.

We can model this as a Bayesian inference problem of the form (2), with parameters $\Theta = (d_1, d_2, r, h)$ and measurement vector $\mathbf{y}$ of the arrival delays. The acoustic propagation model $\mathcal{M}(\Theta)$ predicts the arrival delays $\mathbf{x}$ given the channel parameters $\Theta$. So:

$$\mathbf{y} \sim \mathcal{N}(\mathcal{M}(\Theta), \sigma^2\mathbf{I}),$$

where $\sigma = 0.1$ ms is the measurement noise in arrival delays.

We solve the probabilistic program using Turing and ADVI as in the previous example. The priors used, *ground truth*

Table II: Estimated channel geometry.

| $\Theta$ | Ground truth | Prior | Estimated |
|---|---|---|---|
| $h$ | 20.00 m | $\mathcal{N}(20, 0.1)$ | $19.99 \pm 0.05$ m |
| $r$ | 97.30 m | $\mathcal{U}(0, 250)$ | $97.87 \pm 0.64$ m |
| $d_1$ | 7.20 m | $\mathcal{N}(7, 1)$ | $7.20 \pm 0.15$ m |
| $d_2$ | 12.70 m | $\mathcal{U}(0, 20)$ | $12.67 \pm 0.15$ m |

simulated values, and estimated values are shown in Table II. We see that the estimated values of the parameters agree well with the ground truth.

### D. Inferring sound speed profile from impulse response

In the previous three examples, we worked with isovelocity sound speed environments and hence adopted `PekerisRayModel` for acoustic modeling. In the next example, we simulate a long-range deep water channel with a Munk sound speed profile as shown in Figure 4. We use the `RaySolver` for acoustic modeling.

Consider a setup with a 1 kHz acoustic source at 1 km depth that sends a broadband pulse once every week. A receiver 10 km away at a depth of 800 m measures the impulse response from the received broadband pulse. We assume that we have an initial sound speed profile measurement with a CTD at the start of the experiment. The sound speed profile changes over the weeks of the experiment, and we wish to track the changes using the measured impulse response every week.

Since the sound speed profile is an unknown function of depth, we model it using a small 3-layer neural network with parameters $\Theta$. We initialize the parameters $\Theta$ by training the neural network on the known/estimated sound speed profile from the previous week. We use the neural network to provide sound speed estimates to `RaySolver`, thus effectively creating a model $\mathcal{M}$ that combines a numerical physics-based differential equation solver with a data-driven neural network. The hybrid model returns a vector $\mathbf{x}$ of predicted delays of the first 4 multipath arrivals in the impulse response:

$$\mathbf{x} = \mathcal{M}(\Theta)$$

Let $\mathbf{y}$ be a corresponding vector of measured delays of the multipath arrivals in the impulse response. We wish to minimize a loss function $\mathcal{L}$ that measures the difference between the predictions and measurements:

$$\min_{\Theta} \mathcal{L}(\Theta),$$

where

$$\mathcal{L}(\Theta) = \|\mathcal{M}(\Theta) - \mathbf{y}\|_2^2.$$

The optimization problem is solved using gradient descent, with the gradient $\nabla\mathcal{L}$ obtained using AD through the `RaySolver` acoustic propagation model. This essentially trains the neural network to approximate the sound speed profile via a loss function that utilizes the propagation model $\mathcal{M}$. At the end of each week's training, we get a revised estimate of the sound speed profile.

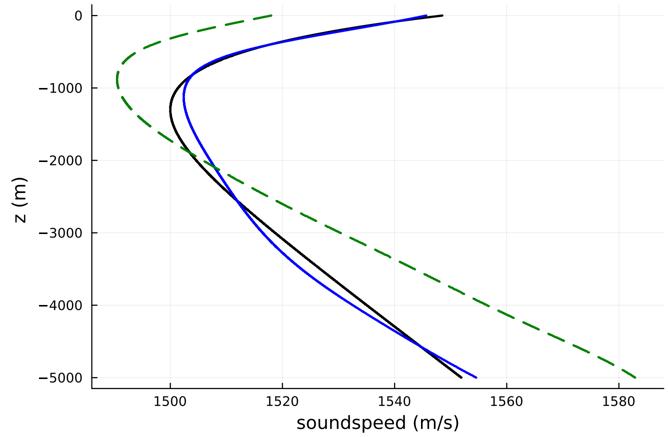



Figure 6: Estimated (blue), ground truth (black) and initial (greed dashed) sound speed profiles. The initial profile is from CTD measurements at the start of the experiment, or the estimated profile from the previous week. The estimated sound speed profile is obtained from the neural network.

The simulated *ground truth* sound speed profile (solid black line), learnt estimate of the profile (solid blue line), and the initialization from previous week (dashed green line) are shown in Figure 6. While we see that the estimate does not match the ground truth perfectly, it is a very good match considering that it was obtained from a single measurement from a single hydrophone!

## IV. Conclusion & Future Work

The availability of an open-source differentiable acoustic propagation model opens up new possibilities for algorithms that exploit the gradient of the modeled quantities with respect to the input parameters of the model. We illustrated the use of AD with such models to solve geoacoustic inversion, source localization and sound speed profile estimation problems using well-known techniques such as ADVI and gradient-descent. We also demonstrated how a physics-based acoustic propagation model can be combined with a neural network into a single hybrid model that can be trained end-to-end. However, this only scratches the surface of what is possible with differentiable acoustic modeling. Other than solving inverse problems, such models can also be used for sensitivity analysis. We cam propagate measurement or knowledge uncertainties through the model, yielding uncertainties on the model prediction.

Differentiable acoustic propagation modeling is still in its infancy. Much remains to be done in terms of improving model accuracy, computational performance, differentiability with respect to all parameters, and compatibility with the scientific machine learning software ecosystem. Normal mode models can also be made differentiable.

The purpose of this paper was to introduce an open-source tool, and demonstrate some ways in which it can be used. The hope is that this leads to exciting new techniques that leverage decades of understanding of how to model underwater acoustic propagation, as well as modern developments in the area of machine learning, Bayesian inference and automatic differentiation. Being an open-source project, we also hope to attract researchers and developers who may be willing to contribute to further development and testing of the tool.

## References


[1] F. B. Jensen, W. A. Kuperman, M. B. Porter, H. Schmidt, and A. Tolstoy, *Computational ocean acoustics*, vol. 794. Springer, 2011.

[2] N. R. Chapman, "Inverse problems in underwater acoustics," in *Handbook of signal processing in acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. New York, NY: Springer New York, 2008, pp. 1723–1735. doi: 10.1007/978-0-387-30441-0_95.

[3] A. B. Baggeroer and W. A. Kuperman, "Matched field processing in ocean acoustics," in *Acoustic signal processing for ocean exploration*, J. M. F. Moura and I. M. G. Lourtie, Eds. Dordrecht: Springer Netherlands, 1993, pp. 79–114. doi: 10.1007/978-94-011-1604-6_8.

[4] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[5] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 2008–2026, 2018.

[6] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[7] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo," *arXiv preprint arXiv:1701.02434*, 2017.

[8] M. D. Hoffman, A. Gelman, *et al.*, "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo." *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, 2014.

[9] C. C. Margossian, "A review of automatic differentiation and its efficient implementation," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 9, no. 4, p. e1305, 2019.

[10] M. B. Porter, "The BELLHOP manual and user's guide: Preliminary draft," *Heat, Light, and Sound Research, Inc., La Jolla, CA, USA, Tech. Rep*, vol. 260, 2011.

[11] M. Giordano, "Uncertainty propagation with functionally correlated quantities," *ArXiv e-prints*, Oct. 2016, Available: https://arxiv.org/abs/1610.08716

[12] H. Ge, K. Xu, and Z. Ghahramani, "Turing: A language for flexible probabilistic inference," in *International conference on artificial intelligence and statistics, AISTATS 2018, 9-11 april 2018, playa blanca, lanzarote, canary islands, spain*, 2018, pp. 1682–1690. Available: http://proceedings.mlr.press/v84/ge18b.html

[13] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, "Automatic differentiation variational inference," *Journal of machine learning research*, 2017.

[14] M. Chitre, "UnderwaterAcoustics: Julia toolbox for underwater acoustic modeling," *GitHub repository*. https://github.com/org-arl/UnderwaterAcoustics.jl; GitHub, 2022.

[15] M. Chitre, "AcousticsToolbox: Julia wrapper to the OALIB acoustic propagation modeling toolbox," *GitHub repository*. https://github.com/org-arl/AcousticsToolbox.jl; GitHub, 2022.

[16] M. Chitre, "A high-frequency warm shallow water acoustic communications channel model and measurements," *The Journal of the Acoustical Society of America*, vol. 122, no. 5, pp. 2580–2586, 2007.

[17] R. E. Francois and G. R. Garrison, "Sound absorption based on ocean measurements. Part II: Boric acid contribution and equation for total absorption," *The Journal of the Acoustical Society of America*, vol. 72, no. 6, pp. 1879–1890, 1982.

[18] L. M. Brekhovskikh, Y. P. Lysanov, and J. P. Lysanov, *Fundamentals of ocean acoustics*. Springer Science & Business Media, 2003.

[19] D. Jackson, "APL-UW high-frequency ocean environmental acoustic models handbook," *Applied Physics Laboratory, University of Washington, Technical Report*, vol. 9407, no. 102, pp. 1499–1510, 1994.

[20] J. Revels, M. Lubin, and T. Papamarkou, "Forward-mode automatic differentiation in Julia," *arXiv:1607.07892 [cs.MS]*, 2016, Available: https://arxiv.org/abs/1607.07892

[21] M. Innes *et al.*, "A differentiable programming system to bridge machine learning and scientific computing." 2019. Available: https://arxiv.org/abs/1907.07587

[22] M. Chitre, "RaySolver: A differentiable Gaussian beam tracer," *GitHub repository*. https://github.com/org-arl/AcousticRayTracers.jl; GitHub, 2022.

[23] L. Collatz, *The numerical treatment of differential equations*, vol. 60. Springer Science & Business Media, 2012.

[24] H. Rosenbrock, “Some general implicit processes for the numerical solution of differential equations,” *The Computer Journal*, vol. 5, no. 4, pp. 329–330, 1963.

[25] F. C. White *et al.*, “JuliaDiff/FiniteDifferences.jl: v0.12.26.” Zenodo, Jan. 2023. doi: 10.5281/zenodo.7516940.

[26] A. D. Gordon, T. A. Henzinger, A. V. Nori, and S. K. Rajamani, “Probabilistic programming,” in *Future of software engineering proceedings*, 2014, pp. 167–181.