Hybrid minimizer: Internal Report

Arthur Adriaens

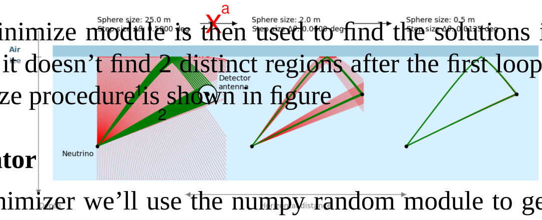
Ghent University

1. Introduction

Upon learning that more complex ice models where needed and after seeing the work that has been done to iteratively ray trace a path (Oeyen, 2022) , I checked out the source code to try and understand the workings. In the source code I saw that a clever but unsuccesful attempt was made to implement the scipy.optimize module as an alternative to the iterative ray tracer. I came up with a way to implement this that succeeded as will be explained below.

2. How it works

The hybrid minimizer can be seen as an extension of the iterative minimizer, it checks after the first loop (as explained in the paper by B. Oeyen et al (Oeyen, 2022). if there are 2 distinct launch regions, if this is the case it breaks out of the loop as visually explained using a modified version of B. Oeyen et al. their figure below:



[a] process is broken out of as 2 distinct launch regions (region 1 & 2) are found.

The scipy.optimize.minimize module is then used to find the solutions in the respective angle intervals, as illustrated below: If it doesn't find 2 distinct regions after the first loop, it falls back on the iterative ray tracer. The scipy optimize procedure is shown in figure

3. random number generator

To test the hybrid minimizer we'll use the numpy random module to generate the random numbers, the considered square (as there is only a z component to the ice model the 3D problem is essentially only a 2D problem) is x:-4km,+4km and z:0,-3km. A good test to see if the generator is both random and uniform is to plot the next element to the previous element, shown in figure for the generated z coordinates, in figure for the generated x coordinates. This clearly is a good random number generator and is the one we'll be using for the testing of the hybrid ray tracer. As a counter-example, a bad random number generator's expected output is shown in figure

4. Performance Pre-Optimalisation

Prior to doing any kind of optimalisation a comparison of the accuracy is shown in figures and . Besides the better accuracy, the hybrid ray tracer is also faster with approximately a 15 increase. Of course there are some parameters that can be tweaked making this percentage go up as explained in the next section.

5. Performance Optimalisation

5.1.

Length of the normal vector As visually explained in figure , the size of the normal vector influences how big the ray tracer's step size is taken close to the detector. This thus influences the convergence and time taken. The results of varying this are shown in figures and . The first optimization conclusion is thus: take the normal vector length to be 1 meter.

5.2. ztol

We'll now change the tolerence on the vertical distance away from the detector which is deemed accepted i.e in figure Δz is below this threshold it's accepted. The results are shown in figures and . From which we can conclude the second optimization conclusion: take ztol to be 0.05 m.

5.3. Sphere Size & Step Size

As explained in Oeyen et al.'s work, the initial rays are sent out in steps of a certain angle and with a sphere around the detector (as can also be seen in figure , but for clarification I again refer to their paper). The sphere size and step size haven't jet been optimized however, as this is the slowest step in the hybrid ray tracer this was optimized here (only the initial sphere and step size as those are relevant for the hybrid raytracer).

References

Oeyen, 2022.

Oeyen, B, Plaisier, I, Nelles, A., Glaser, C., and Winchen, T., *Effects of firn ice models on radio neutrino simulations using a RadioPropa ray tracer,* 37th International Cosmic Ray Conference. 12-23 July 2021. Berlin (2022).