FAU-ECAP-DIPL-2018-001

The Effect of Hole Ice on the Propagation and Detection of Light in IceCube

Diplomarbeit aus der Physik

Vorgelegt von cand. rer. nat.
Sebastian Fiedlschuster

September 2018

Friedrich-Alexander-Universität Erlangen-Nürnberg

Physikalisches Institut

Prof. Dr. Uli Katz

Dr. Thorsten Glüsenkamp

The Effect of Hole Ice on the Propagation and Detection of Light in IceCube

Sebastian Fiedlschuster

September 2018

Erlangen Centre For Astroparticle Physics

Friedrich-Alexander-Universität Erlangen-Nürnberg

Prof. Dr. Uli Katz

Dr. Thorsten Glüsenkamp

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

IceCube South Pole Neutrino Observatory

Printed in Erlangen, Germany.

The LaTeX version of this document can be found online at https://github.com/fiedl/hole-ice-latex.

Additional material can be found at https://github.com/fiedl/hole-ice-study.

Abstract

IceCube is a neutrino observatory at Earth's South Pole that uses glacial ice as detector medium. Secondary particles from neutrino interactions produce Cherenkov light, which is detected by an array of photo detectors deployed within the ice. In distinction from the glacial bulk ice, hole ice is the refrozen water in the drill holes around the detector modules, and is expected to have different optical properties than the bulk ice. Aiming to improve detector precision, this study presents a new method to simulate the propagation of light through the hole ice, introducing several new calibration parameters. The validity of the method is supported by a series of statistical cross checks, and by comparison to measurement and simulation results from other calibration studies. Evaluating calibration data indicates a strongly asymmetric shielding of the detector modules. A preliminary analysis suggests that this cannot be accounted for by the shadow of cables, but can be explained by hole ice with a suitable scattering length, size, and position relative to the detector modules. The hole-ice approximation, which is used in the standard simulation chain is found to disagree with all existing direct-propagation methods and should be recalculated with a new direct-simulation run.

Der Einfluss von Loch-Eis auf die Ausbreitung und Detektion von Licht im IceCube-Neutrino-Observatorium

Sebastian Fiedlschuster

September 2018

Zusammenfassung

Das IceCube-Neutrino-Observatorium am Südpol verwendet das Eis eines Gletschers als Detektor-Medium, in dem Teilchen aus Neutrino-Reaktionen auf ihrem Weg durch das Eis Licht erzeugen, das von Photo-Detektoren im Eis registriert wird. Loch-Eis (engl. „hole ice“) ist das erneut gefrorene Wasser in den Bohrlöchern, in denen die Detektor-Module in das Eis eingelassen wurden, und hat voraussichtlich andere optische Eigenschaften als das übrige Eis des Gletschers.

Um die Detektor-Kalibrierung und damit die Genauigkeit von IceCube zu verbessern, stellt diese Arbeit neue Algorithmen vor, mit denen die Propagation von Photonen durch Loch-Eis mit verschiedenen Eigenschaften simuliert werden kann. Die Tauglichkeit der Algorithmen wird durch eine Reihe von statistischen Überprüfungen sowie durch einen Vergleich mit Messungen und Simulationen anderer Kalibrierungsstudien untermauert.

Als Anwendungsbeispiele werden in dieser Arbeit die Simulation eines oder mehrerer Loch-Eis-Zylinder mit unterschiedlichen Eigenschaften, etwa der Lage, der Größe, der Absorptions- und der Streulänge des Loch-Eises, die Simulation eines lichtabsorbierenden Kabels sowie die beispielhafte Kalibrierung anhand von Daten des Leuchtdioden-Kalibrierungs-Systems von IceCube durchgeführt.

Die großräumige Ausbreitung von Licht durch das Eis des Gletschers erfolgt nahezu unbeeinflusst von den Eigenschaften des Loch-Eises. Allerdings muss jedes Photon, das von IceCube registriert oder vom Kalibrierungssystem abgegeben wird, zunächst durch das Loch-Eis gelangen, sodass sowohl die Detektor-Einheiten als auch die Kalibrierungs-Leuchtdioden in Abhängigkeit von den Eigenschaften des Loch-Eises effektiv abgeschirmt werden, da ein Teil des Lichtes vom Loch-Eis absorbiert, ein anderer Teil reflektiert werden. Für Detektor-Module, die im Loch-Eis nicht völlig zentrisch zum Liegen gekommen sind, ist dieser Effekt abhängig von der Azimuth-Richtung der Photonen, was sich auch in den Kalibrierungsdaten widerspiegelt.

Aus Kalibrierungsdaten ergibt sich, dass die Detektor-Module richtungsabhängig abgeschirmt werden. Vorläufige Ergebnisse dieser Simulations-Studie zeigen, dass die beobachtete Abschirmung nicht von Kabeln verursacht werden

kann, die an der Seite der Detektor-Module verlaufen, sondern die Annahme eines Loch-Eises geeigneter Lage, Größe und optischer Eigenschaften erforderlich ist, um die Kalibrierungsdaten zu erklären.

Die neuen Propagations-Algorithmen werden in Kürze in das Simulations-System von IceCube integriert werden. Studien, die eine geringe Statistik aufweisen, sich also mit Neutrino-Reaktionen befassen, die nur wenig Licht hervorrufen, oder, bei denen das Licht nur von wenigen Detektor-Modulen registriert wird, können mit den in dieser Arbeit vorgestellten Simulationsmethoden systematische Unsicherheiten verringern.

Da die direkte Simulation der Licht-Propagation durch das Loch-Eis jedoch zusätzliche Simulations-Laufzeit erfordert, ist es empfehlenswert, dass Studien mit einer hohen Statistik, die sich also mit Neutrino-Reaktionen befassen, die eine große Menge an Licht hervorrufen, das wiederum von einer Vielzahl von Detektor-Modulen registriert wird, den Einfluss des Loch-Eises durch bereits gebräuchliche Verfahren nur näherungsweise berücksichtigen. Die Parameter dieser Näherungsverfahren sollten jedoch aufgrund der Erkenntnisse aus direkten Simulationen korrigiert werden.

Contents

CD-ROM

For the contents of the CD-ROM, see appendix A.2.

1 Introduction

IceCube is a neutrino observatory located at Earth's South Pole. It uses a cubic-kilometer of glacial ice as detector medium where secondary particles from neutrino interactions produce light as they move through the ice. The light is detected by an array of photo detector modules that are deployed throughout the ice. [Aar+13]

The primary scientific objective of IceCube is the study of neutrinos with energies ranging from 10 TeV to 10 PeV produced in astrophysical processes, and the identification and characterization of their sources. In collaboration with other neutrino detectors as Antares, with optical, x-ray, gamma-ray, radio, and gravitational-wave observatories, IceCube participates in efforts for multi-messenger astronomy. Other objectives include the indirect detection of dark matter, the search for other exotic particles, and the study of neutrino-oscillation physics. [Aar+17b; Aar+13]

As IceCube detects neutrinos indirectly through the interaction with other particles, involving a chain of processes and components, a key requirement for precise measurements is to minimize uncertainties for each process and component involved. Some components such as technical instruments in the detector modules can be tested and calibrated in isolation in laboratories. Other components involved such as the glacial ice cannot be extracted and need to be studied where they are. Uncertainties concerning the properties of the glacial ice can affect the precision for measurements of the direction of the detected neutrinos by several percent. [Wre18]

All light that is detected by the detector modules needs to travel through the refrozen water of the drill holes that were needed to deploy the detector modules within the ice. This so-called *hole ice* may have properties significantly different from the surrounding bulk ice regarding the propagation of light through this medium. The properties of the hole ice are less known than the properties of the bulk ice and pose the largest systematic uncertainty for study of neutrino oscillations and a number of other analyses. [RRK17]

This study aims to provide the necessary tools to improve detector calibration by introducing the means to simulate the propagation of light through the hole ice. By comparing different simulation scenarios, involving hole ice of different respective properties, to calibration data, it is then possible to study the properties of the hole ice and its effect on the propagation of light, and on the detection of light by the detector modules, reducing the systematic uncertainties imposed by the hole ice, and in the long run improving the precision of the IceCube observatory.

After providing some background information in sections 2 to 4, two algorithms and their integration into the existing IceCube software framework will be presented

in section 5 that allow to simulate the direct propagation of photons through hole ice of different properties. The validity of the algorithms will be supported by a series of tests and cross checks in section 5.6.

Examples of application such as the simulation of one or several hole-ice cylinders with different sizes and photon scattering lengths, the simulation of shadowing cables, and a calibration method using LED flasher data are given in section 6. Section 7 presents a brief comparison of methods and preliminary results to other studies. This section also discusses performance considerations and lists ice properties not considered in this study.

The material needed to reproduce this study is provided on the accompanying CD-ROM and can be found online at https://github.com/fiedl/hole-ice-study.

2 Theoretical Concepts

2.1 Neutrinos

Neutrinos are particles that primarily interact with other particles through the *weak interaction* and have very small probabilities of interacting with other particles, allowing them to cross matter almost unhindered. While this makes them interesting messenger particles for observing far-distant astronomical sources and phenomena, because they are long ranging and arrive undeflected and unscattered, it also makes their detection challenging due to the large amount of detector medium required. [Lex98; Aar+17b]

Within the *standard model of particle physics* (see figure 1), the electron neutrino, ν_e, the muon neutrino, ν_μ, and the tau neutrino ν_τ, are *leptons* without electrical charge and couple to the $W^\pm$ and Z^0 bosons.

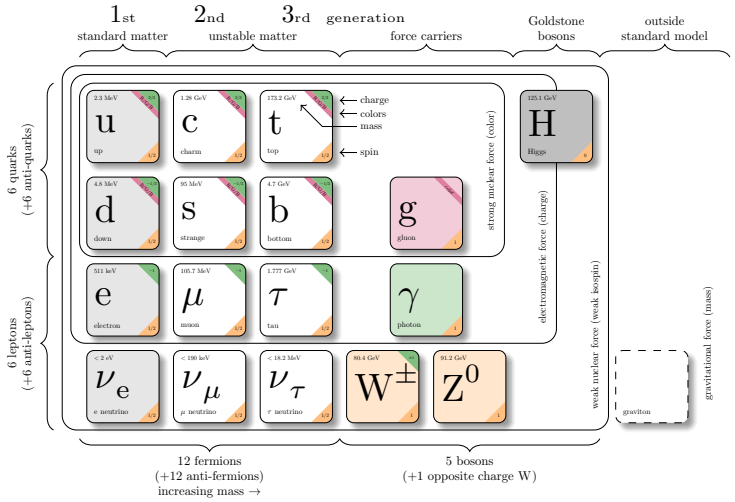


Figure 1: Particles of the standard model of particle physics. Together with the electron, the muon, and the tau, the neutrinos form the group of leptons, which do not participate in *strong interactions*. As neutrinos also do not participate in the *electromagnetic interaction*, their primary interaction channel is through *weak interactions*. Image based on: [Bur12]

As the *mass eigenstates* ν_i of neutrinos that describe neutrinos in the context of their propagation through spacetime, are not identical to the neutrino *flavor eigenstates* ν_ℓ that describe neutrinos participating in weak interactions, neutrinos are subject to quantum mechanical phenomena as *neutrino oscillations*. [GP16]

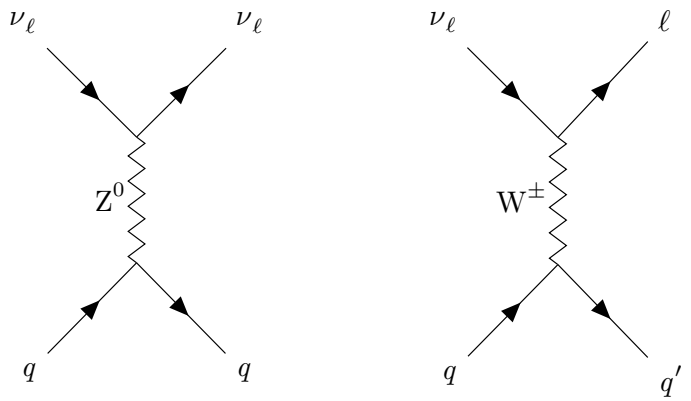
$$|\nu_\ell\rangle = \sum_{i=1}^{N} U_{\ell i} \, |\nu_i\rangle \,, \quad \ell \in \{\mathrm{e}, \mu, \tau\}$$

Determining the mixing matrix $U_{\ell i}$ that connects the mass eigenstates to the flavor eigenstates, is part of IceCube's scientific objectives.

Neutrinos generated by interactions of cosmic-ray particles in the Earth's atmosphere, *atmospherical neutrinos*, have energies in the GeV to TeV scale. *Astrophysical neutrinos*, generated by high-energy phenomena in the universe, have energies up to the PeV scale. [Aar+17b]

2.2 Neutrino Interactions Relevant to IceCube

The primary interaction channel for detecting neutrinos in IceCube is deep-inelastic scattering of neutrinos with quarks of nuclei in the detector material or nearby rock (see figure 2). [Aar+14]



(a) Neutral-current neutrino interaction through Z^0 bosons.

(b) Charged-current neutrino interactions through $\mathrm{W}^\pm$ bosons.

Figure 2: Feynman diagrams showing neutrinos ν_ℓ interacting with quarks q of nuclei of the ice or nearby rock through Z^0 and $\mathrm{W}^\pm$ bosons, producing leptons ℓ (electrons e, muons μ, tau particles τ) and quarks. Time evolves to the right in these diagrams.

These interactions lead to three different kind of event signatures in the IceCube detector, visualized in figure 3.

Shower- or Cascade-Like Events In both, interactions through Z^0 bosons (*neutral-current interactions*) and through $W^{\pm}$ bosons (*charged-current interactions*), hadronic showers are created as energy is transferred from the incoming neutrino to the outgoing quark. Hadronic showers are particle cascades originating from hadron particles. If the outgoing lepton is an electron, this may also create an accompanying electromagnetic shower, which is a particle cascade originating from particles primarily interacting through the electromagnetic interaction. [Aar+14]

$$\begin{aligned} \text{neutral current:} \quad \nu_\ell + \text{nucleon} &\to \nu_\ell + \text{hadron} \qquad \ell \in \{\text{e}, \mu, \tau\} \\ &\to \nu_\ell \text{ (escapes)} \; + \text{hadronic shower} \end{aligned}$$

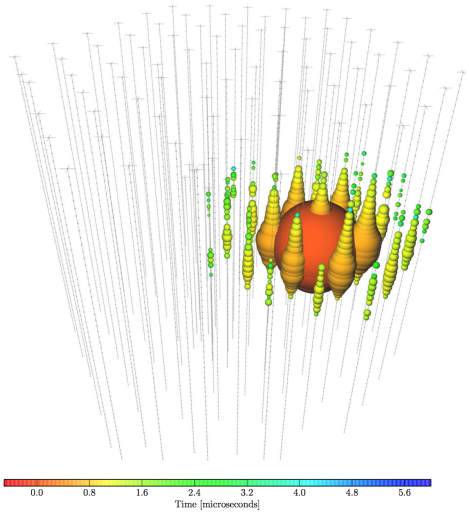
$$\begin{aligned} \text{charged current:} \quad \nu_\text{e} + \text{nucleon} &\to \text{e} + \text{hadron} \\ &\to \text{electromagnetic shower} + \text{hadronic shower} \end{aligned}$$

Track-Like Events If the outgoing lepton is a muon, this creates track-like event signatures as muons travel long distances. TeV muons may travel several kilometers in the antarctic ice. [Aar+17a; CR04]

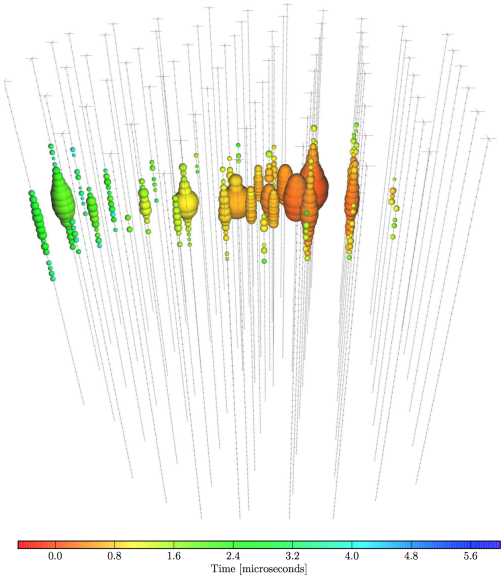
$$\begin{aligned} \text{charged current:} \quad \nu_\mu + \text{nucleon} &\to \mu + \text{hadron} \\ &\to \text{muon track} + \text{hadronic shower} \end{aligned}$$

Double-Bang Events If the outgoing lepton is a tau, the tau will create a track. But the track will only be a couple of meters long as the tau then decays into another hadronic shower. This creates a so-called *double-bang signature*, where two hadronic showers are joined by a short track. The shorter the joining track is, the more similar the event looks to a single-cascade-like event. [Aar+17a; Aar+14; GP16]

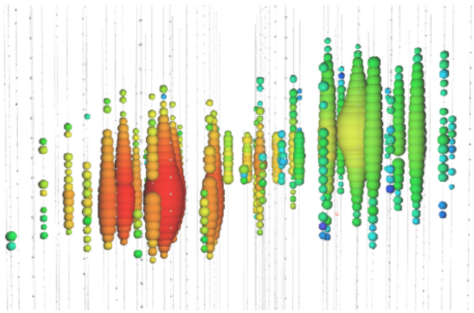
$$\begin{aligned} \text{charged current:} \quad \nu_\tau + \text{nucleon} &\to \tau + \text{hadron} \\ &\to \text{tau track} + \text{hadronic shower} \\ &\to \text{hadronic shower} + \text{hadronic shower} \end{aligned}$$



(a) Cascade-like event: The cascade is completely contained within the detector and deposits a total of 1141 TeV energy within the detector. Image and data source: [Aar+13; Aar+14]



(b) Track-like event: The muon track starts within the detector and deposits a total of 71 TeV of energy within the detector before it leaves the detector volume. Image and data source: [Aar+13; Aar+14]



(c) Simulated double-bang event: The earlier (red) cascade has been created from the primary neutrino interaction vertex. The tau travels from the position of the first cascade for a short distance, creating a track, before decaying into another later (green) cascade. Image source: [Arg18]

Figure 3: Visualization of examples for the different event signatures of neutrino events observed with IceCube. The spheres represent the energy registered by the respective detector module. The color indexes the time information: Red is the beginning of the event, green the middle, and blue the end of the event.

2.3 Cherenkov-Light Emission

When charged particles, both the primary lepton and particles within the cascades, move through the detector medium faster than the phase velocity of light within this medium, they emit so-called *Cherenkov radiation*, which are photons with wavelengths in the visible and the near ultra violet spectrum. [Aar+14; Aar+17b; Aar+17a]

The light emission is not isotropic: Photons are emitted under an angle ϕ relative to the direction of propagation of the charged particle. [Lex98]

$$\cos\phi = \frac{1}{\beta\, n} = \frac{c'}{v}, \quad \beta = \frac{v}{c}, \quad c' = \frac{c}{n}$$

n is the refractive index of the medium, c' the speed of light within the medium, c the speed of light in vacuum.

The virtual photon field of the charged particle moving through the medium polarizes the atoms in the medium. Each resulting electric dipole is a source of electromagnetic radiation. But as each dipole arranges towards the charged particle, integrating over the whole spatial sphere around the charged particle, the net emitted radiation vanishes if the charged particle's velocity v is smaller than the speed of light c' in the medium. For higher velocities, the Coulomb field of the charged particle can only polarize the atoms within a cone with an opening angle of 2ϕ, the *Cherenkov cone*, resulting in a net photon emission perpendicular to the surface of the cone. [Lex98]

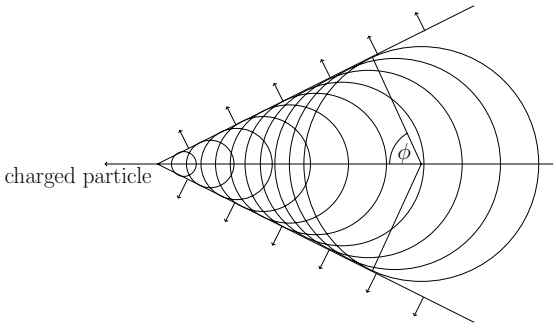


Figure 4: Visualization of the *Cherenkov cone*: A charged particle moves through a medium with a velocity $v > c'$ greater than the phase velocity c' of light within this medium. Due to non-isotropic polarization effects, *Cherenkov photons* are emitted under an angle ϕ relative of the direction of motion of the charged particle.

The spectral distribution of Cherenkov photons is given by equation 1. A particle with $\pm z$ elementary charges that travels a distance $\mathrm{d}x$ emits d^2N Cherenkov

photons with a wavelength range of $[\nu; \nu + \mathrm{d}\nu]$. α is the fine structure constant. [KS12] The spectrum prefers photons with shorter wavelengths ν.

$$\frac{\mathrm{d}^2 N}{\mathrm{d}x\,\mathrm{d}\nu} = \frac{2\pi\,\alpha\,z^2}{\nu^2} \cdot \left(1 - \frac{1}{\beta^2\,n^2}\right) \tag{1}$$

Per GeV of secondary particle shower energy within the detector, an order of 10^5 visible Cherenkov photons are created. [Aar+17b]

2.4 Photon Absorption and Scattering

The propagation of light through a medium depends on the optical properties of that medium, in particular the velocity of light within that medium, the scattering probability and the absorption probability. [Lun+07]

The absorption of light for the relevant wavelength range is caused by electronic and molecular excitation processes [Lun+07] and is quantified by the **absorption length** λ_{abs}, which is the mean of the exponentially distributed free path length to absorption [Lun+07]. Therefore, in accordance with the Beer-Lambert law, the absorption length is the path length that light needs to travel within a medium to have its intensity drop to $1/e$ of its original intensity. [Lex98] Absorption lengths in the South-Polar ice vary between 10 m in dusty regions and 280 m in very clear ice layers. [Ack+06; Chi14; Col+13]

The scattering of light off microscopic scattering centers, such as sub-millimeter-sized air bubbles and micron-sized dust grains [PB97; Ack+06] is the dominant scattering mechanism in glacial ice [Ask+97; Lun+07]. This scattering can be modeled using the more general Mie scattering theory, which describes the scattering of electromagnetic radiation off small, spherical masses of material with refractive indices differing from the refractive index of its surroundings. [Mie08; Ack+06; Lun+07]

Mie scattering gives the distribution of the scattering angle θ for any wavelength and scattering center size. For ice, this distribution is approximated using a one-parameter Henyey-Greenstein (HG) phase function $p_{\mathrm{HG}}(\theta; \tau)$, where the one parameter τ is the mean cosine of the scattering angle. [Lun+07]

$$p_{\mathrm{HG}}(\theta; \tau) = \frac{1 - \tau^2}{2(1 + \tau^2 - 2\tau\,\cos\theta)^{\frac{3}{2}}}, \quad \tau = \langle \cos\theta \rangle$$

The South-Polar ice has shown to be preferentially forward scattering with a mean cosine of the scattering angle of $\langle \cos\theta \rangle = 0.94$ with only a weak dependence on the wavelength. [Ack+06]

The **scattering length** λ_{sca}, which is also called **geometric** scattering length, is the mean of the exponentially distributed free path and thereby the average distance between scatterings. [Ack+06] A related and often used quantity is the **effective scattering length** λ_{e}, which is the distance that light needs to propagate through a turbid medium before the photon directions are completely randomized. [Lun+07]

$$\lambda_{\text{e}} = \frac{\lambda_{\text{sca}}}{1 - \langle \cos\theta \rangle} \tag{2}$$

In a medium with isotropic scattering, the geometric and the effective scattering lengths are the same. In a preferentially forward scattering medium like the South-Polar ice, the original direction of a sample of photons is tendentially retained for several scattering steps until the photon direction of the sample is isotropized. The projection of the net velocity vector on the original direction is decreased on average by $\langle \cos\theta \rangle$ for each scattering. [Lun+07]

After n scatterings, the effectively transported forward distance along the original direction is $\lambda_{\text{sca}} \sum_{i=0}^{n} \langle \cos\theta \rangle^i$, such that in the limit of many scatterings, $n \to \infty$, the effectively transported forward distance becomes the effective scattering length. [Lun+07; Ack+06]

$$\lim_{n\to\infty} \lambda_{\text{sca}} \sum_{i=0}^{n} \langle \cos\theta \rangle^i = \frac{\lambda_{\text{sca}}}{1 - \langle \cos\theta \rangle} = \lambda_{\text{e}}$$

Typical effective scattering lengths within the South-Polar ice are on the order of 25 m [Lun+07] and range from 5 m to 90 m in the detector volume [Col+13], corresponding to the geometric scattering length ranging from 0.3 m to 5.4 m.

Light interference effects are ignored during photon propagation as the average distance between the scattering centers is large compared to the photon wavelength. [Ack+06] Also, despite modeling different ice regions with abrupt boundaries in the simulations of this study, the physical boundaries are assumed such that refractive index variations are continuous. Hence reflection at the medium boundaries is ignored in simulations. [Lun+07]

3 Experimental Concepts

3.1 IceCube Detector

The IceCube neutrino detector is built into a cubic-kilometer of the glacial ice at Earth's South Pole. 5160 photo detecting optical modules have been deployed between 1450 m and 2450 m below the surface. Construction has begun in 2005 with the deployment of the first optical modules. The detector is fully operational since 2010. [Aar+17b]

The optical modules are anchored on 86 vertical cables called *strings*, which are positioned on a triangular grid with an overall hexagonal footprint. The strings are about 125 m apart. Each string holds 60 optical modules with a vertical spacing of 17 m. [Aar+17b]

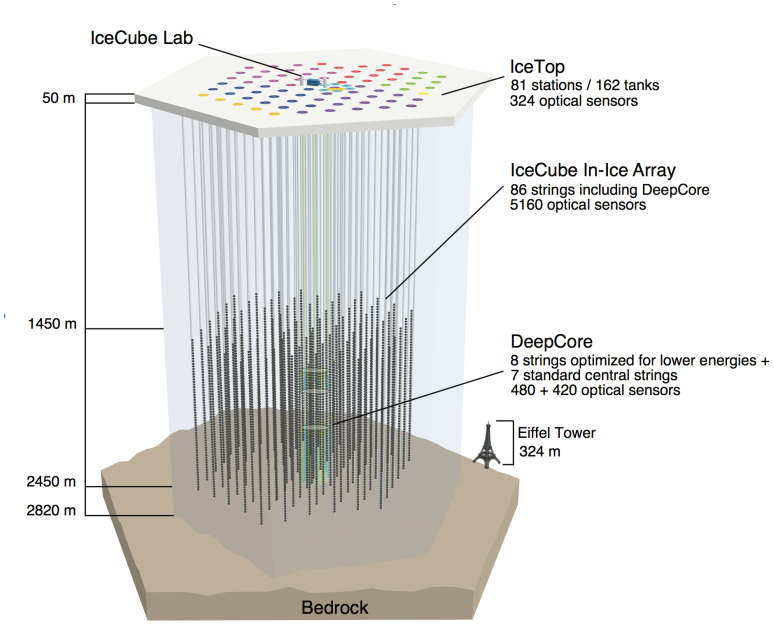
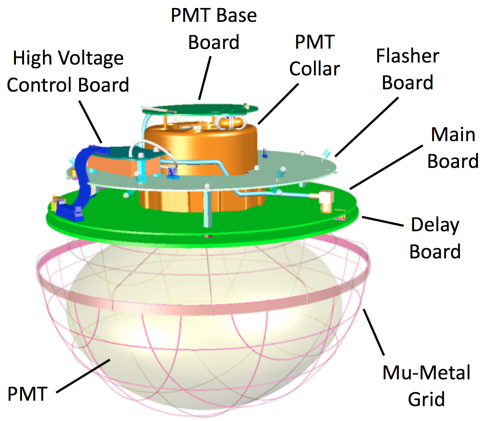


Figure 5: Schematic overview of the IceCube detector. Image source: [Aar+17b]

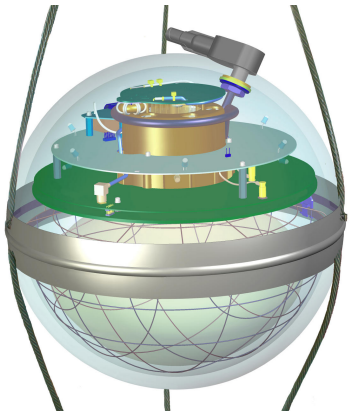
Additional to this *in-ice array*, which is designed to measure neutrinos with energies from the TeV to the PeV scale, the *DeepCore* sub array hold additional optical modules in order to lower the detection energy threshold in this region of the detector to detect neutrinos with energies from 10 GeV to 100 GeV. [Aar+17b]

3.2 Digital Optical Modules (DOMs)

The basic detection unit in IceCube is the *Digital Optical Module* (DOM). Its main components are a photo multiplier tube (PMT), which detects impacting photons, a main electronics board, and a flasher board containing light-emitting diodes (LEDs) for calibration purposes (figure 6 a). The DOM's components are contained within a glass sphere with an outer diameter of about 35 cm that can withstand high pressures (figure 6 b). The space in between is filled with a gel to avoid optical effects at the medium boundaries. [Aar+17b]



(a) Components of the optical module. Image source: [Aar+17b]



(b) Glass sphere containing the components of the optical module. Image source: [Mad13]

Figure 6: Schematic display of a digital optical module (DOM), IceCube's basic detection unit.

The recorded signals from the PMT are digitized within the module before sending the signal to the surface in order to minimize the loss of information from degradation of analog signals sent over long distances. [Ach+06]

The optical modules are optimized for detecting Cherenkov light emitted by particles with energies from 10 GeV to 10 PeV up to 500 m away from the optical module. [Aar+17b]

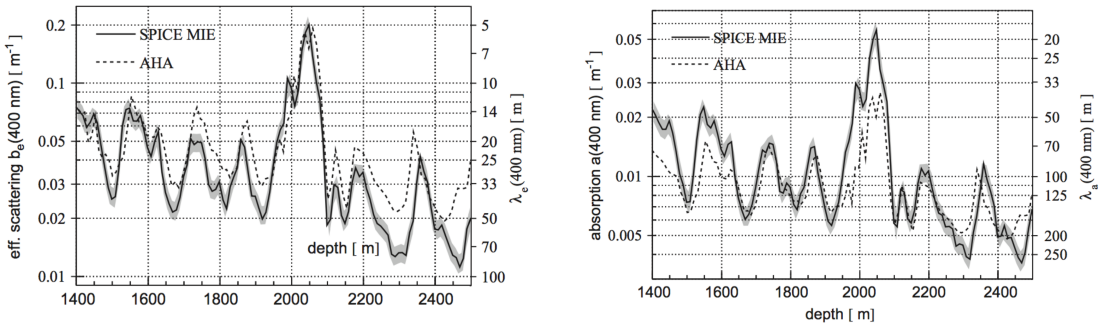
3.3 Properties of South-Polar Ice

The ice at the South Pole has exceptional optical properties, because air bubbles shrink and vanish under large pressure, forming a so-called *clathrate hydrate*, where impurities are enclosed inside the ice crystal structure. [Ron16]

Using IceCube's LED flasher calibration system, the properties of IceCube's glacial ice have been measured: The average distance to absorption, λ_{abs}, the average distance between successive scatters λ_{sca}, and the angular distribution of the new direction after scattering.

To fit these parameters for different depths, the ice has been divided into z-layers of an arbitrary thickness of 10 m. The ice parameters have been fitted for each layer such that the properties are best interpreted as average of their true values over the thickness of the ice layers. [Col+13]

Scattering The measured effective scattering lengths λ_{e} range from 5 m to 90 m, corresponding to the geometric scattering length ranging from 0.3 m to 5.4 m. The depth dependence of the scattering length is shown in figure 7 (a). [Col+13]



(a) Depth dependence of the effective scattering length λ_{e} and the effective scattering coefficient $b_{\text{e}} := {}^1/\lambda_{\text{e}}$.

(b) Depth dependence of the absorption length λ_{abs} and the absorption coefficient $a := {}^1/\lambda_{\text{abs}}$.

Figure 7: Values of the absorption length λ_{abs} and the effective scattering length λ_{e} for different depths, but for a fixed photon wavelength of 400 nm. Plot taken from [Col+13, figure 16]. A detailed data table is given in [Col+13, table C1].

The wavelength dependence is given by equation 3 [Col+13, section 4], where $b_{\text{e}} := {}^1/\lambda_{\text{e}}$ is the effective scattering coefficient, ν the photon wavelength, and α a global fit parameter. $b_{\text{e}}(\nu = 400\,\text{nm})$ is given by figure 7 (a) and [Col+13, table C4]. The global parameter α has been fitted to $\alpha = 0.90 \pm 0.03$ [Ack+06, section 5.1].

$$b_{\text{e}}(\nu) = b_{\text{e}}(\nu = 400\,\text{nm}) \cdot \left(\frac{\nu}{400\,\text{nm}}\right)^{-\alpha} \tag{3}$$

The scattering prefers the forward direction, with a mean cosine of the scattering angle of $\langle \cos\theta \rangle = 0.94$ [Ack+06, paragraph 9] (or $\langle \cos\theta \rangle = 0.90$ [Col+13]).

Absorption The absorption lengths in the South-Polar ice vary between 10 m in dusty regions and 280 m in very clear ice layers. Their depth dependence is given in figure 7 (b). The absorption length, which is governed by the dust concentration, is especially low in the so-called "dust peak" at a depth of about 2000 m. [Ack+06; Chi14; Col+13]

The dependence of the absorption coefficient $a := {}^1/\lambda_{\text{abs}}$ on the photon wavelength ν, the temperature difference $\delta\tau$ is given by equation 4 [Col+13].

$$a(\nu) = a_{\text{dust}}(\nu) + A\,\mathrm{e}^{-B/\nu}\,(1 + 0.01 \cdot \delta\tau) \tag{4}$$

$$a_{\text{dust}}(\nu) = a_{\text{dust}}(\nu = 400\,\text{nm}) \left(\frac{\nu}{400\,\text{nm}}\right)^{-\kappa} \tag{5}$$

The global parameters have been fitted to $A = (6954 \pm 973)\,\text{m}^{-1}$, $B = (6618 \pm 71)\,\text{nm}$, and $\kappa = 1.08 \pm 0.01$ [Ack+06, section 5.2].[1] The temperature difference $\delta\tau(d) = T(d) - T(1730\,\text{m})$ for depths d is given by equation 6 [Col+13].

$$T(d) = 221.5\,\text{K} - 0.00045319\,\frac{\text{K}}{\text{m}} \cdot d + 5.822 \cdot 10^{-6}\,\frac{\text{K}}{\text{m}^2} \cdot d^2 \tag{6}$$

Ice-Layer Tilt and Ice Anisotropy The absorption properties of the ice layers follow the dust concentration, which does not strictly follow the arbitrary z-layers, but is tilted. To model this feature, the ice layers can also be considered tilted by using an effective-z coordinate, $z_{\text{e}}(x, y, z) = z + \text{relief}(x, y, z)$, which is shown in figure 8. [Col+13]

Furthermore, scattering and absorption show also a slight dependency on the propagation direction of the photons, aligned with the ice flow direction of the glacier [RRK17], which is referred to as *ice anisotropy*.

This study considers the dependencies of scattering and absorption length on depth, temperature and wavelength, but does not consider the ice-layer tilt and the ice anisotropy.

An overview of the different ice models used in IceCube is given in [Chi17].

[1] The quantity A in [Col+13] corresponds to the quantity A_{IR} in [Ack+06]. B in [Col+13] corresponds to λ_0 in [Ack+06].

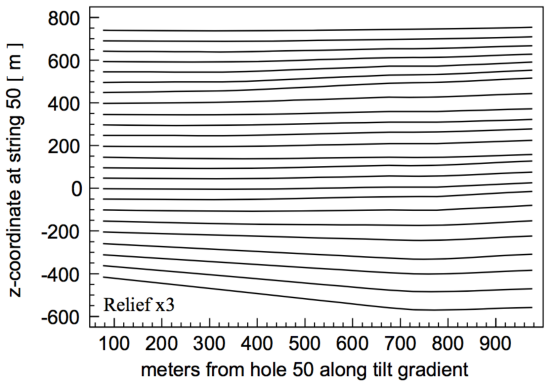


Figure 8: Ice layers along the average gradient direction within the ice. The relief is amplified by a factor of 3 to enhance the clarity of the layer structure. The lowest layer shown exhibits a shift of 56 m between its shallowest and deepest points, which is the largest shift of all layers shown in the figure. Plot and caption taken from [Col+13, figure 14].

3.4 Hole Ice Around the Detector Strings

The so-called *hole ice* is the refrozen water within the drill holes that were necessary to deploy the detector strings with the optical modules.

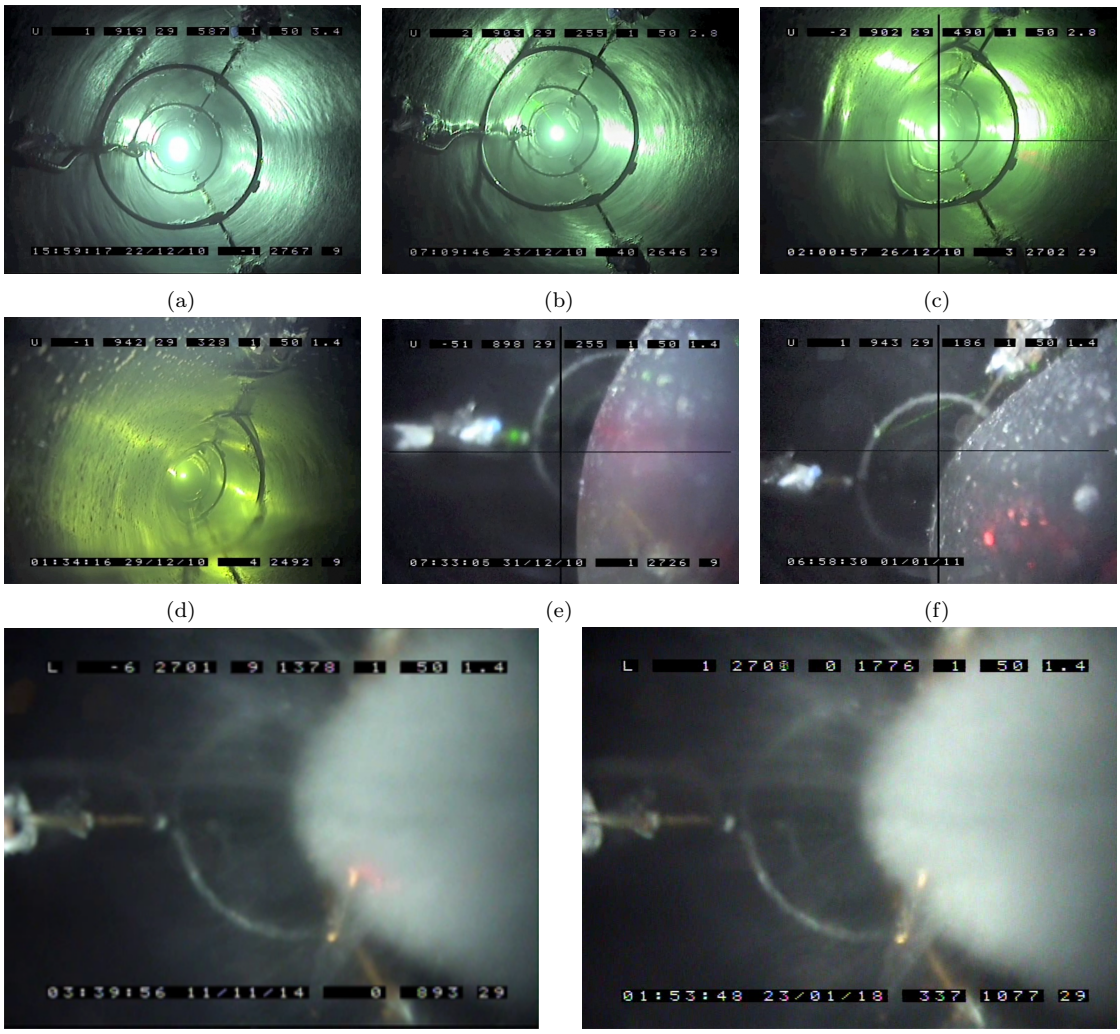
For the IceCube detector, 68 boreholes with an approximate diameter of 60 cm to a depth of about 2500 m were created using a *hot-water drilling* technique. Drilling one hole required about 48 hours time. During the deployment, the drill hole was filled with water. [Aar+17b]

When the glacial ice in the drill holes became water, the structures that were responsible for the specific properties of the bulk-ice layers were destroyed. Thus, the properties of the hole ice are considered largely independent of those of the surrounding bulk ice. The deployed instrumentation become frozen in place and optically coupled to the surrounding ice sheet when the water in the boreholes became ice again. [Aar+17b]

In order to monitor the freeze-in process, a camera system consisting of two video cameras in separate spheres, each also equipped with four LEDs and three lasers, has been deployed along one of the detector strings. The cameras observed that the drill hole became completely refrozen within 15 days. [Aar+17b]

The camera observations suggest two hole-ice components, a clear outer region, and an inner column of a smaller scattering length and a diameter of about 16 cm (figure 9). [Ron16; Aar+17b]

The observed freeze-in process from the outside in is consistent with the model



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 9: Monitoring the freeze-in process using a camera system deployed within string number 80. The drill hole freezes from outside in as seen in (a) to (f). The final configuration that can still be observed in 2018 as seen in (h) still shows a diffuse column, called "bubble column", on the right-hand side of images (g) and (h). Image sources: [RRK17; Fin+11; Ron18]

of *cylindrical freezing*, where impurities or air bubbles are pushed inwards along the freezing boundaries until they merge in the center. [Ron16] After completing the freeze-in process, no long-term changes have been observed from 2010 to 2018. [Aar+17b; Fin+11; Ron18]

In this study, when needing to differentiate between the different components of the hole ice, the outer clear component will be called *drill-hole ice*, or *drill-hole column*. The inner component with a shorter scattering length will be called *bubble column*. Note, however, that there is no established nomenclature in the context of IceCube publications, yet.

In this study, the hole-ice columns will be modeled as cylinders. More complicated geometries such as accounting for the inevitable swinging of the drilling head, or pressure effects that could lead to a vertical gradient in the properties of the hole ice, are not considered in this study.

4 Computing Concepts

4.1 Monte-Carlo Simulations of Photons

A Monte-Carlo simulation is a computational method that utilizes a large amount of random numbers. In order to substitute a complex, possibly unknown probability distribution, samples of random numbers are drawn from one or several basic probability distributions and processed in deterministic calculations. The results are evaluated to gain information about processes or quantities involved. This method is especially useful for systems with many degrees of freedom. [Lex98]

This study needs to determine whether and when detector modules are hit by photons from a given source assuming different ice parameters. In principle, one could devise a mathematical function of input quantities and random variables that determines whether and when a photon is detected by an optical module. This task would be disproportionately complex, however, especially as the function would have to be revised for every change in the underlying models.

Therefore, this study uses Monte-Carlo simulations that propagate photons through the ice in several simulation steps. Based on drawing random numbers from basic probability distributions, the simulation determines in each step, whether and when to scatter or to absorb the photon next, and, in which direction the photon is to be scattered. Checking for DOM collisions in each step, the simulation is able to determine for each photon and each detector module whether and when the photon is detected by the module.

To obtain probability statements from Monte-Carlo simulations, the *law of large numbers* is employed: If an experiment involving random processes is repeated n times, the relative frequency $h_n(A) := {}^{H(A)}/_{n}$ of an event A, which occurs $H(A)$ times in total in these n experiments, approaches the *probability* $p(A)$ of the event A for large numbers n with certainty. [Lex98]

$$\lim_{n\to\infty} \mathrm{P}(|h_n(A) - p(A)| < \epsilon) = 1, \quad \epsilon \in \mathbb{R}$$

Technical progress concerning computational devices, graphics processing units (GPUs) in particular, allows to perform highly parallelized Monte-Carlo simulations on large scale. For a random-walk description of the propagation of photons, see [Ask+97]. A first implementation of a photon-propagation simulation through ice is described in [Lun+07]. A study of propagation simulations using GPUs is presented in [Chi14].

4.2 Parallel Computing on Graphics Processing Units (GPUs)

Graphics processing units (GPUs) are optimized for performing simple calculations for a large number of values in parallel.

The general procedure for GPU calculations is to allocate memory on the GPU, to copy input parameters onto the GPU, perform calculations on the GPU, and then to download the results from the GPU. This procedure is efficient if the time that is spent for allocating, and copying to and from the GPU memory is short compared to the time spent with calculations on the GPU. [Owe13]

The basic units for parallelization are *computational threads*. All operations within a thread are sequential, while all those operations are applied to a set of threads, called *thread block*, in parallel. Each GPU may have one or several thread blocks. [Owe13]

While technological progress is achieved rapidly, GPU memory is still a considerably limited resource. In particular fast memory, which requires a small amount of physical time for reading and writing information, is expensive. Therefore, memory is divided in several categories: The *local memory* that belongs only to one thread, is fastest but most limited. The *shared memory*, which is common to all threads of one thread block, is slower. Next slower is the *global memory*, which is common to all thread blocks on the GPU. Slowest, but in comparison to the GPU memory practically unlimited, is the *host memory*, which is memory not on the GPU but on other components of the computer. Efficient memory use is one of the key concepts for performant GPU programming. *Coalesce memory access*, which means that each thread in a thread block reads or writes from or to a coherent memory block parallel to the other threads of the thread block, increases memory access performance. [Owe13]

Techniques that utilize the internal optimizations of GPUs allow for further performance improvements: Using GPU-native *atomic operations* such as the increment operator that increases the value of a variable by 1 is more performant than using a generic mathematical operation. GPUs support vectors with four components as native data types. Using *native vectorial operations* such as a dot product is more performant than implementing the operation as mathematical function manually. [Owe13]

In parallel computing, the *step complexity* of an algorithm is a measure of the physical time the parallelized algorithm needs to run. The *work complexity* is a measure for the summed computational work that is done by all threads in that time. A pattern to avoid in this context is *thread divergence* where some threads have to stay idle and wait for other threads completing their work. [Owe13]

4.3 IceCube Simulation Framework

This study uses the IceCube Simulation Framework, short IceSim, in Version `V05-00-07`. The framework, written in C++[2] and Python[3], provides the software needed to run simulations, to write and to process IceCube-specific data such as simulated or recorded events and the detector geometry.

> The documentation of the IceCube Simulation Framework can be found at http://software.icecube.wisc.edu/documentation/.
>
> A guide on how to install the IceCube Simulation Framework with all the other tools needed for this study is provided at https://github.com/fiedl/hole-ice-study/blob/master/notes/2018-01-23_Installing_IceSim_in_Zeuthen.md.
>
> The source code of the IceCube Simulation Framework can be found at http://code.icecube.wisc.edu/projects/icecube/browser/IceCube/meta-projects/simulation.

4.4 Photon Propagation With Clsim

The main tool of this study is the photon-tracking simulation software Clsim. This software implements a ray-tracing algorithm described in section 5.2.2, modeling scattering and absorption of light in the deep-glacial ice at the South Pole or in Mediterranean sea water. [Kop13] Written in C++, Python and OpenCL C[4], Clsim uses OpenCL to simulate the photon propagation in a highly parallelized way on graphics processing units (GPUs). [Kop17] Clsim reads the photon sources from the IceSim data, converts the data into a format that can be used on GPUs, uploads the data and the propagation program ("kernel") onto the GPU, and propagates the photons there. Then Clsim downloads the hits and tracks of the propagated photons, performs post-processing operations such as accepting hits on optical modules based on the acceptance criteria of the modules, and converts them back into an IceSim-compatible format.

[2] C++ programming language, https://isocpp.org
[3] Python programming language, https://www.python.org
[4] OpenCL, Open Computing Language, https://www.khronos.org/opencl

The source code of CLSIM, which is released under the Internet Systems Consortium (ISC) license, can be accessed at the CLSIM code repository https://github.com/claudiok/clsim.

In order to simulate the propagation through hole ice, CLSIM has been modified for this study. Until the modified source code has been merged into the main code repository, the modified CLSIM source code can be accessed through the forked code repository at https://github.com/fiedl/clsim.

A guide on how to install the modified version of CLSIM can be found in appendix A.8 and on https://github.com/fiedl/hole-ice-study/blob/master/notes/2018-01-23_Installing_IceSim_in_Zeuthen.md#install-patched-clsim.

4.5 Photon Visualization With STEAMSHOVEL

STEAMSHOVEL (Figure 10) is the event and data viewer software of ICECUBE. It allows to visualize the ICECUBE detector as well as events that have been recorded, reconstructed or simulated in the detector. For example, STEAMSHOVEL can be used to visualize light sources such as muons traveling through the detector producing Cherenkov light, or LED flashes that can be emitted by the optical modules for calibration purposes. STEAMSHOVEL may also visualize the photons propagating through the ice, or the amount of light detected by the optical modules. Data containing timing information can be visualized in an animated manner, watching an event as slow-motion animation.

Most event visualizations in this study are created using STEAMSHOVEL. In order to visualize hole ice and propagating photons for this study, several patches of the standard STEAMSHOVEL code are required.

The required STEAMSHOVEL patches are provided within the code repository of this study: https://github.com/fiedl/hole-ice-study/tree/master/patches/steamshovel

4.6 Other Software Tools

In order to process data, to perform statistical analyses, and to plot data, this study uses several supplementary software libraries, such as NUMPY[5], MATPLOTLIB[6]

[5] NumPy package for scientific computing, http://www.numpy.org

[6] Matplotlib plotting library, http://matplotlib.org

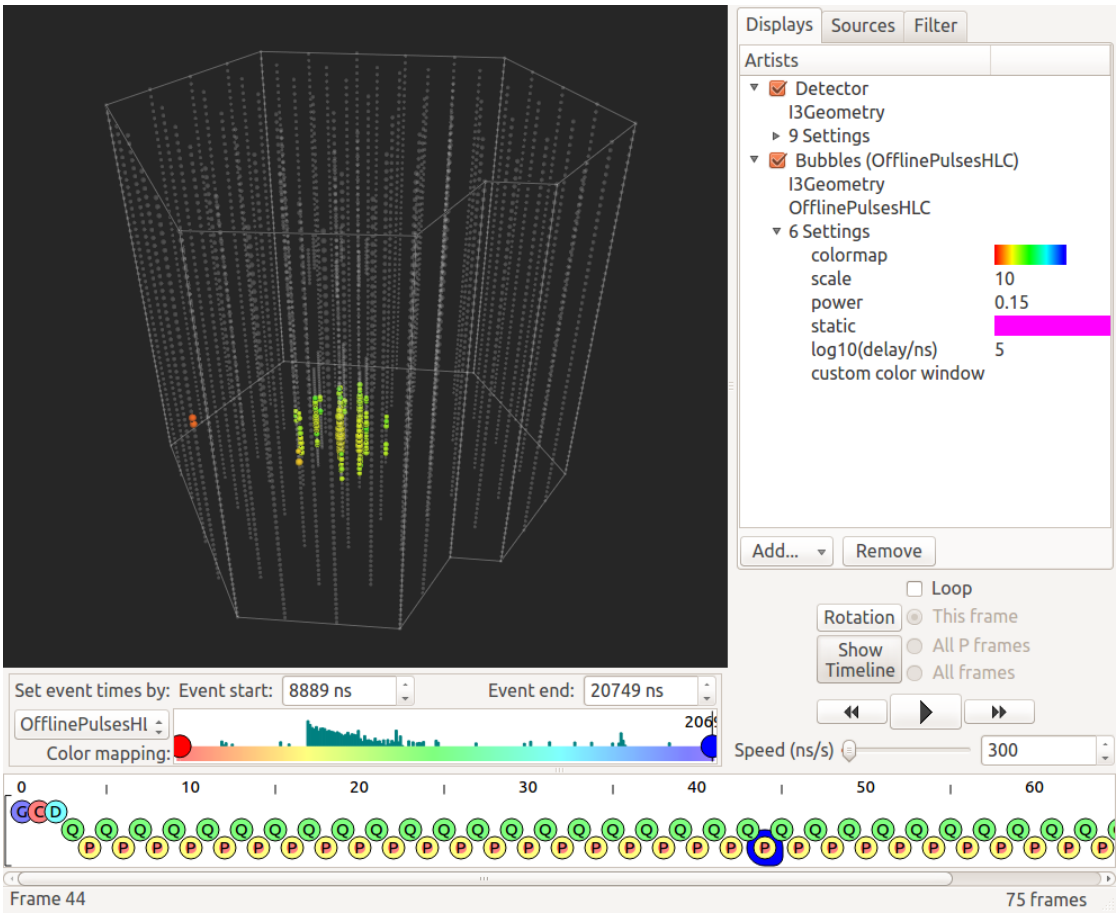


Figure 10: Screenshot of STEAMSHOVEL, the ICECUBE event and data viewer software. The main workspace shows a schematic of the ICECUBE detector with its 86 strings holding 60 optical detector modules each. Control elements allow to hide and display components and to navigate through the event information. Image source: [Van+17]

and PANDAS[7]. For unit tests (section 5.6.1), this study uses the GTEST[8] testing framework.

The necessary scripts and installation instructions can be found in the code repository of this study at https://github.com/fiedl/hole-ice-study.

[7] Pandas Python Data Analysis Library, http://pandas.pydata.org
[8] Google Test Framework, gtest, https://github.com/google/googletest

5 Development of Algorithms For the Direct Photon-Propagation Through Hole Ice With CLSIM

5.1 Modeling Hole Ice as Distinct Cylindrical Ice Volumes

In the ICECUBE simulation framework, photon propagation simulation takes several dependencies into account: The photon absorption length and the photon scattering length may depend on the photon's wavelength. Absorption and scattering length may also depend on the photon's z-coordinate as the South-Polar ice consists of several *ice layers*. These ice layers may also be tilted. Furthermore, the absorption length may depend on the photon's direction of motion, which is called *absorption anisotropy*.

This study adds another ice feature to the simulation: Hole ice may be modeled by adding cylinder-shaped volumes within the surrounding bulk ice where the propagation properties, or, to be specific, the photon's absorption length and scattering length, differ from the propagation properties of the bulk ice.

Figure 11 illustrates such a scenario for one single photon: The photon's trajectory starts in the bulk ice. The photon enters the hole-ice cylinder where the scattering length is shorter as in the bulk ice. Hence the photon scatters more frequently within the cylinder. When the photon leaves the hole-ice cylinder the propagation properties of the bulk ice take effect again, resulting in the photon scattering less frequently after leaving the cylinder.

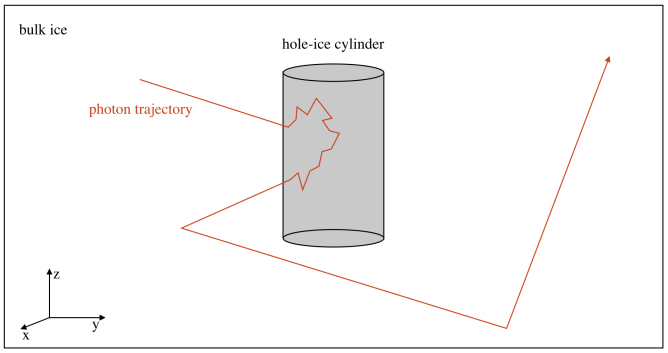


Figure 11: Schematic diagram of a propagating photon. The photon enters a hole-ice cylinder with a scattering length different from the outside ice. When leaving the cylinder the photon assumes the scattering length of the outside ice again.

In this study, cylinders are always defined along the z-axis. Also, the scattering angle is assumed to behave the same within the hole ice as in the bulk ice.

5.2 Propagating Photons Through Different Media

5.2.1 Very Basic Photon-Propagation Algorithm

A first, very basic photon-propagation algorithm, which is not actually implemented in CLSIM, but is presented as comparative example, moves the photon by a small distance δx. At the new photon position, the algorithm checks for detection at an optical module and randomizes as a function of the scattering and absorption lengths whether the photon should be scattered or absorbed by the ice at this position. Then, the photon is propagated again by the same small distance δx. The same loop repeats until the photon either hits an optical module or is absorbed by the ice. Figure 12 illustrates a propagation scenario in a two-dimensional coordinate system. Figure 13 presents the algorithm as flow chart.

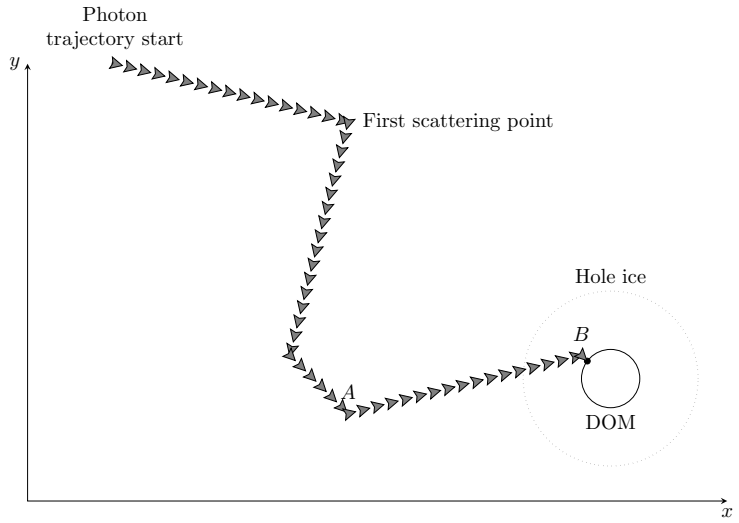


Figure 12: Illustration of a basic photon-propagation algorithm in a two-dimensional coordinate system. The photon is propagated by a small distance in each propagation step. At each position, the algorithm checks for absorption, scattering and whether an optical module has been hit.

This very basic propagation algorithm can handle propagation through different media just by making the scattering and absorption probabilities depend on the current photon position and direction. Ice layers and ice layer tilt can be implemented by making the scattering and absorption probabilities depend on the current photon position, absorption anisotropy by making the absorption probability depend on the current photon direction as well. Hole-ice cylinders can be implemented by checking whether the current photon position is within any of a list of cylinders

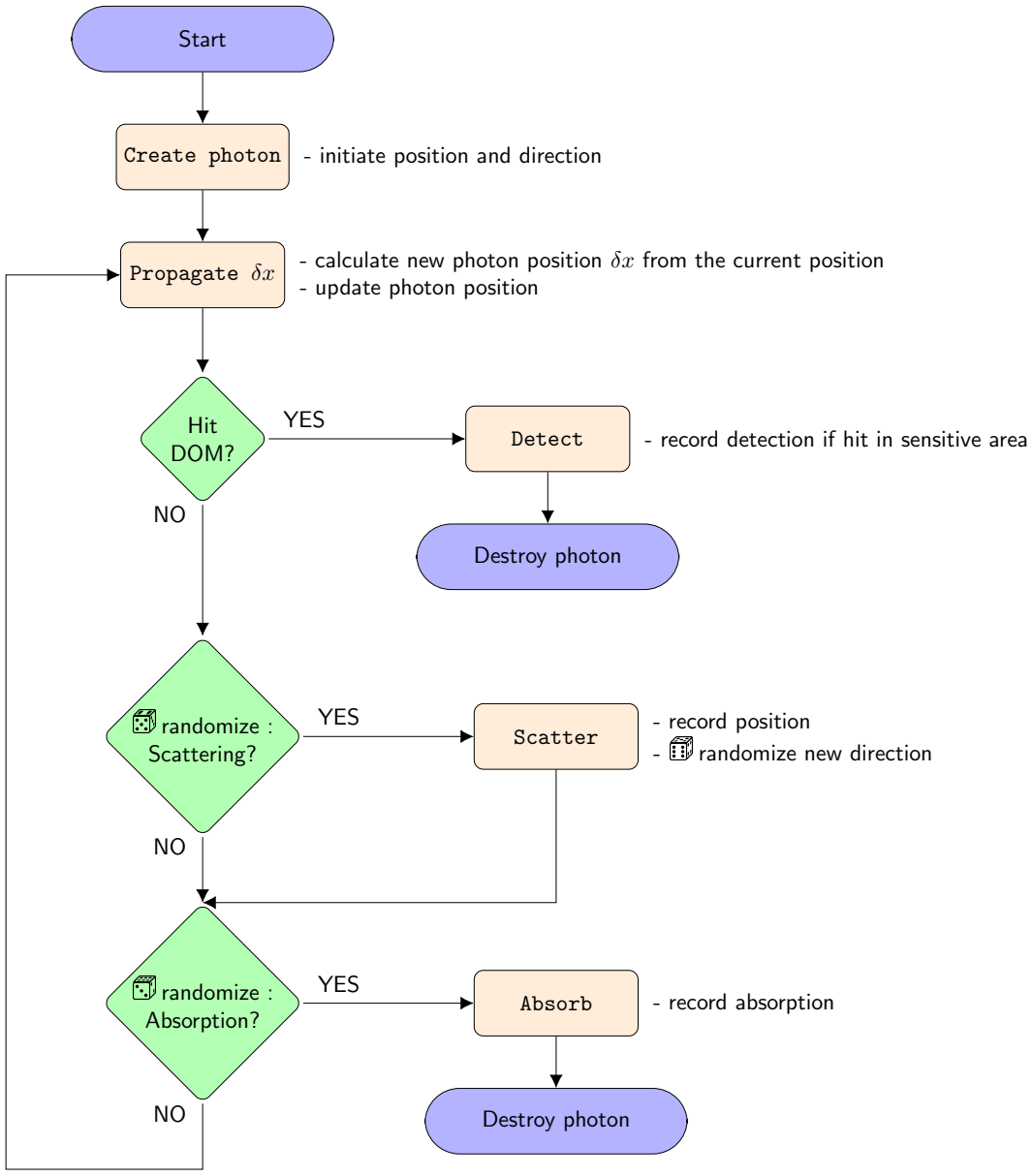


Figure 13: Flow chart of a basic photon-propagation algorithm. Interfaces, where the algorithm begins or ends, are displayed as violet pill shapes. Processes, where the algorithm performs an operation or a calculation, are displayed as brown rounded boxes. Decisions are displayed as green rounded diamond shapes. One propagation step consists of moving the photon by a fixed small distance δx, checking for detection as well as randomizing scattering and absorption within the ice.

defined in any way, either by supplying a list of cylinder coordinates and radii, or by re-using the coordinates of the detector strings.

This basic propagation algorithm, however, is very inefficient regarding computational performance: The algorithm moves the photon over long distances in small steps without changing the direction. For a typical geometric scattering length of two meters, moving the photon in steps of $\delta x = 1\,\mathrm{mm}$ per propagation step would mean performing 2000 propagation steps before changing the direction of motion.

The propagation algorithm can be made more efficient by moving the photon in each propagation step not by a small distance δx but by the whole distance to the next interaction point. This way, the above example would take only one propagation step rather than 2000. This performance improvement, however, comes at the cost that propagating through different media requires a different, more involved computational approach. This more efficient propagation algorithm, which is the standard photon-propagation algorithm in IceCube, will be described in the next section.

5.2.2 Standard Photon Propagation Algorithm

In IceCube's standard photon-propagation algorithm, a propagation step moves the photon not just by a small, fixed distance δx but to the next interaction point at once. An interaction may be the photon scattering within the ice, the photon being absorbed by the ice, or the photon hitting an optical module. [Kop17; Chi14]

At each scattering point, the algorithm randomizes the new photon direction based on the scattering angle distribution, and how far the photon will travel until it is scattered again based on the scattering length. How far the photon will travel until it is absorbed is randomized just once when the photon is created.

For each propagation step, the algorithm checks whether the photon will hit an optical module on the path between two scattering points. Also, the algorithm checks whether the destined distance to absorption will be reached before the next scattering point. If the photon will be destroyed in this step, either by being absorbed in the ice, or by hitting an optical module, the final position is recorded. Otherwise, the photon proceeds to the next scattering position. This loop is repeated until the photon is either absorbed or hits an optical module. [Chi14]

Figure 15 shows a flow chart of this algorithm. Figure 14 shows the same two-dimensional scenario as figure 12 in the previous section, but illustrates how the same photon trajectory is modeled by this algorithm with significantly less simulation steps.

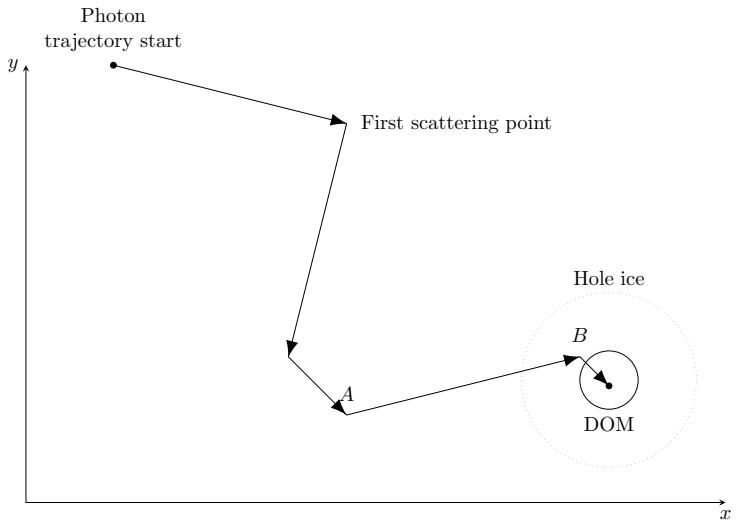


Figure 14: Illustration of IceCube's standard photon-propagation algorithm in a two-dimensional coordinate system. The photon is propagated from one scattering point to the next scattering point in each propagation step. In each step, the algorithm checks for absorption and whether an optical module (DOM) is hit in between the two scattering points.

Assuming the number of calculations in each scattering step is of the same order of magnitude in both described algorithms, the number of simulation steps translates to the total number of calculations the processing unit needs to perform for each photon. Thus, the algorithm that needs much less simulation steps is also much more efficient.

Propagating a photon through several media with different optical properties is more involved in this algorithm than in the basic one. If the algorithm calculates the distance to the next interaction point considering only the interaction properties of the ice at the current position of the photon as suggested by the basic algorithm, then the next interaction point would be calculated inaccurately if the medium properties change between two interaction points. The larger the scattering length gets compared to the size of the ice volumes with constant interaction properties, the worse the inaccuracies become. This can be illustrated by an extreme example where the scattering length and the absorption length within the bulk ice are several meters long, but the photon would hit a cable with a diameter of only a couple of centimeters between two scattering points. The cable would absorb the photon at once. But if the next interaction point is calculated evaluating only the interaction properties at the position of the current scattering point, then the photon ignores the cable and continues on its path without being absorbed by the cable.

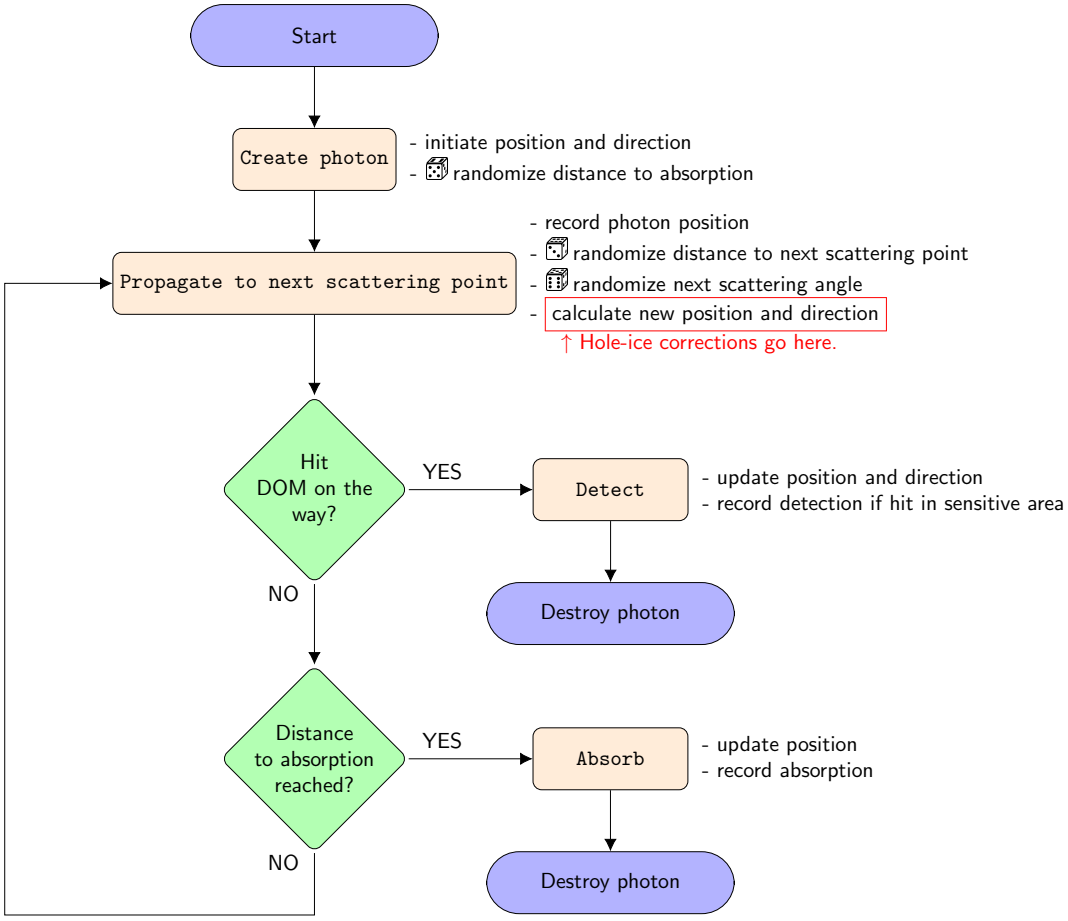


Figure 15: Flow chart of the standard photon-propagation algorithm. One propagation step consists of moving the photon from one scattering point to the next scattering point. If the photon hits an optical module in between the two scattering points, the algorithm records a hit and destroys the photon. If the photon is absorbed in the ice in between the two scattering points, it will only be propagated to the position of absorption and it will not reach the next scattering point. When adding propagation through hole ice with a different scattering length to this algorithm, the calculation of the next scattering point needs to be modified accordingly.

The solution to this problem chosen for the standard propagation algorithm in IceCube is to implement medium transitions in the following manner: Rather than randomizing the geometrical distance $X := \overline{AB}$ from the current interaction point A to the next interaction point B itself, the algorithm randomizes the number $N :\in \mathbb{R}^+$ of interaction lengths the photon will travel to the interaction point, like a budget. This budget is spent by considering the different media on the way from the current interaction point along the photon's direction, and converting the number of interaction lengths into a geometrical distance according to the shares of the different media in the path of the photon between the two interaction points.

Suppose, starting at interaction point A, the photon travels in $m :\in \mathbb{N}$ media M_i with interaction lengths λ_i for a distance of x_i respectively until it reaches the next interaction point B. In each medium M_i, the algorithm spends $n_i : x_i = n_i \lambda_i$ of the budget $N := \sum_1^m n_i$ that has been randomized at interaction point A until it reaches interaction point B in a distance $X := \overline{AB}$.

$$X = \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} n_i \lambda_i, \qquad N = \sum_{i=1}^{m} n_i \tag{7}$$

Each distance x_i is the length of the trajectory the photon spends in medium M_i. The first distance x_1 is the distance from the starting point A to the medium border of M_1 and M_2 along the photon direction. The subsequent distances x_i are the distances of the first medium border (of M_{i-1} and M_i) and the second medium border (of M_i and M_{i+1}) respectively along the photon direction. The last distance x_m is determined by how much of the budget N is left in the last medium, $x_m = n_m \lambda_m$, such that the overall budget $N := \sum_{i=1}^m n_i$ is spent.

Coming back to the extreme example where a cable lies ahead on the photon path, the algorithm now considers all media on the path along the photon direction in order to convert the number of interaction lengths into a geometrical distance to the interaction point. As the absorption length within the cable's medium is set to zero, all of the absorption-length budget is spent at the medium border to the cable, resulting in the final interaction point of the photon being right at the point where the photon enters the cable. Thus, this algorithm lets the photon be absorbed by the cable as intended.

Ice layers, the tilt of the ice layers, and the absorption anisotropy are modeled in this algorithm by implementing the rules on how the interaction budget is spent accordingly. Adding cylinder-shaped volumes to model hole ice or cables means to further extend these rules on how the scattering and absorption budgets are spent along the photon's trajectory.

Two different algorithms for the photon propagation through cylinder-shaped volumes are described in the following sections: Section 5.3 describes a first approach where the propagation through the hole-ice cylinders is added as subsequent correction for the existing medium-propagation algorithm. Section 5.4 describes a second approach where the existing medium-propagation algorithm is rewritten in order to support the propagation through hole-ice cylinders the same way as other medium transitions.

5.3 Algorithm A: Hole-Ice Propagation as Subsequent Correction of the Propagation Without Hole Ice

A first approach to adding propagation through hole-ice cylinders to the standard propagation algorithm (section 5.2.2) implemented in CLSIM is to add a subsequent correction to each simulation step without rewriting the existing algorithm.

The source code of the implementation of this first approach can be found in appendix A.9, in the folder `algorithm_a` on the CD-ROM, as well as in the code repository at https://github.com/fiedl/clsim/tree/sf/hole-ice-2017/resources/kernels/lib/hole_ice.

This approach assumes that the scattering length $\lambda_{\text{sca}}^{\text{H}}$ and the absorption length $\lambda_{\text{abs}}^{\text{H}}$ within the hole ice can be expressed as multiple of the corresponding scattering length λ_{sca} and absorption length λ_{abs} within the surrounding bulk ice.

$$\lambda_{\text{sca}}^{\text{H}} = f_{\text{sca}}\,\lambda_{\text{sca}}, \quad \lambda_{\text{abs}}^{\text{H}} = f_{\text{abs}}\,\lambda_{\text{abs}}$$

The factor $f_{\text{sca}} :\in \mathbb{R}_0^+$ will be called **scattering length factor**, $f_{\text{abs}} :\in \mathbb{R}_0^+$ **absorption length factor**. These factors can be implemented as properties of the individual hole-ice cylinders, or as common property of all cylinders.

If, for example a cylinder has an absorption length factor of $f_{\text{abs}} = 1$, the absorption length within the hole ice is the same as in the surrounding bulk ice. If the absorption length factor is $f_{\text{abs}} = 0$, a photon will be absorbed instantly when entering the hole-ice cylinder. A scattering factor of $f_{\text{sca}} = 0.1$ means that the scattering length within the hole ice is one tenth of the scattering length within the surrounding bulk ice.

Task The task of this **hole-ice-correction algorithm** is to modify the quantities that are effected by the hole ice in each simulation step.

Figure 16 illustrates this task: The standard CLSIM propagation algorithm calculates the next scattering point B without any knowledge of the hole ice. The hole-ice algorithm takes the properties of the hole ice into account and calculates a correction for the next scattering point. CLSIM will use the corrected next scattering point B' rather than the originally calculated point B.

Context The hole-ice-correction algorithm is inserted into the CLSIM propagation algorithm's simulation step right after CLSIM has calculated the distance to the

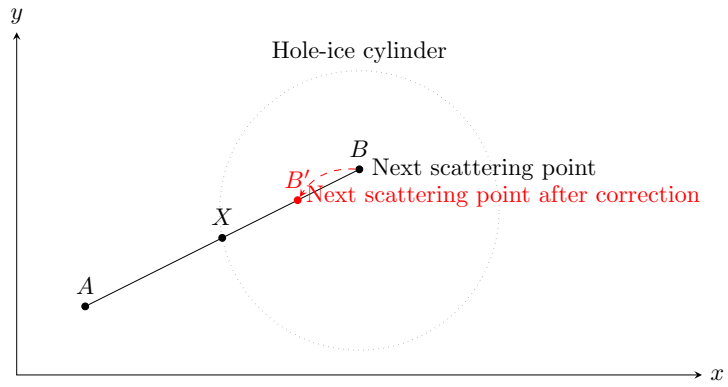


Figure 16: Illustration of the task of the hole-ice correction algorithm. In this two-dimensional scenario, the hole-ice cylinder is represented by a circle. When the photon scatters at point A, the standard-CLSIM algorithm calculates the next scattering point B without knowledge of the hole ice. The hole-ice correction algorithm calculates what fraction of the trajectory AB runs through the hole ice and determines a correction, resulting in a hole-ice-corrected next scattering point B'.

next scattering point and the remaining distance to the absorption point without knowledge of the hole ice. After the hole-ice-correction algorithm follows the check whether the photon hits an optical module on the way from A to B'.

The hole-ice-correction algorithm takes the following input parameters: Current photon position at point A, photon direction after scattering at point A, a list of the hole-ice cylinders with their coordinates and radii, the hole-ice scattering length factor f_{sca} and absorption length factor f_{abs} as common properties of all hole-ice cylinders, the distance the next scattering point calculated by the standard algorithm, and the remaining distance to the absorption point calculated by the standard algorithm. As output parameters, the hole-ice correction algorithm returns the hole-ice-corrected distance to the next scattering point and the hole-ice-corrected remaining distance to the absorption point.

Procedure For each propagation from one scattering point A to the next scattering point B, the hole-ice algorithm calculates the intersection points of the photon trajectory AB and the hole-ice cylinders in range. Based on what portion of the distance AB runs through the hole ice, the algorithm calculates a correction for the distance to the next scattering point and a correction for the remaining distance to the absorption point.

Both calculations, scattering correction and absorption correction, depend on each other: If the photon scatters earlier within the hole ice than determined by the

standard algorithm, then the corrected path within the hole ice is shorter than the original one, which means that the remaining distance to the absorption point will be longer as less of the absorption budget is spent, yet. If, on the other hand, the hole ice absorbs the photon instantly when entering the cylinder, then the corrected final position B' of this simulation step won't be the point determined by the scattering correction but the point where the photon enters the cylinder.

Figure 17 shows a flow chart of this algorithm including the loop over the hole-ice cylinders in range, the calculation of the intersection points of photon path and hole-ice cylinder, the correction of the next scattering point and the correction of the remaining distance to absorption.

Intersection Calculations In order to determine the fraction of the path AB of the photon that runs through the hole-ice cylinder, the intersection points of the photon path and the cylinder need to be calculated. This can either be done solving the geometric equations coordinate-wise for the coordinates of the zero, one or two intersection points, or by treating the same scenario as vectorial problem, calculating the coordinate vectors of the intersection points using other vectorial quantities rather than separate coordinates. The latter approach turns out to be more efficient as it utilizes the support of native vector operations of the graphics processing units. Both approaches are presented in appendix A.11.

3D-2D Projection Intersection points can be calculated in two dimensions by projecting all coordinates from the three-dimensional coordinate system onto the x-y plane along the z-axis, and then calculating the intersection of a line and a circle rather than the intersections of a line and a cylinder. The fraction of the photon path that runs through the hole ice is the same as in three dimensions due to the intercept theorem, because both, the distance to the intersection point, $\overline{AX}_{\mathrm{2D}} := \xi\overline{AX}_{\mathrm{3D}}$, and the distance to the next scattering point, $\overline{AB}_{\mathrm{2D}} := \xi\overline{AB}_{\mathrm{3D}}$, use the same projection factor $\xi \in \mathbb{R}^+$, which itself depends on the photon direction.

$$\frac{\overline{AX}_{\mathrm{2D}}}{\overline{AB}_{\mathrm{2D}}} = \frac{\overline{AX}_{\mathrm{3D}}}{\overline{AB}_{\mathrm{3D}}}$$

The relative distance corrections, $\Delta x / \overline{AB}$, are the same in two and three dimensions due to the same reason. But the absolute distance correction $\Delta x : \overline{AB'} = \overline{AB} + \Delta x$ needs to be expressed in three dimensions when adding the correction to the distances used in the standard algorithm as they are defined for a three-dimensional coordinate system. Keeping this conversion requirement in mind, the geometric

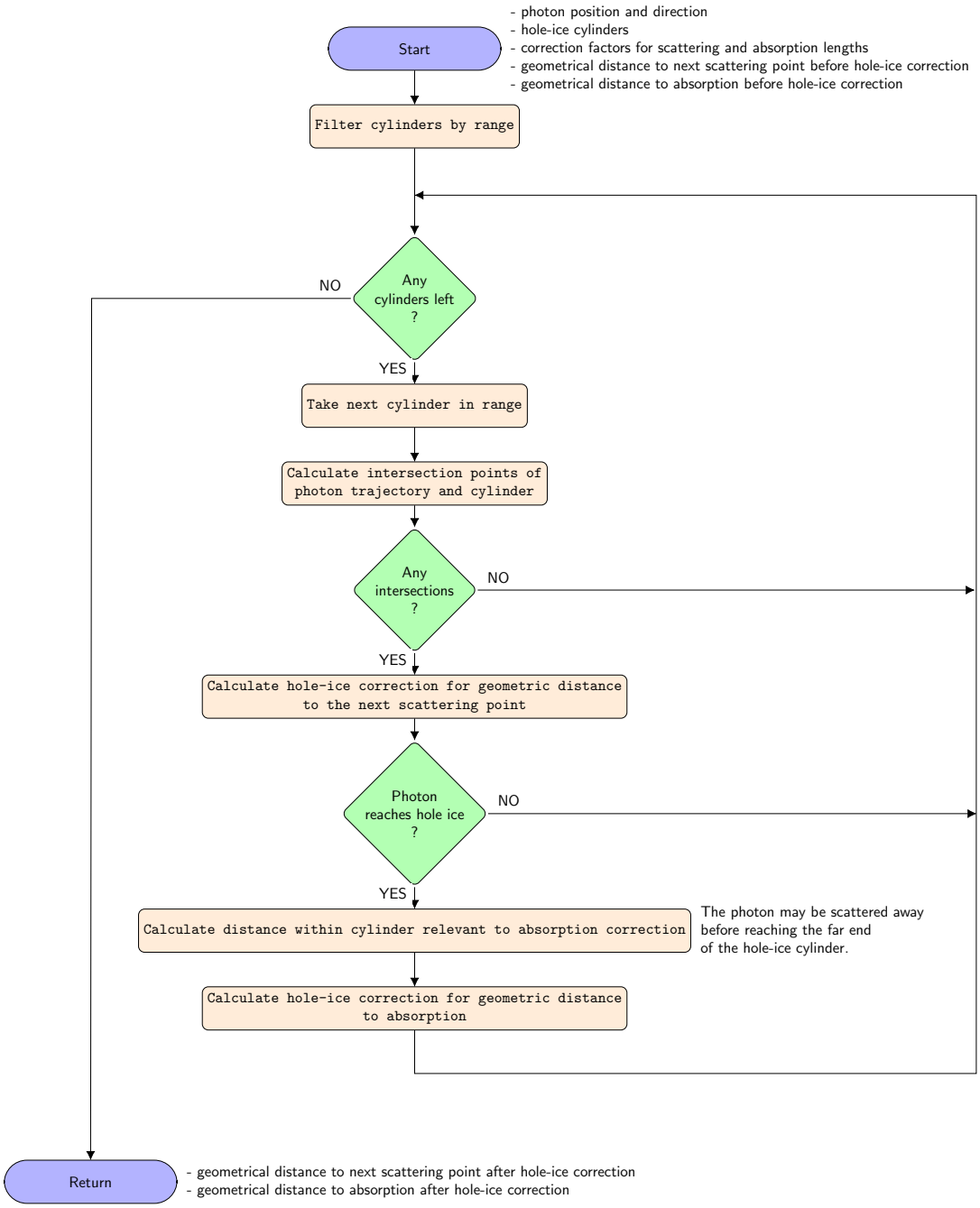


Figure 17: Flow chart of the hole-ice-correction algorithm, which calculates subsequent corrections for the quantities that are effected by the hole ice: the distance to the next scattering point and the remaining distance to absorption of the photon. Both corrections depend on each other: If the photon scatters earlier, the absorption correction will be smaller. The algorithm returns the corrected quantities to the standard algorithm, which uses the corrected quantities from that point on.

cases of the photon path AB running through a hole-ice cylinder can be examined in two dimensions.

Geometric Cases When the hole-ice algorithm calculates the distance correction $\Delta x : \overline{AB'} = \overline{AB} + \Delta x$, there are several geometric cases to consider, depending on the number N of intersections of the photon path AB with the hole-ice cylinder and whether the path starts within the cylinder or outside the cylinder.

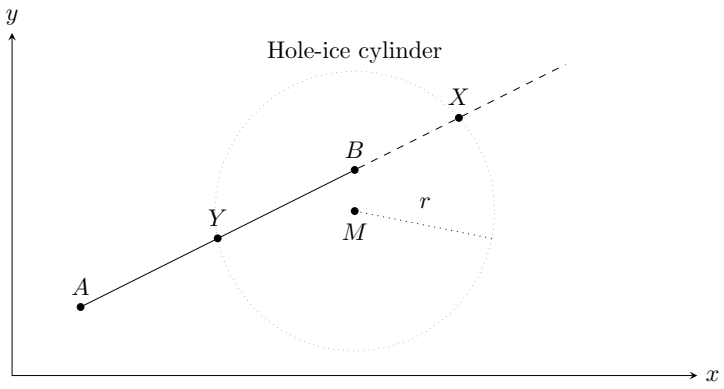


Figure 18: Intersection of the line along the photon path AB with the hole-ice cylinder in two dimensions, where the cylinder is represented by a circle of radius r around the center M. The first intersection point is Y, the second intersection point X. The second intersection point X is beyond the path's end point B. Thus, the number N of intersections is considered to be $N = 1$ in this case.

The distance correction depends on the fraction $w :\in \mathbb{R}_0^+$ of the path length $\overline{AB}$ that runs within the hole-ice cylinder.

Case 1 If the photon starts outside the cylinder and the path has no intersection with the cylinder ($N = 0$), then the path does not run through the cylinder, $w = 0$, and the distance correction Δx needs to be zero: $\Delta x = 0$.

Case 2 If the photon starts inside the cylinder and the path has no intersection with the cylinder ($N = 0$), then the whole path is within the cylinder, $w = 1$. A photon that would travel a distance of λ within the bulk ice, travels a distance of λ^{H} within the hole ice, $\lambda^{\mathrm{H}} = f\,\lambda$. The distance correction Δx for this photon would be $\Delta x = \lambda^{\mathrm{H}} - \lambda = (f-1)\lambda$, or more generally, $\Delta x = (f-1)\overline{AB}$.

Case 3 If the photon starts outside the cylinder, but intersects the cylinder once ($N = 1$), then the first part of the trajectory, AY, stays the same, and only

the path from the first intersection point Y to the destination point B runs within the hole ice, $w = \overline{YB}/\overline{AB}$. As only this fraction needs to be corrected, the distance correction is $\Delta x = (f-1)\overline{YB}$.

Case 4 If the photon intersects the cylinder once ($N = 1$), but starts within the cylinder, only the first part, AC, of the path needs to be corrected.

C is the **termination point** of the trajectory inside the cylinder. In cases where the photon reaches the far end of the cylinder, C is just the second intersection point, $C = X$. If the trajectory within the cylinder is terminated, however, by the other interaction respectively, for example if the photon is scattered away before reaching the far end when calculating the absorption correction, C is the point where the trajectory is terminated by the other interaction.

In contrast to Case 3 where the outside distance $\overline{AY}$ has been fixed, in this case the outside trajectory part $\overline{CB}$ comes after the hole-ice part $\overline{YC}$ and will be the shorter the stronger the hole-ice correction is. Therefore, one needs to use the scaled distance $\overline{AC}/f$ to determine how much of the path will remain outside the cylinder, resulting in a distance correction of $\Delta x = (1 - \frac{1}{f})\overline{AC}$.

Case 5 If the photon starts outside the cylinder and has two intersections ($N = 2$) with the cylinder, this is a combination of Case 3 and Case 4, resulting in a distance correction of $\Delta x = (1 - \frac{1}{f})\overline{YC}$.

A flow chart of the distance-correction algorithm is shown in figure 19.

Pros and Cons As this first hole-ice algorithm only adds subsequent corrections for the standard-CLSIM algorithm, it leaves the standard-CLSIM code, which is already well tested, almost untouched. The additions of the hole-ice algorithm interface with the CLSIM algorithm only through a small surface area, corresponding to a small number of variables passed from one algorithm to the other. This allows to test the hole-ice algorithm with so-called *unit tests* (section 5.6.1) where specific examples with fixed input variables are provided and tested whether the algorithm produces the expected results for the example.

The hole-ice correction algorithm, however, has important limitations. First, the current understanding of the hole ice suggests that the interaction properties of the hole ice are independent of the properties of the surrounding bulk ice (see section 3.4). But with this hole-ice correction algorithm, it is only possible to define the interaction lengths of the hole ice relative to the interaction properties of the surrounding bulk ice. While one could workaround this problem locally by adjusting the correction factors f in relation to the local properties of the bulk

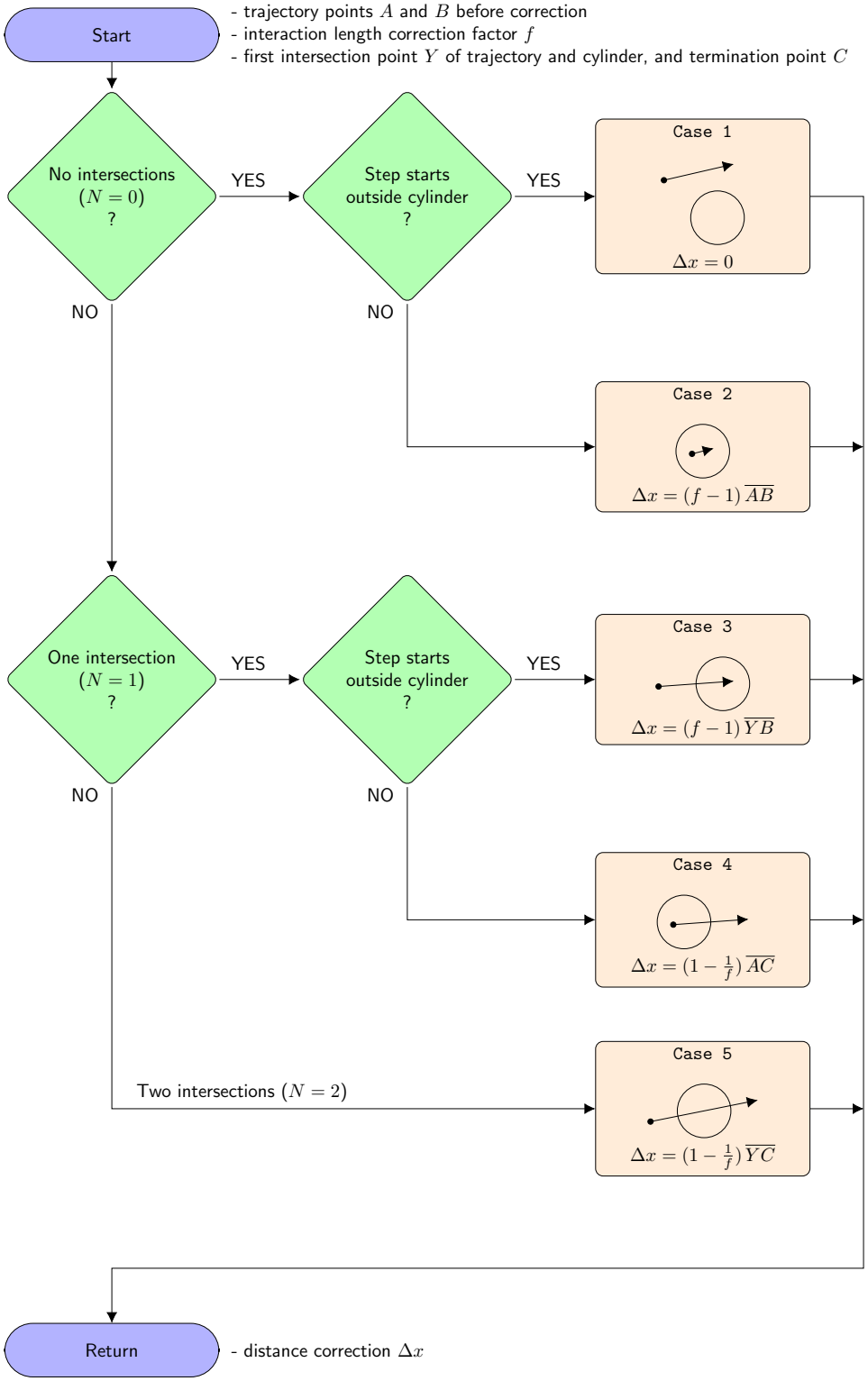


Figure 19: Flow chart for calculating the hole-ice correction for the geometric distance to the next interaction point.

ice, large jumps over several layers of ice with different properties would cause inevitable errors.

Secondly, the algorithm does work only correct in scenarios where the photon crosses only one cylinder between two scattering points. In scenarios where the photon crosses several cylinders, for example when nested cylinders are used to model radially changing properties of the hole ice, or when instantly-absorbing cylinders are used to model cables within a hole-ice cylinder, this algorithm is not able to calculate the corrections properly because it would interpret the interaction factors f not only relative to the surrounding bulk ice but also relative to the cylinders the photon has already crossed in this simulation step.

A different approach to adding propagation through hole-ice cylinders to the standard-propagation algorithm is to treat hole-ice cylinders the same way as other media such as ice layers rather than accounting for the hole-ice effects using subsequent corrections. This second approach gets rid of the above shortcomings, but at the cost that the standard-CLSIM algorithm that handles the propagation through different media needs to be re-written. This second approach is described in the following section.

5.4 Algorithm B: Hole-Ice Propagation by Generalizing the Previous Medium-Propagation Algorithm

A second approach to adding propagation through hole-ice cylinders to the standard propagation algorithm (section 5.2.2) implemented in CLSIM is to rewrite the part of the standard algorithm that propagates the photon through different media, aiming to make this algorithm more generic and support hole-ice cylinders at the same level as other structures such as ice layers.

The source code of the implementation of this second approach can be found in appendix A.10, in the folder `algorithm_b` on the CD-ROM, as well as in the code repository at https://github.com/fiedl/clsim/tree/sf/hole-ice-2018/resources/kernels/lib.

In order to propagate a photon through different media, the standard propagation algorithm converts the randomized number of interaction lengths N into a geometrical distance $X := \sum_i n_i \lambda_i,\ N = \sum_i n_i$ (equation 7 on page 41) based on the interaction lengths λ_i of the different media. In its current implementation in CLSIM, this conversion is hard-coded in a way that does support medium changes in the form of tilted ice layers but does not support other shapes such as cylinders of hole ice.[9] In the approach described in this section, the medium-propagation algorithm is re-implemented in a way that does support cylinder-shaped media.

Task The task of this new **medium-propagation algorithm** is to determine in each simulation step, which media, including ice layers as well as hole-ice cylinders, are on the photon's path along its current direction and to convert the randomized scattering and absorption length budgets to geometrical distances according to the interaction lengths of the different media.

Context The new medium-propagation algorithm is inserted into the CLSIM propagation algorithm's simulation step right after CLSIM has randomized the number of scattering lengths to the next scattering point, replacing CLSIM's original medium-propagation algorithm. After the medium-propagation algorithm follows the check whether the photon hits an optical module on the way from the current scattering point A to the next scattering point B.

The medium-propagation algorithm takes the following input parameters: Current photon position at point A, photon direction after scattering at point A, a list of the

[9] A comparison of the flow charts of standard CLSIM's media-propagation algorithm and the new media-propagation algorithm is given in appendix A.7.

hole-ice cylinders with their coordinates, radii, scattering lengths and absorption lengths, a list of ice layers with their coordinates, scattering lengths and absorption lengths, the randomized number of scattering lengths to the next scattering point, the remaining number of absorption lengths to the absorption point, and the remaining geometric distance to absorption. As output parameters, the medium-propagation algorithm returns the updated number of remaining absorption lengths to the absorption point, the geometric distance to the next scattering point, and the updated remaining geometric distance to absorption.

Procedure The new algorithm works with a generic list of media. For each medium, the algorithm stores the scattering length, the absorption length, and the distance along the photon's direction from the current photon position to the medium's border. This way, the algorithm knows in which distance from the current position which medium properties take effect. Due to this generic structure, homogeneous media with all kinds of shapes could be added to this list. In this first implementation, the algorithm adds un-tilted ice layers and cylinders, which model hole ice and cables. Tilted ice layers as well as absorption anisotropy are not re-implemented in this first implementation, but may be added later.[10] After adding ice layers and hole-ice cylinders to the media list, the algorithm sorts the list by ascending distance from the current photon position in order to have the list in the proper order for the following calculations. Finally, the algorithm loops over the media list and calculates the geometrical distances to the next scattering point and to the absorption point just like the previous media-propagation algorithm has done for ice layers only.

Media Loop In the loop over the different media along the photon's direction, the algorithm calculates the contribution of the media to the geometrical distances to the next scattering point and to the absorption point. The flow chart of the whole media loop is shown in figure 20. The flow chart for the calculation of one medium's contribution is shown separately in figure 21.

Contribution of the Last Medium As shown in figure 20, the last medium is handled separately as the contribution of the last medium is not determined by the size of the medium but by the amount of scattering and absorption budget left. The contributions of the first $(m-1)$ media to the geometrical distance sum up to $\sum_{i=1}^{m-1} x_i$ where each contribution $x_i := n_i \lambda_i$ determines how much

[10] At the time of writing, ice tilt and absorption anisotropy are not re-implemented, yet. To check the current state of this issue, see https://github.com/fiedl/hole-ice-study/issues/48.

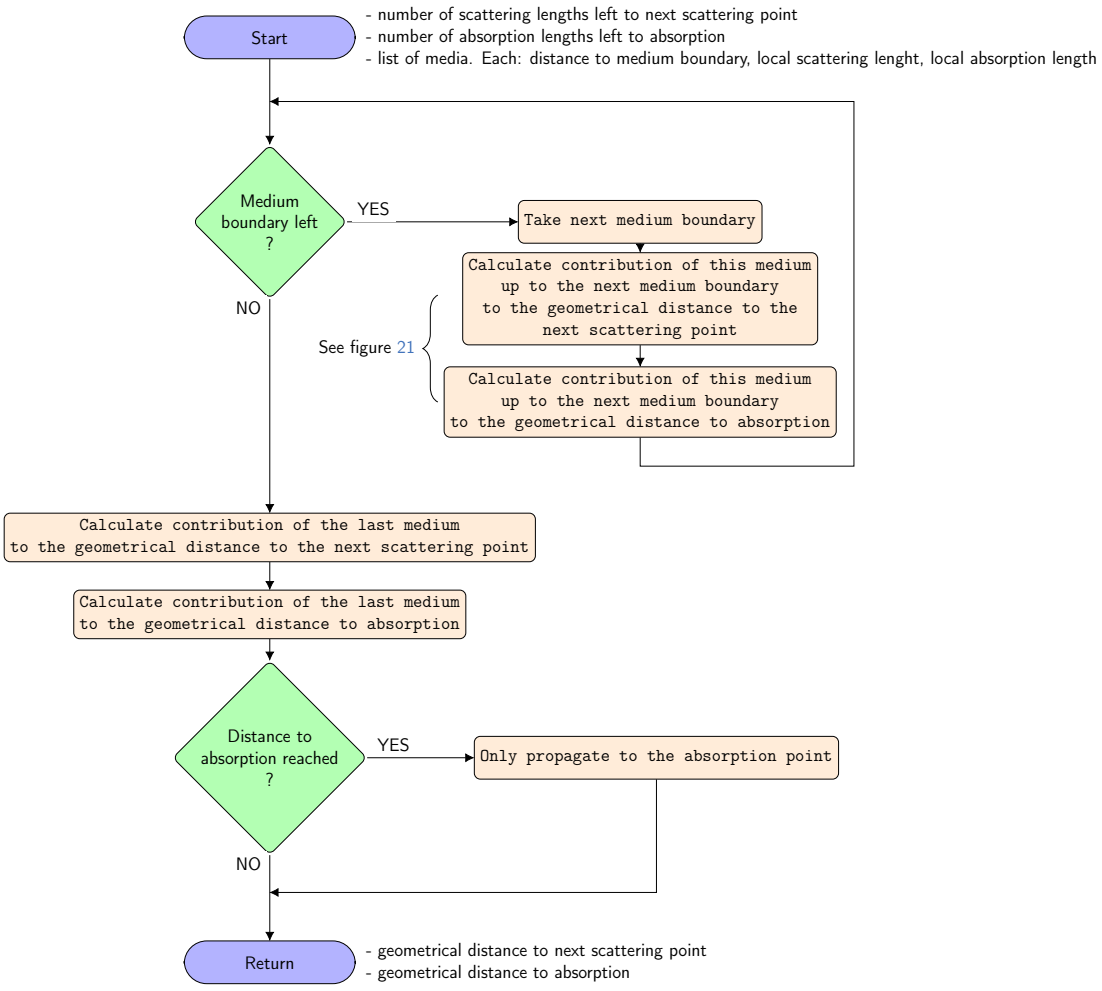


Figure 20: Flow chart of the part of the new media-propagation algorithm that loops over list of media and calculates geometrical distances to the next scattering point and to the absorption point.

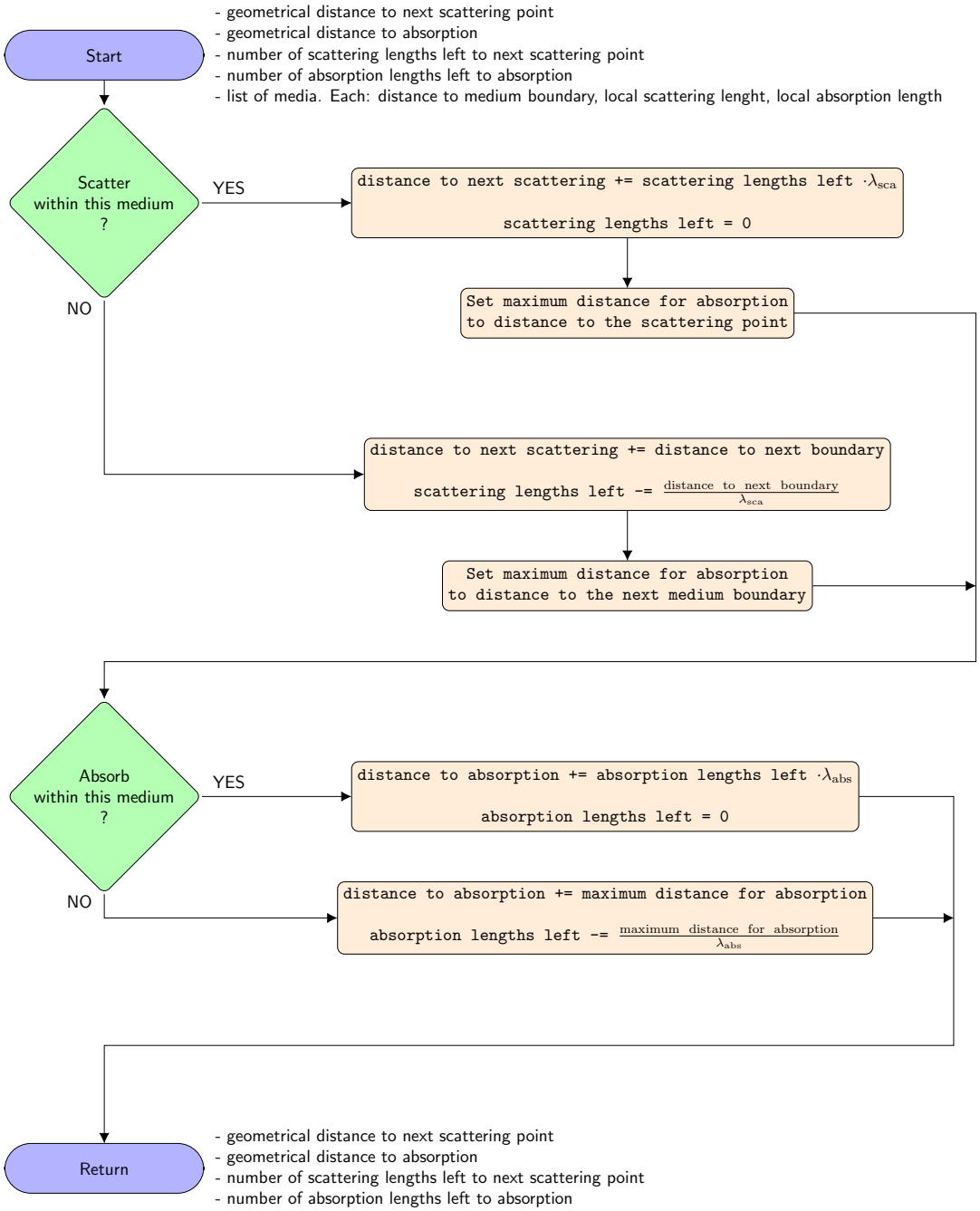


Figure 21: Flow chart of the part of the new media-propagation algorithm that calculates the contribution of one specific medium to the geometrical distances to next scattering point and to the absorption point.

of the budget $N := \sum_{i=1}^{m} n_i$ is spent. After the first $(m-1)$ media have been crossed, $n_m := N - \sum_{i=1}^{m-1} n_i$ of the budget is left for the last medium. Thus, the photon travels a distance of $x_m := n_m \lambda_m$ in the last medium where it reaches the interaction point.

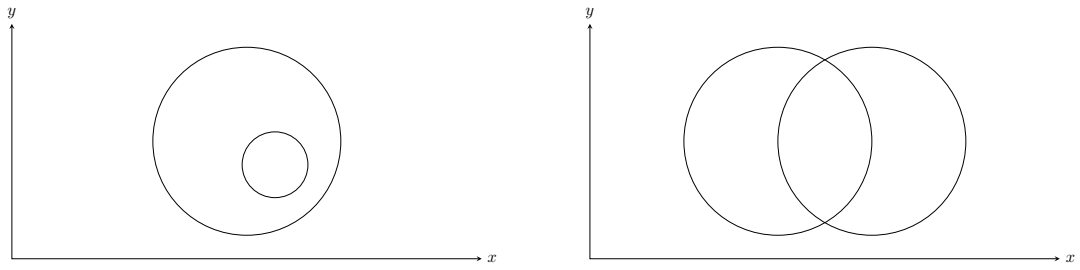
Absorption Before Reaching the Scattering Point When calculating the distance to the next scattering point based on the scattering length budget, if the photon's absorption length budget is spent before reaching the next scattering point, or equivalently, if the distance from the current scattering point A to the absorption point is shorter than the distance from A to the next scattering point, then the next trajectory point B is set to be the absorption point as shown in figure 20. The final trajectory point of the photon is its point of absorption.

Scattering Before Reaching the Absorption Point When calculating the contribution of a medium to the remaining distance to the absorption point as shown in the lower half of figure 21, the algorithm needs to know what distance γ within that medium contributes to spending the absorption length budget. If the photon does not scatter within this medium, the distance γ is just the distance from one medium border to the next medium border along the photon direction.[11] If the photon does scatter within this medium, the distance γ is set to the distance to the scattering point[12] as shown in the upper half of figure 21.

Order and Precedence of Cylinders When nesting or overlapping cylinders as shown in figure 22, the algorithm needs to determine which ice properties to apply for a photon at a position within both cylinders. In the left part (a) of the figure, the intuition might be: The smaller cylinder models a special area within the more general outer cylinder and, therefore, should take precedence. In the right part (b) of the figure, however, both cylinders have the same size. Here it is unclear which ice properties to apply for a position within the cylinders' overlap. To make this situation unambiguous, the algorithm defines that the media added to the list later take precedence over media added earlier. Therefore, when modeling nested cylinders, the algorithm requires to add the larger cylinder before adding the smaller cylinder to the definition list.

[11] This corresponds to the termination point being set to the second intersection point, $C = X$, in Case 4 and Case 5 of the hole-ice-correction algorithm described in section 5.3.

[12] This corresponds to the termination point being set to the point of the other interaction in Case 4 and Case 5 of the hole-ice-correction algorithm described in section 5.3.



(a) In a case where the inner cylinder is smaller, intuition suggests that the inner cylinder represents a special area within a more general area and should take precedence for a photon located within both cylinders.

(b) If both overlapping cylinders have the same size, the choice which properties should be applied to a photon at a position within the overlap of both cylinders, is arbitrary.

Figure 22: Two-dimensional projection of nested or overlapping hole-ice cylinders with different ice properties. If a photon is at a position within the overlap of both cylinders, which ice properties should be applied for the propagation? The algorithm defines that the cylinder added last takes precedence.

Pros and Cons This second approach allows to define cylinder-shaped volumes with absolute scattering and absorption lengths, which meets the requirements of the current understanding of the hole-ice phenomenon. Simulation steps over large z-distances that cross hole-ice cylinders are handled well by this algorithm in contrast to the previously described hole-ice-correction algorithm (section 5.3). Additionally, this algorithm supports nested or overlapping cylinders, which can be utilized to model bubble column together with the drill-hole column as well as the string's cable. Ice tilt and absorption anisotropy have not been ported to this algorithm, yet, but can be added later on.[13]

The down side to this approach is that an important part of standard-CLSIM's media-propagation algorithm needed to be re-implemented, requiring evidence that the new algorithm is still doing what is expected. In order to provide this evidence, section 5.6 describes a series of cross checks to make sure the new algorithm results in the expected scattering and absorption behavior. Also, standard-CLSIM's medium-propagation has been ported to the new interfaces as drop-in replacement for the new medium-propagation algorithm, such that one can switch between both algorithms without a code rollback.[14]

A comparison to other methods to account for the hole-ice effect is presented in section 7.1.

[13] See: https://github.com/fiedl/hole-ice-study/issues/48.

[14] A guide on how to switch the medium-propagation algorithm is presented in appendix A.12.

	Algorithm (a)	**Algorithm (b)**
Approach	Leave CLSIM medium propagation as it is and add **hole-ice effects as correction** afterwards	Unify CLSIM medium propagation through layers and hole ice: Treat them as **generic medium changes**
Hole-ice properties	defined relative to bulk-ice properties	defined absolute
Pros	+ Small surface area of hole-ice code, well testable through unit tests + Standard CLSIM almost untouched	+ Supports nested cylinders and cables
Cons	– Current understanding of hole-ice suggests defining hole-ice properties absolute rather than relative	– Needed rewrite of CLSIM's medium-propagation code – Ice tilt and ice anisotropy not re-implemented, yet

Table 1: Comparison of the hole-ice-correction algorithm (a) presented in section 5.3 and the new generic medium-propagation algorithm (b) presented in section 5.4.

5.5 Technical Issues and Optimizations

Running a simulation with millions of photons propagating and scattering, has to meet certain performance requirements that make sure that the simulations of interest can be run in a reasonable time on current hardware. On the other hand, current graphics processing units (GPUs) afford the opportunity to utilize parallelization capabilities. Both impose technical challenges. To deal with the necessary complexity and, in particular, to make sure that a complex algorithm does what it is supposed to do, unit tests and cross checks are used as will be described in the next section 5.6. This section focuses on dealing with computational limits, introducing the techniques of GPU parallelization, cluster parallelization, thinning, and performance optimizations.

Simulation-Step Parallelization Each simulation step consists of only a few operations: Scattering the photon, checking for absorption, checking for collision with a detector module, and propagating the photon to the next scattering position (see 5.2.2). This small set of operations can be parallelized, performing the same calculations and operations simultaneously for several thousands of simulation steps, each with different values such as coordinates and directions, on graphics processing units, which are designed for exactly this kind of work. Figure 23 illustrates this parallelization of simulation steps.

This parallelization leads to a performance improvement by factors of 150 and more when running the propagation simulation on a single GPU compared to running the same simulation on a single CPU core [Chi14]. Standard CLSIM as well as the hole-ice extended CLSIM can be run on CPUs, GPUs, or both using OPENCL as abstraction layer. In practice, simulating 1000 photons for visualization purposes and recording their paths, which requires a lot of memory, works best on a CPU. Simulating millions of photons, but only recording their final position, where the photon is absorbed within the ice or has it an optical module, works best on GPUs.

Cluster Parallelization A different technique, which adds another layer of parallelization on top of the simulation-step parallelization, is to run the same simulation on a cluster of machines, each equipped with one ore several GPUs. Each machine runs the same simulation, but with different parameters, such as different radii and scattering lengths of the hole-ice cylinders. This study uses this technique for parameter scans (sections 6.3 and 6.7) where the same simulation is performed for different parameter sets in order to find a specific parameter set which best fits some kind of external requirement.

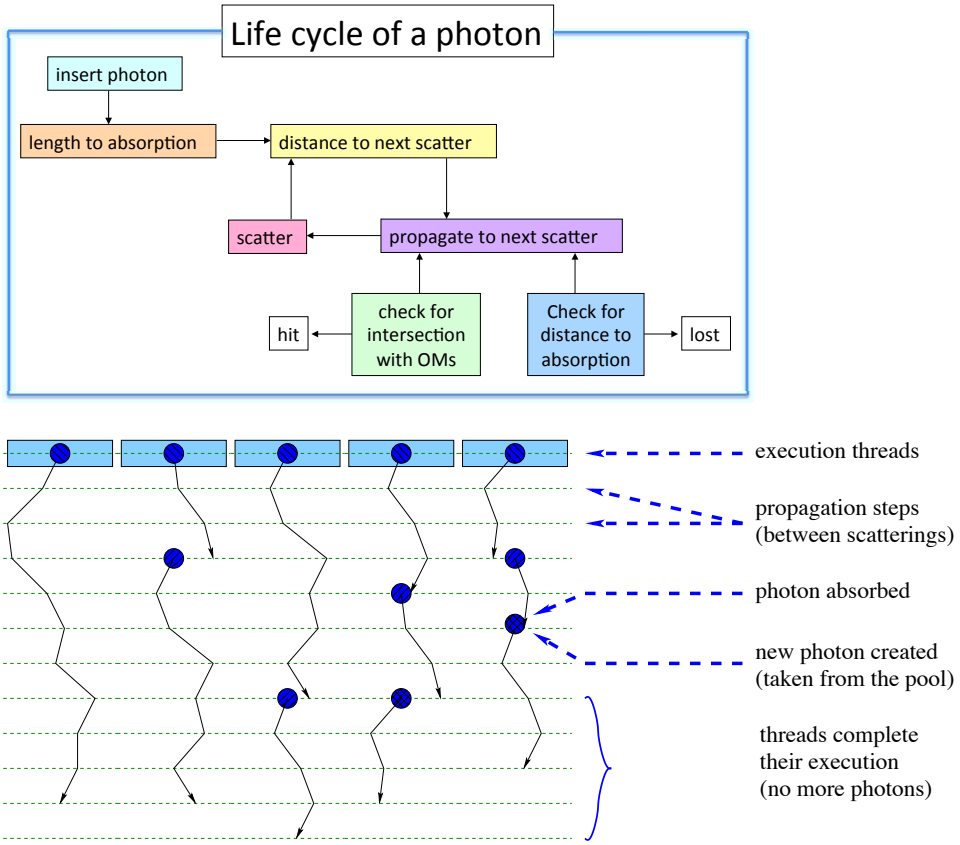


Figure 23: Parallelization of simulation steps: Tracking of photons entails the computationally identical steps: Propagation to the next scatter, calculation of the new direction after scatter, and evaluation of intersection points of the photon track segment with the detector array. These same steps are computed simultaneously for thousands of photons. (Image and caption taken from [Chi14], figure 1.)

Issues Concerning Graphics Processing Units Running and developing a simulation software for graphics processing units poses challenges specific to this architecture. A list of technical issues to watch out for in follow-up studies is provided in appendix A.15.

Parallel-Programming Optimizations Applying parallel-programming optimization techniques listed in section 4.2 may help to improve the performance of the algorithm that runs on the graphics processing units.

When optimizing code, AHMDAL'S LAW gives the theoretical limit on how much a process can be sped up by optimizing one of its components. [Don14]

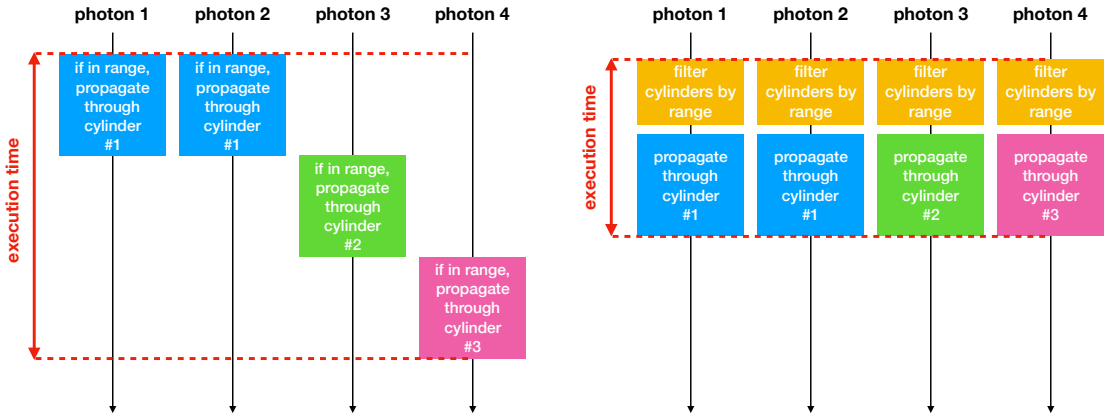
$$s_{\text{total}} = \frac{1}{(1-p) + {}^{p}/_{s_{\text{comp}}}}, \quad s = \frac{t_{\text{before}}}{t_{\text{after}}}, \quad \lim_{s_{\text{comp}} \to \infty} s_{\text{total}} = \frac{1}{1-p}$$

Here, s_{total} is the resulting speedup of the whole process, which is defined as the fraction of the time the process takes before the optimization and the time the same process takes after the optimization has been applied. p is the portion of execution time that the component requires before optimizing. s_{comp} is the speedup the the component that is optimized achieved by the optimization. Even in the limit that the component that is optimized takes zero time after applying the optimization, $s_{\text{comp}} \to \infty$, the speedup of the whole process is limited. Also, HOUSE and WYMAN [Don14] recommend to first write working code, and then optimize in a second step.

One key concept to efficient parallel programming is to avoid executional threads being idle, waiting for other threads to finish before resuming. One application of this concept in the hole-ice algorithm is to **filter hole-ice cylinders by range** in a **separate loop**. The algorithm needs to check in each simulation step, which hole-ice cylinders are in range of the photon and therefore should be considered when calculating the propagation through the hole ice. Without this optimization, the cylinders are processed one by one in each simulation step. While, in each iteration of the cylinder loop, all photons in range of one specific cylinder are propagated through that cylinder, all other photons not in range of the cylinder, wait idly and thereby waste computation time. This is illustrated in figure 24 (a).

Even though the optimization adds time for the additional loop where the indices of the cylinders in range are saved into a local array, the main loop where the propagation through each cylinder is handled is much faster now, because it does not need to iterate over all cylinders in the detector but only over the cylinders in range of the photon and improves parallelization as illustrated in figure 24 (b).

The implementation of this optimization is documented in `issues/30` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/30.



(a) Checking if a cylinder is in range in the same loop as propagating through that cylinder. The cylinders are processed successively. Lots of photons are idle.

(b) Checking if a cylinder is in range in a separate loop. The main loop where the photons are propagated through the cylinders does not loop over all cylinders but only over cylinders marked as in range for each photon respectively.

Figure 24: Processing the propagation of photons through hole-ice cylinders in parallel. While filtering cylinders by range separately adds additional executional time, it improves parallelization and thereby saves execution time overall.

Another optimization used in the hole-ice algorithm is to **order mathematical cases by their frequency of application**. In a case-by-case analysis, each executional thread skips the additional case checks when its case is determined. If the checks for common cases are handled first, there is a good chance that all of the parallel threads fall into common cases such that the checks for the rare cases can be skipped for the whole thread block. This technique is used in the implementation of the geometric cases for both the hole-ice-correction algorithm (5.3) and the new media-propagation algorithm (5.4).

GPU-Specific Optimizations In addition to the general principles of parallel programming (section 4.2), House and Wyman [Don14] list practical tips for optimizing GPU-ray-tracing code. Most notably, passing data structures by reference rather than by value proved to have significant performance impact because **allocating memory on a GPU is expensive**. This also includes simple structures like arrays of numbers, and even re-using the same array for several calculations rather than re-defining it. This technique has caused a performance improvement on GPUs by a factor of six in this study.

This optimization of re-using arrays is documented in `issues/70` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/70.

Optimizing mathematical operations on paper is an important technique as each calculation operation that can be cut down on in a procedure that saves time when executing the procedure many times. In addition to simplifying equations mathematically, avoiding numerically expensive operations such as square roots improves performance.

GPUs support 4-dimensional vectors as native data types as well as native vector operations such as the dot product. Expressing equations in terms of vectors rather than simplifying equations on the coordinate level, resulted in improved performance and numerical accuracy.

This optimization of using vector types rather than scalar types is documented in `issues/28` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/28.

Performance Profiling OWENS recommends profiling tools like GPROF, VTUNE, or VERYSLEEPY [Owe13]. But also basic techniques like printing time differences between executional blocks can help to isolate performance problems. Figure 25 shows the execution time profile of the simulation step of the new medium-propagation algorithm.

Note that printing a profiling result may be more expensive as the computation that is to be profiled itself. In this study, printing the time difference of the beginning and the end of a simulation step takes five times as long as the simulation step itself.

Profiling the simulation step is documented in `issues/69` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/69.

Performance Optimizations for Production Use After implementing and debugging a new algorithm, a lot of debug outputs and other development tools are no longer required when running simulations. For those production simulations, the debug features can be turned off by **compiling the ICECUBE simulation framework for production** use rather than for debugging. Thereby, simulations

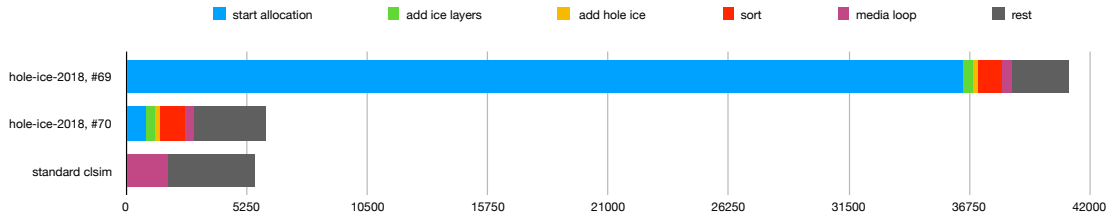


Figure 25: Performance profile of the average simulation step in arbitrary units (GPU clock cycles). Each row of the chart represents a different algorithm. In the first row, which shows an early stage of the new medium-propagation algorithm, the initial memory allocation within each simulation step takes a lot of time, especially when comparing to standard CLSIM (row 3). The second row shows the profile of the new medium-propagation algorithm after a memory issue has been fixed. Notable operations in each simulation step are the initial memory allocation for local variables, adding ice layers, adding hole-ice cylinders, sorting medium boundaries by distance to the photon and looping over the media to convert the interaction budget into geometrical distances.

run significantly faster as debug outputs often cost more time than the actual simulation itself.

Also, activating **kernel caching** on the production machines may improve simulation performance. However, this should only be used when always using the same propagation kernel, and should be avoided when switching between kernel versions or turning kernel features on and off, because the kernel cache will not be reliably reset automatically when changing kernels.

Another way to save computational time when performing simulations, especially large cluster scans, is to scale down the number of photons using a **thinning factor**. A representative fraction of the photons is propagated in the simulation, and then the simulation results, for example the numbers of hits in each detector module, are scaled up accordingly.

- How to switch between a debug build and a production build is documented in `issues/73` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/73.
- How to enable or disable kernel caching is documented in `issues/15` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/15.
- Using a thinning technique in flasher parameter scans is documented in `issues/59` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/59.

5.6 Unit Tests and Cross Checks

There are several techniques to ensure that the propagation algorithm produces results that meet the expectations. First, CLSIM itself implements a number of tests that run in each simulation and check computational quantities for consistency. [Kop17]

Secondly, one can define scenarios with known quantities for single photons and check externally whether the algorithm produces the expected results for those single photons. This is done with *unit tests* and is presented in section 5.6.1.

Thirdly, checks for a sample of photons can be devised where all photons should behave the same way, for example all photons should be absorbed by a medium that is configured for instant absorption. This check is presented in section 5.6.2.

Finally, a sample of propagated photons can be examined regarding their statistical properties, for example the sample's distributions of quantities like arrival time and path length. These tests are presented in sections 5.6.3 to 5.6.7.

5.6.1 Unit Tests With Single Photons

In order to verify that individual components ("units") of the implementation produce results as expected, unit tests were implemented for the algorithm that calculates intersections as well as for the hole-ice-correction algorithm.

The unit tests for the intersection algorithm can be found in the folder `unit_tests` on the CD-ROM and at https://github.com/fiedl/clsim/blob/sf/hole-ice-2017/resources/kernels/lib/intersection/intersection_test.c.

The unit tests for the hole-ice-correction algorithm can be found in the folder `unit_tests` on the CD-ROM and at https://github.com/fiedl/clsim/blob/sf/hole-ice-2017/resources/kernels/lib/hole_ice/hole_ice_test.c.

Task The task of unit tests is to test individual components of a software. In this case, the tests execute separate functions of the new algorithms with fixed input parameters and check whether the functions return the expected results, which, in preparation of the test, have been obtained by other means, either by a separate program, using a separate programming language, a separate algorithm, or via calculations by hand.

In this study, unit tests define single photons that cross hole-ice cylinders in a pre-defined way and check whether the intersection algorithm determines the correct intersection points, whether the 2d-3d projections are handled correctly, and whether the hole-ice-correction algorithm determines the expected corrections for the geometric distances according to the hole-ice parameters provided by the test scenario. Extreme examples can be designed to produce simple results that can be calculated by hand and verified by intuition. For example, for hole-ice cylinders configured for instant absorption, the absorption point is identical to the intersection point of the photon trajectory and the cylinder. More complex examples require more involved calculations by hand or using other software like PYTHON scripts or computer algebra tools for calculating intersections.

Testing Framework This study uses the GTEST[15] testing framework. This framework has been chosen due to its good documentation, wide adoption and slim architecture that made it possible to use the framework to test individual components of the new source code without interfering with the rest of the ICESIM framework.

The introductory documentation of GTEST can be found at https://github.com/google/googletest/blob/master/googletest/docs/primer.md.

Limitations of Unit Tests Unit tests are most useful to ensure stability when adjusting, refactoring or rewriting software components. Even after replacing a large part of the source code, unit tests can make sure that the software still produces the same results as before. Tests also allow for so-called test-driven development where the expected results are specified first, and then the software is built or changed iteratively until it produces the expected results. This technique works best when the components to be tested have small interfaces because the technique requires the tests to provide all input parameters for the components to be tested.

Unexpected issues may arise when unit tests are run on another architecture as the production code. For example, the unit tests used in this study were insensitive to certain driver issues and numerical issues[16] that did only occur when running the software on the GPUs of the computing cluster and did not occur when running the unit tests on the local CPU of the development machine.

[15] Google Test Framework, gtest, https://github.com/google/googletest

[16] See section A.15.

Unit tests are, by design, also insensitive to high-level problems. Each individual component may produce the results expected from this component while still issues may arise when components are not tied together correctly, or because the expectations for an individual component may be wrong, which becomes apparent only when adopting a larger perspective rather than the focused perspective on individual components.

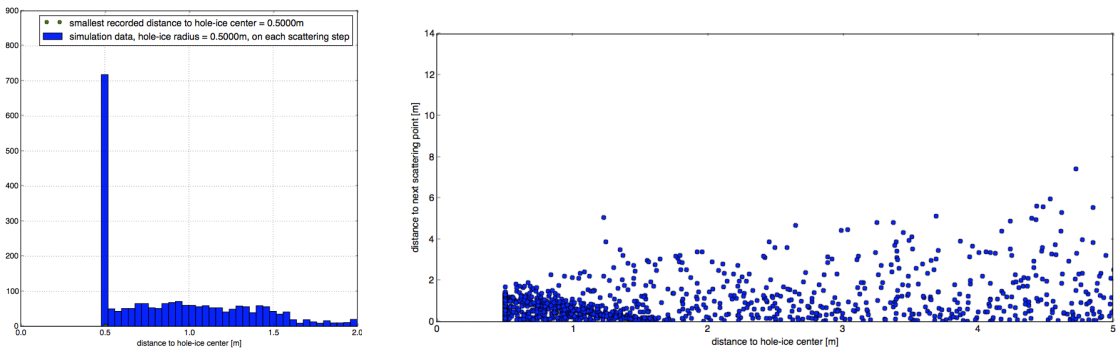
To increase confidence in the new software, therefore, in addition to unit testing, high-level cross checks need to be performed as described in the following subsections.

5.6.2 Instant-Absorption Tests

A simple high-level test can be performed by starting several photons towards a cylinder, which is configured for instant absorption. After propagating the photons using the new media-propagation algorithm and recording the path of each individual photon, one can verify the results making sure than no photon position is recorded inside the instant-absorption cylinder (figure 26). Additionally, one can visualize the simulation using the STEAMSHOVEL event viewer to verify that no photon enters the instant-absorption cylinder as shown in figure 27.

The implementation of this cross check can be found in `issues/67` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/67, `issues/47` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/47, and `issues/22` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/22.

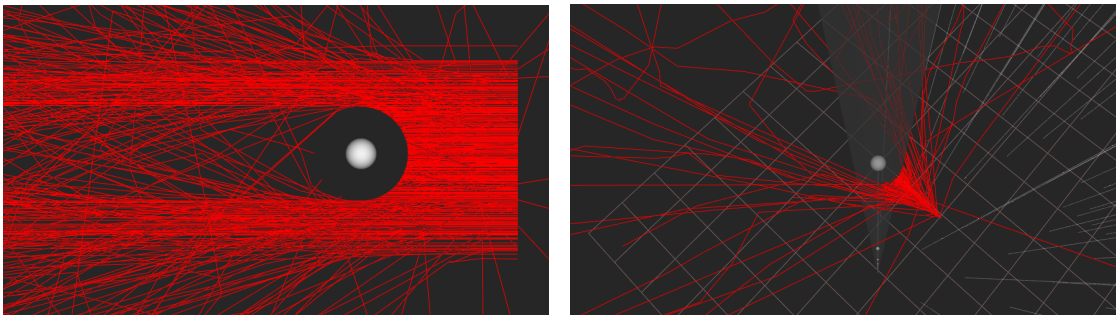
A related, but more complex scenario is starting photons within two nested cylinders where the inner cylinder is configured for a short scattering length and the outer cylinder is configured for instant absorption (figure 28).



(a) Distribution of the distance to the hole-ice center, evaluated for each scattering step.

(b) Distance to the next scattering point and distance to hole-ice center for each scattering step.

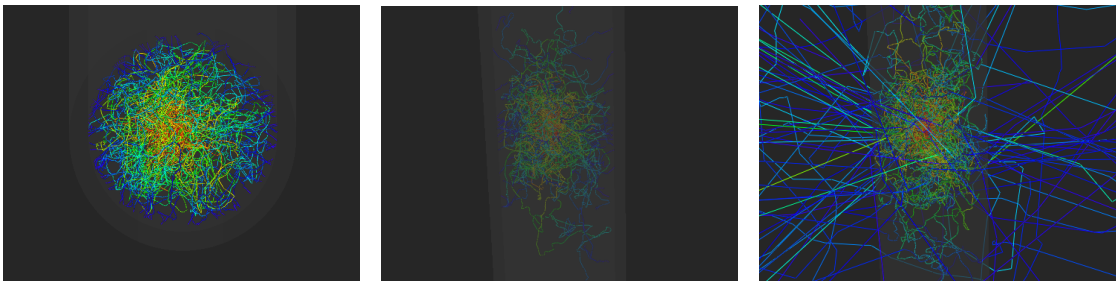
Figure 26: Analyzing the simulation of photons propagating towards a hole-ice cylinder configured for instant absorption. No photon can be recorded within the hole-ice cylinder's radius.



(a) Starting from different positions into the same direction. View from above.

(b) Starting from the same position into different directions.

Figure 27: Visualizing an instant-absorption test using the STEAMSHOVEL event viewer. In the simulation, photons are started towards a cylinder configured for instant absorption. If the medium-propagation algorithm works as expected, no photon can get inside the cylinder.



(a) View from above.

(b) View from the side.

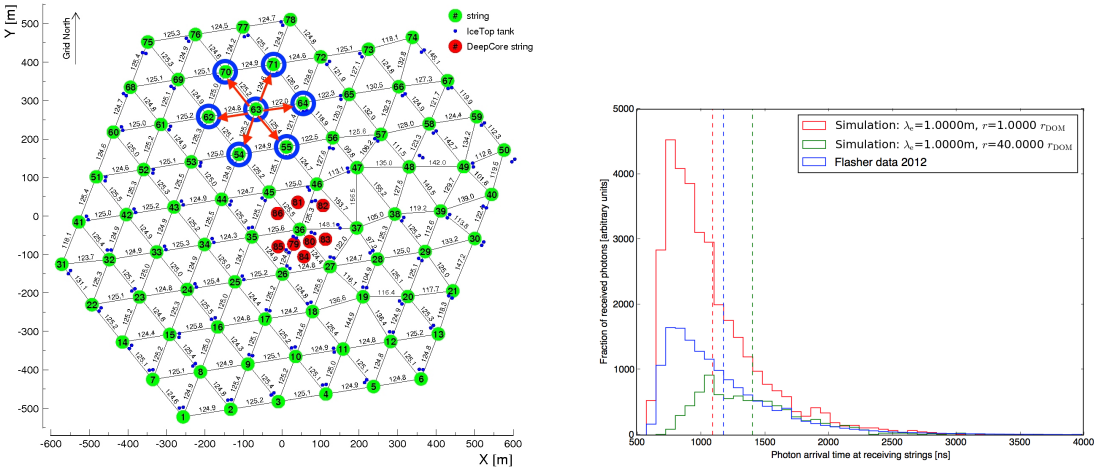
(c) Instant absorption of the outer cylinder turned off.

Figure 28: Visualizing an instant-absorption test where photons are started within two nested cylinders. The outer cylinder is configured for instant absorption. No photon can pass through the area between both cylinders unless the instant absorption is turned off.

5.6.3 Arrival-Time Distributions

One way to test the behavior of statistical properties of a sample of photons is to plot the arrival-time distribution of photons traveling from a central position to receiving optical modules around the starting position. Figure 29 shows arrival-time distributions for this scenario being carried out with a flasher experiment compared to two simulations with different hole-ice configurations each.

> The source for the simulations and for creating these histograms can be found in `issues/91` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/91.



(a) Top view of the detector strings in this simulation. The photons are started at the middle optical module of string 63, and are received by the optical modules of the surrounding strings 70, 71, 64, 55, 54, and 62.

(b) Photon-arrival-time distributions for different hole-ice configurations. The dashed lines indicate the mean arrival times.

Figure 29: Creating arrival-time distributions for photons propagating through hole ice. The photons are started at a central position and detected by optical modules around the starting position.

An estimation based on the mean scattering angle and the radius of the hole-ice cylinder suggests that one would need a difference of the hole-ice cylinder's radius of several meters to cause a difference in the mean arrival time on the order of 100 ns. While this is no realistic scenario as the drill hole's radius is only about 50 cm, the simulation allows to use extreme scenarios to observe the effects more noticeable.

From the comparison of the arrival-time distributions, note that the distribution that is based on real data is comparable to the ones based on simulations, which

suggests that the new algorithm does not introduce unexpected, unphysical effects to the photon propagation.

In the simulation where the hole-ice cylinders have a much larger radius, the effects of the hole ice should be stronger. Indeed, the comparison of the distributions shows three effects to expect from this assumption: First, in the simulation with stronger hole-ice effect, less photons arrive at the receiving optical modules in total as more photons are scattered away by the hole ice. Secondly, for stronger hole ice, the photons arrive later at the receiving optical modules, because they spend more time scattering randomly within the hole-ice cylinder before reaching the receiving optical module. Thirdly, for stronger hole ice, the left-hand side of the distribution histogram is less steep, because with more scattering points on the photon's path, there is a larger number of possible paths, which leads to the arrival time being more distributed.

While these effects confirm the expectations qualitatively, examining other high-level observables allows to compare expectations to the simulations' results even quantitatively. This is the subject of the following sections, beginning with observing the photons' path-length distributions.

5.6.4 Exponential Distribution of the Total Path Length

Let the **Total path length** of a photon be the summed distance from the position where the photon is created, along the photon's path, to the position where the photon is absorbed. The total path length X of photons that propagate in a medium with an absorption probability that is the same everywhere in the medium, for example within a confined volume within the bulk ice, is expected to follow an exponential distribution with a rate parameter corresponding to the absorption probability, or equivalently the absorption length λ within the medium.[17]

$$X: \quad P(X \in [x; x + \mathrm{d}x[) \propto e^{-x/\lambda}, \quad x \in \mathbb{R}_0^+ \tag{8}$$

When propagating photons within a hole-ice cylinder, their path length should also follow an exponential distribution, but with different absorption length λ^{H}, as long as the absorption length is sufficiently small such that most photons do not leave the cylinder before being absorbed.

This behavior can be verified using a simulation. In the simulation, a pencil beam of 10^4 photons is started within a hole-ice cylinder with a radius of 1.0 m. The photons are started with a distance of 0.9 m to the cylinder center towards the cylinder center. Let the effective scattering length within the cylinder be 100.0 m and the absorption length be 0.1 m. For each simulated photon, the total path length is recorded.

Figure 30: STEAMSHOVEL event display of the simulation of photons propagating and being absorbed within the hole-ice cylinder without medium transition. The photons are started within the hole-ice cylinder, which is represented by the grey cylinder, inwards. The detector module, which is represented by the white sphere in the center of the figure, is shown only to illustrate the size of the scenery and is configured not to interact with the simulated photons.

The implementation of this cross check can be found in `issues/64` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/64.

The distribution of the total path length should follow an exponential decay curve.

[17] If the photons do not depend on each other, the number of photons that are absorbed at a certain path length does only depend on the absorption probability, or equivalently on the absorption length λ, and the number of photons remaining at this path length. When a photon is absorbed, the number of remaining photons decreases. Thus, the change in the number of photons is proportional to the number of photons, causing the number of photons absorbed at a certain path length following an exponential distribution. See also appendix A.13.

From the rate of the decay, one should be able to read the absorption length λ (see appendix A.13).

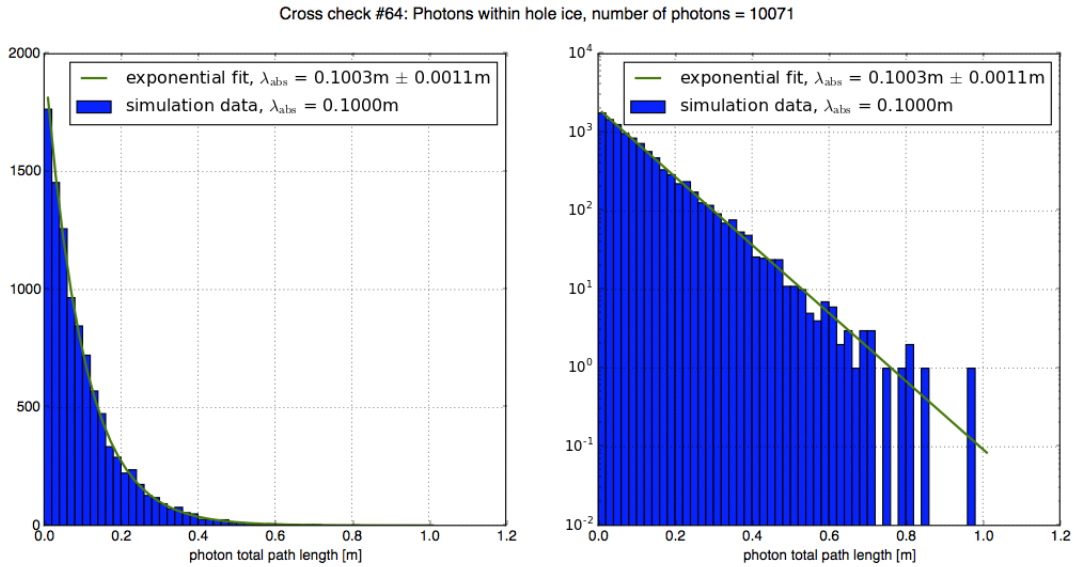


Figure 31: Distribution of the total path length of simulated photons both started and absorbed within a hole-ice cylinder using the new medium-propagation algorithm. The distribution follows an exponential curve. The fitted absorption length is $\lambda_{\text{abs}} = 0.1003\,\text{m} \pm 0.0011\,\text{m}$. The true absorption length used in the simulation is $0.1000\,\text{m}$.

Indeed, the simulation yields the expected distribution of the total path length (figure 31). Via a curve fit, the absorption length λ_{abs} can be determined for this simulation to be $\lambda_{\text{abs}} = 0.1003\,\text{m} \pm 0.0011\,\text{m}$, which is in accordance with the true absorption length of $0.1000\,\text{m}$ used in the simulation.

5.6.5 Piecewise Exponential Distribution of the Total Path Length for One Medium Boundary

This cross check aims to verify that the medium-boundary transition is handled correctly when photons enter a hole-ice cylinder from outside.

In a simulation, a pencil beam of 10^5 photons is started in a distance of 2.0 m to cylinder center towards the hole-ice cylinder with a radius of 1.0 m. The effective scattering length outside is set to 10^6 m, inside to 100 m. The absorption length outside is set to 1.0 m, inside to 0.10 m. As before, for each simulated photon, the total path length, from the starting point of the photon along the photon's trajectory to the point where the photon is absorbed, is recorded.

The implementation of this cross check can be found in `issues/65` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/65.

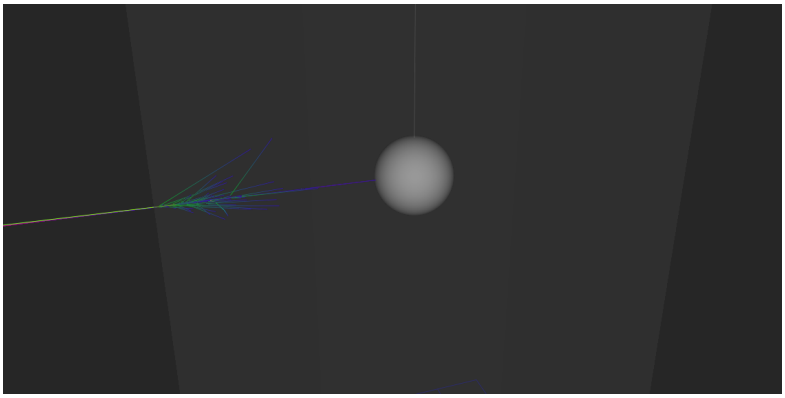


Figure 32: STEAMSHOVEL event display of the simulation of photons propagating through the hole-ice medium boundary. The photons are started outside and towards the hole-ice cylinder on the left hand side. As the scattering length within the cylinder is smaller, the photons scatter a lot more within the cylinder. The detector module is shown as a white sphere only to illustrate the size of the scenery and is configured not to interact with the simulated photons.

As the absorption lengths are different outside and inside the cylinder, the histogram should now follow two separate exponential curves: The left hand side of the histogram, which corresponds to the area outside the cylinder, should follow an exponential curve governed by the absorption length outside the cylinder. The right hand side, which corresponds to the area within the cylinder, should follow an exponential curve governed by the absorption length within the hole-ice cylinder.

Note that the histogram's bins are proportional to the number $m(x) := {}^{-\,\mathrm{d}n(x)}/_{\mathrm{d}x}$ of photons that are absorbed within the path length interval $[x; x + \mathrm{d}x[$, not the

number $n(x)$ of remaining photons. Thus, the histogram is expected to show a jumping discontinuity at the position of the medium border.[18]

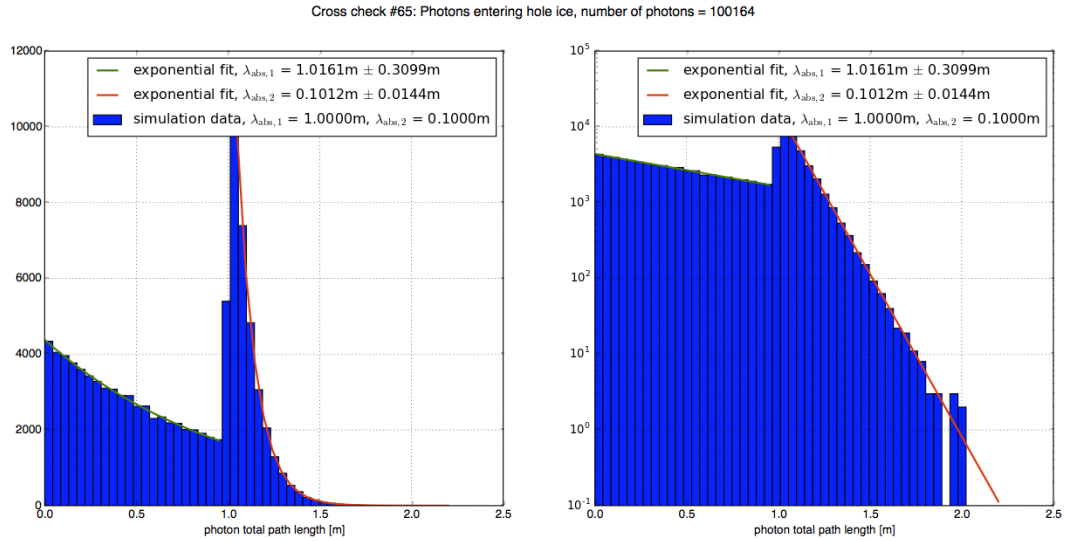


Figure 33: Distribution of the total path length of simulated photons started outside and absorbed within a hole-ice cylinder using the new medium-propagation algorithm. The distribution follows two exponential curves. The fitted absorption lengths are $\lambda_{\text{abs},1} = 1.016\,\text{m} \pm 0.310\,\text{m}$ and $\lambda_{\text{abs},2} = 0.101\,\text{m} \pm 0.014\,\text{m}$. The true absorption length in the simulation is set to 1.000 m outside, and to 0.100 m within the hole-ice cylinder.

The simulation yields the expected distribution of the total path length (figure 33). Via a curve fit, the absorption lengths $\lambda_{\text{abs},1} = 1.016\,\text{m} \pm 0.310\,\text{m}$ and $\lambda_{\text{abs},2} = 0.101\,\text{m} \pm 0.014\,\text{m}$ are in determined in accordance with the true absorption length outside of 1.000 m and within the hole-ice cylinder of 0.100 m set in the simulation.

[18] See also appendix A.13.

5.6.6 Piecewise Exponential Distribution of the Total Path Length for Two Medium Boundaries

This cross check aims to verify that the medium-boundary transition is handled correctly when photons enter and leave a hole-ice cylinder.

In a simulation, a pencil beam of 10^5 photons is started in a distance of 1.5 m to the cylinder center towards the hole-ice cylinder with a radius of 0.5 m. The effective scattering length outside and inside is set to 10^6 m. The absorption length outside is set to 1.0 m, inside to 0.75 m. As before, for each simulated photon, the total path length is recorded.

The implementation of this cross check can be found in `issues/66` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/66.

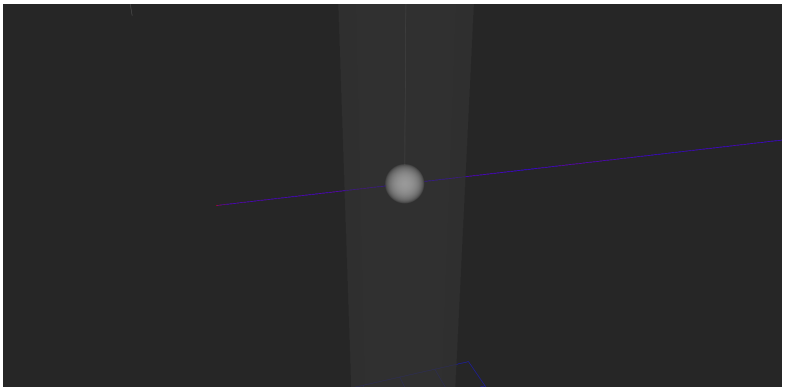


Figure 34: STEAMSHOVEL event display of the simulation of photons propagating through a hole-ice cylinder. As the scattering length is chosen to be very large, the photon path appears as a single line. The photons are started outside the hole-ice cylinder on the left hand side. Some of them are absorbed before entering the cylinder, some within the cylinder and some after leaving the cylinder on the right hand side. The detector module is shown as a white sphere only to illustrate the size of the scenery and is configured not to interact with the simulated photons.

As the simulated photons may now cross two different medium boundaries, the distribution of the path lengths should follow three exponential curves: On the left hand side of the histogram, which corresponds to the photons that are absorbed before entering the hole ice, the histogram should follow an exponential curve governed by the absorption length outside. In the middle, which corresponds to the photons that are absorbed inside the cylinder, the histogram should follow an exponential curve governed by the absorption length within the hole ice. On the right hand side, which corresponds to the photons that are absorbed after

leaving the hole-ice cylinder, the histogram should follow an exponential curve, again, governed by the absorption length outside the cylinder.

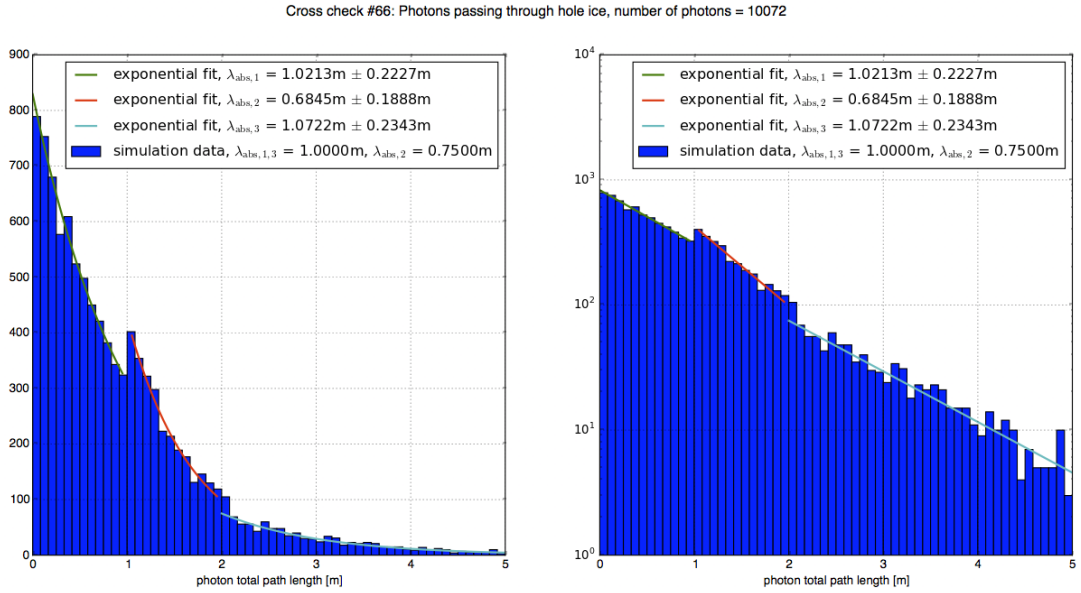


Figure 35: Distribution of the total path length of simulated photons, which start outside the cylinder, and may be absorbed before, within or after passing the cylinder, using the new medium-propagation algorithm. The distribution follows three exponential curves. The fitted absorption lengths are $\lambda_{\text{abs},1} = 1.02\,\text{m} \pm 0.22\,\text{m}$, $\lambda_{\text{abs},2} = 0.68\,\text{m} \pm 0.19\,\text{m}$, and $\lambda_{\text{abs},3} = 1.07\,\text{m} \pm 0.23\,\text{m}$. The true absorption length in the simulation is set to 1.00 m outside, and to 0.75 m within the hole-ice cylinder.

The simulation yields the expected distribution of the total path length (figure 35). The absorption lengths $\lambda_{\text{abs},1} = 1.02\,\text{m} \pm 0.22\,\text{m}$, $\lambda_{\text{abs},2} = 0.68\,\text{m} \pm 0.19\,\text{m}$ and $\lambda_{\text{abs},3} = 1.07\,\text{m} \pm 0.23\,\text{m}$ determined via a curve fit are in accordance with the true absorption length outside of 1.00 m and within the hole-ice cylinder of 0.75 m set in the simulation.

5.6.7 Distance to Next Scattering Point in Relation to the Distance From the Hole-Ice Center

The cross checks of the previous sections refer to the absorption length. To examine the scattering-length behavior of the photons simulated using the new media-propagation algorithm, this cross check observes the distance to the next scattering point in relation to the distance from the hole-ice center at each scattering point.

In a simulation, a plane wave of 100 photons is started towards a hole-ice cylinder. Let the hole-ice radius be 0.50 m. The scattering length within the hole-ice cylinder

is 0.06 m, the scattering length in the surrounding bulk ice is 1.48 m. For each scattering step, the distance to the next scattering point and the current distance to the center of the hole-ice cylinder are recorded.

The implementation of this cross check can be found in `issues/71` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/71.

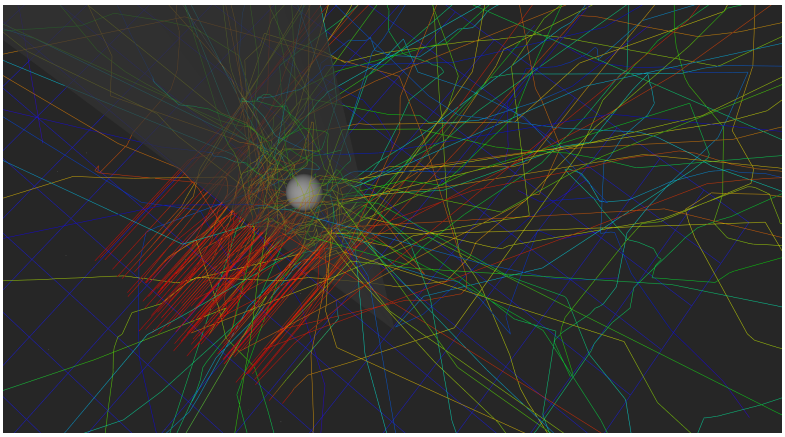
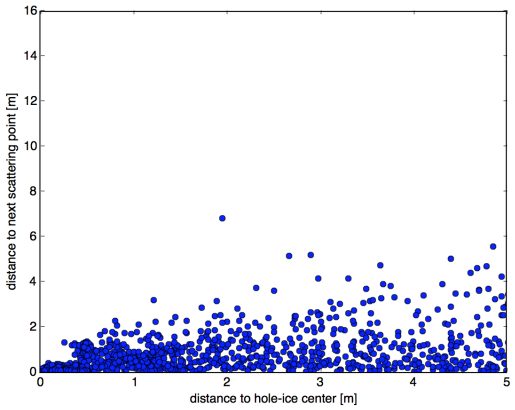


Figure 36: STEAMSHOVEL event display of a simulation of a plane wave of photons propagating towards a hole-ice cylinder. The scattering length within the hole-ice cylinder is different from the scattering length within the surrounding bulk ice.

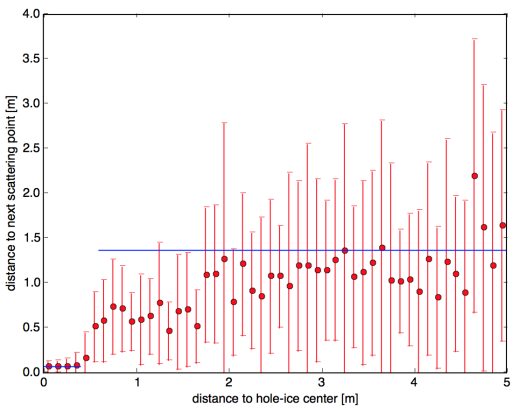
As the distance to the next scattering point should be exponentially distributed, governed by the scattering lengths within and without the hole-ice cylinder, the distance to the next scattering point should differ in the following three regions:

1. Well within the hole-ice radius, the distance to the next scattering point should be in the range of the scattering length of the hole ice.
2. Well outside the hole-ice radius, the distance to the next scattering point should be in the range of the scattering length of the surrounding bulk ice.
3. In the vicinity of the hole-ice-cylinder radius, there should be some kind of transition due to the complex geometrical simulation.

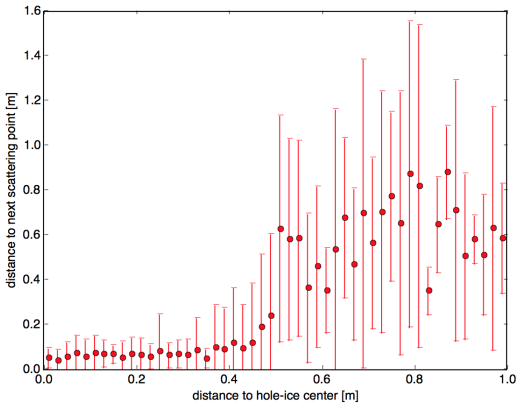
Assuming an exponential distribution of the distance to the next scattering point, the mean distance to the next scattering point within 80 % of the hole-ice radius is fitted to $0.07\,\mathrm{m} \pm 0.07\,\mathrm{m}$, the mean distance to the next scattering point outside of 120 % of the hole-ice radius is fitted to $1.37\,\mathrm{m} \pm 1.37\,\mathrm{m}$ (figure 37 b), which is in the range of the true scattering lengths of 0.06 m and 1.48 m inside and outside the hole ice respectively.



(a) All data points



(b) Averaged for bins of a width of 10 cm



(c) Magnification of the vicinity of the hole-ice border

Figure 37: For each scattering step of a simulation of photons entering a hole-ice cylinder, plot the distance to the next scattering step in relation to the current distance to the cylinder center. The mean distance to the next scattering point within 80 % of the hole-ice radius is fitted to $0.07\,\mathrm{m} \pm 0.07\,\mathrm{m}$. The mean distance to the next scattering point outside of 120 % of the hole-ice radius is fitted to $1.37\,\mathrm{m} \pm 1.37\,\mathrm{m}$. The true scattering length in the simulation is set to 0.06 m inside, and to 1.48 m outside the hole-ice cylinder.

When plotting the local average of the distance to the next scattering point for the vicinity of the hole-ice border (figure 37 c), the curve shows no abrupt jump but a transition between both domains as expected.

6 Examples of Application

This section shows examples of what can be done with the new hole-ice algorithms. Section 6.1 visualizes the scattering of photons within hole-ice cylinders with different scattering lengths. Sections 6.2 and 6.3 demonstrate scanning the effective angular acceptance of optical modules for different hole-ice parameters. Sections 6.4 to 6.6 show examples of shifting and nesting hole-ice cylinders as well as adding cables as cylinders of instant absorption. Section 6.7 shows an example of a flasher-calibration study. Section 6.8 demonstrates a flasher simulation with shadowing cable.

6.1 Visualizing a Hole-Ice Column With Different Scattering Lengths

One of the first things one can use the new propagation algorithms for, is to visualize the scattering behavior of photons entering a hole-ice cylinder.

Simulation In this example, 100 simulated photons are propagated towards and into a hole-ice cylinder. The photons are started from random positions within a 1m-by-1m plane in a distance of 1 m from the cylinder center in parallel towards the cylinder. (See figure 38 a.)

The cylinder radius is 30 cm. The scattering length within the bulk ice is $\lambda_{\text{sca}} = 1.3\,\text{m}$, corresponding to an effective scattering length of $\lambda_{\text{e}} = 21.7\,\text{m}$. The absorption length within the bulk ice is $\lambda_{\text{abs}} = 48.0\,\text{m}$.

The scattering length $\lambda^{\text{H}}_{\text{sca}}$ within the hole ice is defined relative to the bulk-ice scattering length: $\lambda^{\text{H}}_{\text{sca}} = f\,\lambda_{\text{sca}}$, $f \in \mathbb{R}^{+}_{0}$, for example $f = 1/10$ in figure 38 (c), $f = 1/100$ in figure 38 (d), and $f = 1/1000$ in figure 38 (d).

This simulation uses the hole-ice-correction algorithm described in section 5.3. The interaction with the optical detector module is deactivated in this simulation.

The configuration and implementation of this simulation is documented in `issues/40` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/40.

A script for configuring and running this and other simulations of this kind is provided at https://github.com/fiedl/hole-ice-study/tree/master/scripts/FiringRange.

Visualization The photon-propagation simulation records the starting point, each scattering point, and the final trajectory point of each photon. The photon trajectories can be visualized using the STEAMSHOVEL event display software.

Photon trajectories are represented as lines connecting the scattering points. The colors of the trajectory segments indicate simulation steps. A red trajectory segment represents a photon that has just been created. The blue end of the spectrum represents a photon that is about to be absorbed.

Figure 38 shows the STEAMSHOVEL visualizations of the simulation described above for different hole-ice scattering lengths.

An animated visualization of this photon-propagation simulation can be found at https://youtu.be/BhJ6F3B-I1s, and in the folder `video` on the CD-ROM.

Observations If the scattering length of the hole ice is the same as the scattering length of the bulk ice (figure 38 a and b), the photons pass right through the cylinder as if it were not there.

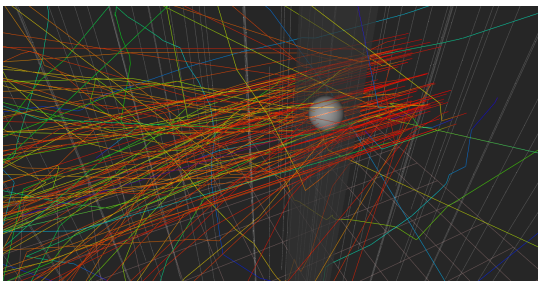
The mean scattering angle in the bulk ice as well as in the hole ice is assumed as 20° [Wos07]. Therefore, for a weak hole ice with a large scattering length, the photons are deflected within the hole ice. See figure 38 (c) in comparison to (b).

For a stronger hole ice with smaller scattering length, several scatterings occur within the hole ice, changing the direction of the scattered photons more drastically. This makes the hole-ice cylinder cause an effective reflection of the incoming photons. See figure 38 (d).

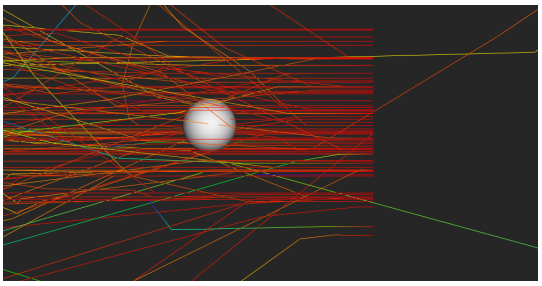
In figure 38 (e), the scattering length is so small that the space between the cylinder border and the optical module is large enough to show the typical "random walk" [Ask+97] of the randomly scattering photons. This random scattering delays the photons on their way to the optical module (compare section 5.6.3 regarding arrival-time distributions).

For even stronger hole ice, which is shown in figures 38 (f), (g), and (h), the hole-ice cylinder around the optical module is shielding the module from the incoming photons. The small scattering length confines the random walk of the photons to the outer region of the cylinder as it is unlikely for a photon to be propagated further into the cylinder. The photons walk in the outer region until they are absorbed, or scattered out of the cylinder and escape.

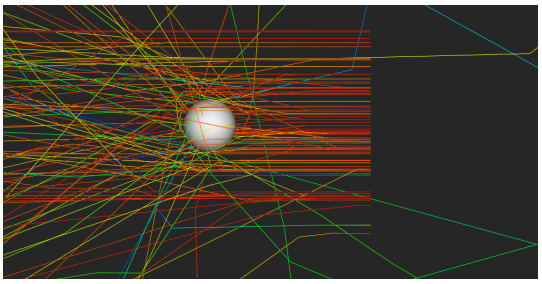
The same effect, confining the photons to the outer region of the cylinder, can



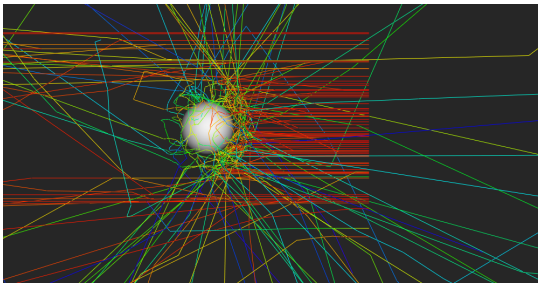
(a) $\lambda_{\text{sca}}^{\text{H}} = 1.0 \cdot \lambda_{\text{sca}}$, 3D view. The hole-ice is represented by the grey cylinder around the optical module represented by a white circle.



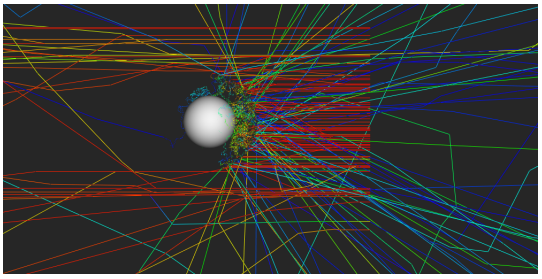
(b) $\lambda_{\text{sca}}^{\text{H}} = 1.0 \cdot \lambda_{\text{sca}}$, view from above



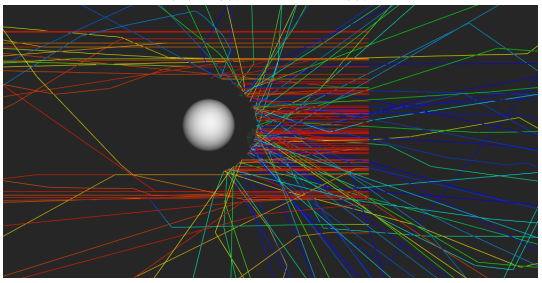
(c) $\lambda_{\text{sca}}^{\text{H}} = 0.1 \cdot \lambda_{\text{sca}}$



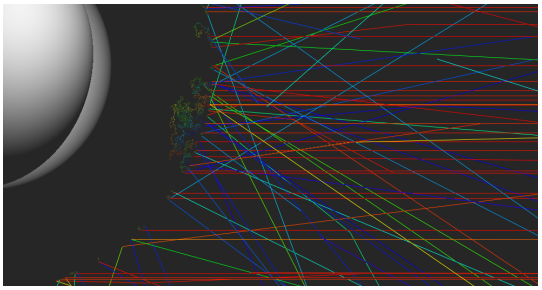
(d) $\lambda_{\text{sca}}^{\text{H}} = 0.01 \cdot \lambda_{\text{sca}}$



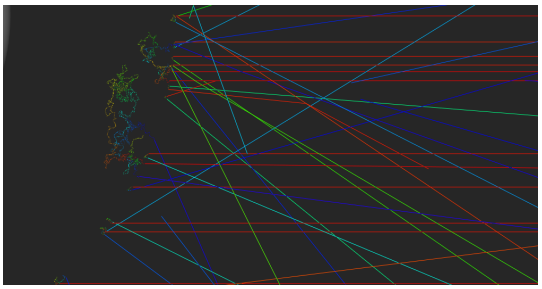
(e) $\lambda_{\text{sca}}^{\text{H}} = 0.001 \cdot \lambda_{\text{sca}}$



(f) $\lambda_{\text{sca}}^{\text{H}} = 0.0001 \cdot \lambda_{\text{sca}}$



(g) $\lambda_{\text{sca}}^{\text{H}} = 0.0001 \cdot \lambda_{\text{sca}}$, zoom



(h) $\lambda_{\text{sca}}^{\text{H}} = 0.0001 \cdot \lambda_{\text{sca}}$, zoom 2

Figure 38: STEAMSHOVEL visualization of a photon-propagation simulation where photons are started from a plane on the right hand side onto a hole-ice cylinder with a radius of 30 cm from 1 m distance. The scattering length $\lambda_{\text{sca}}^{\text{H}}$ within the hole-ice cylinder is given relative to the scattering length λ_{sca} of the surrounding bulk ice. The absorption length is kept the same within the cylinder as in the bulk ice. The geometric scattering length of the bulk ice is $\lambda_{\text{sca}} = 1.3\,\text{m}$, the absorption length is $\lambda_{\text{abs}} = 48.0\,\text{m}$. Colors of the photon trajectories indicate the number of scatterings relative to the total number of scatterings of the photon until absorption. Red: photon just created, blue: photon about to be absorbed. The optical module is shown as a white sphere only to indicate the scale of the scenery. Interaction with the optical module is turned off in the simulation. Animation on YOUTUBE: https://youtu.be/BhJ6F3B-I1s

be achieved by defining the absorption length within the cylinder to $\lambda_{\text{abs}}^{\text{H}} = 5\,\text{cm}$. Then, no photon can reach more than 5 cm into the cylinder (figure 39).

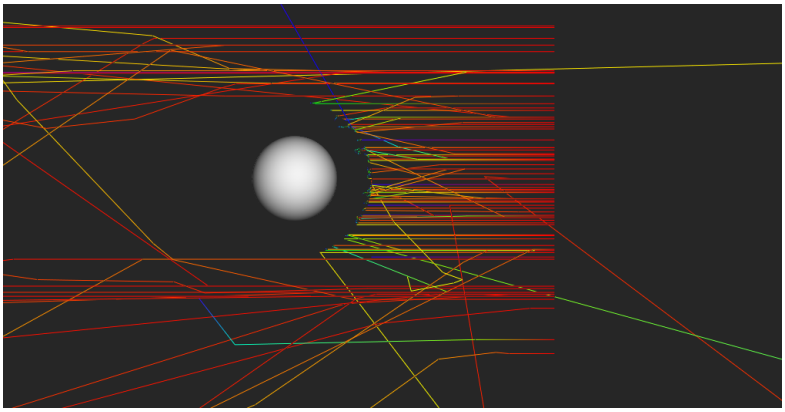
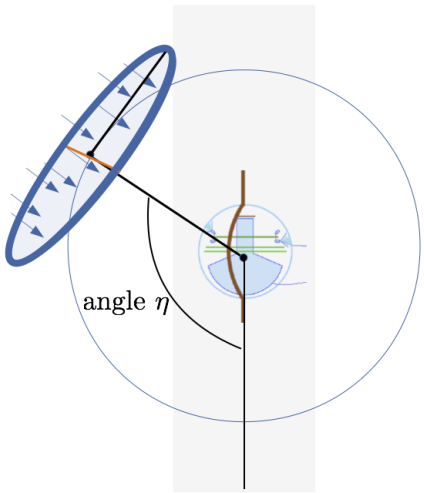


Figure 39: STEAMSHOVEL visualization of a simulation where the absorption length within the cylinder is set to $\lambda_{\text{abs}}^{\text{H}} \approx 5\,\text{cm}$. The scattering length factor is $f_{\text{sca}} = 0.001$. No photon can reach more than 5 cm into the cylinder. Figure 38 (e) on page 81 shows the same simulation without setting the hole-ice absorption length.

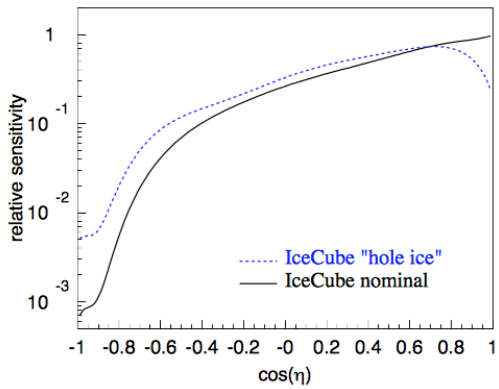
6.2 Scanning the Angular Acceptance of an Optical Module

The angular acceptance, or angular sensitivity to incoming photons, of IceCube's digital optical modules (DOMs) depends on the direction of the incoming photons. This is due to the detector design as the photomultiplier tube (PMT) is facing downwards as illustrated in figure 40 (a). Therefore, a photon coming from below is more likely to hit the PMT and be detected than a photon coming from above. Also, losses due to glass and gel transmission effects as well as the quantum and collection efficiencies of the PMT depend on the impact angle of the incoming photon [Col+13].

Figure 40 (b) shows the measured **angular acceptance** $a_{\text{DOM}}(\eta)$ of the IceCube DOMs by plotting the **relative sensitivity** from lab measurements for each polar angle representing the direction of incoming photons. [Col+13] The **sensitivity** is the fraction of the incoming photons that are registered by the detector. The sensitivity is normalized **relative** to the optimal incoming angle, which is from below, where the relative sensitivity is defined to be 1. Note that the polar angle η is measured from below: For photons coming from below, $\eta = 0$. For photons coming from above, $\eta = \pi$.



(a) Side view of an IceCube digital optical module (DOM). The photomultiplier tube (PMT) in each DOM is facing downwards. Incoming photons are shown in the left upper corner of the illustration. They are moving towards the DOM under a polar angle η measured from below. Image taken from [Ron15].



(b) Angular acceptance $a(\eta)$: Relative sensitivity of the DOM to incoming photons depending on the polar angle η. The "nominal" curve $a_{\text{DOM}}(\eta)$ does not consider hole-ice effects and is based on lab measurements. The "hole ice" curve $a_{\text{DOM,HI}}(\eta)$ effectively contains hole-ice effects and is based on simulations (see footnote 19, page 84). Plot taken from [Col+13], figure 7.

Figure 40: Angular acceptance: The sensitivity of IceCube's digital optical modules (DOMs) depends on the polar angle η of the direction of incoming photons relative to the DOM.

6.2.1 Expected Effect of Hole Ice on the Angular Sensitivity

The effect of hole ice on the detection of incoming photons is expected to depend on the polar angle η of the incoming photons. [Col+13] When the photons are coming from the side, $\eta = \pi/2, \cos\eta = 0$, the distance that incoming photons need to travel through the hole ice is minimal. Therefore, the effect of the hole ice is expected to be minimal for photons from this direction.

For photons approaching the optical module from below, $\eta = 0, \cos\eta = 1$, the hole-ice effect should be strong as the distance that photons need to travel through the hole ice is maximal. The photons are likely to be scattered away before reaching the optical module, effectively reducing the sensitivity for photons approaching the optical module from below as shown in figure 40 (b) by the blue curve in the region of $\cos\eta = 1$.

For photons approaching the optical module from above, $\eta = \pi, \cos\eta = -1$, the hole-ice effect should also be strong as the distance that photons need to travel through the hole ice is maximal. The optical module is very insensitive to registering photons coming from above. But propagating through the hole ice, photons approaching the optical module from above are likely to be scattered away before reaching the optical module. There is a chance that those photons travel around the optical module and finally hit the module at a lower position in the sensitive area. This effectively increases the sensitivity of the module to photons approaching the optical module from above as shown in figure 40 (b) by the blue curve in the region of $\cos\eta = -1$.[19]

[19] The effective "hole-ice" angular-acceptance curve (blue curve in figure 40 b), has been obtained with simulations using the PHOTONICS software [Lun+07], assuming a hole-ice cylinder of 30 cm radius and a geometric scattering length of 50 cm. These are the so-called "H2 parameters" from an AMANDA calibration study [Kar98; Chi16]. See also section 7.1.1 and appendix A.14.

6.2.2 Measuring the Effective Angular Acceptance With Simulations

In order to quantify the hole-ice effect on the angular acceptance of IceCube's optical modules for different hole-ice models, this study performs a series of simulations, starting photons from different angles η_i towards an optical module and counting the photons registered by the optical module.

The simulation records the number $N(\eta_i) := N$ of started photons for each angle η_i, as well as the number $k(\eta_i)$ of registered hits for each angle. The relative hit frequency is $h(\eta_i)$.

$$h(\eta_i) = \frac{k(\eta_i)}{N} \tag{9}$$

To make this relative frequency $h(\eta_i)$ comparable to the DOM's angular acceptance $a_{\mathrm{DOM}}(\eta)$, which is defined such that $a_{\mathrm{DOM}}(\eta = 0) = 1$, this study often uses a renormalized relative frequency $\tilde{h}(\eta_i) := g\, h(\eta_i)$ with a scaling factor $g \in \mathbb{R}$ such that $\tilde{h}(\eta = 0) = 1$.

$$\tilde{h}(\eta_i) = g\, h(\eta_i), \quad g = \frac{1}{h(\eta = 0)} \tag{10}$$

A script to configure and perform these kinds of simulations is provided in `hole-ice-study/scripts/AngularAcceptance` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/tree/master/scripts/AngularAcceptance.

6.2.3 Photon Sources: Pencil Beams and Plane Waves

This study considers two types of photon sources for angular-acceptance studies, pencil beams and plane waves. Both are illustrated in figure 41.

One simulation is performed for each angle η_i. Photons are started from a distance d from the center of the optical module. In the case of pencil beams, photons are started at a fixed position for each angle η_i such that it approaches the optical module under this angle as shown in figure 41 (a).

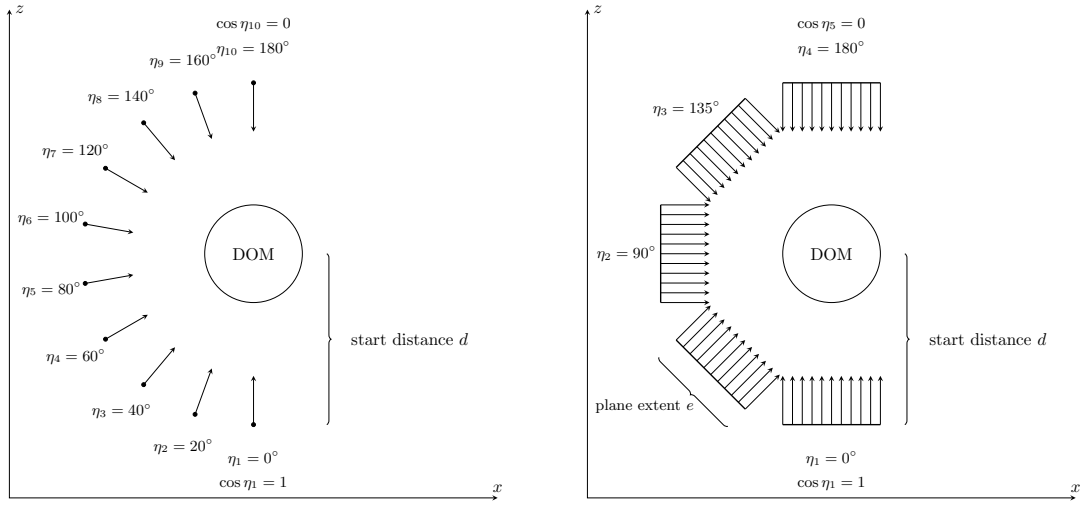
In order to approximate plane waves as photon sources, photons are not started from fixed positions but from random positions within a plane that is located in a distance d from the center of the optical module and oriented perpendicular to the distance vector as shown in figure 41 (b).

Due to the limitation of computational resources, rather than simulating plane waves with infinite extent, an arbitrary extent e of the plane is chosen. RONGEN [Ron15; Ron16] compares the influence of the plane extent e on the angular-acceptance curves ([Ron15], slides 6 and 9). The angular-acceptance simulations described in the section arbitrarily choose a plane extent $e = 1\,\mathrm{m}$ and a start distance $d = 1\,\mathrm{m}$.[20]

Note that the scaling factor g (equation 10) depends on the photon source and needs to be redetermined when switching between pencil beams and plane waves, or when changing the starting distance d, or the plane extent e.

Figure 42 shows a STEAMSHOVEL visualization of simulated photons started from a pencil beam, figure 43 for photons started from a plane.

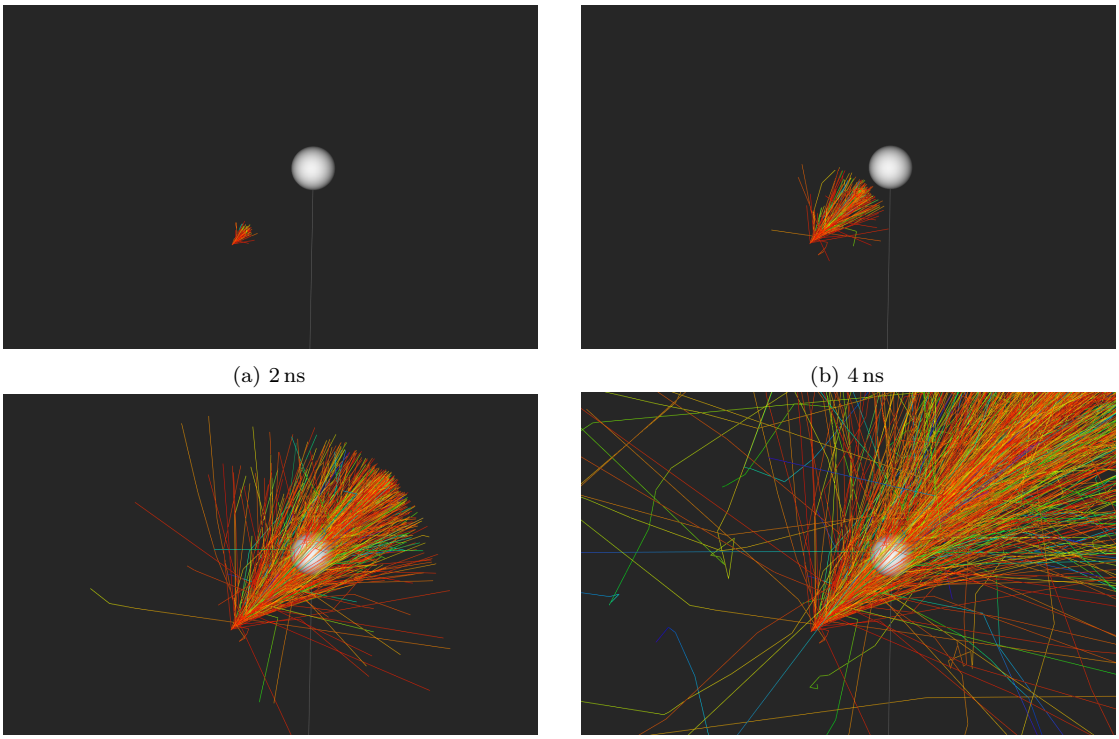
[20] For follow-up studies, an extent of $e = 2\,\mathrm{m}$ and a distance of $d = 2.5\,\mathrm{m}$ should be chosen as photon-source parameters, because the angular-acceptance curves become stable for these or larger parameters. [Ron16]



(a) Pencil beams: Start photons at fixed positions.

(b) Plane waves: Start photons from random positions within planes with extent e approximating plane waves approaching the DOM. The two-dimensional planes are represented by lines in this x-z-projected view.

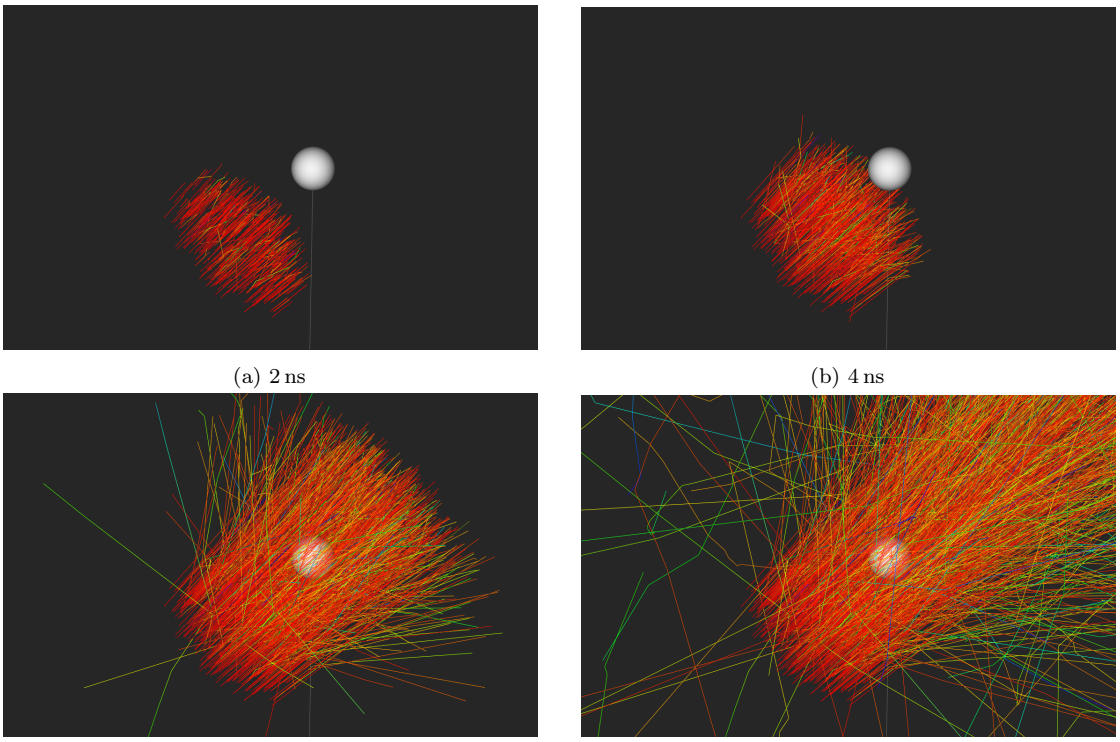
Figure 41: Angular-acceptance scan: In each simulation, start photons from a different polar angle η_i from a distance d towards the digital optical module (DOM) and record the number of photons registered by the DOM. In this illustration, the DOM is viewed from the side. For optical modules aligned along the z-axis, the acceptance is symmetrical with respect to the azimuth angle.



(a) 2 ns (b) 4 ns

(c) 9 ns (d) 100 ns

Figure 42: STEAMSHOVEL visualization of photons started as pencil beam under an angle of $\eta = 45°$ from a distance of $d = 1\,\text{m}$ towards the upmost optical module of a detector string. Snapshots are taken at 2 ns, 4 ns, 9 ns, and 100 ns after starting the photons. The opening angle of the beam is 0.001°. The photon spread seen in the event display is due to scattering along the photon trajectory.



(a) 2 ns (b) 4 ns

(c) 9 ns (d) 100 ns

Figure 43: STEAMSHOVEL visualization of photons started as plane wave with an extent of $e = 1\,\mathrm{m}$ under an angle of $\eta = 45°$ from a distance of $d = 1\,\mathrm{m}$ towards the detector module. Snapshots are taken at 2 ns, 4 ns, 9 ns, and 100 ns after starting the photons.

6.2.4 Acceptance Criterion: A Priori Angular Acceptance or Direct Detection

In each simulation step of the photon-propagation simulation, the algorithm checks whether the photon intersects the sphere representing a optical module between two scattering points (see section 5.2.2). At this time, the intersection position and the direction of the photon are known. Based on that information, in order to model the sensitivity of the optical module, the simulation needs to decide whether to consider the incoming photon as detected, or to ignore the hit.

This study considers two approaches to this decision-making process: Accept the hits based only on the measured **angular acceptance** $a_{\mathrm{DOM}}(\eta)$ of the optical module and the directions of the incoming photons, or accept the hits based only on the location of the sensitive photomultiplier area and the positions of the hits. The latter method is called **direct detection** [Ron17b]. Both approaches are illustrated in figure 44.

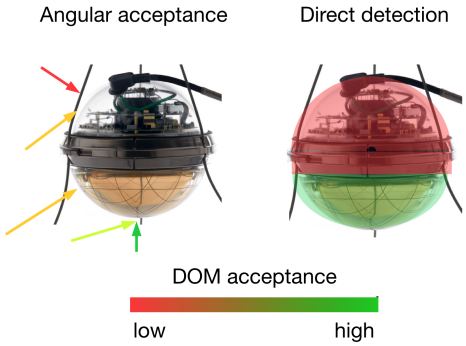


Figure 44: Acceptance criteria: The DOM on the left-hand side accepts or rejects incoming photons as hits based only on the direction of the photon ("angular acceptance"). The DOM on the right-hand side accepts or rejects incoming photons only based on the impact position that determines whether the sensitive area of the photomultiplier tube has been hit ("direct detection"). Image taken from [Ron17b, slide 17].

Using only the photon direction as acceptance criterion, is the default approach in CLSIM. This method has the advantage that both, the location of the photomultiplier tube, and other angular-dependent characteristics such as its quantum efficiency, can be abstracted into a single angular-acceptance function that only depends on one parameter, the photon direction.

In CLSIM, the angular acceptance $a_{\mathrm{DOM}}(\eta)$ of ICECUBE's optical modules has been

implemented as polynomial.

$$a_{\mathrm{DOM}}(\eta) = \sum_{j=0}^{10} b_j \cos(\eta)^j, \quad \eta \in [0; \pi] \tag{11}$$

$$\begin{gathered} b_0 = 0.26266, \quad b_1 = 0.47659, \quad b_2 = 0.15480, \\ b_3 = -0.14588, \quad b_4 = 0.17316, \quad b_5 = 1.3070, \\ b_6 = 0.44441, \quad b_7 = -2.3538, \quad b_8 = -1.3564, \\ b_9 = 1.2098, \quad b_{10} = 0.81569 \end{gathered}$$

The polynomial parameters for different angular-sensitivity models like $a_{\mathrm{DOM}}(\eta)$, also called "nominal" or "h0" model, and other models that include hole-ice approximations, called "h1", "h2", and "h3", can be found in IceCube's source repository within the ICE-MODELS project: http://code.icecube.wisc.edu/projects/icecube/browser/IceCube/projects/ice-models/trunk/resources/models/angsens.

In this study that focuses on the propagation through hole ice, the photons are expected to frequently scatter, and thereby to change their direction in close proximity to the optical module. In this scenario, using the position of a hit rather than only the direction, is of interest, in particular as, according to [Ron17b], direct detection is needed to distinguish different hole-ice models, and has been implemented for this study in CLSIM as alternative to the standard angular-acceptance method.

The implementation of direct detection is documented in `issues/32` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/32.

The most accurate approach to modeling the sensitivity of the optical modules to registering incoming photons would be to take both, the hit position, and the photon direction into account. The position would account for whether the photon would hit the sensitive area of the photomultiplier tube. The impact angle would account for angular-dependent transmission effects and for the quantum efficiency of the tube. Combining both approaches in CLSIM, however, is considered out of scope of this study.

Also, this study does not consider inclined orientations of the optical modules, but always assumes that the modules are oriented along the z-axis.[21]

6.2.5 Angular-Acceptance Simulations Without Hole Ice

In order to compare the different approaches regarding photon sources and acceptance criteria, angular-acceptance simulations have been conducted using the new propagation algorithm, but without any hole-ice cylinders.

> These angular-acceptance simulations without hole ice have been documented in `issues/98` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/98.

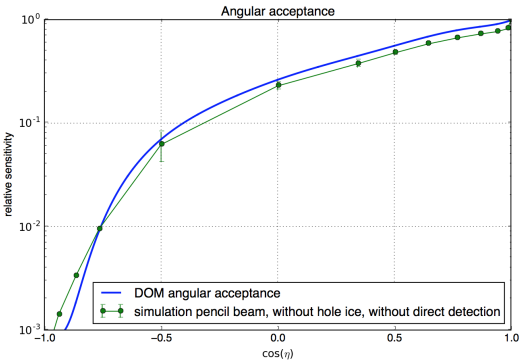
Figure 45 shows the results of the angular-acceptance scans, comparing the above approaches, each also in comparison to the a priori angular-acceptance curve $a_{\text{DOM}}(\eta)$ from [Col+13] ("nominal" curve in figure 40).

The first simulation, using pencil beams and the a priori angular acceptance as acceptance criterion, figure 45 (a), follows the a priori angular acceptance curve $a_{\text{DOM}}(\eta)$ by design. When deactivating scattering and absorption in the bulk ice entirely, all photons would hit the optical module, and all photons would have still the original direction. All non-angular acceptance criteria would be absorbed in the scaling factor g and both curves, the simulation curve and the a priori curve, would match exactly in the limit of many started photons, $N \to \infty$.

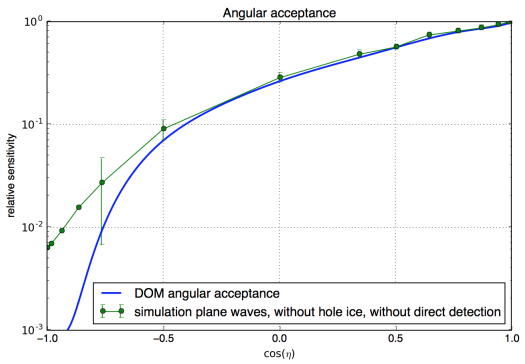
When switching from pencil beams to plane waves, figure 45 (b), more photons approaching the optical module from above, in the region of $\cos\eta = -1$, are accepted because there are photons that start from above, but from a x-y position such that they will fly by the DOM and then be scattered and hit the DOM with an angle that is accepted by the DOM. This effect increases when increasing the extent e of the plane waves.

When switching to direct detection as acceptance criterion, figure 45 (d), the a priori angular acceptance $a_{\text{DOM}}(\eta)$ is no longer put into the simulation. Nevertheless, the simulation curve and the a priori curve have roughly the same shape. Rongen [Ron15] argues that the dominant factor for the shape of the angular-acceptance curve is determined by geometrical considerations, which are reproduced by the simulation with direct detection.

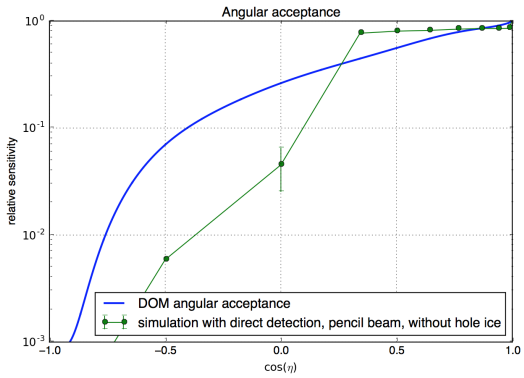
[21] To check if DOM orientations have been implemented at the time of reading, check https://github.com/fiedl/hole-ice-study/issues/53.



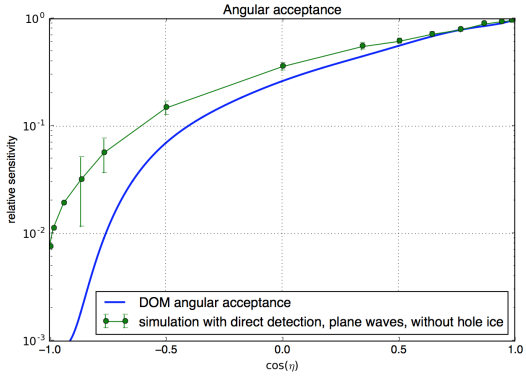
(a) pencil beams, a priori angular acceptance



(b) plane waves, a priori angular acceptance



(c) pencil beams, direct detection



(d) plane waves, direct detection

Figure 45: Comparison of angular-acceptance-scan simulations with different approaches, all without hole ice. In (a), the simulation curve follows the a priori curve by design as the a priori curve is used as DOM acceptance criterion. With direct detection, (c) and (d), the a priori curve is not put into the simulation. The error bars are based on a binomial likelihood (section 6.3).

Further studies could investigate whether finding the proper plane extent e and starting distance d would suffice to make both curves match. Also, the bulk-ice scattering length, which is not infinite in these simulations, could lead to a systematic error, which should be investigated in follow-up studies.[22]

When switching to pencil beams with direct detection, figure 45 (c), the simulation essentially becomes a scan for where the sensitive area of the photomultiplier tube ends in the optical module (compare figure 44).

6.2.6 Angular-Acceptance Simulations With Hole Ice

With the new propagation algorithm, angular-acceptance simulations can be performed, gathering the relative frequencies $\tilde{h}(\eta_i)$ for photons started under an angle η_i being accepted as hits by the optical module, but this with a hole-ice cylinder surrounding the target optical module.

In a first attempt, arbitrary properties for the hole-ice cylinder are used, assuming a cylinder radius $r := r_{\mathrm{DOM}}$ being the same as the radius r_{DOM} of the optical module, and assuming an arbitrary effective scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 1\,\mathrm{m}$.

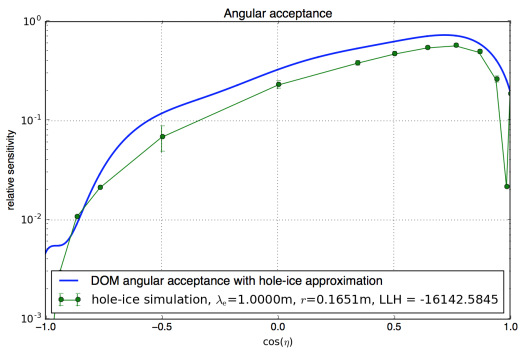
These angular-acceptance simulations with hole ice are documented in `issues/99` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/99.

Figure 46 shows the results of the angular-acceptance scans with hole ice, comparing the different approaches (direct detection vs. angular acceptance, and plane waves vs. pencil beams), each in comparison to the a priori effective angular-acceptance curve $a_{\mathrm{DOM,HI}}(\eta)$ from [Col+13] that approximates the effect of hole ice.

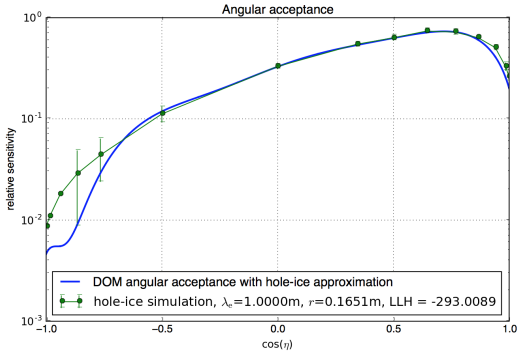
$$a_{\mathrm{DOM,HI}}(\eta) = \sum_{j=0}^{10} b_j \cos(\eta)^j, \quad \eta \in [0;\pi] \tag{12}$$

$$\begin{gathered} b_0 = 0.32813, \quad b_1 = 0.63899, \quad b_2 = 0.20049, \\ b_3 = -1.2250, \quad b_4 = -0.14470, \quad b_5 = 4.1695, \\ b_6 = 0.76898, \quad b_7 = -5.8690, \quad b_8 = -2.0939, \\ b_9 = 2.3834, \quad b_{10} = 1.0435 \end{gathered}$$

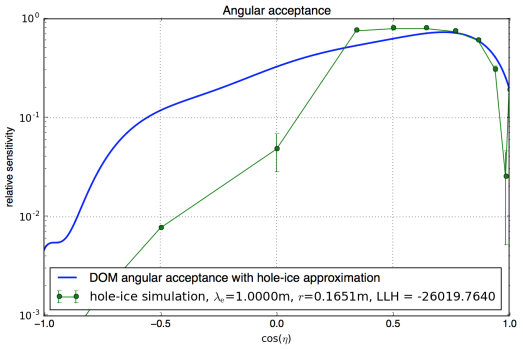
[22] To check for progress on that matter, see https://github.com/fiedl/hole-ice-study/issues/108.



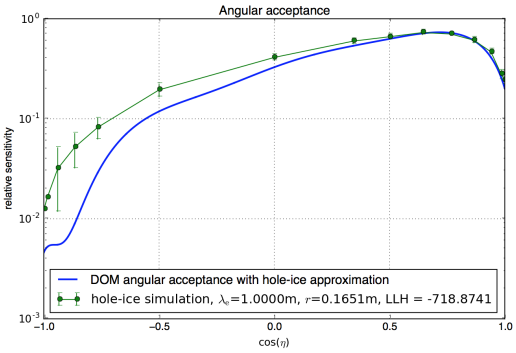
(a) pencil beams, a priori angular acceptance



(b) plane waves, a priori angular acceptance



(c) pencil beams, direct detection



(d) plane waves, direct detection

Figure 46: Comparison of angular-acceptance-scan simulations with different approaches, all with a hole-ice cylinder with arbitrary properties, assuming an effective hole-ice scattering length of $\lambda_{\mathrm{e}}^{\mathrm{H}} = 1\,\mathrm{m}$ and a hole-ice-cylinder radius of $r = r_{\mathrm{DOM}}$.

When using plane waves as photon sources, figure 46 (b) and (d), the behavior of photons approaching the optical module from below matches the a priori curve $a_{\mathrm{DOM,HI}}(\eta)$ [Col+13] already reasonably well for these hole-ice parameters.

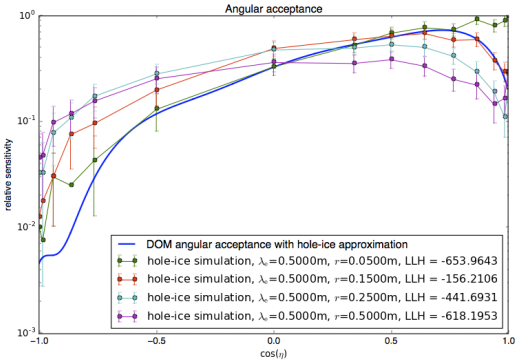
The next sections 6.2.7 and 6.2.8 show effective angular-acceptance curves from simulations with different hole-ice parameters.

6.2.7 Varying the Hole-Ice-Cylinder Radius in Simulations

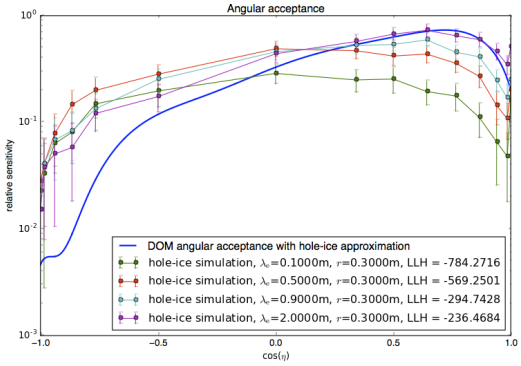
With the new medium-propagation algorithm (section 5.4), the hole-ice parameters can be varied.

> Implementing angular-acceptance simulations for different hole-ice-cylinder radii is documented in `issues/82` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/82.

Figure 47 (a) shows angular-acceptance curves from simulations with different hole-ice-cylinder radii. In these simulations, the hole-ice cylinder's center is set to be the center of the optical module. The effective scattering length of the hole ice is fixed, $\lambda_{\mathrm{e}}^{\mathrm{H}} = 50\,\mathrm{cm}$.



(a) Varying the radius of the hole-ice cylinder.



(b) Varying the scattering length of the hole ice.

Figure 47: Comparison of angular-acceptance simulations with different hole-ice parameters. The blue curve shows the a priori angular-acceptance curve $a_{\mathrm{DOM,HI}}(\eta)$ from [Col+13] that approximates the effect of the hole ice. LLH gives the log-likelihood value of comparing the simulation curve to the a priori curve using a binomial likelihood function (see section 6.3).

When the hole-ice-cylinder radius is larger in the simulations, the hole-ice effect is stronger because the ice volume occupied by the hole-ice cylinder is larger, making it more likely that photons approaching the detector module scatter within the

hole ice. When increasing the hole-ice-cylinder radius, photons approaching the optical module from above (left-hand side of figure 47 a) are increasingly scattered "around" the optical module such that they hit the optical module in the sensitive area at the bottom, resulting in more accepted hits. Photons coming from below (right-hand side of figure 47 a) are increasingly scattered away before reaching the optical module when increasing the hole-ice-cylinder radius, resulting in less hits.

6.2.8 Varying the Hole-Ice Scattering Length in Simulations

In another series of simulations, the scattering length of photons propagating through the hole-ice cylinder is varied.

> Implementing angular-acceptance simulations for different hole-ice scattering lengths is documented in `issues/83` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/83.

Figure 47 (b) shows angular-acceptance curves from simulations with different hole-ice scattering lengths. In these simulations, the hole-ice cylinder's center is set to be the center of the optical module. The radius of the hole-ice cylinder is fixed to $r = 30\,\text{cm}$.

When the scattering length within the hole-ice cylinder is shorter, photons propagating through the cylinder scatter more often. For smaller scattering lengths, photons approaching the optical module from below (right-hand side of figure 47 b) are more likely to be scattered away in the hole ice before reaching the optical module, resulting in less hits. On the left-hand side of figure 47 (b), where the photons are approaching the optical module from above, the effect of varying the scattering length is less prominent as compared to varying the cylinder radius (47 a). The detection of photons approaching the optical module from above requires that photons flying by the optical module are scattered into the sensitive area of the module. For a plane wave of photons approaching the module from above, this is much more likely when increasing the hole-ice radius as compared to shortening the scattering length.

Both series of angular-acceptance simulations, varying the hole-ice scattering length and varying the hole-ice-cylinder radius, confirm qualitatively the expected hole-ice effects (section 6.2.1).

In agreement with figure 47, Rongen [Ron17a] suggests that the scattering length of the hole ice determines the position of the maximum of the angular-sensitivity

curve, the hole-ice radius dominantly determines the strength of the reduction of the sensitivity in the region of $\cos\eta \approx 1$.

Additional angular-acceptance simulations using hole-ice parameters suggested by other studies, will be presented in section 7.2.

6.3 Determining the Hole-Ice Parameters Corresponding to the Current Hole-Ice Approximation

In current CLSIM simulations that do not use the new medium-propagation algorithm introduced by this study, an effective angular-acceptance curve $a_{\mathrm{DOM,HI}}(\eta)$ for optical modules is used to approximate the effect of the hole ice on the detection of photons (section 6.2.6).

The hole-ice properties that have been assumed to create the effective angular-acceptance curve $a_{\mathrm{DOM,HI}}(\eta)$ are a hole-ice-cylinder radius of 30 cm and a geometric hole-ice scattering length of $\lambda_{\mathrm{sca}}^{\mathrm{H}} = 50\,\mathrm{cm}$ (*H2 model*) [Kar98]. These properties can be used in a CLSIM simulation using the new medium-propagation algorithm and direct propagation through the hole ice in order to compare the effective angular-acceptance curve resulting from the simulation to the approximation curve currently in use.

This simulation using the H2 parameters is documented in `issues/80` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/80.

In the simulation, direct detection is used as acceptance criterion, plane waves with an extend of $e = 1\,\mathrm{m}$ and a starting distance of $d = 1\,\mathrm{m}$ are used. The hole-ice absorption length is set to a high fixed value, $\lambda_{\mathrm{abs}}^{\mathrm{H}} = 100\,\mathrm{m}$. Figure 48 shows the result of this simulation. The angular-acceptance curve resulting from the simulation does not match the approximation curve currently in use very well.[23]

By performing a grid scan (see section 5.5) over a range of hole-ice parameters, one can find the best match with the approximation curve currently in use, in order to understand which kind of direct-propagation hole ice best matches the hole ice currently assumed in all simulations using the approximation curve.

This grad scan is documented in `issues/12` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/12.

A script to configure and perform this kind of parameter scan is provided in `hole-ice-study/scripts/ParameterScan` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/tree/master/scripts/ParameterScan.

[23] To make sure this is no matter of confusing effective scattering length and geometric scattering length, the same simulation has been performed with an effective scattering length of 50 cm rather than a geometric scattering length of 50 cm. These curves match even worse. See appendix A.14.

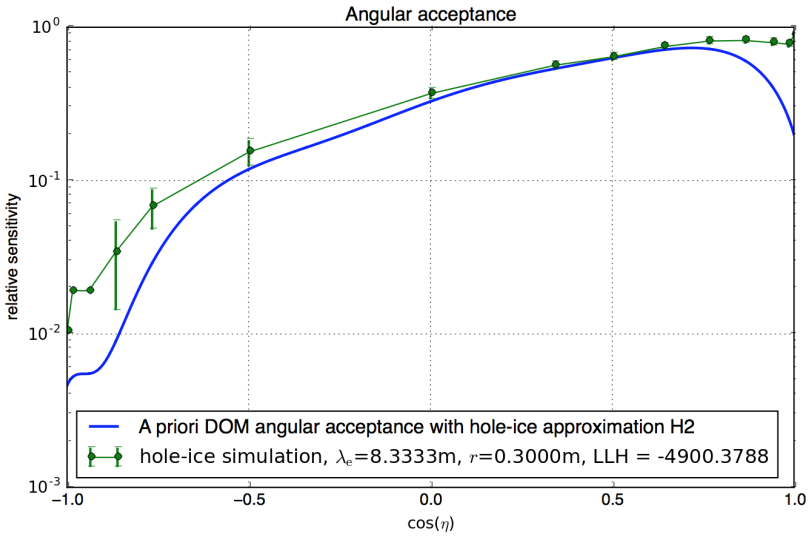


Figure 48: Comparing the effective angular-acceptance curve currently in use for approximating the effect of hole ice on the detection of photons in simulations to a simulation with direct photon propagation through hole ice using the same hole-ice parameters, hole-ice-cylinder radius $r = 30\,\text{cm}$, geometric hole-ice scattering length $\lambda_{\text{sca}}^{\text{H}} = 50\,\text{cm}$. LLH is the log binomial likelihood comparing both curves.

In principle, one could scan over a wide variety of parameters, including the scattering and absorption lengths of the hole ice, the hole-ice-cylinder radius, the distance from which the photons are started towards the optical module, the kind of the photon source (pencil beams, or plane waves), and the extent of the photon source plane. In order to keep the computational effort minimal, however, all conditions are kept the same as above, and only hole-ice-cylinder radius and hole-ice scattering length are varied in the grid scan.

To compare the simulation results to the approximation curve, a binomial likelihood L is used.

$$L = \prod_{i=1}^{N} \binom{n}{k_i} p(\eta_i)^{k_i} \left(1 - p(\eta_i)\right)^{n-k_i} \tag{13}$$

This likelihood can be interpreted as the probability that for a number of N simulations, one simulation for each angle η_i, starting n photons from the angle η_i towards the optical module, k_i photons will be accepted as hit by the optical module if the acceptance probability $p(\eta_i)$ for the angle η_i is given by the approximation curve $a_{\text{DOM,HI}}(\eta)$, $p(\eta_i) = {}^{1}\!/_{g}\, a_{\text{DOM,HI}}(\eta_i)$. g is a scaling factor (see section 6.2.2), which is needed due to the normalization of $a_{\text{DOM,HI}}(\eta)$.

As one series of simulations is performed for each set $\mathcal{H}$ of hole-ice parameters, there is a likelihood $L_{\mathcal{H}}$ for each parameter set $\mathcal{H}$, which depends on the numbers $k_{i,\mathcal{H}}$ of photon hits for photons started under the angle η_i, propagated with hole-ice parameters $\mathcal{H}$.

$$L_{\mathcal{H}} = \prod_{i=1}^{N} \binom{n}{k_{i,\mathcal{H}}} p(\eta_i)^{k_{i,\mathcal{H}}} \left(1 - p(\eta_i)\right)^{n-k_{i,\mathcal{H}}}$$

The simulation resulting in the maximal likelihood $L_{\mathcal{H}_\mathrm{b}}$ corresponds to the set of hole-ice parameters $\mathcal{H}_\mathrm{b}$ that best describe the approximation curve $a_{\mathrm{DOM,HI}}(\eta)$.

Figure 49 shows the result of the parameter grid scan, plotting the likelihood, or rather $-2\Delta\mathrm{LLH} := -2(\ln L_{\mathcal{H}} - \ln L_{\mathcal{H}_\mathrm{b}}) = -2\ln\left(L_{\mathcal{H}}/L_{\mathcal{H}_\mathrm{b}}\right)$ as a more common measure of agreement, against the effective scattering length $\lambda_\mathrm{e}^\mathrm{H}$ of the hole ice on the one axis, and the radius r of the hole-ice cylinder on the other axis. For the optimal parameters $\mathcal{H}_\mathrm{b}$, ΔLLH is zero.

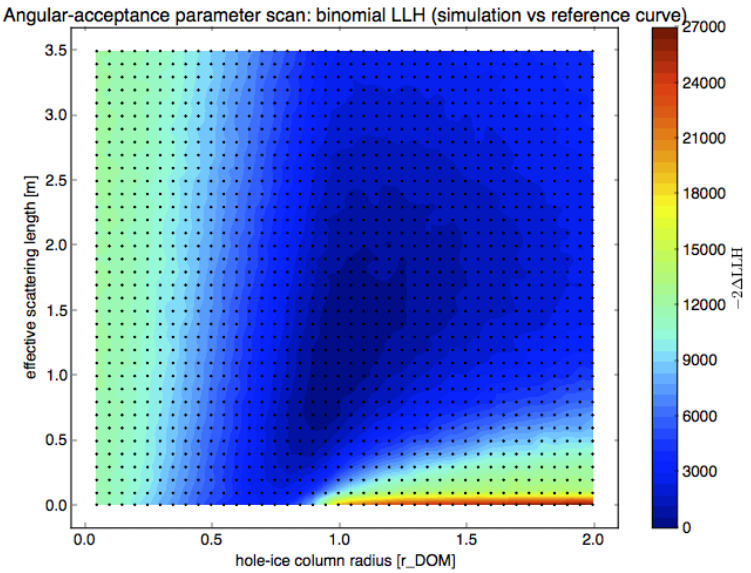
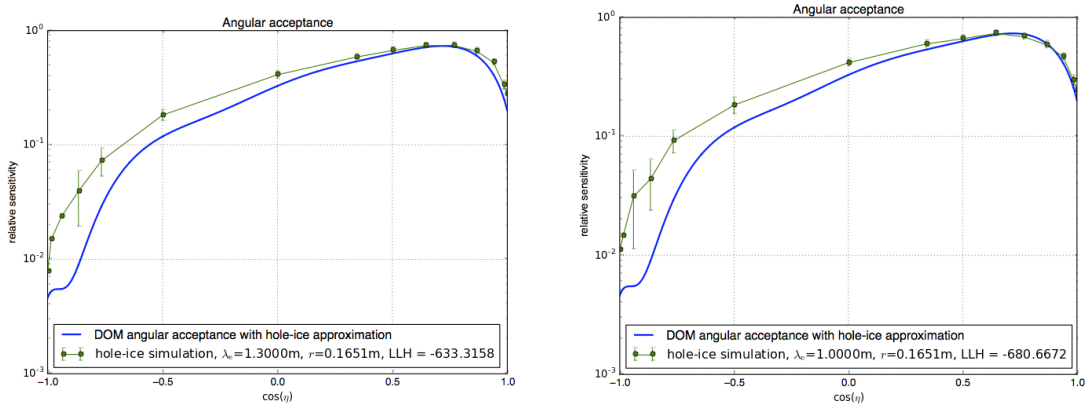


Figure 49: Agreement of the fixed hole-ice-approximation angular-acceptance curve $a_{\mathrm{DOM,HI}}(\eta)$ and direct hole-ice simulations using the new CLSIM medium-propagation algorithm with hole-ice-cylinder radii r given in units of the radius r_{DOM} of the optical module, and effective hole-ice scattering lengths $\lambda_\mathrm{e}^\mathrm{H}$. Each dot in the plot represents one parameter set $\mathcal{H}$ and one corresponding set of simulations.

The hole-ice parameters $\mathcal{H}_\mathrm{b}$ that lead to a simulation, which results in an angular-acceptance curve that best agrees with the approximation curve $a_{\mathrm{DOM,HI}}(\eta)$, are

a hole-ice-cylinder radius $r = 1.0\,r_{\mathrm{DOM}}$ where r_{DOM} is the radius of an optical module, and an effective hole-ice scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 1.3\,\mathrm{m}$. Figure 50 shows an angular-acceptance curve for this parameter set, as well as for the nearby parameter set of $r = 1.0\,r_{\mathrm{DOM}}$, $\lambda_{\mathrm{e}}^{\mathrm{H}} = 1.0\,\mathrm{m}$.



(a) Simulation with hole-ice radius $r = 1.0\,r_{\mathrm{DOM}}$ and effective hole-ice scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 1.3\,\mathrm{m}$.

(b) Simulation with hole-ice radius $r = 1.0\,r_{\mathrm{DOM}}$ and effective hole-ice scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 1.0\,\mathrm{m}$.

Figure 50: Angular-acceptance curves from simulations best matching the hole-ice-approximation angular-acceptance curve $a_{\mathrm{DOM,HI}}(\eta)$. The LLH is the log binomial likelihood for the agreement of the simulation curve and the reference curve.

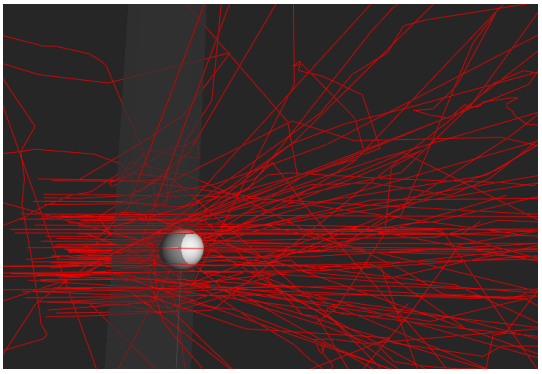
For photons approaching the optical module from below (right-hand side of the plots), the simulations follow the approximation curve. For lower angles, the simulation sees more photon hits, which, however, is the case for all simulations in this grid scan.

To summarize this result, choosing a hole-ice cylinder of the same size as the optical module, and an effective scattering length of about 1 m for a direct hole-ice simulation, comes closest to using the hole-ice-approximation angular-acceptance curve, which is currently default for CLSIM simulations.

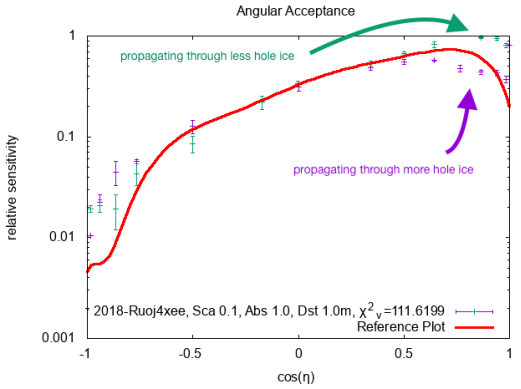
6.4 Simulating the Displacement of Optical Modules Relative to the Hole-Ice Columns

Optical modules do not necessarily need to be positioned well-centered relative to the hole ice. In practice, when the drill hole is refreezing, the optical module may be deposited displaced both relative to the drill hole as well as to the bubble column.

The hole-ice effect on the detection of photons by an optical module that is not symmetrically positioned relative to the hole ice cannot be modeled by an effective angular-acceptance curve because these curves always assume azimuthal symmetry. Nevertheless, the the asymmetry of the effect can be visualized using an angular-acceptance simulation, plotting the acceptance for polar angles $\eta \in [0; \pi[$ and for polar angles $\eta \in [\pi; 2\pi[$ in different colors in the same plot (figure 51 b).



(a) Steamshovel display of the simulation scenario for $\eta = \pi$. The photons are started from a plane on the left-hand side of the image.



(b) Effective angular acceptance resulting from this simulation. One simulation curve shows the acceptance of photons arriving from a polar angle $\eta \in [0; \pi[$, the other simulation curve shows the acceptance of photons arriving from $\eta \in [\pi; 2\pi[$. The red curve shows the H2 hole-ice-approximation angular-acceptance curve from [Col+13].

Figure 51: Simulation of a hole-ice cylinder, which is shifted relative to the position of the optical module. The optical module is shifted beyond the cylinder border such that it is partly within and partly outside of the cylinder. Photons approaching the optical module from one direction, $\eta = \pi$, need to travel through the maximal distance through the hole ice, photons from the opposite direction, $\eta = -\pi$, hit the optical module before reaching the hole-ice cylinder.

To model the displacement of optical modules relative to the hole ice in production simulations, both the positions of the optical modules and the positions of the hole-ice cylinders, or even cylinder sections for specific z-ranges can be configured independently.

In this example simulation, the cylinder is shifted relative to the optical module while the photon sources, which start photons from different directions towards the optical module, are rotated around the optical module, not around the position of the cylinder.

The simulation with a shifted hole-ice cylinder is documented in `issues/8` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/8.

In this example, which uses the hole-ice-correction algorithm (section 5.3), the hole ice is modeled as cylinder with 30 cm radius and a hole-ice scattering length of $^1/_{10}$ of the scattering length of the surrounding bulk ice. The optical module is shifted by 20 cm from the central position, and therefore is shifted beyond the border of the hole ice in order to demonstrate an extreme effect. In practice, the optical module can only be shifted by a maximum of about 15 cm from the central position, because the drill hole is only about 60 cm in diameter.

Figure 51 (a) shows the simulation scenario in STEAMSHOVEL, figure 51 (b) shows the resulting angular-acceptance curves from the simulation, one for the angular range $\eta \in [0; \pi[$ and one for $\eta \in [\pi; 2\pi[$. One of these curves shows a more extreme hole-ice effect as more photons need to propagate through the hole ice to hit the optical module in comparison to the other curve.

RONGEN [RRK17] has performed a series of simulations where the optical module is shifted randomly relative to the hole-ice column for each data point. The results shown in figure 52 indicate a wide spread of the effective acceptance curve for angles all lower angles, $\cos\eta > 0$.

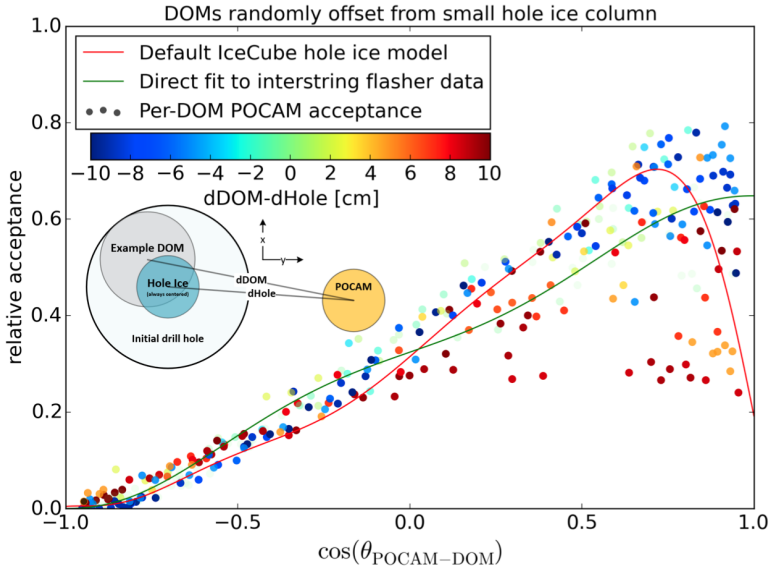


Figure 52: Results from a series of simulations where the position of the optical module is shifted randomly relative to the hole-ice column for each data point. The displacement of the module especially widens the spread of the angular-acceptance curve for all lower angles, $\cos\eta > 0$. The "Default IceCube hole ice model" refers to the a priori angular-acceptance curve with H2 parameters from [Col+13]. The "Direct fit to interstring flasher data" refers to the model described in section 7.1.2. Image source: [RRK17]

6.5 Simulating Nested Hole-Ice Columns

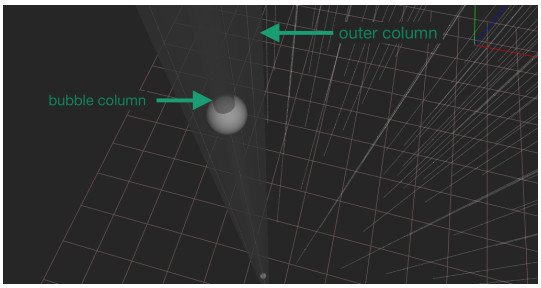
Current descriptions of the hole ice characterize it as a relatively clear outer region with a radius of the entire drill hole, and a central column of about 8 cm radius filled with air bubbles, resulting in a small scattering length in this region. [Aar+17b; RRK17; Ron16; Ron17b]

The new medium-propagation algorithm (section 5.4) allows to simulate both ice columns as independent cylinders. How to arrange, shift, and size those cylinders, and how to configure their scattering and absorption properties in production simulations, needs to be determined in follow-up studies.

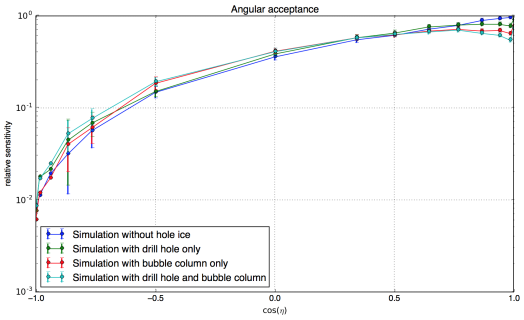
As an example, the following simulation models two nested cylinders, an outer cylinder of 30 cm radius with a moderate geometric scattering length of 50 cm that resembles the drill hole, and an inner cylinder of 8 cm radius and a small geometric scattering length of 1 cm that represents the bubble column. With this configuration, the simulation scans the effective angular acceptance of the optical module, which is embedded in the center of the cylinders.

The implementation of this simulation is documented in `issues/7` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/7.

The same simulation then is repeated, once with only the bubble column, and once with only the drill hole. Figure 53 shows the resulting effective angular-acceptance curves.



(a) Steamshovel visualization of the simulation scenario. The outer column represents the entire drill hole. The inner column is filled with small air bubbles and is therefore called bubble column.



(b) Effective angular acceptance of the optical module for different cylinder configurations, in the order of increasing hole-ice effect: Without hole-ice cylinders, with only a drill hole, with only a bubble column, and with drill hole and bubble column together.

Figure 53: Simulation of photon propagation through nested hole-ice cylinders.

The simulation that implements both, drill hole and bubble column, shows the

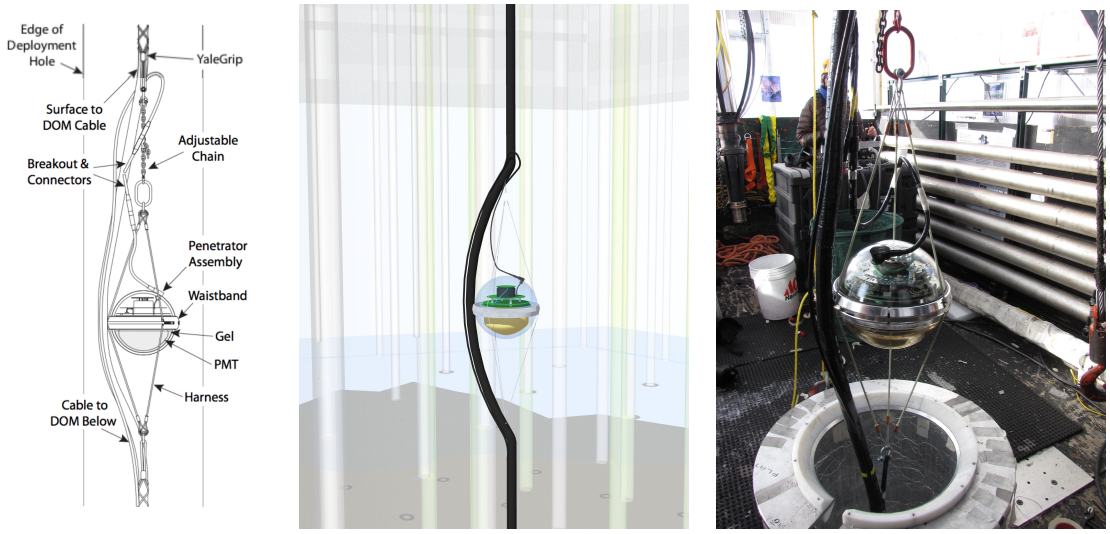
strongest hole-ice effect. The next strongest hole-ice effect shows the simulation that only implements the bubble column. The least strongest effect shows the simulation that implements only the drill hole. As reference, a simulation without any hole ice is shown.

Related studies favor different hole-ice models: A camera deployed in the drill hole shows a diffuse ice volume, which only covers a part of the module sphere, indicating the presence of a bubble column that is not around the center of the optical modules but asymmetrically arranged. [Aar+17b; Ron16] LED measurements, however, that measure the azimuthal dependencies of the ice properties in the ice surrounding the optical modules, do not support the model of a displaced bubble column that would effect the light from some of the LEDs of the optical module but not all of them. [Ron16]

The tool set provided by this study to recreate the different hole-ice models with independent, displaced ice cylinders, allows follow-up studies to compare the different models to flasher data, helping to better understand the characteristics of the hole ice.

6.6 Simulating Shadowing Cables as Opaque Cylinders

The main cable of each detector string (figure 54) allows to power the detector modules and to transfer data from the modules to the surface. The cable has a diameter of 46 mm [Aar+17b] and, as it is located in close proximity to the optical modules, shields the optical modules from a non-negligible amount of incoming photons. The cable's mantle is black such that it will absorb photons rather than reflect them.



(a) Optical module deployment schematics. Image source: [Aar+17b]

(b) Rendered image of the optical module and the main cable. Image source: [Yan09]

(c) Photo: Optical module deployment. Image source: [Mal10]

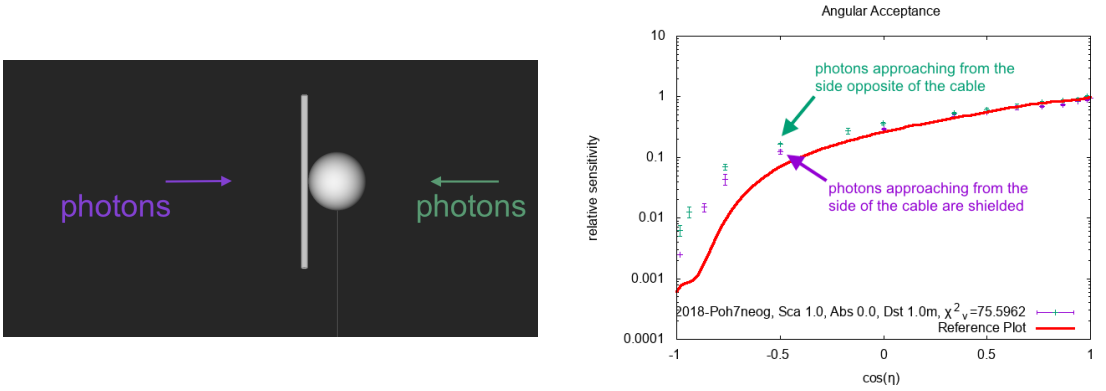
Figure 54: IceCube's optical module in relation to the main cable.

6.6.1 Asymmetric Shadowing Effect Caused by the Cable

Using the new medium-propagation algorithm, the cable can be simulated as one or several cylinders with limited z-range, which are configured for instant absorption.

As a first test, a simulation is performed to verify that the cable causes an asymmetric shadowing effect, shielding photons approaching from one direction while not shielding photons approaching from the opposite direction. Figure 55 shows the effective angular sensitivity of the optical module measured in this simulation and confirms that the asymmetric shadowing can be observed in simulations.

This simulation, placing an opaque cylinder part besides the optical module, propagating the photons with the hole-ice-correction algorithm (section 5.3) and scanning the effective angular acceptance, is documented in `issues/35` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/35.



(a) Steamshovel visualization of the simulation scenario. Photons approaching from different angles are either shielded by the cable or can reach the optical module unhindered.

(b) Effective angular acceptance of the optical module measured in the simulation. Photons approaching from the side of the cable are less likely to reach the optical module and be registered as a hit.

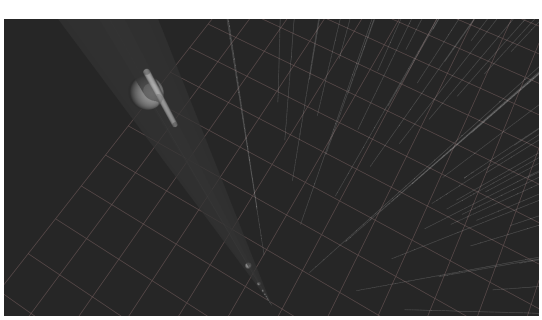
Figure 55: Simulation with the main cable modeled as opaque cylinder placed besides the optical module.

6.6.2 Cable Inside the Bubble Column

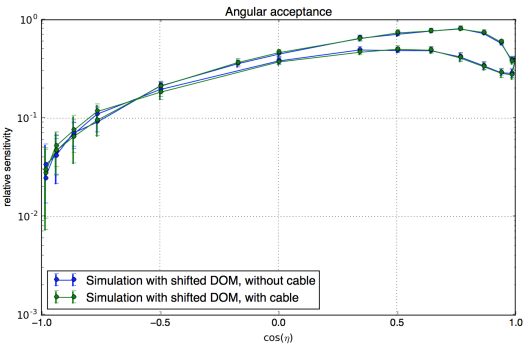
In one class of hole-ice models that has not been studied with simulations, yet, the optical module is shifted such that the cable resides fully or partially within the bubble column. [Ron17b]

Placing a minimal opaque cable cylinder besides the optical module partially inside the bubble column in a simulation, using plane waves as photon sources, direct detection as hit acceptance criterion, and the new medium-propagation algorithm (section 5.4), the cable shadowing effect appears to be subdominant when measuring the effective angular acceptance (figure 56) compared to the effect of the bubble column and the drill hole ice. For photons approaching the optical module from the side where the cable is located, the number of photon hits is decreased compared to a simulation without cable. But the effect is smaller than the statistical error.

This simulation is documented in `issues/110` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/110.



(a) Steamshovel visualization of the simulation scenario. An opaque cable cylinder of 1 m height is placed besides the optical module, fully within the drill hole ice, and partially inside the bubble column.



(b) Effective angular acceptance of the optical module measured in simulations with and without cable.

Figure 56: Simulation with opaque cable partially inside bubble column.

6.6.3 Can Cable Effects Account for the Observed Hole-Ice Effects?

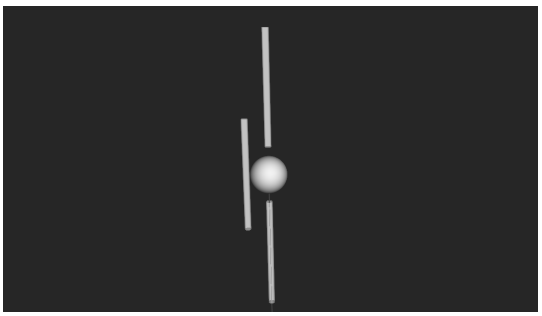
An open question raised by the IceCube Calibration Group is whether the shielding by the main cable is sufficient to account for the effects observed by other studies that are typically attributed to the hole ice.[24]

The following series of simulations compares the effective angular acceptance for a scenario where only a shadowing cable is considered, and a typical drill-hole-ice scenario. This simulation uses the new medium-propagation algorithm (section 5.4), plane waves as photon sources, and direct detection as hit acceptance criterion. The cable is modeled as three opaque cylinder parts (see figure 57 (a) in comparison to the illustrations in figure 54). The drill-hole ice is modeled with properties suggested by the so-called *H2 model*, a hole-ice-cylinder radius of 30 cm and a geometric scattering length of 50 cm [Kar98].

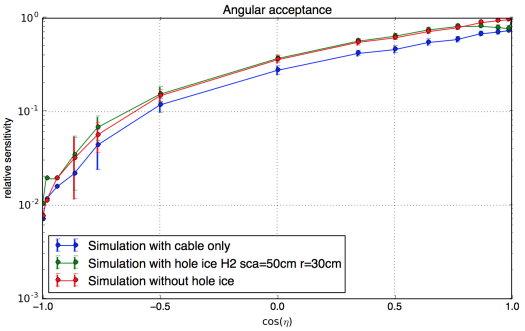
This series of simulations is documented in `issues/101` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/101.

As shown in figure 57 (b), the cable can account for a total reduction in photon hits. For the decreased number of hits for photons approaching from below, and an increased number of hits for photons approaching from above, however, which are expected from earlier hole-ice studies (see section 6.2.6), the simulated cable cannot account for.

[24] Internal correspondence. See for example https://icecube-spno.slack.com/archives/C4FV72473/p1522086838000178.



(a) Steamshovel visualization of the simulation scenario with cables: The cable is modeled as three opaque cylinder parts. See also figure 54 for comparison.

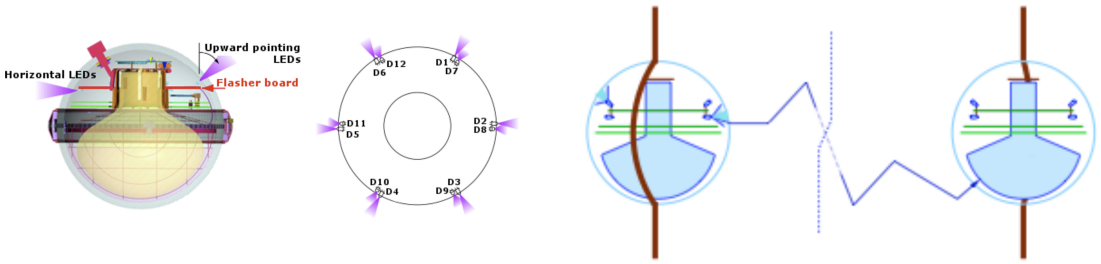


(b) Effective angular acceptance curves for the optical module measured in simulations without hole ice, with only a cable, and with a hole-ice-cylinder of 30 cm radius and a geometric hole-ice scattering length of 50 cm (H2 model parameters).

Figure 57: Measuring the effective angular acceptance of an optical module in simulations: (1) without hole ice, (2) with only a drill hole, (3) with only a cable.

6.7 Scanning Hole-Ice Parameters for Optimal Agreement with Flasher-Calibration Data

For calibration purposes, each optical module in the IceCube detector is equipped with a set of flasher light-emitting diodes (LEDs) as illustrated in figure 58. Using these flasher LEDs, a known amount of light can be produced at a given position and time within the detector that can be measured by the optical modules and can thereby be used to calibrate the detector. [Col+13]



(a) Schematic diagram of an optical module. In the upper hemisphere, the flasher board is located, sustaining six horizontally emitting LEDs and six upward pointing LEDs. Image source: [Ron16]

(b) Principle of operation of the LED flasher calibration system: Photons emitted by the LED of one optical module propagate through the ice and are detected by another optical module. The amount of light observed at the receiving optical modules depends on the properties of the ice the photons propagate through on the path from the sending to the receiving optical modules. Image source: [Col+13]

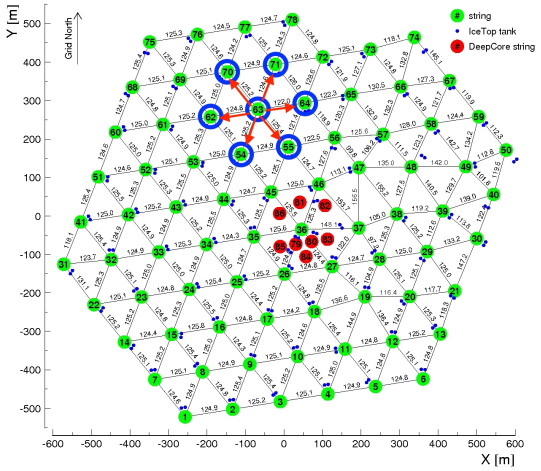
Figure 58: IceCube's light-emitting diode (LED) flasher calibration system.

Light that is emitted at one optical module and detected by another module has to travel through the ice in between those modules. The amount of light arriving at the target module depends on the amount of light absorbed or scattered away on the way from the sending to the receiving optical module. Both, the properties of the bulk ice of the South Polar glacier, and the properties of the hole ice surrounding the sending and the receiving optical modules, contribute to this effect.

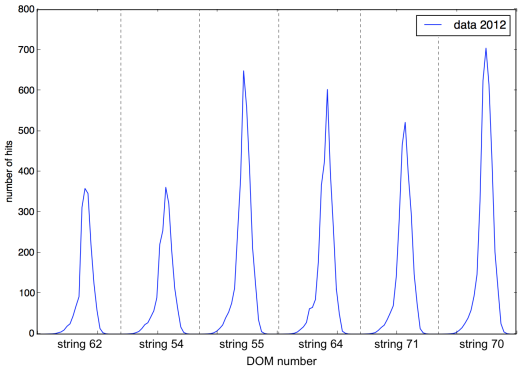
Figure 59 shows 7-string calibration data.[25] To produce this data set, all LEDs on `DOM 60_30`, which is the middle optical module on string number 30, have been activated at once. The plot shows the amount of light received at the optical modules of the surrounding strings 62, 54, 55, 64, 71, and 70.

The aim of the following series of simulations is to adjust the simulated hole-ice parameters until the the simulation is able to reproduce the calibration data in order to find the hole-ice parameters that best match the calibration data.

[25] Data source: [Chi12]



(a) Top view of the detector strings. Each green circle represents one detector string, containing 60 optical modules each. In this flasher study, the LEDs of the central optical module on string 63 have been activated, and the amount of light arriving at the optical modules of the surrounding strings 62, 54, 55, 64, 71, and 70 has been measured. Image based on [Wos11]



(b) Amount of light measured at the receiving optical modules on strings 62, 54, 55, 64, 71, and 70. Each column of this plot represents one receiving string. Within each column, the left most value represents the top most optical module of the string, the right most value represents the bottom most module of the string. Only optical modules are shown that receive at least one photon hit.

Figure 59: 7-string flasher data: A central optical module is emitting light using the mounted flasher LEDs. The surrounding optical modules are measuring the amount of light arriving there.

This preliminary 7-string flasher study is documented in `issues/59` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/59.

A script to configure and perform these kinds of simulations is provided in `hole-ice-study/scripts/FlasherSimulation` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/tree/master/scripts/FlasherSimulation.

A script for performing flasher simulations as grid scan is provided in `hole-ice-study/scripts/FlasherParameterScan` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/tree/master/scripts/FlasherParameterScan.

As shown in figure 59 (b), the amount of light received at strings 62 and 54 is significantly reduced compared to the other strings. The distances of the sending optical module to the receiving strings range from 122.0 m to 125.4 m and average out at 124.5 m, such that the different distances can only cause a peak reduction of about 2 % rather than the observed reduction of about 30 %. The properties of the bulk ice cannot account for the asymmetric reduction either. The reduction may be caused by some kind of asymmetric shadowing effect, either by shadowing cables[26], or by asymmetric hole ice in proximity of the sending optical module, such that the

[26] Flasher simulation with cable: See section 6.8.

emitted light is suppressed in the direction to strings 62 and 54. The reduction may also be caused by the hole ice of strings 62 and 54 having other optical properties than the hole ice of the other strings, or by the position of the hole-ice cylinders of strings 62 and 54, shielding those strings less from the direction of the sending string 63. A systematic simulation study to investigate those scenarios is out of scope of this study, but is suggested as a follow-up study.

In this flasher study, all optical modules are assumed to be positioned central in their respective hole-ice cylinders. In the simulation, hits are recorded in all optical modules of the detector. For a first attempt, however, only the receiving strings 55, 64, 71, and 70 are used to fit the properties of the hole ice.[27] Figure 60 shows STEAMSHOVEL visualizations of the simulation scenario.

Likelihood As a measure for comparing the calibration data to the simulation, a Poisson likelihood L is used as a basis.

$$L = \prod_i \mathrm{P}_{\lambda_i}(k_i) = \prod_i \frac{\lambda_i^{k_i}\, \mathrm{e}^{-\lambda_i}}{k_i!} \tag{14}$$

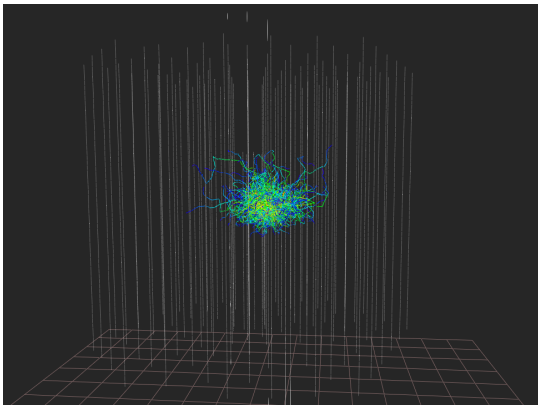
The index i refers to the individual receiving optical modules. Each factor $\mathrm{P}_{\lambda_i}(k_i)$ represents the probability that module i registers k_i hits, given by the calibration data, while λ_i, which is the mean of the Poisson distribution, represents the expected number of hits from the simulation.

To account for the finite statistics of the simulation, this likelihood can be generalized. Rather than using just the number λ_i of expected hits, which assumes infinite statistics for the simulation, one uses the number $k_{i,\mathrm{MC}}$ of Monte-Carlo hits, which is the number of hits seen in the simulation, and a weight w, which quantifies the ratio of calibration-data statistics and simulation statistics, to account for the results from the simulations. [Glü18, equation 21]

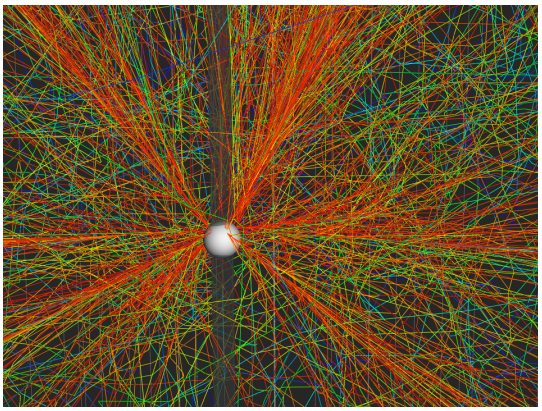
$$L = \prod_i \frac{\left(\frac{1}{w}\right)^{k_{i,\mathrm{MC}}} \cdot \left(k_i + k_{i,\mathrm{MC}} - 1\right)!}{\left(k_{i,\mathrm{MC}} - 1\right)! \cdot k_i! \cdot \left(1 + \frac{1}{w}\right)^{k_i + k_{i,\mathrm{MC}}}} \tag{15}$$

If, in order to improve accuracy, the simulation has 10 times the number of photons as compared to the calibration data, the weight would be $w = 1/10$ as each simulated

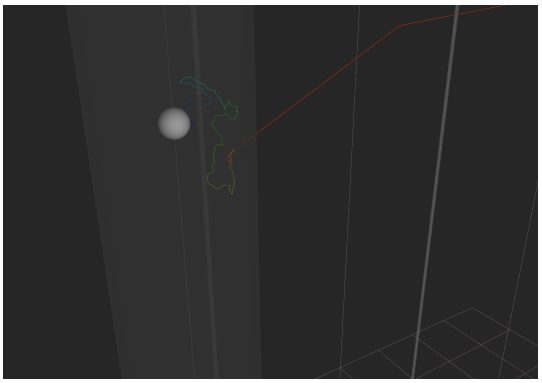
[27] If the dominant effect causing the asymmetry is the displacement of the sending optical module within the hole ice, excluding strings 62 and 54 leads to a systematic error. If the dominant effect causing the asymmetry is the shadowing cable, including those strings leads to a systematic error. Further studies are required to account for both possibilities.



(a) Total view of the detector. The flasher LEDs have been activated with full brightness. To see separate photon tracks in the image, only a fraction of 10^{-8} of the photons is shown in this image. The detector strings are about 125 m apart. The optical modules can detect light emitted up to 500 m away. [Aar+17b]



(b) Light emitted by the LEDs at the sending optical module. The modules is positioned centered in a bubble column with a diameter of 60 % of the diameter of the optical module. Only a fraction of 10^{-8} of the photons is shown in this image.



(c) To emphasize the effect of the hole ice, a single photon is isolated in this image, entering a hole-ice cylinder with a diameter of 500 % of the diameter of the optical module (larger than in the real simulation). The scattering probability increases significantly within the cylinder.

Figure 60: STEAMSHOVEL visualization of the flasher scenario. LEDs at a central optical module emit light that propagates through the ice and is then received at surrounding optical modules.

photon would correspond only to $^1/_{10}$ real photons. If, in order to save computation time, only $^1/_{10}$ of the number of photons from the calibration data are simulated (section 5.5), the weight is $w = 10$ as each simulated photon corresponds to 10 photons from the calibration data.

Results Figure 61 shows a contour plot of the conducted parameter scan. The likelihood $L_{\mathcal{H}}$ of agreement of calibration data and the simulation using the hole-ice parameters $\mathcal{H}$ given by the coordinate axes, or rather $-2\Delta\text{LLH} := -2(\ln L_{\mathcal{H}} - \ln L_{\mathcal{H}_\text{b}}) = -2\ln\left(L_{\mathcal{H}}/L_{\mathcal{H}_\text{b}}\right)$, is plotted against the hole-ice parameters. $L_{\mathcal{H}_\text{b}}$ is the maximum likelihood that corresponds to simulation with the hole-ice parameters $\mathcal{H}_\text{b}$ that best describe the calibration data, such that $\Delta\text{LLH} = 0$ corresponds to the simulation that best agrees with the calibration data.

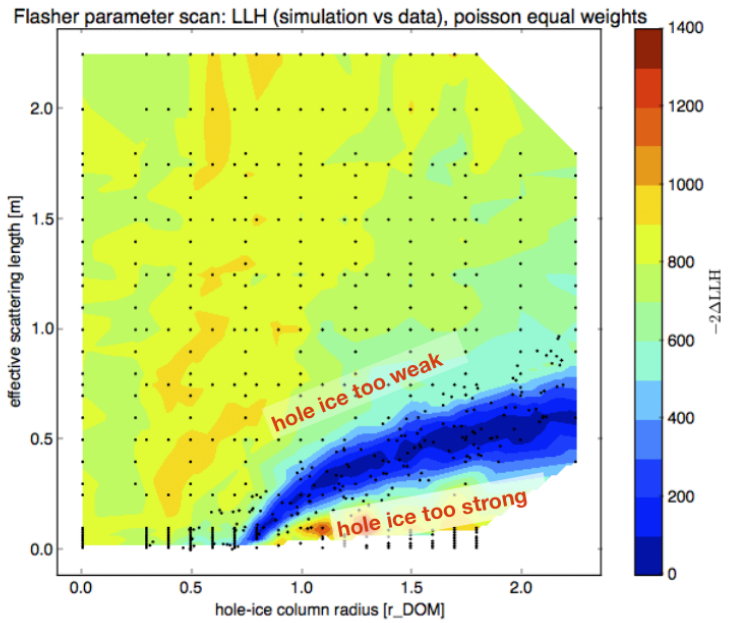


Figure 61: Agreement of flasher calibration data with flasher simulations for different hole-ice parameters. The axes show the radius of the simulated hole-ice column and the effective scattering length of the hole ice. The blue "valley" shows the region of best agreement of data and simulation. Above the valley, the simulated hole ice is too weak, below the valley it is too strong to account for the amount of photon hits in the calibration data.

The contour plot shows a valley (blue area) where the simulated hole ice leads to good agreement of simulation and calibration data. If the radius of the simulated hole-ice cylinder is too small, or the scattering length of the hole ice is too large, the hole-ice effect is too weak and the amount of light arriving at the receiving

optical modules is too high compared to the flasher calibration data. If, on the other hand, the hole-ice cylinder is too large, or the scattering length of the hole ice is too small, the hole-ice effect is too strong and the amount of light arriving at the receiving optical modules is too low compared to the flasher calibration data.

The optimal agreement of simulation and calibration data is achieved with hole-ice parameters $\mathcal{H}_\mathrm{b}$: $r = 0.83\, r_\mathrm{DOM}, \lambda_\mathrm{e}^\mathrm{H} = 0.10\,\mathrm{m}$, with a log likelihood $\mathrm{LLH} = \ln L_{\mathcal{H}_\mathrm{b}} = -278.66$. Note, however, that this example study is only to be considered a proof of concept for this kind of flasher parameter scan. Due to a number of systematics, in particular due to the unrealistic assumption that all optical modules are symmetrically centered within their hole-ice columns, it is expected that the study does not fit the correct hole-ice properties.

Systematics First, an outdated configuration for the positions of the optical module has been used in these simulations. Preliminary investigations show that based on newer calibration data, the positions of the optical modules have been updated on the order of 1 m. Thus, newer position calibration information should be used for follow-up studies.

Next, an outdated set of ice-model parameters has been used in this set of simulations. In this study, the `spice_mie` ice model from 2010 has been used. For a proper flasher study, however, current ice models with improved fits for the scattering and absorption lengths, which depend on the ice layer, in particular the dust concentration in these layers, the ice temperature, photon wavelength, and photon direction, should be used. (See [Col+13; Chi17].) The tilt of the ice layers and the absorption anisotropy (section 3.3) have not been included in these simulations, because those features are not yet re-implemented for the new medium-propagation algorithm.[28] These ice properties are negligible for short propagation distances, but become important for distances larger than about 10 m, which is given for flasher studies where the spacing of sending and receiving optical module is on the order of 100 m.

Most importantly, the displacement of the optical modules relative to their hole-ice columns has not been considered in this study and should be included in follow-up simulations. In this study, two receiving strings have been deliberately excluded due to the observed asymmetry (figure 59). After determining the effect of the shadowing cable in simulations, however, the displacement of the optical module along the direction suggested by the observed asymmetry should be included as fit parameter. Preliminary investigations show that neglecting the displacement of

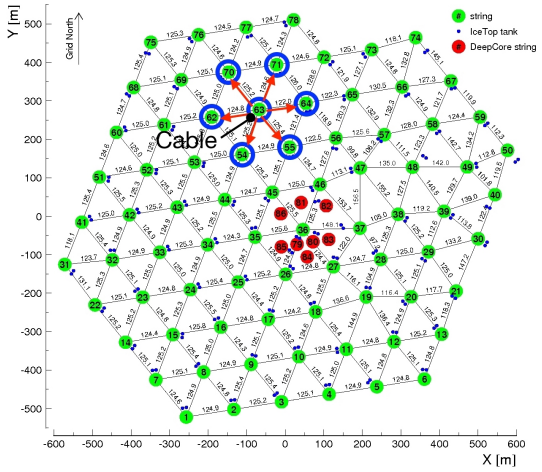
[28] For the current status of re-implementing ice layers and ice anisotropy, see https://github.com/fiedl/hole-ice-study/issues/48.

the optical module causes systematic uncertainties for the fitted hole-ice scattering length on the order of 30 %.

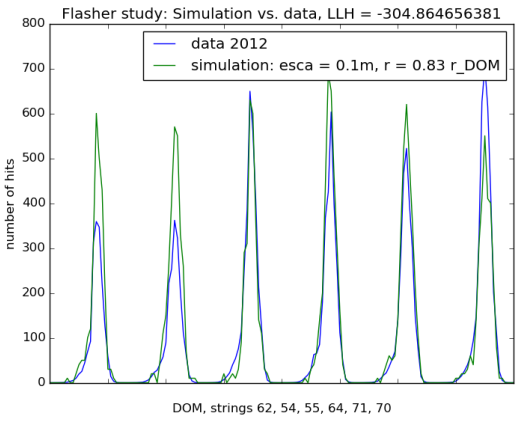
6.8 Comparing Flasher-Calibration Data to a Flasher Simulation With Cable

As shown in section 6.7, evaluating calibration data indicates a strongly asymmetric shadowing effect when flashing the LEDs of a central optical module and measuring the amount of light detected by the surrounding optical modules. This effect may either be caused by the LEDs of the sending optical module being asymmetrically shielded, or by the receiving optical modules in two of the receiving strings being more shielded than in the other receiving strings, or by a combination of the two.

As the main cable at the side of the sending optical module causes asymmetric shielding as shown in section 6.6, the question arises whether the cable shadow is sufficient to account for the asymmetries observed in the flasher calibration data. Figure 62 (a) shows a flasher-simulation scenario where an opaque cable with 4 cm diameter is placed besides the sending optical module such that it would reduce the emitted light in the direction of the two strings that show a reduced amount of detected light in the calibration data.



(a) Top view of the detector strings with position of the shadowing cable. Image based on [Wos11].



(b) Amount of light seen by the receiving optical modules. From left to right each peak corresponds to one of the strings 62, 54, 55, 64, 71, and 70.

Figure 62: Flasher-simulation scenario with a shadowing cable besides the sending optical module.

This flasher simulation with shadowing cable is documented in `issues/97` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/97.

In this flasher simulation, the sending and the receiving optical modules are embedded in a bubble column of a radius of 83 % of the optical module and an effective bubble-column scattering length of 10 cm, which are the best-fit parameters of the flasher scan in section 6.7 that only considers the supposedly unshadowed receiving strings 55, 64, 71, and 70. All bubble columns are positioned symmetrically relative to the embedded optical modules, such that the bubble columns only cause an overall reduction, but no azimuthal asymmetry. The distances of the sending optical module to the receiving strings range from 122.0 m to 125.4 m and their differences can only cause a peak reduction of about 2 %. Therefore, in this scenario, the shadowing cable is the only remaining candidate to cause the observed peak reduction of about 32 % observed in the calibration data. However, comparing results of the flasher simulation with shadowing cable to the flasher-calibration data (figure 62 b) makes it implausible that the shadowing cable alone could be the cause of the azimuthal asymmetry.

Hole ice, on the other hand, is a suitable candidate to account for the observed asymmetric peak reduction. Assuming the entire drill hole being filled with hole ice, varying the effective hole-ice scattering from 60 cm to 40 cm can cause a peak reduction in the observer magnitude of 30 %. As section 6.4 has shown that the displacement of the optical modules relative to the hole ice may cause strong asymmetric effects, it is plausible that a displacement of the optical modules relative to a hole ice with suitable optical properties is required to explain the asymmetric effects observed in calibration data.

7 Discussion

7.1 Comparison to Other Hole-Ice-Simulation Methods

7.1.1 A Priori Modification of Angular-Acceptance Curves

Using a modified angular-acceptance curve $a(\eta)$ as acceptance criterion of the optical modules in simulations to effectively account for the effect of hole ice on the detection of photons, is the default approach in CLSIM (see sections 6.2.4 and 6.2.6). The respective angular-acceptance curves have been obtained using lab measurements and simulations using the PHOTONICS photon-propagation software. [Col+13; Lun+07; Wos04]

> Information on the nominal and the hole-ice-approximating angular-acceptance curves are provided or linked in `issues/10` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/10.

Figure 63 shows three modified angular-acceptance curves that approximate the hole-ice effect in comparison to the optical module's angular acceptance based on the lab measurement. The modified curves H1, H2, and H3 assume that the entire drill hole is filled with hole ice, corresponding to a hole-ice-cylinder radius of 30 cm, and assume geometric hole-ice scattering lengths of $\lambda_{\text{sca}}^{\text{H}} = 100\,\text{cm}$ (H1), $\lambda_{\text{sca}}^{\text{H}} = 50\,\text{cm}$ (H2), and $\lambda_{\text{sca}}^{\text{H}} = 30\,\text{cm}$ (H3) respectively.

Using modified angular-acceptance curves for the optical modules to approximate the hole-ice effect has the advantage that new parameters gained from calibration studies can be inserted into the existing simulations just by replacing the polynomial parameters used for the angular acceptance of the optical modules. Also, using modified acceptance curves is faster than simulating the hole ice directly as the many scattering steps that correspond to the photons scattering within the hole ice are not actually simulated and just effectively accounted for using when the algorithm decides whether a photon hit is accepted by an optical module.

This method, however, relies on strong assumptions: The lab measurements the acceptance curves are based on, have been performed using manufactured ice rather than South-Polar ice. The hole-ice modifications are based purely on simulations and do not use any ICECUBE calibration information. [Col+13] Furthermore, this method assumes all optical modules having exactly the same properties regarding positioning and relative orientation. The optical modules are assumed to be perfectly centered within the drill hole as well as in the bubble column.

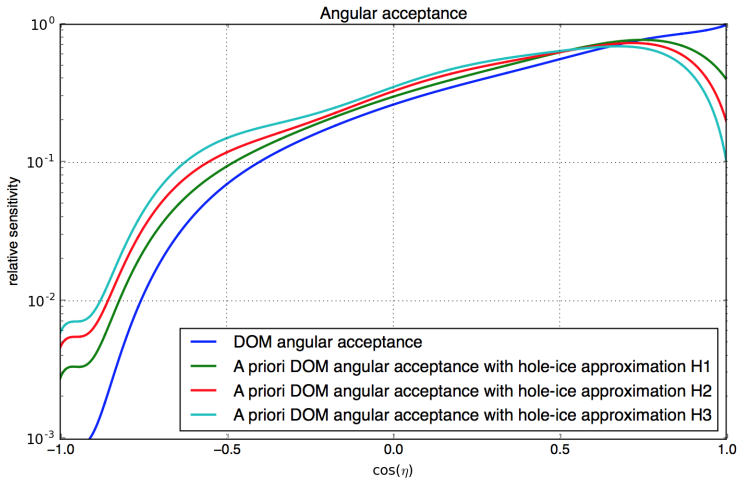


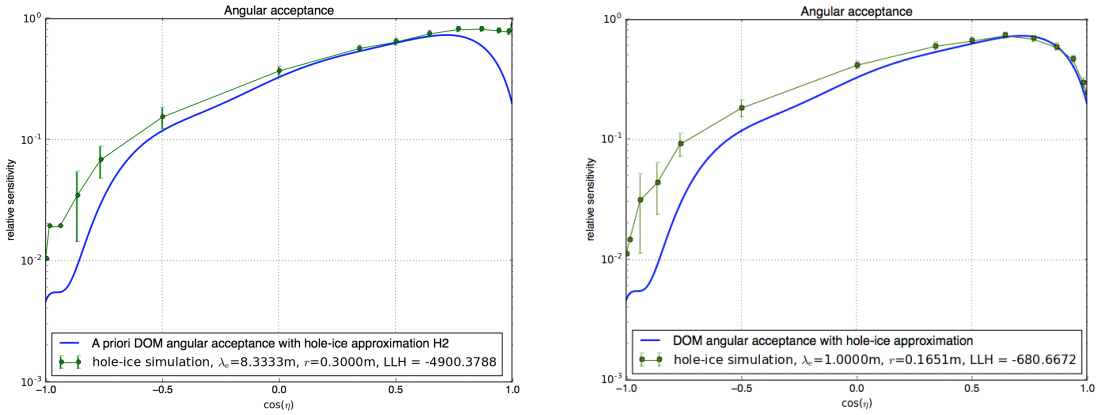
Figure 63: A priori angular-acceptance curves: blue: without hole ice, H1: assuming a hole-ice-cylinder radius if 30 cm and a geometric hole-ice scattering length of $\lambda_{\text{sca}}^{\text{H}} = 1.00\,\text{m}$, corresponding to an effective hole-ice scattering length of $\lambda_{\text{e}}^{\text{H}} = 16.67\,\text{m}$, H2: $\lambda_{\text{sca}}^{\text{H}} = 0.50\,\text{m}, \lambda_{\text{e}}^{\text{H}} = 8.33\,\text{m}$, H3: $\lambda_{\text{sca}}^{\text{H}} = 0.30\,\text{m}, \lambda_{\text{e}}^{\text{H}} = 5.00\,\text{m}$. [Col+13; Kar98; Chi16]

The a priori H2 acceptance curve, which is used in standard-CLSIM simulations by default, is used as reference curve for other angular-acceptance plots in this study. However, comparing this reference curve, which itself assumes a hole-ice-cylinder radius of 30 cm and a geometric scattering length of 50 cm (H2 parameters) [Chi16], to a CLSIM simulation using the same H2 parameters (figure 64), both angular-acceptance curves do not match. Instead, the reference curve best matches a CLSIM simulation, which assumes a hole-ice radius of the same size as the optical module, $r = 0.17\,\text{m}$, and an effective hole-ice scattering length of $\lambda_{\text{e}}^{\text{H}} \approx 1.0\,\text{m}$ (section 6.3).

7.1.2 Angular-Acceptance Fitting Using Flasher Data

CHIRKIN [Chi17] suggests a hole-ice model, here referred to as *Dima's model*, resulting in an angular-acceptance curve, $a_{\text{DOM,HI}}^{\text{Dima}}(\eta; p)$, different from the a priori curves for the H1, H2, H3 parameters. This angular-acceptance curve can be used as acceptance criterion of the optical modules in simulations instead of the a priori curves described in section 7.1.1.

The angular-acceptance curve of Dima's model is parameterized using a single parameter $p \in \mathbb{R}$ that has been determined as $p \in [0.2; 0.4]$ using flasher studies. [And16] In this model, the hole-ice-cylinder radius r and the scattering length $\lambda_{\text{sca}}^{\text{H}}$



(a) CLSIM simulation with H2 hole-ice parameters: $r = 30\,\text{cm}, \lambda_{\text{sca}}^{\text{H}} = 50\,\text{cm}, \lambda_{\text{sca}}^{\text{H}} = 8.33\,\text{m}$

(b) CLSIM simulation with parameters $r = r_{\text{DOM}} = 0.1651\,\text{m}, \lambda_{\text{sca}}^{\text{H}} = 6\,\text{cm}, \lambda_{\text{e}}^{\text{H}} = 1.0\,\text{m}$

Figure 64: Comparing the a priori angular-acceptance curve assuming H2 hole-ice parameters from [Col+13; Kar98] to CLSIM simulations.

within the hole ice are unknown.[29]

$$\begin{aligned} a_{\text{DOM,HI}}^{\text{Dima}}(\eta; p) = {} & 0.34 \cdot (1 + 1.5\,\cos(\eta) - \cos(\eta)^{3/2}) \\ & + p \cdot \cos(\eta) \cdot (\cos(\eta)^2 - 1)^3 \\ & p \in [0.2; 0.4] \end{aligned} \tag{16}$$

Figure 65 shows this angular-acceptance curve, $a_{\text{DOM,HI}}^{\text{Dima}}(\eta; p)$, for the parameters $p = 0.2$, $p = 0.3$, and $p = 0.4$, compared to the a priori angular-acceptance curve $a_{\text{DOM,HI}}(\eta)$ of the H2 model and the angular-acceptance curve of an optical module $a_{\text{DOM}}(\eta)$ without hole ice. Notably, the effect of the hole ice on photons approaching the optical module directly from above or below is weaker in this model compared to the H2 approximation curve. However, the overall acceptance in the lower half sphere is higher, and the overall acceptance in the upper half sphere is lower as in the H2 approximation curve.

This approach does not rely on specific assumptions about the properties of the hole ice, but uses flasher calibration data to measure the mean angular-acceptance behavior of ICECUBE's optical modules. Nevertheless, this method relies on the same assumptions regarding the equality of all optical modules: All optical modules

[29] As the hole-ice parameters are not given by Dima's model, one cannot directly compare the model to CLSIM simulations with direct hole-ice propagation. A parameter scan to find the best hole-ice parameters for Dima's model is suggested as follow-up study. See: https://github.com/fiedl/hole-ice-study/issues/114

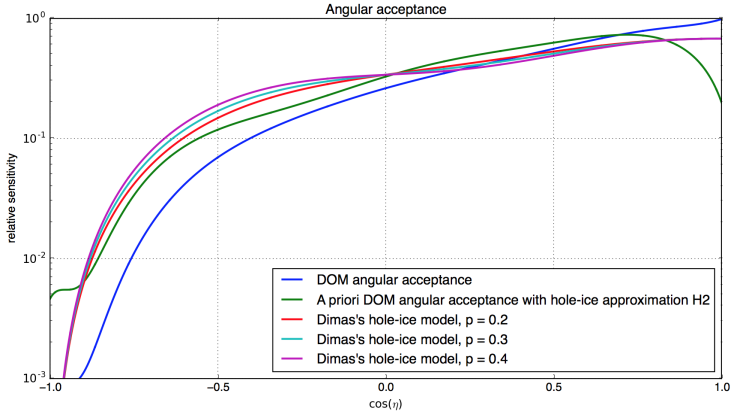


Figure 65: Comparing the a priori angular-acceptance curve H2 from [Col+13] to Dimas's model [Chi17]. Also, the angular acceptance of the optical module (DOM) without any hole ice is shown.

are assumed to have the same relative position and orientation with respect to the hole ice.

7.1.3 Direct Photon Propagation With Ppc

The IceCube Simulation Framework includes two independent photon-propagation-simulation tools, Clsim and Ppc (Photon Propagation Code). [Chi14; Chi18; Lan18] Similar to this study, which uses Clsim, simulations with direct propagation through hole ice have been performed using Ppc. [Ron17b; Ron15; Ron17a; RRK17; Chi14]

The source code of Ppc can be found at http://code.icecube.wisc.edu/projects/icecube/browser/IceCube/projects/ppc.

Simulating the propagation of photons through the hole ice by simulating the scattering directly drops the assumption that all optical modules need to be equally positioned and oriented with respect to the hole ice. All optical modules can be individually calibrated using flasher data. [Ron17b]

The basic propagation algorithm is the same in Ppc and Clsim. The algorithm for propagating the photons through different media, relies on the same principle of converting randomized numbers of interaction lengths to a geometric distances, but is implemented differently: Ice layers and hole-ice cylinders are hard-coded in Ppc whereas in Clsim's new medium-propagation algorithm, they are treated as generic medium changes (see section 5.4). Both, Ppc and Clsim, can be run on clusters

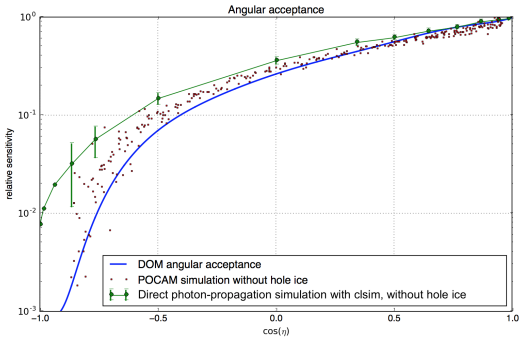
of graphics processing units. [Chi14; Chi18; Lan18; Kop17] Hardware requirements and performance of PPC and CLSIM are comparable. Due to micro optimizations, PPC tends to run slightly faster.[30] PPC does only support one hole-ice cylinder per string. The main cable of the detector string is not implemented as absorbing object but rather accounted for by using a direction-dependent shadowing factor for incoming photons. [Chi18; Lan18] In particular, the cable cannot be modeled to reside inside the bubble-column cylinder. [Ron17b]

Since both propagation softwares, PPC and CLSIM with the new medium-propagation mechanism, are performing the same task, it is important to know whether both simulations produce the same results regarding the effect of the hole ice. A detailed comparison of PPC and CLSIM results is out of scope of this study. But a first comparison can be achieved using existing PPC simulation results. In feasibility studies for the proposed PRECISION OPTICAL CALIBRATION MODULE (POCAM), direct photon propagation simulations with PPC have been performed, assuming different hole-ice parameters, and producing effective angular-acceptance curves corresponding to those hole-ice parameters. [Ron17a; RRK17] Assuming the same hole-ice parameters respectively, angular-acceptance simulations can be performed in CLSIM, allowing for a direct comparison of the PPC and the CLSIM results (figure 66).

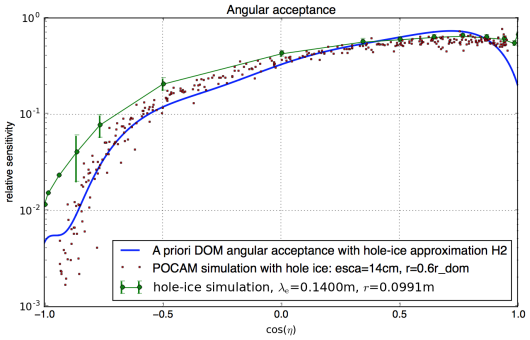
The CLSIM angular-acceptance simulations with hole-ice parameters from the POCAM simulations are documented in `issues/103` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/103.

At a rough view, both simulations produce similar effective angular-acceptance results for different hole-ice parameters. For upper angles, however, this study's simulation systematically sees more photon hits than PPC. As this has already been observed in section 6.2.5 (see figure 45) when introducing plane waves as photon sources, this effect could be a systematic problem of the implementation of the angular-acceptance simulations rather than of the propagation algorithm. Further investigations on this matter are required, but are out of scope of this study.

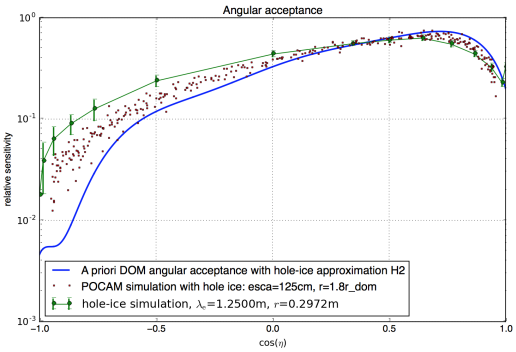
[30] Internal correspondence suggests a performance factor of 1.5 to 2.0. See also section 7.3 for performance considerations.



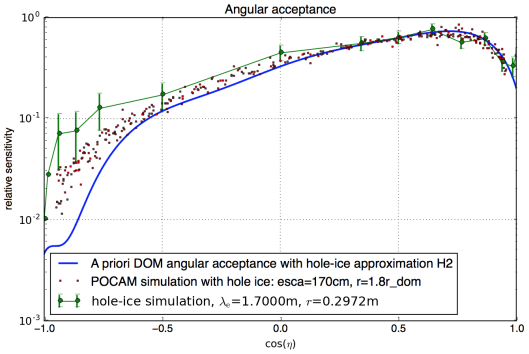
(a) No hole ice in both simulations.



(b) Hole-ice-cylinder radius $r = 0.6\,r_{\mathrm{DOM}} = 10\,\mathrm{cm}$, hole-ice effective scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 14\,\mathrm{cm}$, corresponding to a geometric scattering length $\lambda_{\mathrm{sca}}^{\mathrm{H}} = 0.8\,\mathrm{cm}$ in both simulations.



(c) Hole-ice-cylinder radius $r = 1.8\,r_{\mathrm{DOM}} = 30\,\mathrm{cm}$, hole-ice effective scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 125\,\mathrm{cm}$, corresponding to a geometric scattering length $\lambda_{\mathrm{sca}}^{\mathrm{H}} = 7.5\,\mathrm{cm}$ in both simulations.



(d) Hole-ice-cylinder radius $r = 1.8\,r_{\mathrm{DOM}} = 30\,\mathrm{cm}$, hole-ice effective scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 170\,\mathrm{cm}$, corresponding to a geometric scattering length $\lambda_{\mathrm{sca}}^{\mathrm{H}} = 10.2\,\mathrm{cm}$ in both simulations.

Figure 66: Comparison of effective angular-acceptance curves, which include hole-ice effects for varying hole-ice parameters, created with two independent propagation tools, Ppc and Clsim. As reference, the a priori effective angular-acceptance curve for H2 parameters from [Col+13] is shown. Data source of the Ppc simulations is a series of POCAM simulations [RRK17]. The radius of the optical module is $r_{\mathrm{DOM}} = 16.510\,\mathrm{cm}$.

7.2 Compatibility of Hole-Ice Parameters From Other Studies

Simulations with direct photon propagation through hole ice of different parameter configurations $\mathcal{H}$ allow to rate the compatibility of different hole-ice models. For example, generating simulation-based angular-acceptance curves allows to calculate the likelihood of two models being compatible. This, however, would require a more involved study of systematics of the specific simulation design, which is out of scope of this study, but should be considered by follow-up studies.

As a first step, this section shows angular-acceptance curves from simulations with hole-ice parameters from different studies. Table 2 at the end of this section (page 133) summarizes the findings.

7.2.1 YAG H2 Parameters

Based on laser measurements, using a Yttrium-Aluminium-Granat (YAG) laser, [Kar98] suggested a number of hole-ice models, from which the so called *H2 model* has become the dominant hole-ice model in IceCube. This model suggests a geometric scattering length of 50 cm for the hole ice, and a hole-ice-cylinder radius of 30 cm, and is the basis for the modified angular-acceptance curve with hole-ice approximation described in section 7.1.1.

Using the new direct photon propagation though hole ice with Clsim, one can simulate the propagation through hole ice with parameters suggested by the H2 model, scan the angular acceptance of an optical module within the simulation, and compare the result to the existing a priori angular-acceptance curve (section 7.1.1) [Col+13] as well as the angular-acceptance curve suggested by Dima's model (section 7.1.2) [Chi17]. Figure 67 shows the simulation results.

This angular-acceptance simulation using the H2 hole-ice parameters is documented in `issues/80` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/80.

For photons approaching the optical module from below (right-hand side of the plot in figure 67), the simulation rather follows Dima's model, but counts more detected photons in total. For photons approaching from above (left-hand side of the plot), the form of the simulation curve is similar to the a priori curve, in particular with respect to the plateau on the left hand side, but, again, the simulation counts more photons in total.

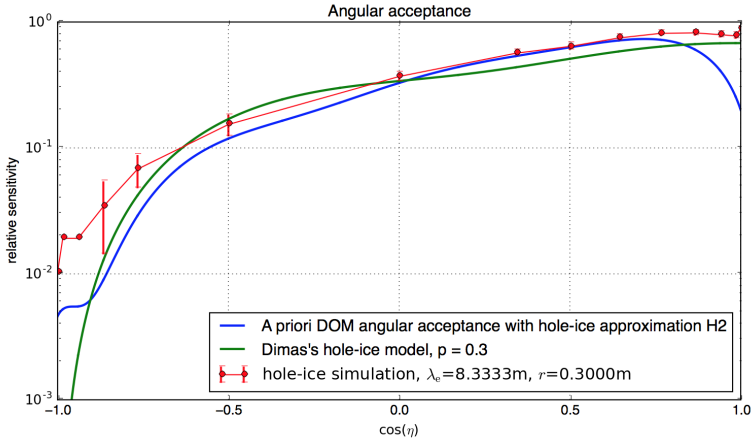


Figure 67: Angular-acceptance simulation with hole-ice parameters from the so called *H2 model* [Kar98], which describes the hole ice as cylinder of 30 cm radius filling the entire drill hole, with a geometric scattering length of 50 cm, corresponding to an effective scattering length of $\lambda_e^H = 8.33\,\text{m}$, using the new medium-propagation algorithm (section 5.4) with direct detection as acceptance criterion and plane waves as photon sources. For comparison, the a priori angular-acceptance curve from [Col+13] and the angular-acceptance curve of Dima's model [Chi17] are shown.

7.2.2 DARD Parameters

In the DARD study (Data Acquisition for a flasheR DOM), LED measurements are used to determine the hole-ice properties. The different flasher LEDs of an optical module are turned on in sequence, and the light detected by the same DOM is measured. The process is then compared to detailed Geant4 simulations[31] for different hole-ice parameters in order to find the optimal hole-ice parameters to explain the data. [Ron15; Ron17b]

The DARD study assumes a fixed hole-ice radius of $r = 30\,\text{cm}$. The best fit value for the geometric scattering length of the hole ice has been determined as $\lambda_{\text{sca}}^H = 10\,\text{cm}$, which is considerably lower than other estimates. [Ron15]

Using the new direct photon propagation though hole ice with Clsim, one can simulate the propagation through hole ice with parameters suggested by DARD, scan the angular acceptance of an optical module within the simulation, and compare the result to the existing a priori angular-acceptance curve [Col+13] as well as the angular-acceptance curve suggested by Dima's model [Chi17]. Simulation results are shown in figure 68.

[31] Geant4 toolkit for the simulation of the passage of particles through matter, https://geant4.web.cern.ch

This angular-acceptance simulation for the hole-ice parameters suggested by the DARD study, is documented in `issues/105` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/105.

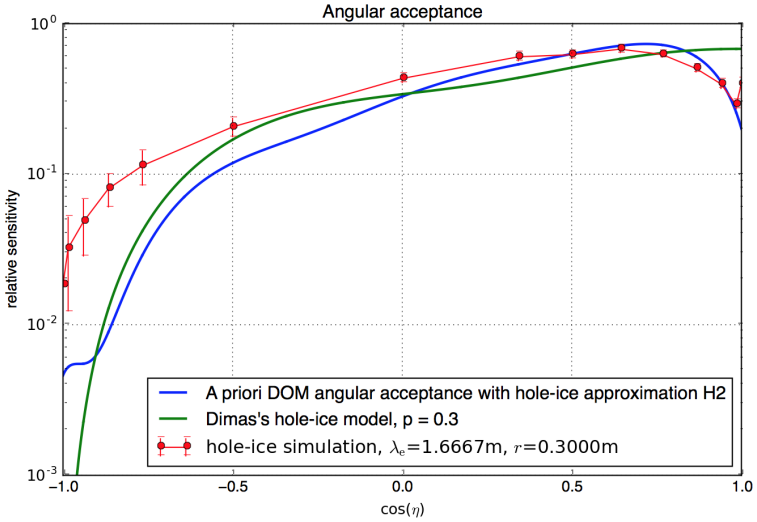


Figure 68: Angular-acceptance simulation with hole-ice parameters from the DARD study [Ron15], which suggest a hole-ice cylinder of 30 cm radius and a geometric scattering length of $\lambda_{\text{sca}}^{\text{H}} = 10\,\text{cm}$, corresponding to an effective scattering length of $\lambda_{\text{e}}^{\text{H}} = 1.67\,\text{m}$, using the new medium-propagation algorithm (section 5.4) with direct detection as acceptance criterion and plane waves as photon sources. For comparison, the a priori angular-acceptance curve from [Col+13] and the angular-acceptance curve of Dima's model [Chi17] are shown.

For photons approaching the optical module from below (right-hand side of the plot in figure 68), the simulation curve rather follows the a priori acceptance curve, but shows a less sharp hole-ice drop-off effect. For higher angles, $\cos\eta < 0.5$, the simulation registers relatively more photons, both in comparison to the a priori curve and to Dima's model.

7.2.3 SpiceHD Parameters

The SpiceHD study (South Pole ICE model with Hole ice fitting and Direct detection) uses flasher data and direct photon propagation simulations with Ppc to determine the parameters of the hole ice. In this 7-string flasher study, LED flashes are sent from a central optical module and the numbers and time distributions of photon hits are observed at the optical modules of the six surrounding strings.[32]

[32] SpiceHD uses the same methods as [Col+13; Chi13]

Comparing the flasher data to simulations with different hole-ice parameters, scanning over the hole-ice-cylinder radius and the hole-ice scattering length, a range of hole-ice parameters is found to be suitable (section 7.2.4). When assuming a compatible hole-ice column radius as observed by camera images [Ron16], the best fit hole-ice parameters have been determined as a hole-ice-cylinder radius of 60 % of the radius of an optical module, $r = 0.6\, r_{\mathrm{DOM}} = 10\,\mathrm{cm}$, and the hole-ice effective scattering length $\lambda_{\mathrm{e}}^{\mathrm{H}} = 14\,\mathrm{cm}$, which corresponds to a geometric scattering length of $\lambda_{\mathrm{sca}}^{\mathrm{H}} = 0.84\,\mathrm{cm}$, which is even smaller than the geometric scattering length of 10 cm suggested by DARD. [Ron17b]

Using the new direct photon propagation though hole ice with CLSIM, one can simulate the propagation through hole ice with parameters suggested by SPICEHD, scan the effective angular acceptance of an optical module within the simulation, and compare the result to the existing a priori angular-acceptance curve [Col+13] as well as the angular-acceptance curve suggested by Dima's model [Chi17] (figure 69).

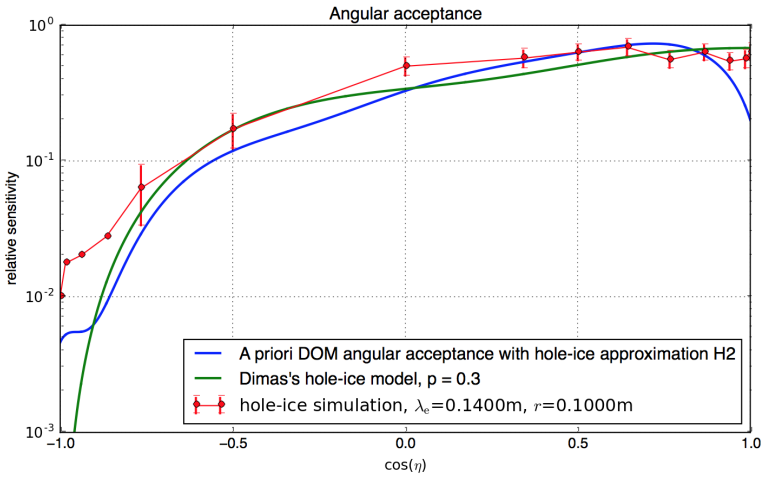


Figure 69: Angular-acceptance simulation with hole-ice parameters suggested by the SPICEHD study [Ron17b]: The dominant hole ice is assumed as a bubble column with radius $r = 0.6\, r_{\mathrm{DOM}}$ where r_{DOM} is the radius of the optical module. The bubble column is thinner than the optical module. The hole-ice effective scattering length is $\lambda_{\mathrm{e}}^{\mathrm{H}} = 14\,\mathrm{cm}$. The simulation uses the new medium-propagation algorithm (section 5.4) with direct detection as acceptance criterion and plane waves as photon sources. The simulation results are shown in comparison to the a priori angular-acceptance curve from [Col+13] and the angular-acceptance curve of Dima's model [Chi17].

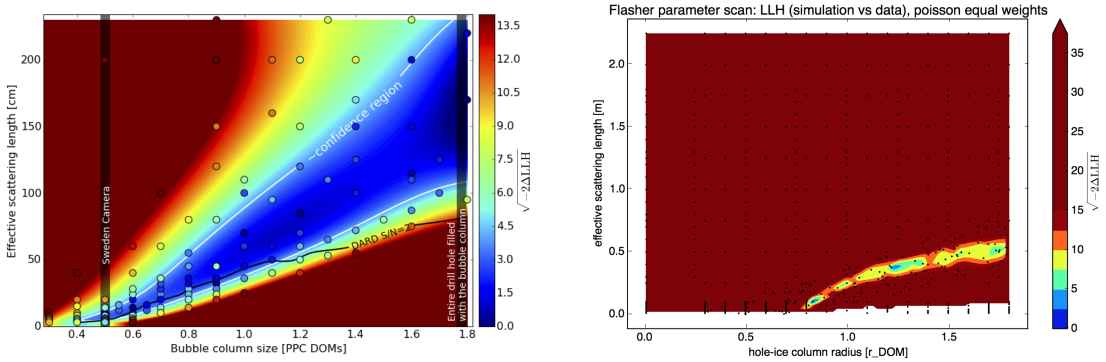
This angular-acceptance simulation for the hole-ice parameters suggested by the SPICEHD study, is documented in `issues/87` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/87.

For photons that approach the optical module from below (right-hand side of the plot in figure 69), the simulation shows a hole-ice effect between the a priori curve and Dima's model. For photons approaching from above, the simulation shows a plateau. For $\cos\eta < 0.5$, the simulation shows more photon hits than both, the a priori curve and Dima's model.

7.2.4 SpiceHD Flasher-Scan Contours

The SpiceHD study (South Pole ICE model with Hole ice fitting and Direct detection, see also section 7.2.3) performs a similar flasher study as described in section 6.7. Both studies compare simulations using direct photon propagation through hole ice of varying parameters to flasher calibration data. Whereas section 6.7 uses the new medium-propagation algorithm for Clsim, SpiceHD uses Ppc to propagate the photons. [Ron17b]

The most important difference between the two studies is that the flasher scan of section 6.7 assumed all optical modules being centered relative to their hole-ice column, but SpiceHD uses the positions of the optical modules relative to the hole ice as additional fit parameters. These additional fit parameters make SpiceHD's contour plot (figure 70 a) smoother and the range of good fit parameters wider. This can best be seen when adjusting the likelihood scale of the flasher scan of section 6.7 to match the scale of SpiceHD, as shown in figure 70 (b).



(a) SpiceHD study using direct hole-ice propagation with Ppc [Ron17b], fitting the size and scattering length of the hole ice and the position of the optical modules relative to the hole ice.

(b) Flasher study with direct hole-ice propagation with Clsim (section 6.7), fitting only the size and the scattering length of the hole ice.

Figure 70: Fitting flasher simulations for different hole-ice parameters to calibration data.

Both studies show a "valley"-like region of good agreement of data and simulation. In the parameter region above the valley, where scattering lengths are longer or hole-ice radii are smaller, the simulated hole ice is too weak. In the parameter

region below the valley, the simulated hole ice is too strong. However, both, the parameter region of the valley and the curvature are different.

As CLSIM and PPC show good agreement when comparing angular-acceptance curves (section 7.1.3), this difference is expected to be caused by the systematics of the preliminary flasher study, discussed in section 6.7, and the different assumptions regarding the positions of the optical modules relative to the hole ice. Preliminary estimations show that this difference accounts roughly for a factor of 3 for the fitted scattering length when assuming large hole-ice radii. This agrees with the behavior seen in figure 70. For $r = 1.8\, r_{\mathrm{DOM}}$, figure 70 (b) shows an effective scattering length of 0.5 m. Figure 70 (a) shows about 1.5 m, which is three times as much.

The SPICEHD contours are compatible to camera observations (section 3.4) that show a hole-ice radius of about 10 cm. [Ron16] The matching SPICEHD hole-ice parameters are $r = 0.6\, r_{\mathrm{DOM}} = 10\,\mathrm{cm}$, $\lambda_{\mathrm{sca}} = 0.84\,\mathrm{cm}$, $\lambda_{\mathrm{e}} = 14\,\mathrm{cm}$. [Ron17b]

7.2.5 Compatibility of Hole-Ice Models

Direct photon-propagation-simulation through hole ice allows to compare different hole-ice models and hole-ice parameters. Both photon-propagation-simulation tools, PPC and CLSIM, produce matching effective angular-acceptance curves for direct photon propagation through hole ice. For photons approaching the optical module from above, the systematic uncertainties of the CLSIM angular-acceptance scans appear too large to consider this angular range. Therefore, this section focuses on the angular range $\cos\eta > 0$ with large statistics, when assessing the compatibility of different hole-ice models.

The SPICEHD flasher study is compatible to camera observations. The resulting hole-ice parameters $\mathcal{H}_{\mathrm{SpiceHD}}$ as well as the parameters $\mathcal{H}_{\mathrm{YAG\ H2}}$ suggested by earlier laser calibration studies, both suggest a weaker hole-ice effect than approximated in the modified a priori angular-acceptance curve used in standard CLSIM simulations. Dima's hole-ice model that only uses flasher-calibration data and does not rely on direct hole-ice simulations, also suggests a weaker hole ice. However, using the hole-ice parameters $\mathcal{H}_{\mathrm{DARD}}$ suggested by the single-LED calibration study DARD, angular-acceptance curves from direct simulations look similar to the a priori angular-acceptance curve. Table 2 shows a summary (page 133).

This leaves no obvious conclusion on which current hole-ice model is to be preferred. But as both direct hole-ice-simulation tools produce compatible results, both tools can now be used to study the various hole-ice scenarios in detail.

Method	Method Parameters	CLSIM+PPC $\mathcal{H}_{\text{YAG H2}}$	CLSIM+PPC $\mathcal{H}_{\text{DARD}}$	CLSIM+PPC $\mathcal{H}_{\text{SpiceHD}}$	A priori $\mathcal{H}_{\text{YAG H2}}$	Dima –
CLSIM+PPC	$\mathcal{H}_{\text{YAG H2}}$					
CLSIM+PPC	$\mathcal{H}_{\text{DARD}}$	✗				
CLSIM+PPC	$\mathcal{H}_{\text{SpiceHD}}$	✓	✗			
A priori	$\mathcal{H}_{\text{YAG H2}}$	✗	✓	✗		
Dima's model	–	✓	✗	✓	✗	

Table 2: Compatibility of angular-acceptance curves $a(\eta)$. **Methods**: CLSIM: Angular-acceptance simulation with CLSIM using the new medium-propagation algorithm with plane waves as photon sources and direct detection as acceptance criterion (section 6.2). PPC: POCAM-angular-acceptance simulation with PPC using direct photon propagation through hole ice and direct detection (section 7.1.3). CLSIM+PPC: Simulations performed with both, CLSIM and PPC, producing similar results. A priori: A priori angular-acceptance curve $a_{\text{DOM,HI}}(\eta)$ for $\mathcal{H}_{\text{YAG H2}}$ parameters from [Col+13] (section 6.2.6). Dima, Dima's model: Angular-acceptance curve $a^{\text{Dima}}_{\text{DOM,HI}}(\eta; p)$ from flasher unfolding studies [Chi17] (section 7.1.2). **Hole-Ice Parameters**: Each parameter set $\mathcal{H}$ consists of a hole-ice-cylinder radius r and a geometric hole-ice scattering length λ_{sca}, corresponding to an effective scattering length λ_{e}. $\mathcal{H}_{\text{YAG H2}}$: from YAG laser measurements [Kar98], also called H2 model parameters, $r = 30\,\text{cm}, \lambda_{\text{sca}} = 50\,\text{cm}, \lambda_{\text{e}} = 8.33\,\text{m}$ (section 7.2.1). $\mathcal{H}_{\text{DARD}}$: from single-DOM LED measurements [Ron15], $r = 30\,\text{cm}$, $\lambda_{\text{sca}} = 10\,\text{cm}, \lambda_{\text{e}} = 1.67\,\text{m}$ (section 7.2.2). $\mathcal{H}_{\text{SpiceHD}}$: from 7-string LED flasher studies [Ron17b] and camera observations [Ron16], $r = 0.6\,r_{\text{DOM}} = 10\,\text{cm}$, $\lambda_{\text{sca}} = 0.84\,\text{cm}$, $\lambda_{\text{e}} = 14\,\text{cm}$ (section 7.2.3). **Rating**: ✓: Curves look roughly similar. ✗: Curves do not look similar.

7.3 Performance Considerations

The performance of the different propagation algorithms, quantified by the time it takes to simulate a certain number of propagation steps, is about the same. The number of propagation steps, however, which is determined by the number of scatterings, has a significant effect on the total simulation time.

Figure 71 shows a comparison of the standard-CLSIM algorithm (section 5.2.2), the hole-ice-correction algorithm (section 5.3), and the new medium-propagation algorithm with hole ice (section 5.4), each running a simulation propagating 10^5 photons on a CPU.[33]

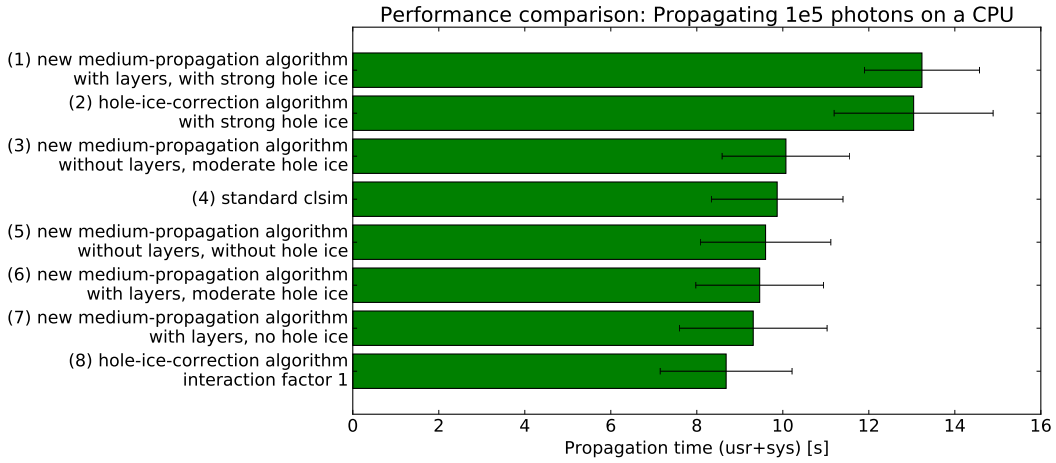


Figure 71: Performance comparison different hole-ice algorithms and different hole-ice scenarios.

> The implementation of this performance measurement is documented in `issues/49` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/49.

The difference of the total simulation time for each algorithm is smaller than the statistical uncertainty of the time measurement as long as the scattering length of the hole-ice cylinder is kept moderate (rows 3 to 8 in figure 71). When moving to smaller scattering lengths within the hole-ice, however, the total simulation time increases (rows 1 and 2 in figure 71), as the total number of simulation steps increases, because one simulation step propagates the photon from one scattering point to the next scattering point. For a fixed hole-ice-cylinder radius of 30 cm,

[33] Test system configuration for CPU simulation: OS: macOS Sierra 10.12.6 16G1212 x86_64, Kernel: 16.7.0, CPU: Intel i7-4870HQ (8) @ 2.50GHz, GPU: Intel Iris Pro, NVIDIA GeForce GT 750M, RAM: 16384MiB.

going from a geometric hole-ice scattering length of 20 m to 0.005 m increases CPU propagation time per photon by about 15 %.

The same effect can be observed when comparing the run time of flasher simulations on a GPU.[34] Running a flasher simulation with standard CLSIM using hole-ice approximation takes about 11 minutes. Running the same simulation with the same number of photons, with the new medium-propagation algorithm, but without any hole-ice cylinders takes about 10 minutes. Running the same simulation, but adding hole-ice cylinders with 36 cm radius and a scattering length of 1/10 of the surrounding bulk ice takes about 15 minutes, increasing the simulation time by about 50 %.

As the number of simulation steps is the dominant factor for the required simulation time, it is desirable to further optimize the implementation of the simulation step (section 5.5), because even small improvements, multiplied by the number increased number of scattering steps for hole-ice simulations, cause a considerable total performance gain.

These performance considerations also lead to the question when it is most useful to use hole-ice propagation algorithms and when to use the default hole-ice approximation with effective angular-sensitivity curves. The approximation does not consider the effective reflection of photons when entering the hole ice (section 6.1), but only removes a certain percentage of photons for each incoming angle. Also, the approximation considers each optical module perfectly centered within the hole ice rather than accounting for the individual displacement of each module. In high-energy events with large statistics where many photons and many different optical modules are involved, the displacement effects are expected to cancel out and the reflection effects are expected to be negligible. For these scenarios, in particular for events with long high-energy muon tracks, the hole-ice approximation is expected to be sufficient. For lower-energy events and events where only a few optical modules are involved, using a calibrated direct photon propagation simulation is expected to reduce simulation systematics. Further studies, however, are required to quantify this reduction of systematics, or equivalently to quantify the gain in precision.

[34] Test system configuration for GPU simulation: Scientific Linux release 7.4 (Nitrogen) x86_64, Kernel: 3.10.0-862.6.3.el7.x86_64, CPU: Intel Xeon E5-2660 0 (32) @ 3.000GHz, GPU: NVIDIA Tesla K20m, RAM: 64215 MiB

7.4 Features Not Considered in This Study

This study provides the means for a *direct propagation of photons through hole ice* and, in general, through cylinders of different ice properties (section 5.4). *Direct detection* is added as acceptance criterion of the optical modules (section 6.2.4).

For the propagation over large distances, the *ice-layer tilt* and the *ice anisotropy* (section 3.3) are omitted in this study and need to be re-implemented into the new medium-propagation algorithm.[35]

The hole ice is modeled as homogeneous cylinder with exact boundaries. A gradient of the hole-ice properties in the radial direction, which could model the radial gradient in the concentration of air bubbles in the bubble column (section 3.4) is not considered. Also, a gradient in z-direction to model a pressure-caused gradient in the ice properties of the hole ice (also section 3.4) is not included in this study.

The hole-ice cylinders are implemented perfectly aligned along the z-direction. A tilt of the cylinders, or even deformations of the cylinders which could be caused by oscillations of the drilling head are not considered in this study.

For the direct-detection acceptance criterion (section 6.2.4), the optical modules are assumed perfectly aligned along the z-direction. Tilted orientations of the optical modules are not accounted for in this study, even though the orientation of the optical modules could be fitted using calibration data. Also, the impact angle is completely ignored when using direct detection, even if transmission or quantum efficiency effects might be direction dependent.

This study also does not provide a ready-to-use geometry definition of the whole detector with all optical-module positions, hole-ice cylinders and cable positions.[36] Also, the performance impact when using such a full geometry file in production simulations is not examined in this study and needs to be evaluated in a follow-up study.

[35] For the current status of the re-implementation of ice-layer tilt and ice anisotropy, check https://github.com/fiedl/hole-ice-study/issues/48.

[36] For the current status of the creation of a full geometry file, see https://github.com/fiedl/hole-ice-study/issues/61.

8 Conclusion

This study presented a new method to simulate the propagation of photons through the hole ice around the detector strings of the IceCube neutrino observatory as an extension of the standard Clsim photon-propagation-simulation software.

Two algorithms were introduced to accomplish this task. As a first approach, an algorithm was developed that accounts for the different optical properties of the hole ice by adding subsequent corrections to the calculations of the standard-Clsim algorithm. The changes to the well-tested standard-Clsim codebase are kept minimal. The algorithm requires the definition of the hole-ice properties to be relative to the properties of the surrounding ice, which, however, does not allow to implement more current hole-ice models. Therefore, a second algorithm was devised that handles the transition from bulk ice to hole ice and vice versa the same way it handles other medium transitions like the propagation through ice layers, and allows to simulate hole ice with absolute scattering and absorption lengths. Because this approach is more generic, it also allows more complex hole-ice scenarios where a photon crosses more than one hole-ice cylinder between two scatters. This approach, however, required to rewrite the previous Clsim media-propagation code. The layer tilt and ice anisotropy are left out in the second algorithm for now, but can be added in the future. The validity of the algorithms is supported by unit tests and a series of statistical cross checks. Clsim hole-ice simulation results were found to agree with simulation results of Ppc, which is a separate IceCube simulation software that also supports direct hole-ice propagation, but uses a different approach to implementing the medium transitions. The performance difference per scattering step of the standard-Clsim media propagation and the new hole-ice algorithms is negligible within the statistical uncertainties. Simulating the direct photon propagation through hole ice with small scattering lengths, however, adds more scattering steps to the simulation, resulting in a longer total simulation time. The performance of large-scale simulations can still be improved by applying GPU-programming and memory optimizations.

In general, the new method allows to simulate the propagation through hole-ice cylinders with different absorption lengths, scattering lengths, and radii. The optical modules can be positioned with individual displacements relative to the hole ice. The method supports the nesting of several hole-ice cylinders of different properties and the simulation of light-absorbing cables. Optical modules support direct detection and thereby can accept photon hits based on whether the photon hit the sensitive area of the optical module rather than based on the impact angle, as done in standard Clsim. These features allow to fit additional calibration parameters such as the positions, sizes, and scattering lengths of the individual

hole-ice columns in the IceCube detector.

Direct-photon-propagation simulations indicate that light propagation through the IceCube detector on large scales is mostly unaffected by the hole ice. Each photon, however, that is eventually detected by the optical modules, and every photon that is emitted by the calibration LEDs, needs to propagate through the hole ice and is affected by the properties of the hole ice. For sufficiently sized hole-ice columns with small scattering lengths, the optical modules are effectively shielded by the hole ice. A fraction of the photons is absorbed during the random walk through the hole ice, or effectively reflected by the hole ice. Evaluating calibration data indicates a strongly asymmetric shielding of the detector modules. Preliminary flasher simulations with direct photon propagation hint that this cannot be accounted for by the shadow of cables, but can be explained by simulating hole ice with a suitable scattering length, size and position relative to the detector modules.

The new simulation method will be integrated into IceCube's simulation framework. Low-statistics studies can use the new simulation method to propagate all photons without hole-ice approximations in order to reduce systematic uncertainties. As the direct hole-ice propagation requires additional simulation time, high-statistics studies should continue to use approximation techniques, which modify the angular-acceptance behavior of the simulated optical modules to effectively account for the average effect of the propagation through hole-ice columns. Comparing the current standard hole-ice approximation to direct simulations, both methods were found to disagree. Assuming the same hole-ice properties, the new simulations indicate a less pronounced hole-ice effect than the approximation curves. As recent calibration studies contradict each other regarding the strength of the hole-ice properties, there is no clear indication towards one of these models. With the new simulation tools, however, more detailed hole-ice-simulation studies can be performed aiming to find a hole-ice description that allows to reproduce all calibration observations. After that, the approximation curves can be updated accordingly to match the new hole-ice model.

A Appendix

A.1 References

[Aar+13] M. G. Aartsen et al. "Evidence for High-Energy Extraterrestrial Neutrinos at the IceCube Detector". In: *Science* 342.6161 (Nov. 2013), pp. 1242856–1242856. DOI: 10.1126/science.1242856. URL: https://doi.org/10.1126/science.1242856.

[Aar+14] M G Aartsen et al. "Energy reconstruction methods in the IceCube neutrino telescope". In: *Journal of Instrumentation* 9.03 (Mar. 2014), P03009–P03009. DOI: 10.1088/1748-0221/9/03/p03009. URL: https://doi.org/10.1088/1748-0221/9/03/p03009.

[Aar+17a] M. G. Aartsen et al. "All-sky Search for Time-integrated Neutrino Emission from Astrophysical Sources with 7 yr of IceCube Data". In: *The Astrophysical Journal* 835.2 (Jan. 2017), p. 151. DOI: 10.3847/1538-4357/835/2/151. URL: https://doi.org/10.3847/1538-4357/835/2/151.

[Aar+17b] M.G. Aartsen et al. "The IceCube Neutrino Observatory: instrumentation and online systems". In: *Journal of Instrumentation* 12.03 (Mar. 2017), P03012–P03012. DOI: 10.1088/1748-0221/12/03/p03012. URL: https://doi.org/10.1088/1748-0221/12/03/p03012.

[Ach+06] A. Achterberg et al. "First year performance of the IceCube neutrino telescope". In: *Astroparticle Physics* 26.3 (Oct. 2006), pp. 155–173. DOI: 10.1016/j.astropartphys.2006.06.007. URL: https://doi.org/10.1016/j.astropartphys.2006.06.007.

[Ack+06] M. Ackermann et al. "Optical properties of deep glacial ice at the South Pole". In: *Journal of Geophysical Research* 111.D13 (2006). DOI: 10.1029/2005jd006687. URL: https://doi.org/10.1029/2005jd006687.

[And15] João Pedro Athayde Marcondes de André. *Idea to change the DOM angular acceptance (or hole ice model)*. https://docushare.icecube.wisc.edu/dsweb/Get/Document-75847/20151116_jpamdeandre_DC_holeice.pdf. 2015.

[And16] João Pedro Athayde Marcondes de André. *MSU Forward Hole Ice*. https://wiki.icecube.wisc.edu/index.php/MSU_Forward_Hole_Ice. 2016.

[Arg18] Carlos Argüelles. *Recent particle physics results from IceCube*. 2018.

[Ask+97] Per Askebjer et al. "Optical properties of deep ice at the South Pole: absorption". In: *Applied Optics* 36.18 (June 1997), p. 4168. DOI: 10.1364/ao.36.004168.

[Bro+06] Bronstein et al. *Taschenbuch der Mathematik*. 2006.

[Bur12] Carsten Burgard. *Standard model of physics*. http://www.texample.net/tikz/examples/model-physics/. 2012.

[Chi12] Dmitry Chirkin. *All-purpose flasher data set 2012*. 2012. URL: cobalt:/home/dima/DIMA/ice/dat/flasher/*.

[Chi13] Dmitry Chirkin. *Likelihood description for comparing data with simulation of limited statistics*. 2013. URL: https://arxiv.org/abs/1304.0735v3.

[Chi14] Dmitry Chirkin. *Photon Propagation with GPUs in IceCube*. http://dx.doi.org/10.3204/DESY-PROC-2014-05/40. 2014. DOI: 10.3204/DESY-PROC-2014-05/40.

[Chi16] Dmitry Chirkin. *IceCube ice-models angular sensitivity source files*. http://code.icecube.wisc.edu/projects/icecube/browser/IceCube/projects/ice-models/trunk/resources/models/angsens. 2016.

[Chi17] Dmitry Chirkin. *Flasher data-derived ice models, calibration call 2017-01-20*. https://docushare.icecube.wisc.edu/dsweb/Get/Document-79091/ice.pdf. 2017.

[Chi18] Dmitry Chirkin. *ppc Soure Code*. http://code.icecube.wisc.edu/projects/icecube/browser/IceCube/projects/ppc. 2018.

[Col+13] IceCube Collaboration et al. *Measurement of South Pole ice transparency with the IceCube LED calibration system*. https://arxiv.org/pdf/1301.5361. 2013. DOI: 10.1016/j.nima.2013.01.054. eprint: arXiv:1301.5361.

[Col17] IceCube Collaboration. *IceCube Simulation Framework Documentation*. http://software.icecube.wisc.edu/documentation/. 2017.

[Col18] IceCube Collaboration. *IceCube Simulation Framework Sourcecode*. http://code.icecube.wisc.edu/projects/icecube/browser/IceCube/meta-projects/simulation. 2018.

[CR04] Dmitry Chirkin and Wolfgang Rhode. "Propagating leptons through matter with Muon Monte Carlo (MMC)". In: (2004). URL: https://arxiv.org/abs/hep-ph/0407075.

[Don14] Chris Wyman Donald H. House. *Computer Graphics, Optimizing Raytracing Code, Tips for Optimizing C/C++ Code*. https://people.cs.clemson.edu/~dhouse/courses/405/papers/optimize.pdf. 2014.

[Ell17] Joshua P. Ellis. "Ti k Z-Feynman: Feynman diagrams with Ti k Z". In: *Computer Physics Communications* 210 (Jan. 2017), pp. 103–123. DOI: 10.1016/j.cpc.2016.08.019. URL: https://doi.org/10.1016/j.cpc.2016.08.019.

[Fin+11] C. Finley et al. *Freezing in the IceCube camera in string 80, 22 Dec - 1st Jan*. 2011.

[Glü16] Thorsten Glüsenkamp. *Search for a cumulative neutrino flux from 2LAC-blazar populations using 3 years of IceCube data*. en. 2016. DOI: 10.18452/17462.

[Glü18] Thorsten Glüsenkamp. "Probabilistic treatment of the uncertainty from the finite size of weighted Monte Carlo data". In: *The European Physical Journal Plus* 133.6 (June 2018). DOI: 10.1140/epjp/i2018-12042-x. URL: https://doi.org/10.1140/epjp/i2018-12042-x.

[GP16] Particle Data Group and C. Patrignani. "Review of Particle Physics". In: *Chinese Physics C* 40.10 (Oct. 2016), p. 100001. DOI: 10.1088/1674-1137/40/10/100001. URL: https://doi.org/10.1088/1674-1137/40/10/100001.

[Gri08] David J Griffiths. *Introduction to elementary particles; 2nd rev. version*. Physics textbook. New York, NY: Wiley, 2008. URL: https://cds.cern.ch/record/111880.

[HK10] Francis Halzen and Spencer R. Klein. "Invited Review Article: IceCube: An instrument for neutrino astronomy". In: *Review of Scientific Instruments* 81.8 (Aug. 2010), p. 081101. DOI: 10.1063/1.3480478. URL: https://doi.org/10.1063/1.3480478.

[Kar98] Albrecht Karle. *Hole Ice Studies with YAG*. http://icecube.berkeley.edu/kurt/interstring/hole-ice/yak.html. 1998.

[Kat18] U. Katz. *Future neutrino telescopes in water and ice*. 2018.

[Kop13] Claudio Kopper. *clsim README*. https://github.com/claudiok/clsim/blob/master/README.md. 2013.

[Kop17] Claudio Kopper. *clsim Sourcecode*. https://github.com/claudiok/clsim. 2017.

[KS12] U.F. Katz and Ch. Spiering. "High-energy neutrino astrophysics: Status and perspectives". In: *Progress in Particle and Nuclear Physics* 67.3 (July 2012), pp. 651–704. DOI: 10.1016/j.ppnp.2011.12.001. URL: https://doi.org/10.1016/j.ppnp.2011.12.001.

[Lan18] Justin Lanfranchi. *ppc for Humans*. http://code.icecube.wisc.edu/projects/icecube/browser/IceCube/projects/ppc/branches/ppc_for_humans/private/ppc. 2018.

[Leo94] W.R. Leo. *Techniques for Nuclear and Particle Physics Experiments: A How-to Approach*. Springer, 1994. ISBN: 9780387572802.

[Lex02] *Lexikon der Mathematik in sechs Bänden*. Spektrum Akademischer Verlag, Heidelberg Berlin, 2002.

[Lex98] *Lexikon der Physik in sechs Bänden*. Spektrum Akademischer Verlag, Heidelberg Berlin, 1998.

[Lun+07] J. Lundberg et al. "Light tracking through ice and water – Scattering and absorption in heterogeneous media with Photonics". In: (2007). DOI: 10.1016/j.nima.2007.07.143. eprint: arXiv:astro-ph/0702108.

[Mad13] Megan Madsen. *IceCube Gallery, DOM No Harness Whiteback*. https://gallery.icecube.wisc.edu/internal/v/GraphicRe/graphics/dom/DOMNoHarnessWhiteback_lg.jpg.html?g2_imageViewsIndex=1. 2013.

[Mal10] Evelyn Malkus. *IceCube Public Gallery, Digital Optical Module (DOM) Deployment*. https://icecube.wisc.edu/gallery/view/153. 2010.

[Mie08] Gustav Mie. "Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen". In: *Annalen der Physik* 330.3 (1908), pp. 377–445. DOI: 10.1002/andp.19083300302. URL: https://doi.org/10.1002/andp.19083300302.

[OW01] Philip Olbrechts and Christopher Wiebusch. *On the Angular Sensitivity of Optical Modules in the Ice*. https://web.physik.rwth-aachen.de/~wiebusch/Publications/Amanda/updn.ps.gz. 2001.

[Owe13] John Owens. *Introduction to Parallel Programming*. https://youtube.com/playlist?list=PLGvfHSgImk4aweyWlhBXNF6XISY3um82_. 2013.

[PB97] P. Buford Price and Lars Bergström. "Optical properties of deep ice at the South Pole: scattering". In: *Applied Optics* 36.18 (June 1997), p. 4181. DOI: 10.1364/ao.36.004181. URL: https://doi.org/10.1364/ao.36.004181.

[Ron15] Martin Rongen. *DARD Update, Calibration Call 2015-11-06*. 2015.

[Ron16] Martin Rongen. "Measuring the optical properties of IceCube drill holes". In: *EPJ Web of Conferences* 116 (2016). Ed. by A. Capone et al., p. 06011. DOI: 10.1051/epjconf/201611606011. URL: https://doi.org/10.1051/epjconf/201611606011.

[Ron17a] Martin Rongen. *Calibrating the hole ice with the POCAM*. 2017.

[Ron17b] Martin Rongen. *Status and future of SpiceHD and DARD, Calibration Workshop August 2017*. 2017.

[Ron18] Martin Rongen. *The 2018 Sweden Camera run — light at the end of the ice*. 2018.

[RRK17] Elisa Resconi, Martin Rongen, and Kai Krings. "The Precision Optical CAlibration Module for IceCube-Gen2: First Prototype". In: *Proceedings of 35th International Cosmic Ray Conference — PoS(ICRC2017)*. Sissa Medialab, Aug. 2017. DOI: 10.22323/1.301.0934. URL: https://doi.org/10.22323/1.301.0934.

[Van+17] Sander Vanheule et al. *Stemshovel Documentation*. http://software.icecube.wisc.edu/documentation/projects/steamshovel/basics.html. 2017.

[Wos03] Kurt Woschnagg. *Review of Ice Models*. http://icecube.berkeley.edu/kurt/talks/IceModels_Madison_June_03.pdf. 2003.

[Wos04] Kurt Woschnagg. *Photonics Web Portal*. http://icecube.berkeley.edu/kurt/photonics/. 2004.

[Wos06] Kurt Woschnagg. *Optical Properties of Ice in AMANDA and IceCube*. http://icecube.berkeley.edu/kurt/ice2000/. 2006.

[Wos07] Kurt Woschnagg. *Effective scattering length*. https://wiki.icecube.wisc.edu/index.php/Effective_scattering_length. 2007.

[Wos11] Kurt Woschnagg. *IceCube-86 (78+8) interstrng (surface) distances*. https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg. 2011.

[Wos98] Kurt Woschnagg. *The Interesting Hole-Ice Properties Page*. http://icecube.berkeley.edu/kurt/interstring/hole-ice/. 1998.

[Wre18] Gerrit Wrede. *Effect of the ice model and propagator/spline difference on ideal muon reconstruction*. 2018.

[Yan09] Jamie Yang. *IceCube Gallery, Array Graphics, Sketch Up, DOM Close Up*. 2009.

A.2 Contents of the CD-ROM

`algorithm_a`	Source code of the hole-ice-correction algorithm (section 5.3 and appendix A.9)
`algorithm_b`	Source code of the new medium-propagation algorithm with hole ice (section 5.4 and appendix A.10)
`clsim`	Source code of standard CLSIM in version `V05-00-07` of the ICECUBE simulation framework
`hole-ice-study`	scripts and data files used in this study, corresponding to the repository https://github.com/fiedl/hole-ice-study
`issues`	Documentation of issues corresponding to https://github.com/fiedl/hole-ice-study/issues
`latex`	LaTeX version of this document, corresponding to the repository https://github.com/fiedl/hole-ice-latex
`papers`	Papers from the bibliography, except for ICECUBE-internal material
`pdf`	This document in portable document format (PDF)
`unit_tests`	Unit tests for the hole-ice-correction algorithm (section 5.6.1)
`video`	Animated visualization of the photon-propagation through a hole-ice cylinder (section 6.1), corresponding to https://youtu.be/BhJ6F3B-I1s

A.3 List of Abbreviations

CLSIM	OpenCL-based photon-tracking simulation using a (source-based) ray tracing algorithm modeling scattering and absorption of light in the deep glacial ice at the South Pole or Mediterranean sea water. See [Kop13; Kop17].
DOM	Digital Optical Module, primary instrumentation and detection unit in ICECUBE, see section 3.2.
H0	Hypothesis stating that no hole ice exists in the ICECUBE detector. See [Kar98].
H1	Hypothesis stating that hole ice exists in the ICECUBE detector with a hole-ice radius of 30 cm, filling the entire drill hole, and a geometric hole-ice scattering length of 100 cm. See [Kar98].
H2	Hypothesis stating that hole ice exists in the ICECUBE detector with a hole-ice radius of 30 cm, filling the entire drill hole, and a geometric hole-ice scattering length of 50 cm. See [Kar98].
H3	Hypothesis stating that hole ice exists in the ICECUBE detector with a hole-ice radius of 30 cm, filling the entire drill hole, and a geometric hole-ice scattering length of 30 cm. See [Kar98].
H4	Hypothesis stating that hole ice exists in the ICECUBE detector with a hole-ice radius of 30 cm, filling the entire drill hole, and a geometric hole-ice scattering length of 10 cm. See [Kar98].
LED	Light-emitting diode, part of ICECUBE's flasher calibration system, see section 6.7.
PPC	photon propagation code. A photon-propagation simulation software. See [Chi14; Chi18; Lan18].
PMT	Photo Multiplier Tube, instrument to detect light, component of ICECUBE's optical modules, see section 3.2.
YAG	Yttrium-Aluminium-Garnet, synthetic material, used as laser medium in ice studies, see section 7.2.1.

A.4 List of Quantities

λ_{sca}	geometric scattering length, in general or within the bulk ice, see section 2.4.
$\lambda_{\text{sca}}^{\text{H}}$	geometric scattering length within the hole ice.
λ_{e}	effective scattering length, in general or within the bulk ice, $\lambda_{\text{e}} = \frac{\lambda_{\text{sca}}}{1-\langle\cos\theta\rangle}$, see section 2.4.
$\lambda_{\text{e}}^{\text{H}}$	effective scattering length within the hole ice, $\lambda_{\text{e}}^{\text{H}} = \frac{\lambda_{\text{sca}}^{\text{H}}}{1-\langle\cos\theta\rangle}$.
λ_{abs}	absorption length, in general or within the bulk ice, see section 2.4.
$\lambda_{\text{abs}}^{\text{H}}$	absorption length within the hole ice.
r	radius of the hole-ice cylinder, see section 3.4.
r_{DOM}	radius of IceCube's digital optical modules (DOMs), see section 3.2.

A.5 List of Units

Units of Length

m	meter, base unit of length in the international system of units
cm	centimeter, 10^{-2} m
mm	millimeter, 10^{-3} m
nm	nanometer, 10^{-9} m

Units of Time

s	second, base unit of time in the international system of units
ns	nanosecond, 10^{-9} s

Units of Energy

eV	electron volt, unit of energy, $1\,\text{eV} = 1.6021766208 \cdot 10^{-19}$ joules, which is the basic unit of energy in the international system of units
MeV	mega electron volt, $10^6\,\text{eV}$
GeV	giga electron volt, $10^9\,\text{eV}$
TeV	tera electron volt, $10^{12}\,\text{eV}$
PeV	peta electron volt, $10^{15}\,\text{eV}$

A.6 List of Mathematical Symbols

:	property operator. $A : B$ means that object A has the property B. The associativity is focused on A, such that "The radius of the cylinder is $r : B$" means "The radius of the cylinder is r and r has the property B."
$=$	equality. $A = B$ means that the quantities A and B have the same value.
$:=$	short cut for property operator in conjunction with the equality operator. $A := B$ means $A : A = B$.
$\approx$	approximate equality. $A \approx B$ means that the quantities A and B have approximately the same value, where the precision of the approximation is given by the context of the statement.
$\equiv$	definition equality operator. $A \equiv B$ means that the symbols A and B represent the same object.
$\sum$	sum. $\sum_{i=1}^{n} a_i$ means $a_1 + a_2 + \cdots + a_n$
$\prod$	product. $\prod_{i=1}^{n} a_i$ means $a_1 \cdot a_2 \cdot \cdots \cdot a_n$
π	circle constant, representing the ratio of a circle's circumference to its diameter, $\pi \approx 3.14159$
$\{\}$	set. $\{A, B\}$ represents the set of the objects A and B.
$\mathbb{N}$	set of all natural numbers
$\mathbb{R}$	set of all real numbers
$\mathbb{R}^+$	set of all positive real numbers
$\mathbb{R}_0^+$	set of all positive real numbers and zero
$\|A\rangle$	quantum mechanical bra-ket notation for the quantum state of A as abstract column vector

A.7 Additional Flow Charts

A.7.1 Standard CLSIM's Media Propagation Algorithm

Figure 72 shows a flow chart of standard CLSIM's media-propagation algorithm (section 5.2.2) that does not support cylinder-shaped media.

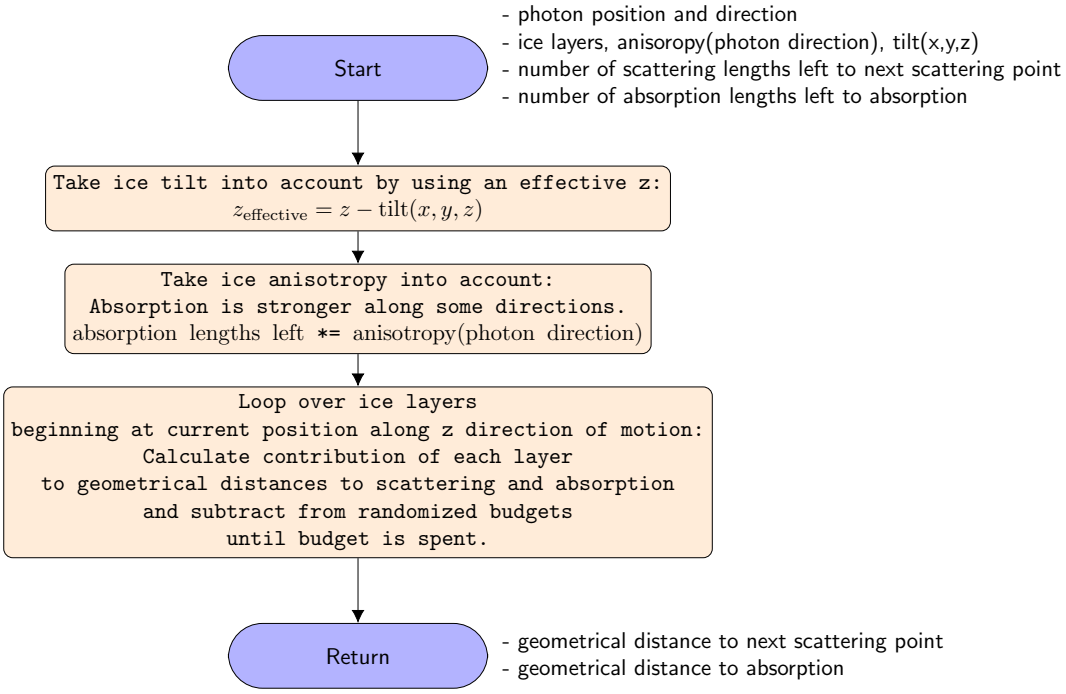


Figure 72: Flow chart of standard-CLSIM's medium-propagation algorithm. Tilted ice layers and ice anisotropy are hard-coded. Media with other geometries such as hole-ice cylinders are not supported. This medium-propagation algorithm is replaced by the new algorithm described in section 5.4.

A.7.2 New Media-Propagation Algorithm

A flow chart of the new medium-propagation algorithm introduced in section 5.4 that replaces standard-CLSIM's media-propagation algorithm is shown in figure 73.

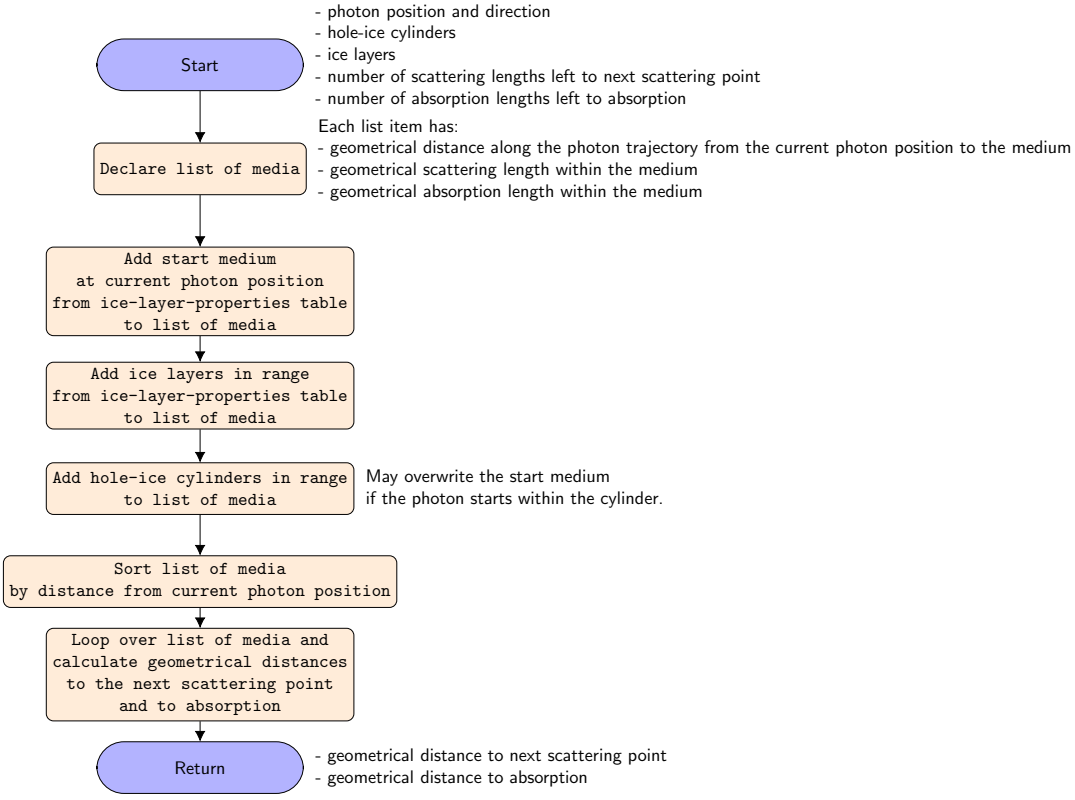


Figure 73: Flow chart of the new medium-propagation algorithm. The algorithm determines all medium changes (layers, hole ice) between two interaction points. Then the algorithm loops over all medium changes with different ice properties and calculates the geometrical distances to the next interaction points.

A.8 How to Install the Modified CLSIM

This study needs a modified version of CLSIM that does support hole-ice simulations. Until this modified version has been merged into the main ICECUBE Simulation Framework source code, these patched version needs to be installed manually.

```
# Get clsim fork
git clone git@github.com:fiedl/clsim.git $SCRATCH/clsim
cd $SCRATCH/clsim

# Symlink it into the icesim source
cd $ICESIM_ROOT/src
rm -rf clsim
ln -s $SCRATCH/clsim clsim

# Compile it
cd $ICESIM_ROOT/build
make -j 6
```

Detailed instructions on how to install the software on the Zeuthen computing center can be found at https://github.com/fiedl/hole-ice-study/blob/master/notes/2018-01-23_Installing_IceSim_in_Zeuthen.md#install-patched-clsim.

Instructions on how to install the software locally on macOS can be found at https://github.com/fiedl/hole-ice-study/blob/master/notes/2016-11-15_Installing_IceSim_on_macOS_Sierra.md.

A.9 Source Code of Algorithm A

This is the source code of the first propagation algorithm through different media, described in section 5.3.

The source code can also be found online at
https://github.com/fiedl/clsim/tree/sf/hole-ice-2017/resources/kernels/lib.

```
1  // https://github.com/fiedl/clsim/blob/sf/hole-ice-2017/resources/kernels/lib/
   ↪  hole_ice/hole_ice.c
2
3  #ifndef HOLE_ICE_C
4  #define HOLE_ICE_C
5
6  #include "hole_ice.h"
7  #include "../intersection/intersection.c"
8
9  inline bool is_between_zero_and_one(floating_t a) {
10   return ((!my_is_nan(a)) && (a > 0.0) && (a < 1.0));
11 }
12
13 inline bool not_between_zero_and_one(floating_t a) {
14   return (! is_between_zero_and_one(a));
15 }
16
17 inline unsigned int number_of_medium_changes(HoleIceProblemParameters_t p)
18 {
19   // These are ordered by their frequency of occurrence in order
20   // to optimize for performance.
21   if (not_between_zero_and_one(p.entry_point_ratio) &&
     ↪  not_between_zero_and_one(p.termination_point_ratio)) return 0;
22   if (not_between_zero_and_one(p.entry_point_ratio) ||
     ↪  not_between_zero_and_one(p.termination_point_ratio)) return 1;
23   if (p.entry_point_ratio == p.termination_point_ratio) return 0; // tangent
24   return 2;
25 }
26
27 inline floating_t hole_ice_distance_correction(HoleIceProblemParameters_t p)
28 {
29   // Depending on the fraction of the distance the photon is traveling
30   // within the hole ice, there are six cases to consider.
31   //
32   // N  denotes the number of intersections.
33   // H  means that the trajectory starts within hole ice.
34   // !H means that the trajectory starts outside of the hole ice.
35   //
```

```
36    // Case 1: The trajectory is completely outside of the hole ice (!H, N=0).
37    // Case 2: The trajectory is completely within the hole ice (H, N=0).
38    // Case 3: The trajectory begins outside, but ends inside the hole ice (!H,
      ↪  N=1).
39    // Case 4: The trajectory begins inside, but ends outside the hole ice (H,
      ↪  N=1).
40    // Case 5: The trajectory starts and ends outside, but passes through the
      ↪  hole ice (!H, N=2).
41    // Case 6: The trajectory begins within one hole-ice cylinder, passes through
42    //           normal ice and ends in another hole-ice cylinder (H, N=2).
43    //
44    // For further information, please have look at:
45    // https://github.com/fiedl/clsim/tree/sf/master/resources/kernels/lib/hole_
      ↪  ice
46
47    const unsigned int num_of_medium_changes = number_of_medium_changes(p);
48
49    // Case 1: The trajectory is completely outside of the hole ice.
50    // Thus, needs no correction.
51    if ((num_of_medium_changes == 0) && !p.starts_within_hole_ice) {
52      return 0;
53    }
54
55    const floating_t ac = p.distance * p.termination_point_ratio;
56
57    if (p.starts_within_hole_ice) {
58
59      // Case 4: The trajectory begins inside, but ends outside the hole ice.
60      if (p.interaction_length_factor * p.distance > ac) {
61        return (1.0 - 1.0 / p.interaction_length_factor) * ac;
62
63      // Case 2: The trajectory is completely within the hole ice.
64      } else {
65        return (p.interaction_length_factor - 1.0) * p.distance;
66      }
67
68    } else {
69
70      const floating_t yb = p.distance * (1.0 - p.entry_point_ratio);
71      floating_t yc = p.distance * (p.termination_point_ratio -
      ↪  p.entry_point_ratio);
72
73      if (yc < 0) {
74        printf("WARNING: YC SHOULD NOT BE NEGATIVE, BUT YC=%f\n", yc);
75        yc = 0;
76      }
77
78      // Case 5: The trajectory starts and ends outside, but passes through the
      ↪  hole ice.
```

```
79      if (p.interaction_length_factor * yb > yc) {
80        return (1.0 - 1.0 / p.interaction_length_factor) * yc;
81
82      // Case 3
83      } else {
84        return (p.interaction_length_factor - 1.0) * p.distance * (1.0 -
          ↪  p.entry_point_ratio);
85      }
86    }
87
88  #ifdef PRINTF_ENABLED
89    printf("WARNING: UNHANDLED INTERSECTION CASE. This point should not be
      ↪  reached.");
90  #endif
91
92    return my_nan();
93  }
94
95  inline floating_t
    ↪  hole_ice_distance_correction_for_intersection_problem(floating_t distance,
    ↪  floating_t interaction_length_factor, IntersectionProblemParameters_t p)
96  {
97    calculate_intersections(&p);
98    HoleIceProblemParameters_t hip = {
99      distance,
100     interaction_length_factor,
101     intersection_s1(p), // entry_point_ratio
102     intersection_s2(p), // termination_point_ratio
103     intersecting_trajectory_starts_inside(p) // starts_within_hole_ice
104   };
105   return hole_ice_distance_correction(hip);
106 }
107
108 #ifdef HOLE_ICE_TEST_H
109 inline floating_t apply_hole_ice_correction(floating4_t photonPosAndTime,
    ↪  floating4_t photonDirAndWlen, unsigned int numberOfCylinders, floating4_t
    ↪  *cylinderPositionsAndRadii, floating_t holeIceScatteringLengthFactor,
    ↪  floating_t holeIceAbsorptionLengthFactor, floating_t *distancePropagated,
    ↪  floating_t *distanceToAbsorption)
110 #endif
111 #ifndef HOLE_ICE_TEST_H
112 inline floating_t apply_hole_ice_correction(floating4_t photonPosAndTime,
    ↪  floating4_t photonDirAndWlen, unsigned int numberOfCylinders, __constant
    ↪  floating4_t *cylinderPositionsAndRadii, floating_t
    ↪  holeIceScatteringLengthFactor, floating_t holeIceAbsorptionLengthFactor,
    ↪  floating_t *distancePropagated, floating_t *distanceToAbsorption)
113 #endif
114 {
115
```

```
116   if (!(my_is_nan(photonPosAndTime.x) || my_is_nan(*distancePropagated))) {
117
118     // Find out which cylinders are in range in a separate loop
119     // in order to improve parallelism and thereby performance.
120     //
121     // See: https://github.com/fiedl/hole-ice-study/issues/30
122     //
123     #ifdef NUMBER_OF_CYLINDERS
124       // When running this on OpenCL, defining arrays using a constant
125       // as array size is not possible. Therefore, we need to use a
126       // pre-processor makro here.
127       //
128       // See: https://github.com/fiedl/hole-ice-study/issues/38
129       //
130       int indices_of_cylinders_in_range[NUMBER_OF_CYLINDERS];
131     #else
132       int indices_of_cylinders_in_range[numberOfCylinders];
133     #endif
134     {
135       unsigned int j = 0;
136       for (unsigned int i = 0; i < numberOfCylinders; i++) {
137         indices_of_cylinders_in_range[i] = -1;
138       }
139       for (unsigned int i = 0; i < numberOfCylinders; i++) {
140         if (sqr(photonPosAndTime.x - cylinderPositionsAndRadii[i].x) +
141             sqr(photonPosAndTime.y - cylinderPositionsAndRadii[i].y) <=
142             sqr(*distancePropagated + cylinderPositionsAndRadii[i].w /* radius
                ↪  */))
143         {
144
145           // If the cylinder has a z-range check if we consider that cylinder
146           // to be in range. https://github.com/fiedl/hole-ice-study/issues/34
147           //
148           if ((cylinderPositionsAndRadii[i].z == 0) ||
              ↪  ((cylinderPositionsAndRadii[i].z != 0) && !((photonPosAndTime.z
              ↪  < cylinderPositionsAndRadii[i].z - 0.5) && (photonPosAndTime.z +
              ↪  *distancePropagated * photonDirAndWlen.z <
              ↪  cylinderPositionsAndRadii[i].z - 0.5)) || ((photonPosAndTime.z >
              ↪  cylinderPositionsAndRadii[i].z + 0.5) && (photonPosAndTime.z +
              ↪  *distancePropagated * photonDirAndWlen.z >
              ↪  cylinderPositionsAndRadii[i].z + 0.5)))))
149           {
150             indices_of_cylinders_in_range[j] = i;
151             j += 1;
152           }
153         }
154       }
155     }
156
```

```
157       // Now loop over all cylinders in range and calculate corrections
158       // for `distancePropagated` and `distanceToAbsorption`.
159       //
160       for (unsigned int j = 0; j < numberOfCylinders; j++) {
161         const int i = indices_of_cylinders_in_range[j];
162         if (i == -1) {
163           break;
164         } else {
165
166           IntersectionProblemParameters_t p = {
167
168             // Input values
169             photonPosAndTime.x,
170             photonPosAndTime.y,
171             cylinderPositionsAndRadii[i].x,
172             cylinderPositionsAndRadii[i].y,
173             cylinderPositionsAndRadii[i].w, // radius
174             photonDirAndWlen,
175             *distancePropagated,
176
177             // Output values (will be calculated)
178             0, // discriminant
179             0, // s1
180             0  // s2
181
182           };
183
184           calculate_intersections(&p);
185
186           // Are intersection points possible?
187           if (intersection_discriminant(p) > 0) {
188
189             const floating_t scatteringEntryPointRatio = intersection_s1(p);
190             const floating_t scatterintTerminationPointRatio =
                ↪  intersection_s2(p);
191
192             HoleIceProblemParameters_t scatteringCorrectionParameters = {
193               *distancePropagated,
194               holeIceScatteringLengthFactor,
195               scatteringEntryPointRatio,
196               scatterintTerminationPointRatio,
197               intersecting_trajectory_starts_inside(p)
198             };
199
200             const floating_t scaCorrection =
                ↪  hole_ice_distance_correction(scatteringCorrectionParameters);
201             *distancePropagated += scaCorrection;
202
```

```
203             // For the absorption, there are special cases where the photon is
                ↪  scattered before
204             // reaching either the first or the second absorption intersection
                ↪  point.
205             floating_t absorptionEntryPointRatio;
206             floating_t absorptionTerminationPointRatio;
207             floating_t absCorrection = 0.0;
208             if (!(not_between_zero_and_one(scatteringCorrectionParameters.entry_
                ↪  point_ratio) &&
                ↪  !scatteringCorrectionParameters.starts_within_hole_ice)) {
209               // The photon reaches the hole ice, i.e. the absorption correction
210               // needs to be calculated.
211               p.distance = *distanceToAbsorption;
212               calculate_intersections(&p);
213
214               // If the photon is scattered away before reaching the far and of
215               // the hole ice, the affected trajectory is limited by the
216               // point where the photon is scattered away.
217               absorptionEntryPointRatio = intersection_s1(p);
218               absorptionTerminationPointRatio = min(
219                 *distancePropagated / *distanceToAbsorption,
220                 intersection_s2(p)
221               );
222
223               HoleIceProblemParameters_t absorptionCorrectionParameters = {
224                 *distanceToAbsorption,
225                 holeIceAbsorptionLengthFactor,
226                 absorptionEntryPointRatio,
227                 absorptionTerminationPointRatio,
228                 intersecting_trajectory_starts_inside(p),
229               };
230
231               absCorrection =
                  ↪  hole_ice_distance_correction(absorptionCorrectionParameters);
232
233             }
234             *distanceToAbsorption += absCorrection;
235
236           }
237
238         }
239       }
240     }
241
242   }
243
244   #endif
```

```c
// https://github.com/fiedl/clsim/blob/sf/hole-ice-2017/resources/kernels/lib/hole_ice/hole_ice.h

#ifndef HOLE_ICE_H
#define HOLE_ICE_H

typedef struct HoleIceProblemParameters {
  floating_t distance;
  floating_t interaction_length_factor;
  floating_t entry_point_ratio;
  floating_t termination_point_ratio;
  bool starts_within_hole_ice;
} HoleIceProblemParameters_t;

#endif
```

```c
// https://github.com/fiedl/clsim/blob/sf/hole-ice-2017/resources/kernels/lib/intersection/intersection.c

#include "intersection.h"

inline void calculate_intersections(IntersectionProblemParameters_t *p)
{
  // Step 1
  const floating4_t vector_AM = {p->mx - p->ax, p->my - p->ay, 0.0, 0.0};
  const floating_t xy_projection_factor = my_sqrt(1 - sqr(p->direction.z));
  const floating_t length_AMprime = dot(vector_AM, p->direction) / xy_projection_factor;

  // Step 2
  p->discriminant = sqr(p->r) - dot(vector_AM, vector_AM) + sqr(length_AMprime);

  // Step 3
  const floating_t length_XMprime = my_sqrt(p->discriminant);

  // Step 4
  const floating_t length_AX1 = length_AMprime - length_XMprime;
  const floating_t length_AX2 = length_AMprime + length_XMprime;
  p->s1 = length_AX1 / p->distance / xy_projection_factor;
  p->s2 = length_AX2 / p->distance / xy_projection_factor;
}

inline floating_t intersection_s1(IntersectionProblemParameters_t p)
{
  return p.s1;
```

```
28  }
29
30  inline floating_t intersection_s2(IntersectionProblemParameters_t p)
31  {
32    return p.s2;
33  }
34
35  inline floating_t intersection_discriminant(IntersectionProblemParameters_t p)
36  {
37    return p.discriminant;
38  }
39
40  inline floating_t intersection_x1(IntersectionProblemParameters_t p)
41  {
42    if ((p.s1 > 0) && (p.s1 < 1))
43      return p.ax + p.direction.x * p.distance * p.s1;
44    else
45      return my_nan();
46  }
47
48  inline floating_t intersection_x2(IntersectionProblemParameters_t p)
49  {
50    if ((p.s2 > 0) && (p.s2 < 1))
51      return p.ax + p.direction.x * p.distance * p.s2;
52    else
53      return my_nan();
54  }
55
56  inline floating_t intersection_y1(IntersectionProblemParameters_t p)
57  {
58    if ((p.s1 > 0) && (p.s1 < 1))
59      return p.ay + p.direction.y * p.distance * p.s1;
60    else
61      return my_nan();
62  }
63
64  inline floating_t intersection_y2(IntersectionProblemParameters_t p)
65  {
66    if ((p.s2 > 0) && (p.s2 < 1))
67      return p.ay + p.direction.y * p.distance * p.s2;
68    else
69      return my_nan();
70  }
71
72  inline bool
    ↪  intersecting_trajectory_starts_inside(IntersectionProblemParameters_t p)
73  {
74    return (intersection_s1(p) <= 0) &&
75        (intersection_s2(p) > 0) &&
```

```
76        (intersection_discriminant(p) > 0);
77  }
78
79  inline bool
    ↪  intersecting_trajectory_starts_outside(IntersectionProblemParameters_t p)
80  {
81    return ( ! intersecting_trajectory_starts_inside(p));
82  }
83
84  inline bool intersecting_trajectory_ends_inside(IntersectionProblemParameters_t
    ↪  p)
85  {
86    return (intersection_s1(p) < 1) &&
87        (intersection_s2(p) >= 1) &&
88        (intersection_discriminant(p) > 0);
89  }
```

```
1   // https://github.com/fiedl/clsim/blob/sf/hole-ice-2017/resources/kernels/lib/
    ↪  intersection/intersection.h
2
3   #ifndef INTERSECTION_H
4   #define INTERSECTION_H
5
6   typedef struct IntersectionProblemParameters {
7
8     // Input values
9     //
10    floating_t ax;
11    floating_t ay;
12    floating_t mx;
13    floating_t my;
14    floating_t r;
15    floating4_t direction;
16    floating_t distance;
17
18    // Output values, which will be calculated in
19    // `calculate_intersections()`.
20    //
21    floating_t discriminant;
22    floating_t s1;
23    floating_t s2;
24
25  } IntersectionProblemParameters_t;
26
27  #endif
```

A.10 Source Code of Algorithm B

This is the source code of the second propagation algorithm through different media, described in section 5.4.

The source code can also be found online at
https://github.com/fiedl/clsim/tree/sf/hole-ice-2018/resources/kernels/lib.

```
1  // https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/propagation_through_media/propagation_through_media.c
2
3  #ifndef PROPAGATION_THROUGH_MEDIA_C
4  #define PROPAGATION_THROUGH_MEDIA_C
5
6  #include "propagation_through_media.h"
7  #include "../ice_layers/ice_layers.c"
8  #ifdef HOLE_ICE
9    #include "../hole_ice/hole_ice.c"
10 #endif
11
12
13 // PROPAGATION THROUGH DIFFERENT MEDIA 2018: Layers, Cylinders
14 //
   ↪ ------------------------------------------------------------------------
15
16 // We know how many scattering lengths (`sca_step_left`) and
17 // absorption lengths (`abs_lens_left`) the photon will
18 // travel in this step.
19 //
20 // Because the mean scattering and absorption lengths are local
21 // properties, i.e. depend on the ice layer or whether the photon
22 // is within a hole-ice cylinder, we need to convert `sca_step_left`
23 // and `abs_lens_left` to geometrical distances in order to determine
24 // where the next interaction point is, i.e. how far to propagate
25 // the photon in this step.
26
27 inline void apply_propagation_through_different_media(
28   floating4_t photonPosAndTime, floating4_t photonDirAndWlen,
29   #ifdef HOLE_ICE
30     unsigned int numberOfCylinders, __constant floating4_t *cylinderPositionsAndRadii,
31     __constant floating_t *cylinderScatteringLengths, __constant floating_t *cylinderAbsorptionLengths,
32   #endif
33   floating_t *distances_to_medium_changes, floating_t *local_scattering_lengths, floating_t *local_absorption_lengths,
```

```
34    floating_t *sca_step_left, floating_t *abs_lens_left,
35    floating_t *distancePropagated, floating_t *distanceToAbsorption)
36  {
37
38    int number_of_medium_changes = 0;
39    distances_to_medium_changes[0] = 0.0;
40    int currentPhotonLayer = min(max(findLayerForGivenZPos(photonPosAndTime.z),
      ↪  0), MEDIUM_LAYERS-1);
41    local_scattering_lengths[0] = getScatteringLength(currentPhotonLayer,
      ↪  photonDirAndWlen.w);
42    local_absorption_lengths[0] = getAbsorptionLength(currentPhotonLayer,
      ↪  photonDirAndWlen.w);
43
44    // To check which medium boundaries are in range, we need to estimate
45    // how far the photon can travel in this step.
46    //
47    const floating_t photonRange = *sca_step_left * local_scattering_lengths[0];
48
49    add_ice_layers_on_photon_path_to_medium_changes(
50      photonPosAndTime,
51      photonDirAndWlen,
52      photonRange,
53
54      // These values will be updates within this function:
55      &number_of_medium_changes,
56      distances_to_medium_changes,
57      local_scattering_lengths,
58      local_absorption_lengths
59    );
60
61    #ifdef HOLE_ICE
62      add_hole_ice_cylinders_on_photon_path_to_medium_changes(
63        photonPosAndTime,
64        photonDirAndWlen,
65        photonRange,
66        numberOfCylinders,
67        cylinderPositionsAndRadii,
68
69        // These values will be updates within this function:
70        &number_of_medium_changes,
71        distances_to_medium_changes,
72        local_scattering_lengths,
73        local_absorption_lengths
74      );
75    #endif
76
77    sort_medium_changes_by_ascending_distance(
78      number_of_medium_changes,
79
```

```
80      // These values will be updates within this function:
81      distances_to_medium_changes,
82      local_scattering_lengths,
83      local_absorption_lengths
84    );
85
86    loop_over_media_and_calculate_geometrical_distances_up_to_the_next_scattering_point(
87      number_of_medium_changes,
88      distances_to_medium_changes,
89      local_scattering_lengths,
90      local_absorption_lengths,
91
92      // These values will be updates within this function:
93      sca_step_left,
94      abs_lens_left,
95      distancePropagated,
96      distanceToAbsorption
97    );
98  }
99
100 inline void sort_medium_changes_by_ascending_distance(int number_of_medium_changes, floating_t *distances_to_medium_changes,
    floating_t *local_scattering_lengths, floating_t *local_absorption_lengths)
101 {
102   // Sort the arrays `distances_to_medium_changes`, `local_scattering_lengths` and
103   // `local_absorption_lengths` by ascending distance to have the medium changes
104   // in the right order.
105   //
106   // https://en.wikiversity.org/wiki/C_Source_Code/Sorting_array_in_ascending_and_descending_order
107   //
108   for (int k = 0; k <= number_of_medium_changes; k++) {
109     for (int l = 0; l <= number_of_medium_changes; l++) {
110       if (distances_to_medium_changes[l] > distances_to_medium_changes[k]) {
111         floating_t tmp_distance = distances_to_medium_changes[k];
112         floating_t tmp_scattering = local_scattering_lengths[k];
113         floating_t tmp_absorption = local_absorption_lengths[k];
114
115         distances_to_medium_changes[k] = distances_to_medium_changes[l];
116         local_scattering_lengths[k] = local_scattering_lengths[l];
117         local_absorption_lengths[k] = local_absorption_lengths[l];
118
119         distances_to_medium_changes[l] = tmp_distance;
120         local_scattering_lengths[l] = tmp_scattering;
121         local_absorption_lengths[l] = tmp_absorption;
122       }
```

```
123       }
124     }
125   }
126
127   inline void loop_over_media_and_calculate_geometrical_distances_up_to_the_
      ↪  next_scattering_point(int number_of_medium_changes, floating_t
      ↪  *distances_to_medium_changes, floating_t *local_scattering_lengths,
      ↪  floating_t *local_absorption_lengths, floating_t *sca_step_left, floating_t
      ↪  *abs_lens_left, floating_t *distancePropagated, floating_t
      ↪  *distanceToAbsorption)
128   {
129     // We know how many scattering lengths (`sca_step_left`) and how many
130     // absorption lengths (`abs_lens_left`) we may spend when propagating
131     // through the different media.
132     //
133     // Convert these into the geometrical distances `distancePropagated`
      ↪  (scattering)
134     // and `distanceToAbsorption` (absorption) and decrease `sca_step_left` and
135     // `abs_lens_left` accordingly.
136     //
137     // Abort when the next scattering point is reached, i.e. `sca_step_left ==
      ↪  0`.
138     // At this point, `abs_lens_left` may still be greater than zero, because
139     // the photon may be scattered several times until it is absorbed.
140     //
141     for (int j = 0; (j < number_of_medium_changes) && (*sca_step_left > 0); j++)
      ↪  {
142       floating_t max_distance_in_current_medium =
        ↪  distances_to_medium_changes[j+1] - distances_to_medium_changes[j];
143
144       if (*sca_step_left * local_scattering_lengths[j] >
        ↪  max_distance_in_current_medium) {
145         // The photon scatters after leaving this medium.
146         *sca_step_left -= my_divide(max_distance_in_current_medium,
          ↪  local_scattering_lengths[j]);
147         *distancePropagated += max_distance_in_current_medium;
148       } else {
149         // The photon scatters within this medium.
150         max_distance_in_current_medium = *sca_step_left *
          ↪  local_scattering_lengths[j];
151         *distancePropagated += max_distance_in_current_medium;
152         *sca_step_left = 0;
153       }
154
155       if (*abs_lens_left * local_absorption_lengths[j] >
        ↪  max_distance_in_current_medium) {
156         // The photon is absorbed after leaving this medium.
157         *abs_lens_left -= my_divide(max_distance_in_current_medium,
          ↪  local_absorption_lengths[j]);
```

```
158         *distanceToAbsorption += max_distance_in_current_medium;
159       } else {
160         // The photon is absorbed within this medium.
161         *distanceToAbsorption += *abs_lens_left * local_absorption_lengths[j];
162         *abs_lens_left = 0;
163       }
164     }
165
166     // Spend the rest of the budget with the last medium properties.
167     if (*sca_step_left > 0) {
168       *distancePropagated += *sca_step_left *
        ↪  local_scattering_lengths[number_of_medium_changes];
169       *distanceToAbsorption += *abs_lens_left *
        ↪  local_absorption_lengths[number_of_medium_changes];
170       *abs_lens_left -= my_divide(*distancePropagated,
        ↪  local_absorption_lengths[number_of_medium_changes]);
171     }
172
173     // If the photon is absorbed, only propagate up to the absorption point.
174     if (*distanceToAbsorption < *distancePropagated) {
175       *distancePropagated = *distanceToAbsorption;
176       *distanceToAbsorption = ZERO;
177       *abs_lens_left = ZERO;
178     }
179 }
180
181 #endif
```

```
1  // https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/
   ↪  propagation_through_media/propagation_through_media.h
2
3  #ifndef PROPAGATION_THROUGH_MEDIA_H
4  #define PROPAGATION_THROUGH_MEDIA_H
5
6  inline void apply_propagation_through_different_media(
7    floating4_t photonPosAndTime, floating4_t photonDirAndWlen,
8    #ifdef HOLE_ICE
9      unsigned int numberOfCylinders, __constant floating4_t
       ↪  *cylinderPositionsAndRadii,
10     __constant floating_t *cylinderScatteringLengths, __constant floating_t
       ↪  *cylinderAbsorptionLengths,
11   #endif
12   floating_t *distances_to_medium_changes, floating_t *local_scattering_lengths,
     ↪  floating_t *local_absorption_lengths,
13   floating_t *sca_step_left, floating_t *abs_lens_left,
14   floating_t *distancePropagated, floating_t *distanceToAbsorption);
15
```

```
inline void sort_medium_changes_by_ascending_distance(int number_of_medium_changes, floating_t *distances_to_medium_changes, floating_t *local_scattering_lengths, floating_t *local_absorption_lengths);

inline void loop_over_media_and_calculate_geometrical_distances_up_to_the_next_scattering_point(int number_of_medium_changes, floating_t *distances_to_medium_changes, floating_t *local_scattering_lengths, floating_t *local_absorption_lengths, floating_t *sca_step_left, floating_t *abs_lens_left, floating_t *distancePropagated, floating_t *distanceToAbsorption);

#endif
```

```
// https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/ice_layers/ice_layers.c

#ifndef ICE_LAYERS_C
#define ICE_LAYERS_C

#include "ice_layers.h"

inline void add_ice_layers_on_photon_path_to_medium_changes(floating4_t photonPosAndTime, floating4_t photonDirAndWlen, floating_t photonRange, int *number_of_medium_changes, floating_t *distances_to_medium_changes, floating_t *local_scattering_lengths, floating_t *local_absorption_lengths)
{

  // The closest ice layer is special, because we need to check how far
  // it is away from the photon. After that, all photon layers are equidistant.
  //
  floating_t z_of_closest_ice_layer_boundary =
      mediumLayerBoundary(photon_layer(photonPosAndTime.z));
  if (photonDirAndWlen.z > ZERO) z_of_closest_ice_layer_boundary +=
      (floating_t)MEDIUM_LAYER_THICKNESS;

  *number_of_medium_changes += 1;
  distances_to_medium_changes[*number_of_medium_changes] =
      my_divide(z_of_closest_ice_layer_boundary - photonPosAndTime.z,
       photonDirAndWlen.z);
  int next_photon_layer =
      photon_layer(z_of_closest_ice_layer_boundary + photonDirAndWlen.z);
  local_scattering_lengths[*number_of_medium_changes] =
      getScatteringLength(next_photon_layer, photonDirAndWlen.w);
  local_absorption_lengths[*number_of_medium_changes] =
      getAbsorptionLength(next_photon_layer, photonDirAndWlen.w);

```

```
29    // Now loop through the equidistant layers in range.
30    //
31    const floating_t max_trajectory_length_between_two_layers =
32        my_divide((floating_t)MEDIUM_LAYER_THICKNESS,
          ↪  my_fabs(photonDirAndWlen.z));
33    while (distances_to_medium_changes[*number_of_medium_changes] +
      ↪  max_trajectory_length_between_two_layers < photonRange)
34    {
35      *number_of_medium_changes += 1;
36      distances_to_medium_changes[*number_of_medium_changes] =
37          distances_to_medium_changes[*number_of_medium_changes - 1]
38          + max_trajectory_length_between_two_layers;
39      next_photon_layer = photon_layer(photonPosAndTime.z
40          + (distances_to_medium_changes[*number_of_medium_changes] + 0.01) *
            ↪  photonDirAndWlen.z);
41      local_scattering_lengths[*number_of_medium_changes] =
42          getScatteringLength(next_photon_layer, photonDirAndWlen.w);
43      local_absorption_lengths[*number_of_medium_changes] =
44          getAbsorptionLength(next_photon_layer, photonDirAndWlen.w);
45    }
46
47  }
48
49  inline int photon_layer(floating_t z)
50  {
51    return min(max(findLayerForGivenZPos(z), 0), MEDIUM_LAYERS-1);
52  }
53
54  #endif
```

```
1   // https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/
    ↪  ice_layers/ice_layers.h
2
3   #ifndef ICE_LAYERS_H
4   #define ICE_LAYERS_H
5
6   inline void add_ice_layers_on_photon_path_to_medium_changes(floating4_t
    ↪  photonPosAndTime, floating4_t photonDirAndWlen, floating_t photonRange, int
    ↪  *number_of_medium_changes, floating_t *distances_to_medium_changes,
    ↪  floating_t *local_scattering_lengths, floating_t
    ↪  *local_absorption_lengths);
7
8   inline int photon_layer(floating_t z);
9
10  #endif
```

```
// https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/hole_ice/hole_ice.c

#ifndef HOLE_ICE_C
#define HOLE_ICE_C

#include "hole_ice.h"
#include "../intersection/intersection.c"

inline void add_hole_ice_cylinders_on_photon_path_to_medium_changes(floating4_t photonPosAndTime, floating4_t photonDirAndWlen, floating_t photonRange, unsigned int numberOfCylinders, __constant floating4_t *cylinderPositionsAndRadii, int *number_of_medium_changes, floating_t *distances_to_medium_changes, floating_t *local_scattering_lengths, floating_t *local_absorption_lengths)
{
  // Find out which cylinders are in range in a separate loop
  // in order to improve parallelism and thereby performance.
  //
  // See: https://github.com/fiedl/hole-ice-study/issues/30
  //
  #ifdef NUMBER_OF_CYLINDERS
    // When running this on OpenCL, defining arrays using a constant
    // as array size is not possible. Therefore, we need to use a
    // pre-processor makro here.
    //
    // See: https://github.com/fiedl/hole-ice-study/issues/38
    //
    int indices_of_cylinders_in_range[NUMBER_OF_CYLINDERS];
  #else
    int indices_of_cylinders_in_range[numberOfCylinders];
  #endif
  {
    unsigned int j = 0;
    for (unsigned int i = 0; i < numberOfCylinders; i++) {
      indices_of_cylinders_in_range[i] = -1;
    }
    for (unsigned int i = 0; i < numberOfCylinders; i++) {
      if (sqr(photonPosAndTime.x - cylinderPositionsAndRadii[i].x) +
          sqr(photonPosAndTime.y - cylinderPositionsAndRadii[i].y) <=
          sqr(photonRange + cylinderPositionsAndRadii[i].w /* radius */))
      {

        // If the cylinder has a z-range check if we consider that cylinder
        // to be in range. https://github.com/fiedl/hole-ice-study/issues/34
        //
```

```
41          if ((cylinderPositionsAndRadii[i].z == 0) ||
            ↪  ((cylinderPositionsAndRadii[i].z != 0) && !(((photonPosAndTime.z <
            ↪  cylinderPositionsAndRadii[i].z - 0.5) && (photonPosAndTime.z +
            ↪  photonRange * photonDirAndWlen.z < cylinderPositionsAndRadii[i].z -
            ↪  0.5)) || ((photonPosAndTime.z > cylinderPositionsAndRadii[i].z +
            ↪  0.5) && (photonPosAndTime.z + photonRange * photonDirAndWlen.z >
            ↪  cylinderPositionsAndRadii[i].z + 0.5)))))
42          {
43            indices_of_cylinders_in_range[j] = i;
44            j += 1;
45          }
46        }
47      }
48    }
49
50    // Now loop over all cylinders in range and calculate corrections
51    // for `*distancePropagated` and `*distanceToAbsorption`.
52    //
53    for (unsigned int j = 0; j < numberOfCylinders; j++) {
54      const int i = indices_of_cylinders_in_range[j];
55      if (i == -1) {
56        break;
57      } else {
58
59        IntersectionProblemParameters_t p = {
60
61          // Input values
62          photonPosAndTime.x,
63          photonPosAndTime.y,
64          cylinderPositionsAndRadii[i].x,
65          cylinderPositionsAndRadii[i].y,
66          cylinderPositionsAndRadii[i].w, // radius
67          photonDirAndWlen,
68          1.0, // distance used to calculate s1 and s2 relative to
69
70          // Output values (will be calculated)
71          0, // discriminant
72          0, // s1
73          0  // s2
74
75        };
76
77        calculate_intersections(&p);
78
79        if (intersection_discriminant(p) > 0) {
80          if ((intersection_s1(p) <= 0) && (intersection_s2(p) >= 0)) {
81            // The photon is already within the hole ice.
82            local_scattering_lengths[0] = cylinderScatteringLengths[i];
83            local_absorption_lengths[0] = cylinderAbsorptionLengths[i];
```

```
84          } else if (intersection_s1(p) > 0) {
85            // The photon enters the hole ice on its way.
86            *number_of_medium_changes += 1;
87            distances_to_medium_changes[*number_of_medium_changes] =
              ↪  intersection_s1(p);
88            local_scattering_lengths[*number_of_medium_changes] =
              ↪  cylinderScatteringLengths[i];
89            local_absorption_lengths[*number_of_medium_changes] =
              ↪  cylinderAbsorptionLengths[i];
90          }
91          if (intersection_s2(p) > 0) {
92            // The photon leaves the hole ice on its way.
93            *number_of_medium_changes += 1;
94            distances_to_medium_changes[*number_of_medium_changes] =
              ↪  intersection_s2(p);
95            if (i == 0) // there is no larger cylinder
96            {
97              const int photonLayerAtTheCylinderBorder =
98                  photon_layer(photonPosAndTime.z + photonDirAndWlen.z *
                    ↪  intersection_s2(p));
99              local_scattering_lengths[*number_of_medium_changes] =
100                 getScatteringLength(photonLayerAtTheCylinderBorder,
                    ↪  photonDirAndWlen.w);
101             local_absorption_lengths[*number_of_medium_changes] =
102                 getAbsorptionLength(photonLayerAtTheCylinderBorder,
                    ↪  photonDirAndWlen.w);
103           } else {
104             // There is a larger cylinder outside this one, which is the one
                ↪  before in the array.
105             // See: https://github.com/fiedl/hole-ice-study/issues/47
106             //
107             local_scattering_lengths[*number_of_medium_changes] =
                ↪  cylinderScatteringLengths[i - 1];
108             local_absorption_lengths[*number_of_medium_changes] =
                ↪  cylinderAbsorptionLengths[i - 1];
109           }
110         }
111       }
112
113     }
114   }
115
116 }
117
118 #endif
```

```
1 // https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/
  ↪  hole_ice/hole_ice.h
```

```

#ifndef HOLE_ICE_H
#define HOLE_ICE_H

inline void add_hole_ice_cylinders_on_photon_path_to_medium_changes(floating4_t photonPosAndTime, floating4_t photonDirAndWlen, floating_t photonRange, unsigned int numberOfCylinders, __constant floating4_t *cylinderPositionsAndRadii, int *number_of_medium_changes, floating_t *distances_to_medium_changes, floating_t *local_scattering_lengths, floating_t *local_absorption_lengths);

#endif
```

```
// https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/intersection/intersection.c

#include "intersection.h"

inline void calculate_intersections(IntersectionProblemParameters_t *p)
{
  // Step 1
  const floating4_t vector_AM = {p->mx - p->ax, p->my - p->ay, 0.0, 0.0};
  const floating_t xy_projection_factor = my_sqrt(1 - sqr(p->direction.z));
  const floating_t length_AMprime = dot(vector_AM, p->direction) / xy_projection_factor;

  // Step 2
  p->discriminant = sqr(p->r) - dot(vector_AM, vector_AM) + sqr(length_AMprime);

  // Step 3
  const floating_t length_XMprime = my_sqrt(p->discriminant);

  // Step 4
  const floating_t length_AX1 = length_AMprime - length_XMprime;
  const floating_t length_AX2 = length_AMprime + length_XMprime;
  p->s1 = length_AX1 / p->distance / xy_projection_factor;
  p->s2 = length_AX2 / p->distance / xy_projection_factor;
}

inline floating_t intersection_s1(IntersectionProblemParameters_t p)
{
  return p.s1;
}

inline floating_t intersection_s2(IntersectionProblemParameters_t p)
{
```

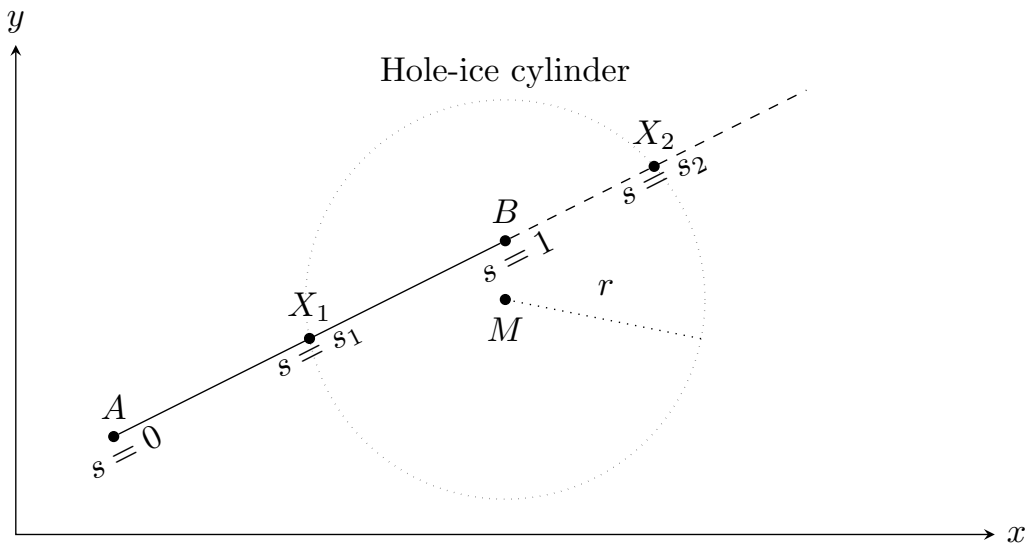
```
32    return p.s2;
33  }
34
35  inline floating_t intersection_discriminant(IntersectionProblemParameters_t p)
36  {
37    return p.discriminant;
38  }
39
40  inline floating_t intersection_x1(IntersectionProblemParameters_t p)
41  {
42    if ((p.s1 > 0) && (p.s1 < 1))
43      return p.ax + p.direction.x * p.distance * p.s1;
44    else
45      return my_nan();
46  }
47
48  inline floating_t intersection_x2(IntersectionProblemParameters_t p)
49  {
50    if ((p.s2 > 0) && (p.s2 < 1))
51      return p.ax + p.direction.x * p.distance * p.s2;
52    else
53      return my_nan();
54  }
55
56  inline floating_t intersection_y1(IntersectionProblemParameters_t p)
57  {
58    if ((p.s1 > 0) && (p.s1 < 1))
59      return p.ay + p.direction.y * p.distance * p.s1;
60    else
61      return my_nan();
62  }
63
64  inline floating_t intersection_y2(IntersectionProblemParameters_t p)
65  {
66    if ((p.s2 > 0) && (p.s2 < 1))
67      return p.ay + p.direction.y * p.distance * p.s2;
68    else
69      return my_nan();
70  }
71
72  inline bool
    ↪  intersecting_trajectory_starts_inside(IntersectionProblemParameters_t p)
73  {
74    return (intersection_s1(p) <= 0) &&
75        (intersection_s2(p) > 0) &&
76        (intersection_discriminant(p) > 0);
77  }
78
```

```
79 inline bool
   ↪  intersecting_trajectory_starts_outside(IntersectionProblemParameters_t p)
80 {
81   return ( ! intersecting_trajectory_starts_inside(p));
82 }
83
84 inline bool intersecting_trajectory_ends_inside(IntersectionProblemParameters_t
   ↪  p)
85 {
86   return (intersection_s1(p) < 1) &&
87       (intersection_s2(p) >= 1) &&
88       (intersection_discriminant(p) > 0);
89 }
```

```
1  // https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/
   ↪  intersection/intersection.h
2
3  #ifndef INTERSECTION_H
4  #define INTERSECTION_H
5
6  typedef struct IntersectionProblemParameters {
7
8    // Input values
9    //
10   floating_t ax;
11   floating_t ay;
12   floating_t mx;
13   floating_t my;
14   floating_t r;
15   floating4_t direction;
16   floating_t distance;
17
18   // Output values, which will be calculated in
19   // `calculate_intersections()`.
20   //
21   floating_t discriminant;
22   floating_t s1;
23   floating_t s2;
24
25 } IntersectionProblemParameters_t;
26
27 #endif
```

A.11 Calculating Intersections of Photon Trajectories With Hole-Ice Cylinders

A.11.1 Analytic Approach



Consider the starting point $A := (A_x, A_y)$ and the ending point $B := (B_x, B_y)$ of the trajectory.

The length $\overline{AB}$ of the trajectory is given by the norm $\|.\|$ of the vector difference $\overrightarrow{AB}$ of starting and ending point.

$$\overline{AB} = \left\|\overrightarrow{AB}\right\|, \quad \overrightarrow{AB} \equiv \overrightarrow{B} - \overrightarrow{A}, \quad \|\overrightarrow{v}\| \equiv \sqrt{v_x^2 + v_y^2}$$

In order to find the intersection points X_1 and X_2, solve the vectorial system of equations

$$\overrightarrow{A} + s\,\overrightarrow{AB} = \overrightarrow{M} + \overrightarrow{MX} \tag{17}$$

$$\left\|\overrightarrow{MX}\right\| = r \tag{18}$$

for the scalar parameter s. The equation system is quadratic in s resulting in zero, one or two solutions.

$$s_{1,2} = \frac{-\beta \mp \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \tag{19}$$

$$\alpha = (B_x - A_x)^2 + (B_y - A_y)^2 \tag{20}$$

$$\beta = 2\,A_x(B_x - A_x) + 2\,A_y(B_y - Ay) - 2\,M_x(B_x - A_x) - 2\,M_y(B_y - A_y) \tag{21}$$

$$\gamma = A_x^2 - 2\,A_x\,M_x^2 + M_x^2 + A_y^2 - 2\,A_y\,M_y + M_y^2 - r^2 \tag{22}$$

The term under the square root is also called **discriminant** D.

$$D = \beta^2 - 4\alpha\gamma \tag{23}$$

For $D < 0$, the square root does not exist in $\mathbb{R}$ and therefore, no intersection point exists. For $D = 0$, there is only one intersection point, which is a tangent point. For $D > 0$, there are two intersection points.

Note that $s = 0$ at the starting point A, $s = 1$ at the ending point B, $s = s_1$ at the first intersection point X_1 and $s = s_2$ at the second intersection point X_2.

Therefore, the intersection point coordinate vectors $\overrightarrow{X_1}$ and $\overrightarrow{X_2}$ are given by:

$$\overrightarrow{X_1} = \overrightarrow{A} + s_1\,\overrightarrow{AB} \tag{24}$$

$$\overrightarrow{X_2} = \overrightarrow{A} + s_2\,\overrightarrow{AB} \tag{25}$$

This result can be verified using a symbolic mathematics library like SymPy[37].

```
# python

from sympy import *
init_printing(use_unicode=True)

# Variables
```

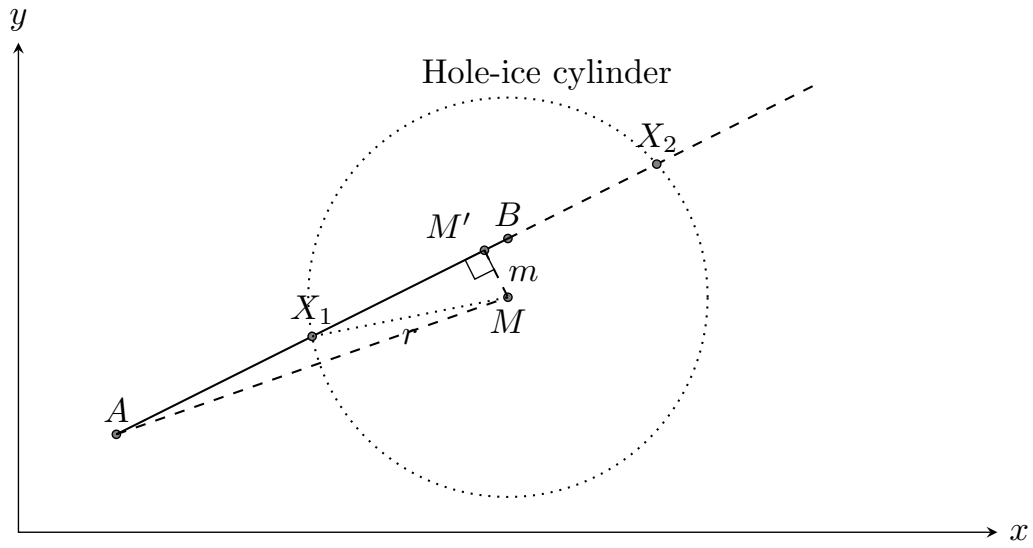
[37] **SymPy** is a Python library for symbolic mathematics. http://sympy.org

```
# ---------

Ax, Ay = symbols("Ax Ay")
Bx, By = symbols("Bx By")
Mx, My = symbols("Mx My")
Px, Py = symbols("Px Py")
r = symbols("r")
s = symbols("s")

vx = Bx - Ax
vy = By - Ay

# First ansatz
# ------------

lhs = (Ax + s * vx - Mx)**2 + (Ay + s * vy - My)**2 - r**2
expanded_lhs = expand(lhs)

intersection_equation = Eq(lhs, 0)

solution = solve(intersection_equation, s)
s1 = solution[0]
s2 = solution[1]

# Second ansatz
# -------------

alpha = (By - Ay)**2 + (Bx - Ax)**2
beta = 2 * Ay * (By - Ay) + 2 * Ax * (Bx - Ax) - 2 * My * (By - Ay) - 2 * Mx *
↪  (Bx - Ax)
gamma = Ay**2 - 2 * Ay * My + My**2 - r**2 + Ax**2 - 2 * Ax * Mx + Mx**2
discriminant = (beta**2 - 4 * alpha * gamma)

ss1 = (- beta - sqrt(beta**2 - 4 * alpha * gamma)) / (2 * alpha)
ss2 = (- beta + sqrt(beta**2 - 4 * alpha * gamma)) / (2 * alpha)

# Compare ansatzes
# ----------------

# See also: http://docs.sympy.org/latest/tutorial/gotchas.html

# # Does not work, because `==` checks object (structural) equality
# s1 == ss1
# s1 == ss2
# s2 == ss1
# s2 == ss2

# Should be 0 for the matching pairs:
simplify(s1 - ss1)  # => 0
```

```
simplify(s1 - ss2)
simplify(s2 - ss1)
simplify(s2 - ss2)  # => 0

# Or numerically:
s1.equals(ss1)  # True
s2.equals(ss2)  # True
```

A.11.2 Geometric Approach

It turns out, the simulation on the GPU is faster and more precise if one does not treat the coordinates, e.g. A_x, A_y as separate quantities but keeps the calculations as much as possible vectorial and uses the GPU-native vector operation functions like `dot` for the vector scalar product ("dot product").



Goal The goal is to find the distances $\overline{AX_1}, \overline{AX_2}$ from the starting point A to the intersection points X_1, X_2.

Relations Consider a projection of M onto AB. The projected point is called M'.

Given $\overline{AM'}$ and $\overline{X_1M'}$, the distances to the intersection points can be calculated as:

$$\overline{AX_{1,2}} = \overline{AM'} \mp \overline{X_1M'} \tag{26}$$

Given $\overrightarrow{AM}$ and the unit vector $\overrightarrow{v}$ in $\overrightarrow{AB}$ direction, $\overline{AM'}$ can be calculated using a vector projection:

$$\overline{AM'} = \overrightarrow{AM} \cdot \overrightarrow{v} \tag{27}$$

The operator $\cdot$ is the scalar product (dot product). The unit vector $\overrightarrow{v}$ in $\overrightarrow{AB}$ direction is $\overrightarrow{v} = \overrightarrow{AB}/\overline{AB}$.

The distance $m := \overline{MM'}$ on right hand side of the triangle $\triangle AM'M$ can be calculated using the pythagorean theorem:

$$m : \overline{AM'}^2 + m^2 = \overline{AM}^2 \tag{28}$$

The distance $\overline{X_1M}$, interpreted as part of the triangle $\triangle X_1M'M$ can be calculated using the pythagorean theorem:

$$\overline{X_1M'} : \overline{X_1M'}^2 + m^2 = r^2 \tag{29}$$

The length $\overline{AM}$ of the vector $\overrightarrow{AM}$ can be calculated using the scalar product, i.e. projecting the vector on itself, and taking the square root:

$$\overline{AM} = \sqrt{\overrightarrow{AM} \cdot \overrightarrow{AM}} \tag{30}$$

Using both pythagorean equations, eliminating m, one finds:

$$\begin{aligned} \overline{X_1M'} &= \sqrt{r^2 - m^2} \\ &= \sqrt{r^2 - (\overline{AM}^2 - \overline{AM'}^2)} \\ &= \sqrt{r^2 - \overline{AM}^2 + \overline{AM'}^2} \\ &= \sqrt{r^2 - \overrightarrow{AM} \cdot \overrightarrow{AM} + \overline{AM'}^2} \end{aligned} \tag{31}$$

Algorithm Using these relations, the desired distances $\overline{AX_1}, \overline{AX_2}$ can be calculated with the following steps:

1. Calculate the length $\overline{AM'}$ by projecting $\overrightarrow{AM}$ onto the unit vector $\overrightarrow{v}$ in $\overrightarrow{AB}$-direction:
$$\overline{AM'} = \overrightarrow{AM} \cdot \overrightarrow{v}$$

2. Calculate $\overline{X_1M'}^2$:
$$\overline{X_1M'}^2 = r^2 - \overrightarrow{AM} \cdot \overrightarrow{AM} + \overline{AM'}^2$$

3. Calculate $\overline{X_1M'}$:
$$\overline{X_1M'} = \sqrt{\overline{X_1M'}^2}$$
In this approach, $\overline{X_1M'}^2$ plays the role of the **discriminant**. If it is greater than zero, there are two intersection points. If it is zero, the intersection point X falls onto M', i.e. the line is actually a tangent. If the discriminant is less than zero, there is no intersection, because the radius r is too small, i.e. the line is out of reach, resulting in the discriminant becoming negative.

4. Calculate the desired distances $\overline{AX_1}$ and $\overline{AX_2}$:
$$\overline{AX_{1,2}} = \overline{AM'} \mp \overline{X_1M'}$$

Or, as C code using the OpenCL native vector functions:

```
// Step 1
const floating4_t vector_AM = (const floating4_t)(M.x - A.x, M.y - A.y, 0, 0);
const floating_t length_AMprime = dot(vector_AM, v);

// Step 2
const floating_t discriminant = r * r - dot(vector_AM, vector_AM) +
↪  length_AMprime * length_AMprime;

// Step 3
const floating_t length_XMprime = native_sqrt(discriminant);

// Step 4
const floating_t length_AX1 = length_AMprime - length_XMprime;
const floating_t length_AX2 = length_AMprime + length_XMprime;
```

A.12 How to Switch Back to Standard-CLSIM's Media-Propagation Algorithm

In order to switch back to the standard CLSIM algorithm, even after the new code has been merged into the ICECUBE simulation framework, the standard CLSIM code is provided as drop-in replacement.

The standard CLSIM code as drop-in replacement can be found at https://github.com/fiedl/clsim/blob/sf/hole-ice-2018/resources/kernels/lib/propagation_through_media/standard_clsim.c.

To apply this code, switch in `propagation_kernel.c.cl`
from calling `apply_propagation_through_different_media()`
to `apply_propagation_through_different_media_with_standard_clsim()`.

Also check https://github.com/fiedl/hole-ice-study/issues/115 whether an `IceTray` switch has already been implemented.

A.13 Exponential Distribution of the Total Photon Path Length

This section shows why the total-path-length distribution of photons propagating through a homogeneous medium described in section 5.6.4 is expected to follow an exponential curve.

Let N be the total number of started within the medium. Let λ_{abs} be the photon absorption length within the medium.

The probability $p(x) := f\,\mathrm{d}x$ for a photon to be absorbed within its path length interval $[x; x + \mathrm{d}x[$ is the same as long as the photon stays within the homogeneous medium.

Let $n(x)$ be the number of photons with a path length of x or more, which is the number of photons that still exist with a path length greater or equal x.

The number of photons $m(x)$ that are absorbed within the interval $[x; x + \mathrm{d}x[$ is determined by the change of the number of remaining photons, which in the limit of many photons is proportional to the absorption probability $p(x)$.

$$m(x) = -\mathrm{d}n(x) = p(x)\ n(x) = f\,\mathrm{d}x\,n(x)$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\ n(x) = -f\ n(x)$$

As the derivative of n is proportional to n, n is an exponential.

$$n(x) = a\,\mathrm{e}^{b\,x}, \quad \frac{\mathrm{d}}{\mathrm{d}x}\,n(x) = b\,a\,\mathrm{e}^{b\,x} = b\,n(x) = -f\,n(x)$$

The absorption length λ_{abs} is defined as the distance after that the number of photons has dropped to $1/\mathrm{e}$ of the original number.

$$n(x) = a\,\mathrm{e}^{b\,x}, \quad n(\lambda_{\text{abs}}) = \frac{1}{\mathrm{e}}\,n(0) \quad \Rightarrow \quad b = {}^{-1}/_{\lambda_{\text{abs}}}$$

$$n(x) = n(0) \cdot \mathrm{e}^{\,-x/\lambda_{\text{abs}}}, \quad n(0) = N$$

In a histogram of the total path lengths X, the bin height is proportional to the number $m(x)$ of photons that are absorbed within the interval $[x; x + \mathrm{d}x[$, not the number $n(x)$ of remaining photons.

$$m(x) = p(x)\, n(x) = p(x)\, N\, \mathrm{e}^{-x/\lambda_{\mathrm{abs}}}$$

In the case of a homogeneous medium where $p(x)$ is constant for all x, $p(x) = p_0$, the histogram should also follow an exponential curve.

$$m(x) = p_0\, N\, \mathrm{e}^{-x/\lambda_{\mathrm{abs}}}$$

From the rate of the exponential decay, one can read the absorption length λ_{abs}.

If, on the other hand, there is a medium border at x_0 such that the absorption probability is piecewise defined,

$$p(x) = \begin{cases} p_0 & : x \leq x_0 \\ p_1 & : x > x_0 \end{cases}$$

then the histogram follows a piecewise defined curve, consisting of two exponential curves.

$$m(x) = \begin{cases} p_0\, N\, \mathrm{e}^{-x/\lambda_0} & : x \leq x_0 \\ p_1\, N\, \mathrm{e}^{-x/\lambda_1} & : x > x_0 \end{cases}$$

From the rates of the exponential decays, one can read the absorption lengths λ_0 and λ_1 of both media from the histogram.

Note that as the histogram shows the number $m(x)$ of decayed photons within an interval, not the number $n(x)$ of remaining photons, the curve the histogram follows does not need to be continuous but may have a jump discontinuity at the position x_0 of the medium border.

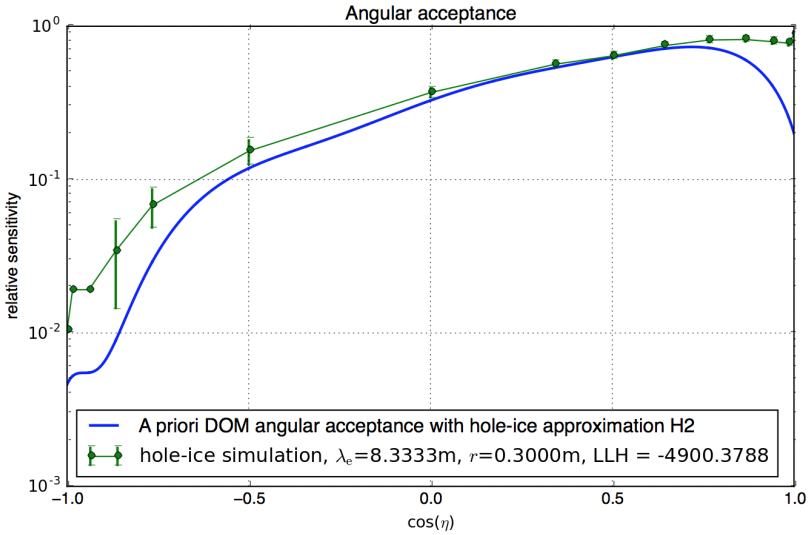
The number $n(x)$ of remaining photons, however, is continuous, also at the position x_0 of the medium border.

$$n(x) = N - \int_0^x m(x) = \begin{cases} N - f_0\, N\, \lambda_0\, \mathrm{e}^{-x/\lambda_0} & : x \leq x_0 \\ n(x_0) - f_1\, N\, \lambda_1\, \mathrm{e}^{-x/\lambda_1} & : x > x_0 \end{cases}$$

A.14 Angular-Acceptance Simulation For H2 Parameters

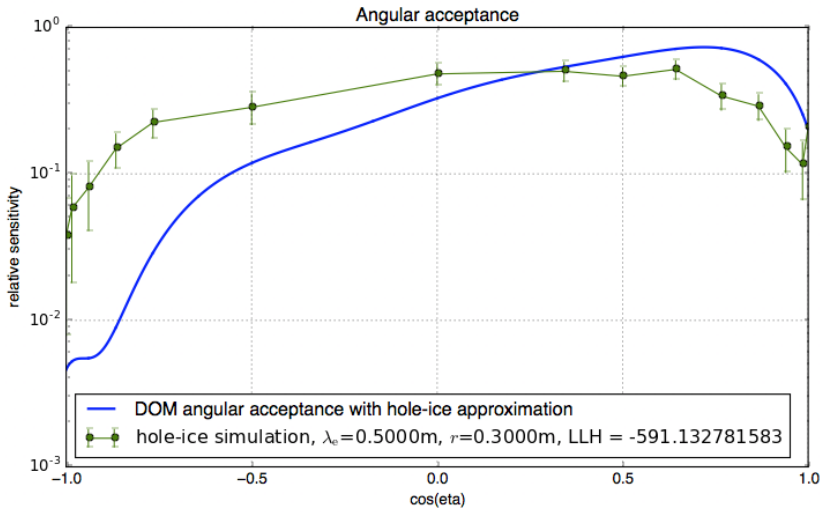
The so-called H2 hole-ice parameters assume a hole-ice radius of 30 cm, corresponding to the entire drill hole being filled with hole ice, and a geometric hole-ice scattering length of 50 cm, corresponding to an effective scattering length of 8.33 m. [Kar98]

Using the new medium-propagation algorithm (section 5.4) with direct detection and plane waves as photon sources (section 6.2) to generate an effective angular-acceptance curve for these H2 hole-ice parameters:



The blue a priori curve is based on previous PHOTONICS simulations assuming the same H2 hole-ice parameters. [Lun+07; Col+13]

The same simulation assuming an effective hole-ice scattering length of 50 cm, corresponding to a geometric scattering length of 3 cm:



These simulations are documented in `issues/80` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/80.

A.15 Technical Issues Concerning Graphics Processing Units

Driver Issue Concerning `isnan` With some GPUs or GPU drivers, basic functions like `bool isnan(float a)` may have bugs and may not always return the expected results. These kind of issues are especially hard to track down because they do not show up in unit tests when running the tests on a different architecture, for example on the local CPU. They also don't raise exceptions.

In this case, the workaround has been to use a custom implementation of the `isnan` function that uses another basic operator to determine the result and circumvents the native implementation of `isnan`.

```
1 // Fixed custom replacement of `isnan`:
2 bool my_is_nan(floating_t a) { return (a != a); }
```

The issue is documented in `issues/14` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/14. The workaround is documented in `issues/16` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/16.

Driver Issue Concerning Kernel Caching When changing the kernel code has no effect when running the simulation again, this could be due to a caching issue: The cache for kernels that are included using the `#include` pre-processor statement is not reset automatically when the included files change.

The workaround is to disable caching using an environment variable, or to reset the cache by hand.

```
# Disable caching globally. Despite the name this also works with OpenCL.
export CUDA_CACHE_DISABLE=1

# Reset cache manually:
rm -r ~/.nv/ComputeCache
```

This problem and the workaround are documented in `issues/15` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/15.

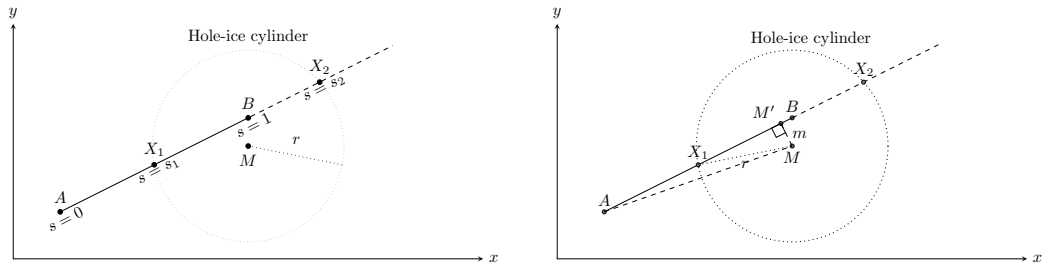
Numerical Inconsistencies Computations that are mathematically equivalent are not guaranteed to be numerically equivalent as well. Performing the same calculation using different formula, or calculating the same quantity using two different, but mathematically equivalent algorithms, may lead to different results.

If such two calculations are performed within one algorithm, this may lead to meaningless results or simulation crashes.

For example, one part of the hole-ice-correction algorithm used the PYTHAGOREAN THEOREM to calculate the distance of the photon from the hole-ice center. Comparing this to the hole-ice radius, the algorithm decided whether the photon is within or outside the cylinder. Another part of the algorithm calculated intersections of the photon trajectory with the hole-ice cylinder. Evaluating the scale parameters s_1 and s_2 (see figure 74 a), the algorithm decided again whether the photon is within or outside the hole ice.

If the photon is near the hole-ice radius, one mechanism may come to the conclusion that the photon is within the hole-ice cylinder while the other mechanism considers the photon outside the cylinder. If both mechanisms are used in the same algorithm, this leads to inconsistencies.

The solution to this problem is to only use one way to determine whether the photon is within or outside the cylinder. The decision may still be "wrong" compared to a high-precision calculation, but it will be consistent and not cause the program to crash.



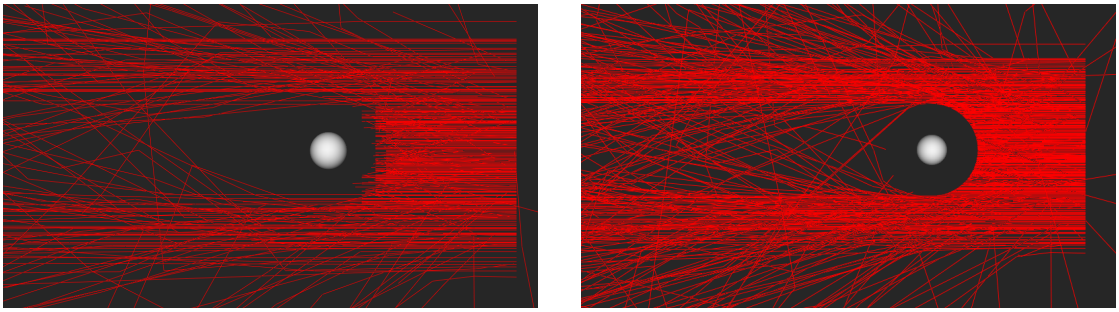
(a) using scalar quantities

(b) using vectorial quantities

Figure 74: Calculating intersections of photon trajectory AB and the hole-ice cylinder represented as circle.

Another, less destructive, but even more demonstrative example for numerical inconsistencies is comparing two ways of calculating intersections as shown in figure 74.

Both methods are described in section A.11. The important difference is that one method uses scalar quantities to calculate the intersection points, the other method uses vectorial quantities and native vector operations. While both methods are mathematically equivalent, their results differ in their numerical precisions, which is visualized in figure 75: Both simulations show photons hitting an instantly absorbing cylinder. While native vector operations cut the photons cleanly at the cylinder border, using only scalar quantities shows a scraggy border.



(a) using scalar quantities to calculate the intersection points coordinate-wise

(b) using native vector operations to calculate the intersection points

Figure 75: Simulation of photons propagating towards a cylinder configured for instant absorption. Both simulations are the same except for the method how intersections of photon rays and cylinders are calculated. Using native vector operations (b) leads to numerically more precise results.

This issue is documented in `issues/28` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/28.

Memory Issues: "OpenCL worker thread died unexpectedly" While the user might wholeheartedly agree that the crash is rather unexpected, the error message does not help to find the underlaying issue, which is most probably a memory issue, for example allocating too few or too much memory on the GPU.

To circumvent this issue, this study adds the propagation configuration parameter `MaxNumOutputPhotonsCorrectionFactor` to CLSIM, which acts as a factor within the calculation that determines how much memory will be allocated for storing photons.

As a rule of thumb, using values from `MaxNumOutputPhotonsCorrectionFactor = 0.1` to `MaxNumOutputPhotonsCorrectionFactor = 1e-3` may resolve the issue. But the memory might then be insufficient to store all propagated photons, especially when photon paths are to be saved as well as compared to only the hits.

Another workaround when creating visualizations is to propagate less photons and/or run the simulation on a CPU. The memory is much larger when running on a CPU such that this issue might not occur on the CPU. But the simulation time will be longer on the CPU.

This issue is documented in `issues/23` on the CD-ROM as well as online at https://github.com/fiedl/hole-ice-study/issues/23.

Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit „The Effect of Hole Ice on the Propagation and Detection of Light in IceCube“ von mir eigenständig und nur unter Zuhilfenahme der angegebenen Quellen und bereitgestellten Hilfsmittel angefertigt wurde.

Ort, Datum	Sebastian Fiedlschuster