# Acoustic raytracing comparisons in the context of geodetic precise off-shore positioning experiments : Supplemental Materials

Pierre Sakic[1], Valérie Ballu[1], Wayne Crawford[2], and Guy Wöppelmann[1]

[1]Laboratoire Littoral Environnement et Sociétés, UMR7266 – Univ La Rochelle and CNRS, 17000 La Rochelle, France

[2]Institut de Physique du Globe de Paris, Sorbonne Paris Cité, UMR7154 – CNRS, 75005 Paris, France

February 7, 2018

**Keywords :** Absolute position, Acoustic ranging, Plate motion, Geodesy, Seafloor positioning, Sound speed profile

Here are presented in details mathematical formulations and algorithms for the eikonal and Snell-Descartes ray tracing methods.

## 1 Eikonal equation

### 1.1 Formulation

The fundamental relation of underwater acoustics is the eikonal equation, linking the wavefront of the emitted rays set with an equal phase $\phi$, thus an equal propagation time $\tau$.

$$|\nabla\phi|^2 = \frac{1}{c^2(\mathbf{x})} \tag{1}$$

$s$ is the arc length of the ray. By definition, $\nabla\phi$ is perpendicular to the wavefront, so :

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} = c\;\nabla\phi \tag{2}$$

Then we differentiate this expression along $s$ for each component $x_i$ of $\mathbf{x}$.

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}x_i}{ds}\right) = \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial\phi}{\partial x_i}\right) \tag{3}$$

$$= \sum_j \frac{\mathrm{d}x_j}{\mathrm{d}s}\cdot\frac{\partial}{\partial x_j}\left(\frac{\partial\phi}{\partial x_i}\right) \tag{4}$$

$$= \sum_j c\frac{\partial\phi}{\partial x_j}\cdot\frac{\partial}{\partial x_i}\left(\frac{\partial\phi}{\partial x_j}\right) \tag{5}$$

$$= c\sum_j \frac{1}{2}\frac{\partial}{\partial x_i}\left(\frac{\partial\phi}{\partial x_j}\right)^2 \tag{6}$$

$$= \frac{c}{2}\frac{\partial}{\partial x_i}\sum_j\left(\frac{\partial\phi}{\partial x_j}\right)^2 \tag{7}$$

$$= \frac{c}{2}\frac{\partial}{\partial x_i}\left(\frac{1}{c}\right)^2 \tag{8}$$

$$= -\frac{1}{c^2}\frac{\partial c}{\partial x_i} \tag{9}$$

So we have for each component :

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}x}{\mathrm{d}s}\right) = -\frac{1}{c^2}\frac{\partial c}{\partial x} \qquad \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}y}{\mathrm{d}s}\right) = -\frac{1}{c^2}\frac{\partial c}{\partial y} \qquad \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}z}{\mathrm{d}s}\right) = -\frac{1}{c^2}\frac{\partial c}{\partial z} \tag{10}$$

Or in a vector notation :

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s}\right) = -\frac{1}{c^2}\nabla c \tag{11}$$

We substitute the sound speed $c(\mathbf{x})$ by his reciprocal, the sound slowness $\sigma(\mathbf{x})$ so as $\sigma(\mathbf{x}) = \dfrac{1}{c(\mathbf{x})}$. We can also define the sound slowness as a vector $\boldsymbol{\sigma} = \begin{bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\end{bmatrix}$. Because $\sigma = \dfrac{\mathrm{d}t}{\mathrm{d}s}$ and $\sigma_{x_i} = \dfrac{\mathrm{d}t}{\mathrm{d}x_i}$, we have $\dfrac{\sigma_{x_i}}{\sigma} = \dfrac{\mathrm{d}x_i}{\mathrm{d}s}$.

Using equation (10), we have :

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\sigma\frac{\mathrm{d}x}{\mathrm{d}s}\right) = \frac{\partial\sigma}{\partial x} \qquad \frac{\mathrm{d}}{\mathrm{d}s}\left(\sigma\frac{\mathrm{d}y}{\mathrm{d}s}\right) = \frac{\partial\sigma}{\partial y} \qquad \frac{\mathrm{d}}{\mathrm{d}s}\left(\sigma\frac{\mathrm{d}z}{\mathrm{d}s}\right) = \frac{\partial\sigma}{\partial z} \tag{12}$$

And by the following substitution :

$$\frac{\mathrm{d}\sigma_x}{\mathrm{d}s} = \frac{\partial\sigma}{\partial x} \qquad \frac{\mathrm{d}\sigma_y}{\mathrm{d}s} = \frac{\partial\sigma}{\partial y} \qquad \frac{\mathrm{d}\sigma_z}{\mathrm{d}s} = \frac{\partial\sigma}{\partial z} \tag{13}$$

The final set of differential equations to be solved is :

$$\frac{\mathrm{d}x}{\mathrm{d}s} = c(s)\sigma_x(s) = \frac{\sigma_x(s)}{\sigma(s)} \qquad \frac{\mathrm{d}\sigma_x}{\mathrm{d}s} = -\frac{1}{c^2}\frac{\partial c}{\partial x} = \frac{\partial \sigma}{\partial x} \tag{14}$$

$$\frac{\mathrm{d}y}{\mathrm{d}s} = c(s)\sigma_y(s) = \frac{\sigma_y(s)}{\sigma(s)} \qquad \frac{\mathrm{d}\sigma_y}{\mathrm{d}s} = -\frac{1}{c^2}\frac{\partial c}{\partial y} = \frac{\partial \sigma}{\partial y} \tag{15}$$

$$\frac{\mathrm{d}z}{\mathrm{d}s} = c(s)\sigma_z(s) = \frac{\sigma_z(s)}{\sigma(s)} \qquad \frac{\mathrm{d}\sigma_z}{\mathrm{d}s} = -\frac{1}{c^2}\frac{\partial c}{\partial z} = \frac{\partial \sigma}{\partial z} \tag{16}$$

And can be simplified in the vector form :

$$\frac{\mathrm{d}\mathbf{Y}}{\mathrm{d}s} = \mathbf{f}\left(s, \mathbf{Y}\right) \tag{17}$$

$$\mathbf{Y} = \begin{bmatrix} x \\ y \\ z \\ \sigma_x \\ \sigma_y \\ \sigma_z \end{bmatrix} \tag{18}$$

and

$$\mathbf{f} = \frac{\mathrm{d}\mathbf{Y}}{\mathrm{d}s} = \begin{bmatrix} \frac{\mathrm{d}x}{\mathrm{d}s} \\ \frac{\mathrm{d}y}{\mathrm{d}s} \\ \frac{\mathrm{d}z}{\mathrm{d}s} \\ \frac{\mathrm{d}\sigma_x}{\mathrm{d}s} \\ \frac{\mathrm{d}\sigma_y}{\mathrm{d}s} \\ \frac{\mathrm{d}\sigma_z}{\mathrm{d}s} \end{bmatrix} = \begin{bmatrix} \frac{\sigma_x}{\sigma} \\ \frac{\sigma_y}{\sigma} \\ \frac{\sigma_z}{\sigma} \\ \frac{\partial \sigma}{\partial x} \\ \frac{\partial \sigma}{\partial y} \\ \frac{\partial \sigma}{\partial z} \end{bmatrix} = \begin{bmatrix} c\ \sigma_x \\ c\ \sigma_y \\ c\ \sigma_z \\ -\frac{1}{c^2}\frac{\partial c}{\partial x} \\ -\frac{1}{c^2}\frac{\partial c}{\partial y} \\ -\frac{1}{c^2}\frac{\partial c}{\partial z} \end{bmatrix} \tag{19}$$

## 1.2 Solving Methods

Here are described the different methods used to solve the set of differential equations above.

### 1.2.1 Euler method

The Euler Method is the simplest ODE resolution method existing. With a stepwise approach, new values at point $s+h$ can be obtained using values and their derivatives at point $s$, where $h$ is the step of integration :

$$\mathbf{Y}_{i+1}(s+h) = \mathbf{Y}_i(s) + h \cdot \frac{\mathrm{d}\mathbf{Y}_i}{\mathrm{d}s} \tag{20}$$

$$\begin{bmatrix} x_{i+1}(s+h) \\ y_{i+1}(s+h) \\ z_{i+1}(s+h) \\ \sigma_{x,i+1}(s+h) \\ \sigma_{y,i+1}(s+h) \\ \sigma_{z,i+1}(s+h) \end{bmatrix} = \begin{bmatrix} x_i(s) \\ y_i(s) \\ z_i(s) \\ \sigma_{x,i}(s) \\ \sigma_{y,i}(s) \\ \sigma_{z,i}(s) \end{bmatrix} + h \cdot \begin{bmatrix} c(s)\ \sigma_{x,i}(s) \\ c(s)\ \sigma_{y,i}(s) \\ c(s)\ \sigma_{z,i}(s) \\ -\frac{1}{c(s)^2}\frac{\partial c(s)}{\partial x} \\ -\frac{1}{c(s)^2}\frac{\partial c(s)}{\partial y} \\ -\frac{1}{c(s)^2}\frac{\partial c(s)}{\partial z} \end{bmatrix} \tag{21}$$

### 1.2.2 Runge-Kutta 4 (RK4) method

The main idea of Runge-Kutta method is to dispatch the points where $\mathbf{f}$ is evaluated between $s$ and $s+h$ to gain in precision. The method is called a fourth-order one, meaning that the total accumulated error is order $O(h^4)$.

Knowing $\mathbf{Y}_i$ at point $s$, values $\mathbf{Y}_{i+1}$ at point $s+h$ may be determined using the following formulas (Butcher, 1963) :

$$\mathbf{Y}_{i+1}(s+h) = \mathbf{Y}_i(s) + \frac{h}{6} \cdot (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{22}$$

where :

$$\mathbf{k}_1 = \frac{\mathrm{d}\mathbf{Y}_i(s)}{\mathrm{d}s} \tag{23}$$

$$\mathbf{k}_2 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_2}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_2} = \mathbf{Y}_i(s) + \frac{h}{2} \cdot \mathbf{k}_1 \tag{24}$$

$$\mathbf{k}_3 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_3}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_3} = \mathbf{Y}_i(s) + \frac{h}{2} \cdot \mathbf{k}_2 \tag{25}$$

$$\mathbf{k}_4 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_3}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_4} = \mathbf{Y}_i(s) + h \cdot \mathbf{k}_3 \tag{26}$$

We also implement the simpler Order 2 Runge-Kutta Method, in order to compare the differences between both methods.

### 1.2.3 Adaptive Runge Kutta Integration, the Fehlberg method

The aim of an adaptive method is to validate the stability of the ODE solving, by estimating two forward steps with a different order, and use them to find the optimal step for the next iteration. It allows a stable final result with a minimum number of iterations, thus a reduced processing time.

Here we describe the Runge-Kutta-Fehlberg (RKF45) (Fehlberg, 1969), which estimates a forward step $\mathbf{Y}_{i+1}$ of order 4, and a second one $\tilde{\mathbf{Y}}_{i+1}$ of order 5, along with an optimized step.

We have :

$$\mathbf{Y}_{i+1}(s+h) = \mathbf{Y}_i(s) + h \cdot \left(\frac{25}{216}\mathbf{k}_1 + \frac{1408}{2565}\mathbf{k}_3 + \frac{2197}{4104}\mathbf{k}_4 - \frac{1}{5}\mathbf{k}_5\right) \tag{27}$$

$$\tilde{\mathbf{Y}}_{i+1}(s+h) = \mathbf{Y}_i(s) + h \cdot \left(\frac{16}{135}\mathbf{k}_1 + \frac{6656}{12825}\mathbf{k}_3 + \frac{28561}{56430}\mathbf{k}_4 - \frac{9}{50}\mathbf{k}_5 - \frac{2}{55}\mathbf{k}_6\right) \tag{28}$$

where :

$$\mathbf{k}_1 = \frac{\mathrm{d}\mathbf{Y}_i(s)}{\mathrm{d}s} \tag{29}$$

$$\mathbf{k}_2 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_2}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_2} = \mathbf{Y}_i(s) + \frac{\mathbf{k}_1}{4} \tag{30}$$

$$\mathbf{k}_3 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_3}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_3} = \mathbf{Y}_i(s) + \frac{3}{32}\mathbf{k}_1 + \frac{9}{32}\mathbf{k}_2 \tag{31}$$

$$\mathbf{k}_4 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_4}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_4} = \mathbf{Y}_i(s) + \frac{1932}{2197}\mathbf{k}_1 - \frac{7200}{2197}\mathbf{k}_2 + \frac{7296}{2197}\mathbf{k}_3 \tag{32}$$

$$\mathbf{k}_5 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_5}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_5} = \mathbf{Y}_i(s) + \frac{439}{216}\mathbf{k}_1 - 8\mathbf{k}_2 + \frac{3680}{513}\mathbf{k}_3 - \frac{845}{4104}\mathbf{k}_4 \tag{33}$$

$$\mathbf{k}_6 = \mathbf{f}\left(\mathbf{Y}_{\mathbf{k}_6}\right) \text{ with } \mathbf{Y}_{\mathbf{k}_6} = \mathbf{Y}_i(s) - \frac{8}{27}\mathbf{k}_1 + 2\mathbf{k}_2 - \frac{3544}{2565}\mathbf{k}_3 + \frac{1859}{4104}\mathbf{k}_4 - \frac{11}{40}\mathbf{k}_5 \tag{34}$$

the step $h$ is adapted using the following algorithm. We first determine two coefficients $R$ and $\delta$ :

$$R = \frac{1}{h}|\mathbf{Y}_{i+1} - \tilde{\mathbf{Y}}_{i+1}| \tag{35}$$

$$\delta = 0.84\left(\frac{\epsilon}{R}\right)^{1/4} \tag{36}$$

If $R \leq \epsilon$ we keep $\tilde{\mathbf{Y}}_{i+1}$ as the step solution, and the next solution will be determined using the step size $\delta \cdot h$.

If $R > \epsilon$ another step solution shall be determined, using $\delta \cdot h$ as step size.

We have also implemented the Cash-Karp method (RKCK) (Cash and Karp, 1990) and Dormand-Prince method (RKDP) (Dormand and Prince, 1986), based on the same principle, but with different coefficients, in order to compare the effective differences between those adaptive methods.

### 1.3 Implementation of the eikonal method direct problem solver

The implementation of a direct problem solver for an eikonal ray tracing is described in algorithm 1.

## 2 Snell-Descartes's Law

With the hypothesis of a horizontally layered ocean like described in section 2 of the article, meaning that there is only a sound speed gradient along the $z$ axis, the problem may be simplified using a radial symmetry, so the planimetrics components $x$ and $y$ becomes equivalent to a radial component $r$. Hence, the azimuthal angle $\phi$ becomes superfluous, thus the propagation direction can only be described with a unique dip angle $\theta(s)$.

### 2.1 Formulation

Starting from the equations 10, we have :

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}r}{\mathrm{d}s}\right) = -\frac{1}{c^2}\frac{\partial c}{\partial r} \tag{37}$$

If there is no variation of sound speed along $r$, $\dfrac{\partial c}{\partial r} = 0$, so :

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}r}{\mathrm{d}s}\right) = 0 \tag{38}$$

$$\frac{1}{c}\frac{\mathrm{d}r}{\mathrm{d}s} = k \text{ constant} \tag{39}$$

$$\frac{\cos\theta(s)}{c(s)} = k \text{ constant for each point of the path} \tag{40}$$

This relation, linking the incident angle with the sound speed is known as the Snell-Descartes law.

The constant $k$, called the ray parameter, only depends on the sound speed in the water column and the initial emission angle, and we have $k = \dfrac{\cos\theta(0)}{c(0)}$. This approach is used in the GNSS/A technique (Chadwell and Sweeney, 2010).

**Algorithm 1** Algorithm used to solve the eikonal method direct problem

$i \leftarrow 0$
$s_0 \leftarrow 0$
$h_{new} \leftarrow h_0$
$\mathbf{Y}_0 = \left[x_S, y_S, z_S, \frac{\cos\theta_0 \cos\phi_0}{c_0}, \frac{\cos\theta_0 \sin\phi_0}{c_0}, \frac{\sin\theta_0}{c_0}\right]$
*end of path* $\leftarrow$ **false**
*border zone* $\leftarrow$ **false**
*change direction* $\leftarrow$ **false**
**while not** *end of path* **do**
  **if** $s_i + h_{new} \geq s_{max}$ **then**
    $h_i = \|s_{max} - s_i\|$
  **else if** *border zone* **then**
    $h_i = s_i$
  **else**
    $h_i = h_{new}$
  **end if**
  **if** *change direction* **then**
    *change direction* $\leftarrow$ **false**
    *border zone* $\leftarrow$ **false**
  **end if**
  *good step* $\leftarrow$ **false**
  **while not** *good step* **do**
    $\mathbf{Y}_{new}, h_{new} \leftarrow IntegrationFunction(\mathbf{Y}_i, s_i, h_i)$
    **if** $z_{min} < z_{new} < z_{max}$ **then**
      *good step* $\leftarrow$ **true**
      **break**
    **else if** *change direction* **then**
      *good step* $\leftarrow$ **true**
      **break**
    **else**
      *border zone* $\leftarrow$ **true**
      **if** $h_i = H[\text{last element}]$ **then**
        **if** $z_{new} > z_{max}$ **then**
          $h_i = \dfrac{z_{max} - z_i}{c_i \sigma_z}$
          *change direction* $\leftarrow$ **true**
        **else if** $z_{new} > z_{max}$ **then**
          $h_i = \dfrac{z_{min} - z_i}{c_i \sigma_z}$
          *change direction* $\leftarrow$ **true**
        **end if**
      **else if** $h_i$ **in** $H[\text{last element}]$ **then**
        $h_i \leftarrow H[\text{next element}]$
      **else**
        $h_i \leftarrow H[\text{first element}]$
      **end if**
    **end if**
  **end while**
  $\mathbf{Y}_{i+1} \leftarrow \mathbf{Y}_{new}$
  $s_{i+1} \leftarrow s_i + h_i$
  **if** $\|s_{i+1} - s_{max}\| < 10^{-6}$ **then**
    *end of path* $\leftarrow$ **true**
  **end if**
  $i \leftarrow i + 1$
**end while**

**Direct problem** In the case of a discretized SSP $c(z)$, the water column may be divided in layers of depth $[z_i, z_{i+1}]$. In each layer $i$ a sound speed gradient may be defined.

$$g(z) = \frac{\mathrm{d}c(z)}{\mathrm{d}z} = \frac{c_{i+1} - c_i}{z_{i+1} - z_i} \tag{41}$$

In the layer $i$, the sound speed variation is $i$ :

$$c(z) = c_i + g(z - z_i) \tag{42}$$

The equation 41 leads directly to the relation :

$$\mathrm{d}c = g\,\mathrm{d}z \tag{43}$$

And differencing the relation 40 gives :

$$\mathrm{d}c = -\frac{\sin\theta}{k}\,\mathrm{d}\theta \tag{44}$$

### 2.1.1 Ray progression into a layer

Taking account of the identity $\sin^2 x + \cos^2 x = 1$, we have :

$$\sin\theta = \sqrt{1 - k^2c^2} \tag{45}$$

$$\cot\theta = \frac{k^2c^2}{\sqrt{1 - k^2c^2}} \tag{46}$$

The propagation time in the layer $i$ is :

$$\begin{aligned}
\Delta\tau_i = \tau_{i+1} - \tau_i &= \int_{z_i}^{z_{i+1}} \frac{\mathrm{d}s}{c} && (47)\\
&= \int_{z_i}^{z_{i+1}} \frac{\mathrm{d}z}{c(z)\sin\theta(z)} && (48)\\
&= \int_{z_i}^{z_{i+1}} \frac{\mathrm{d}z}{c(z)\sqrt{1 - k^2c^2(z)}} && (49)\\
&= \boxed{\frac{1}{|g_i|}\ln\left[\frac{c(z_{i+1})}{c(z_i)}\,\frac{1 + \sqrt{1 - k^2c^2(z_i)}}{1 + \sqrt{1 - k^2c^2(z_{i+1})}}\right]} && (50)
\end{aligned}$$

The spatial propagation along the radial component $r$ is :

$$\begin{aligned}
\Delta r_i = r_{i+1} - r_i &= \int_{z_i}^{z_{i+1}} \cot\theta(z)\,\mathrm{d}z && (51)\\
&= \int_{z_i}^{z_{i+1}} \frac{kc\,\mathrm{d}z}{\sqrt{1 - k^2c^2(z)}} && (52)\\
&= \boxed{\frac{1}{kg_i}\left[\sqrt{1 - k^2c^2(z_i)} - \sqrt{1 - k^2c^2(z_{i+1})}\right]} && (53)
\end{aligned}$$

The length of the ray path $s$ in the layer $i$ is :

$$\Delta s_i = s_{i+1} - s_i = \int_{z_i}^{z_{i+1}} \frac{\mathrm{d}z}{\sin\theta(z)} \tag{54}$$

$$= \frac{c_i}{g_i \cos\theta_i}(\theta_i - \theta_{i+1}) \tag{55}$$

$$= \boxed{\frac{1}{kg_i}\left(\arccos\left(kc_i\right) - \arccos\left(kc_{i+1}\right)\right)} \tag{56}$$

If $k^2c^2(z_{i+1}) \geq 1$ (leading to $\cos\theta \geq 1$, which is impossible), then the direction of propagation is inverted along $z$, and the previous equations become (Hovem, 2013) :

$$\Delta t_i = \frac{2}{|g_i|} \ln\left[\frac{1+\sqrt{1-k^2c^2(z_i)}}{kc(z_i)}\right] \tag{57}$$

$$\Delta r_i = \frac{2}{kg_i}\left[\sqrt{1-k^2c^2(z_i)}\right] \tag{58}$$

Finally, the total propagation time, the total radial propagation distance, and the total path length are the sum of each parameter in each layer.

$$\tau = \sum_i^n \Delta\tau_i \qquad r = \sum_i^n \Delta r_i \qquad s = \sum_i^n \Delta s_i \tag{59}$$

If we use the propagation time $t_{max}$ as the stop parameter, we need to determine the sound speed $c_{\tau_n}$ in the last layer at $\tau_n$ to have $\tau = \sum_i^{n-1} \Delta\tau_i + \Delta\tau_n$. It is given by :

$$c_{\tau_n} = 2\,\frac{\mathrm{e}^{tg} c(z_i)\left(1+\sqrt{-\left(kc(z_i)-1\right)\left(kc(z_i)+1\right)}\right)}{k^2\left(\mathrm{e}^{tg}\right)^2 c^2(z_i) + 2 + 2\sqrt{-\left(kc(z_i)-1\right)\left(kc(z_i)+1\right)} - k^2c^2(z_i)} \tag{60}$$

We can also use a maximum depth $z_{max}$ as a stop parameter. In this case, the sound speed is "cut" at the depth $z_{max}$, the corresponding sound speed is given by $c(z_{max}) = g(z_{max}) \cdot (z_{max} - z_i)$, and a ray tracing is performed in the whole water column.

## 2.2 Implementation of the Snell-Descartes method direct problem solver

The implementation of a direct problem solver for a Snell-Descartes ray tracing is described in algorithm 2.

**Algorithm 2** Algorithm used to solve the Snell-Descartes method direct problem

$a \leftarrow \theta_{0,max,init} = 88°$
$b \leftarrow \theta_{0,min,init} = 0°$
$r_{stop} \leftarrow 10^9$
$\epsilon \leftarrow 10^9$
$\Delta r \leftarrow \|r_R - r_S\|$
**while** $\epsilon > \epsilon_{min}$ **do**
  $r_{R,a} \leftarrow f\left(r_S, \mathbf{C}, a, z_R\right)$
  $r_{R,b} \leftarrow f\left(r_S, \mathbf{C}, b, z_R\right)$
  $\delta r_{R,a} \leftarrow r_{R,a} - \Delta r$
  $\delta r_{R,b} \leftarrow r_{R,b} - \Delta r$
  $c \leftarrow a - \dfrac{a-b}{\delta r_{R,a} - \delta r_{R,b}} \cdot \delta r_{R,a}$
  $b \leftarrow a$
  $a \leftarrow c$
  $r_{stop} \leftarrow r_{R,a}$
  $\epsilon \leftarrow |\Delta r - r_{stop}|$
**end while**
$\theta_0 \leftarrow c$
$r_R, \tau, s \leftarrow f\left(r_S, \mathbf{C}, \theta_0, z_R\right)$

# 3 Comparison of the different eikonal ray tracing methods

## 3.1 Direct problem

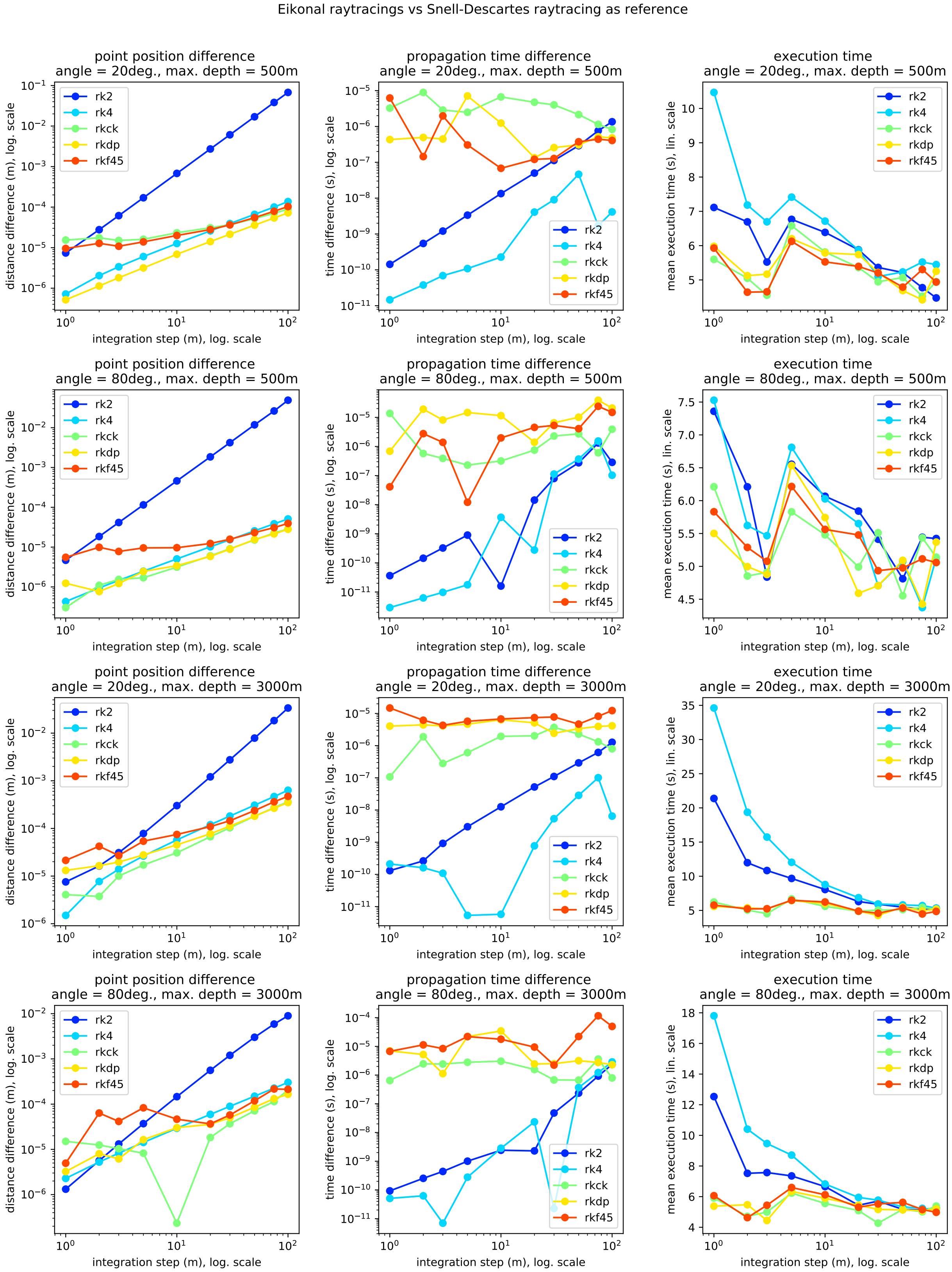


Figure 1: Direct ray tracing using the eikonal method compared to the Snell-Descartes one depending on integration steps $h$ and different initial parameters (emission angle equal to $40\,°$and $80\,°$, and maximum depth equal to $500\,\mathrm{m}$ and $3000\,\mathrm{m}$).

## 3.2 Inverse problem

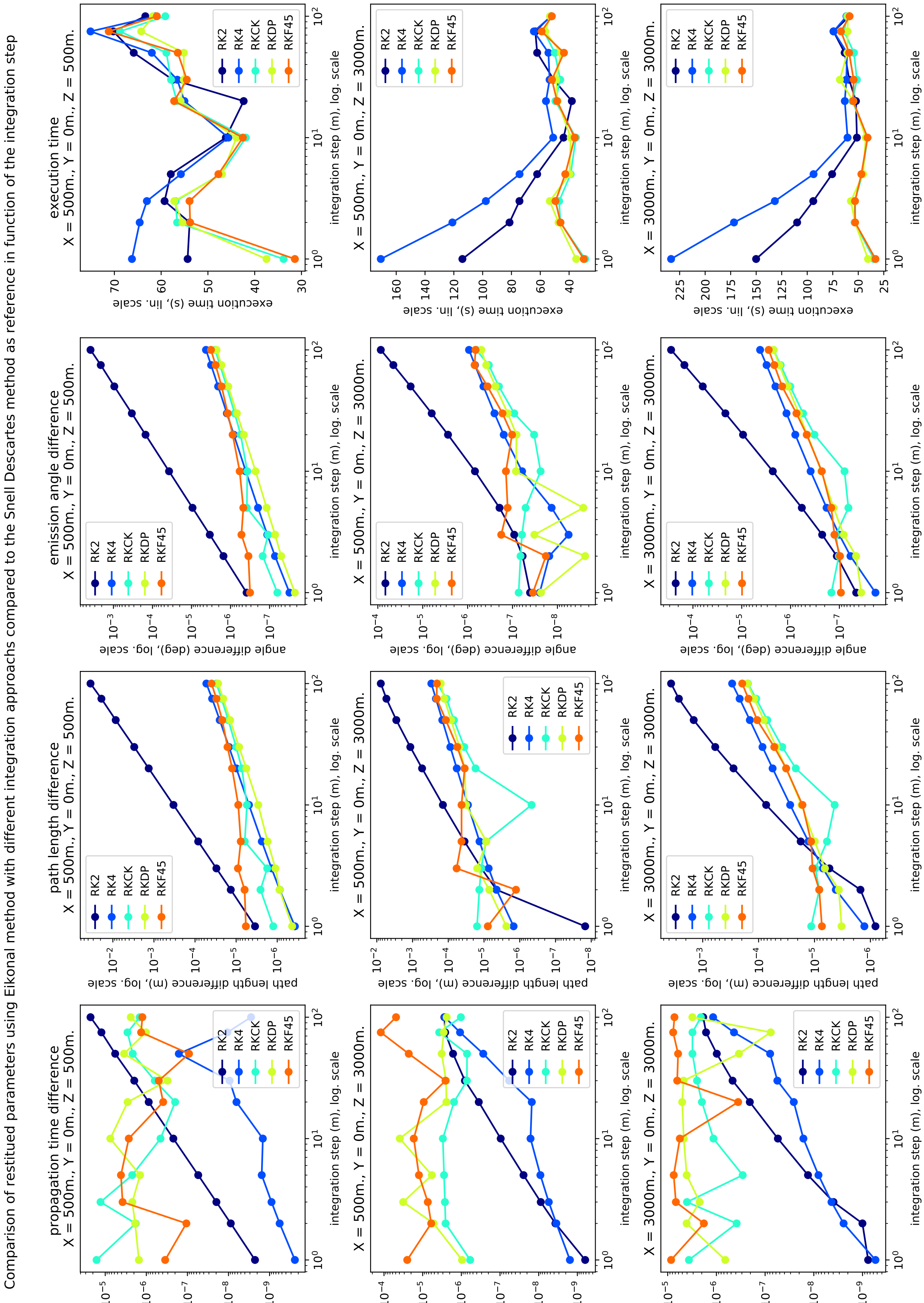


Figure 2: Inverse ray tracing using the eikonal method compared to the Snell-Descartes one depending on integration steps $h$ and different initial parameters ($x_R$ equal to 500 m and 3000 m, and $z_R$ equal 500 m to 3000 m).

## References


Butcher, J. C. (1963). Coefficients for the study of Runge-Kutta integration processes. *Journal of the Australian Mathematical Society*, 3(02):185.

Cash, J. and Karp, A. (1990). A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software*, 16(3):201–222.

Chadwell, C. D. and Sweeney, A. D. (2010). Acoustic Ray-Trace Equations for Seafloor Geodesy. *Marine Geodesy*, 33(2-3):164–186.

Dormand, J. R. and Prince, P. J. (1986). A reconsideration of some embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 15(2):203–211.

Fehlberg, E. (1969). Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems. *NASA technical report*, (July):R–315.

Hovem, J. M. (2013). Ray Trace Modeling of Underwater Sound Propagation. In Beghi, M. G., editor, *Modeling and Measurement Methods for Acoustic Waves and for Acoustic Microdevices*. InTech.