Update on Direct Hole-Ice Simulations With Clsim

IceCube Fall Collaboration Meeting Madison 2022

Work by: Tianlu Yuan & Sebastian Fiedlschuster

Talk by: Sebastian Fiedlschuster (`@fiedl`)
`https://github.com/fiedl/hole-ice-scripts`
`sebastian.fiedlschuster@fau.de`

2022-09-21

Document 2022-uch8wuS7

Erlangen Centre for Astroparticle Physics

Resources

Thesis (2018-09-05) can be found at:

https://arxiv.org/abs/1904.08422

Example scripts and usage info:

https://github.com/fiedl/hole-ice-scripts

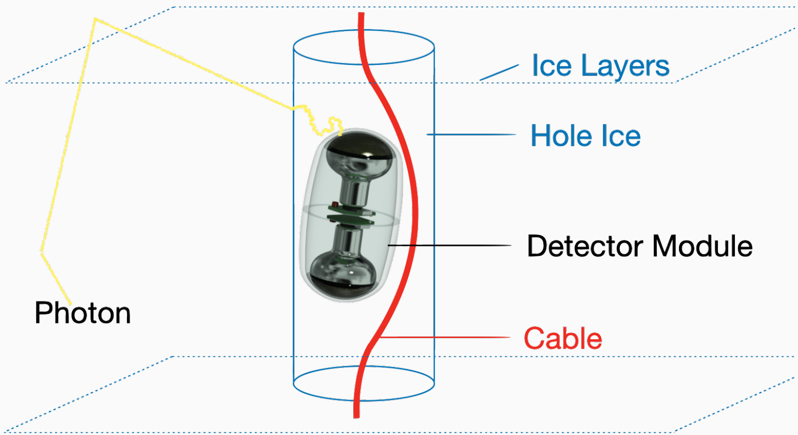
Previous talks on this topic:

https://github.com/fiedl/hole-ice-talk/releases

LaTeX version of these presentation slides:

https://github.com/fiedl/hole-ice-talk

Motivation and Scope

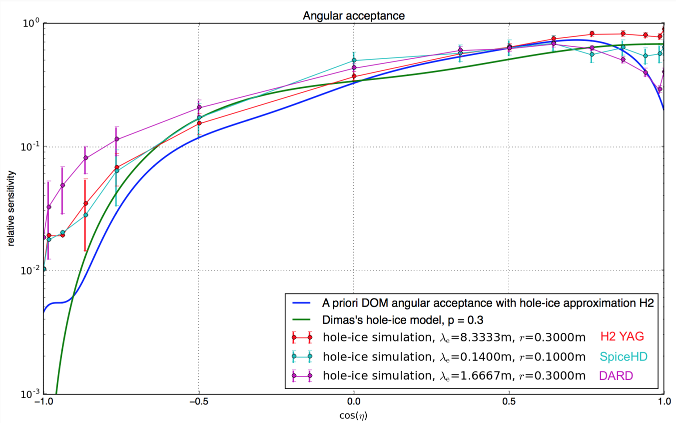


- Topic: Light-propagation simulation in vicinity of detector modules, considering:
  - ice properties in vicinity, esp. in hole ice
  - opaque cables
  - non-spherical detector modules of variable position
- Usually: Effective modification of module sensitivity
- Here: Direct ray-tracing algorithm in clsim

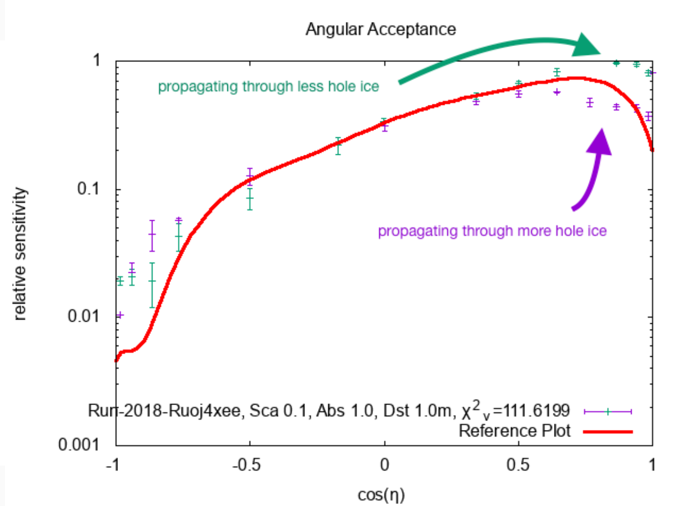
Thesis: `https://arxiv.org/abs/1904.08422`

D-Egg-detector-module image: Pfeiffer, New optical sensors for IceCube-Gen2, 2016

Motivation and Scope



Different hole-ice properties lead to angular-acceptance profiles possibly different from the a-priory approximation.



Geometric asymmetries with respect to hole ice impact effective acceptance.

Motivation and Scope

Some history:

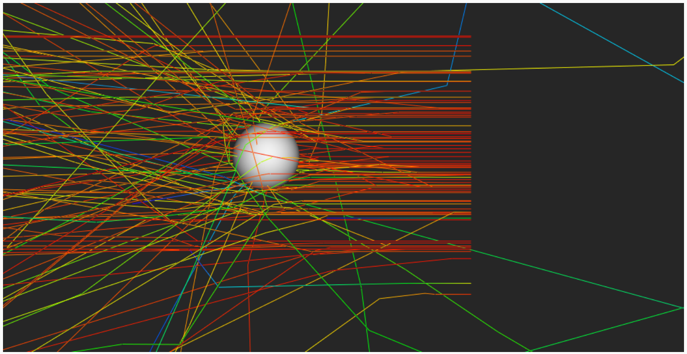
- Thesis 2018
- Kernel compatible, but interface not compatible to current icetray $\rightarrow$ hard to use
- 2021: Attempted port to ppc
- today: Compatible to current icetray

Contents

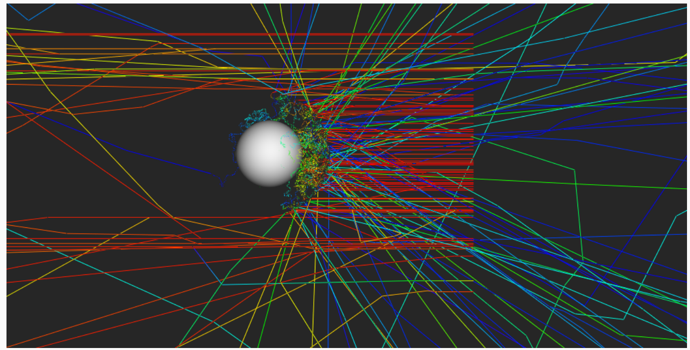
What can it do: Examples of Application

Trying out different hole-ice scattering lengths

The exact optical properties of the hole ice are unknown. With the simulation, one can try out different properties, e.g. scattering length.



Scattering length $\lambda_{\text{sca,hole-ice}} = 10^{-1}\,\lambda_{\text{sca,bulk}}$.
Absorption length $\lambda_{\text{abs,hole-ice}} = \lambda_{\text{sca,bulk}}$.



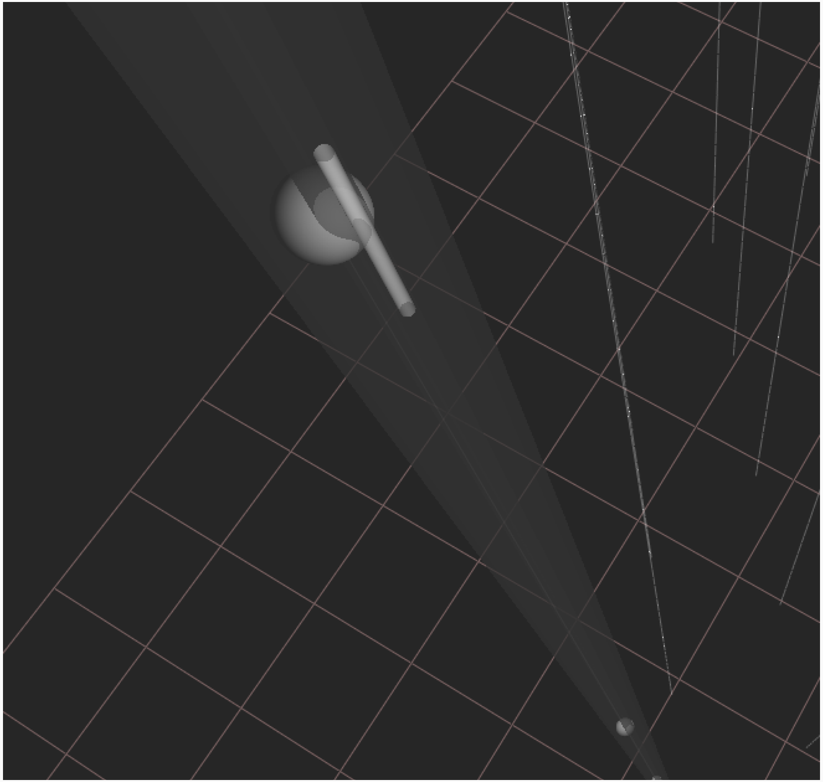
Scattering length $\lambda_{\text{sca,hole-ice}} = 10^{-3}\,\lambda_{\text{sca,bulk}}$.
Absorption length $\lambda_{\text{abs,hole-ice}} = \lambda_{\text{sca,bulk}}$.

Animation on youtube: `https://youtu.be/BhJ6F3B-I1s`

View from top onto a detector module within a hole-ice cylinder. Colors indicate simulation steps, i.e. number of scatterings relative to the total number until absorption.
Red: Photon just created, blue: Photon about to be absorbed.

Source: `https://github.com/fiedl/hole-ice-study/issues/39`

Realistic simulation scenario

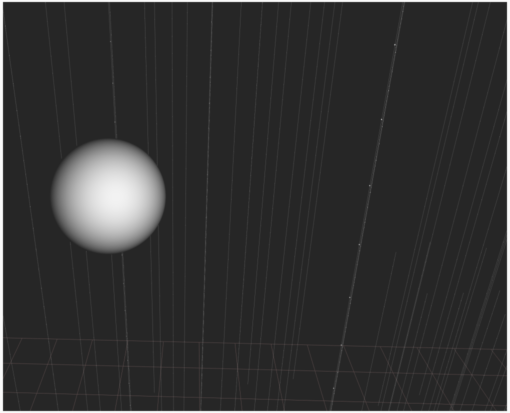


- DOM: radius 16.5 cm, shifted by 12.0 cm against the center of the bore hole
- bubble column: radius 8.0 cm
- drill-hole column: radius 30.0 cm
- cable: radius 3.0 cm, placed next to the DOM, partially within the bubble column

See also: `https://github.com/fiedl/hole-ice-study/issues/110`

How can I use it?

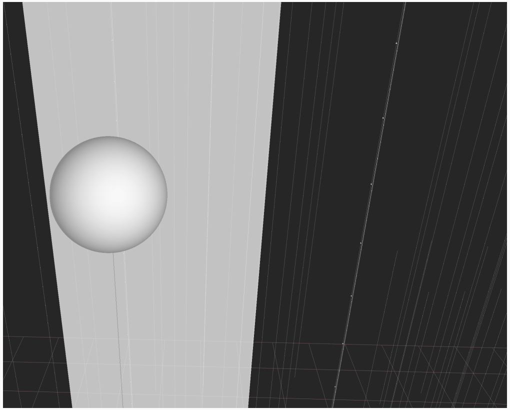
How can I configure the simulation?



```
from icecube.simclasses import I3CLSimMediumCylinder,
↪   I3CLSimMediumCylinderSeries
```

README: https://github.com/fiedl/hole-ice-scripts

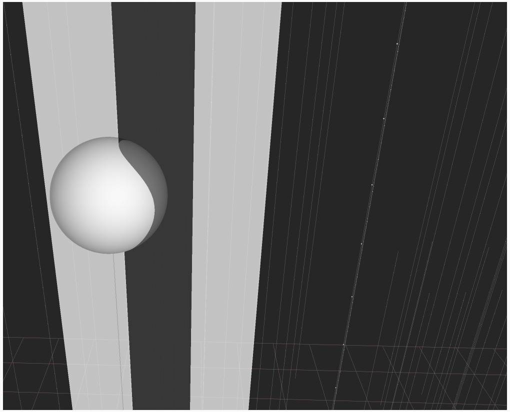
How can I configure the simulation?



```python
from icecube.simclasses import I3CLSimMediumCylinder,
↪  I3CLSimMediumCylinderSeries

drill_hole_cylinder = I3CLSimMediumCylinder()
drill_hole_cylinder.x = -256.14 + (0.30 - 0.16510) # m
drill_hole_cylinder.y = -521.08 # m
drill_hole_cylinder.radius = 0.30 # m
drill_hole_cylinder.scattering_length = 0.004 # m
```

README: https://github.com/fiedl/hole-ice-scripts

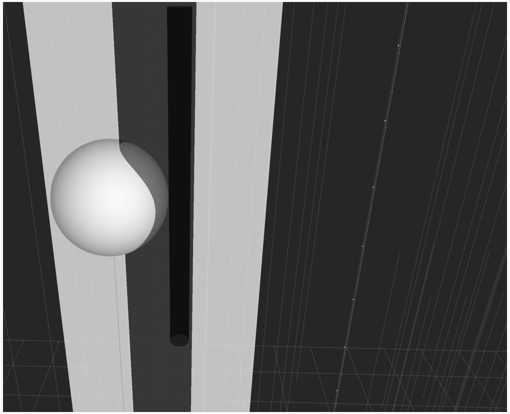
How can I configure the simulation?



```python
from icecube.simclasses import I3CLSimMediumCylinder,
↪  I3CLSimMediumCylinderSeries

drill_hole_cylinder = I3CLSimMediumCylinder()
drill_hole_cylinder.x = -256.14 + (0.30 - 0.16510) # m
drill_hole_cylinder.y = -521.08 # m
drill_hole_cylinder.radius = 0.30 # m
drill_hole_cylinder.scattering_length = 0.004 # m

bubble_column_cylinder = I3CLSimMediumCylinder()
bubble_column_cylinder.x = -256.14 + (0.30 - 0.16510) # m
bubble_column_cylinder.y = -521.08 # m
bubble_column_cylinder.radius = 0.10 # m
bubble_column_cylinder.scattering_length = 0.001 # m
```

README: `https://github.com/fiedl/hole-ice-scripts`

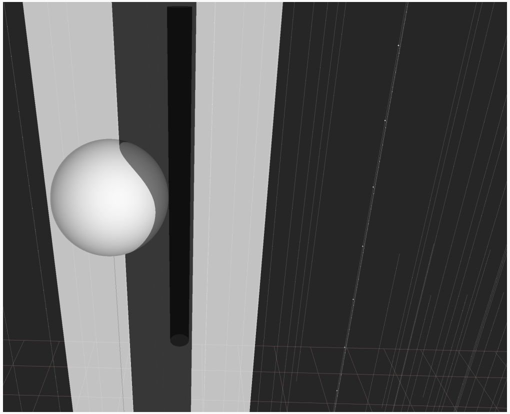
How can I configure the simulation?



```
from icecube.simclasses import I3CLSimMediumCylinder,
↪  I3CLSimMediumCylinderSeries

drill_hole_cylinder = I3CLSimMediumCylinder()
drill_hole_cylinder.x = -256.14 + (0.30 - 0.16510) # m
drill_hole_cylinder.y = -521.08 # m
drill_hole_cylinder.radius = 0.30 # m
drill_hole_cylinder.scattering_length = 0.004 # m

bubble_column_cylinder = I3CLSimMediumCylinder()
bubble_column_cylinder.x = -256.14 + (0.30 - 0.16510) # m
bubble_column_cylinder.y = -521.08 # m
bubble_column_cylinder.radius = 0.10 # m
bubble_column_cylinder.scattering_length = 0.001 # m

cable_cylinder = I3CLSimMediumCylinder()
cable_cylinder.x = -256.14 + (0.16510 + 0.03) # m
cable_cylinder.y = -521.08 # m
cable_cylinder.top = 496.03 + 0.5 # m
cable_cylinder.bottom = 496.03 - 0.5 # m
cable_cylinder.radius = 0.03 # m
cable_cylinder.absorption_length = 0.0 # m
```

README: `https://github.com/fiedl/hole-ice-scripts`

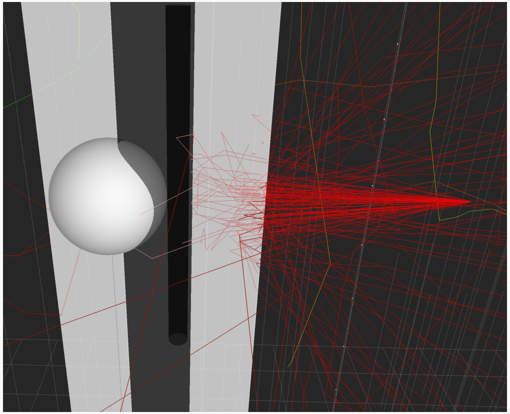
How can I configure the simulation?



```python
from icecube.simclasses import I3CLSimMediumCylinder,
↪  I3CLSimMediumCylinderSeries

drill_hole_cylinder = I3CLSimMediumCylinder()
drill_hole_cylinder.x = -256.14 + (0.30 - 0.16510) # m
drill_hole_cylinder.y = -521.08 # m
drill_hole_cylinder.radius = 0.30 # m
drill_hole_cylinder.scattering_length = 0.004 # m

bubble_column_cylinder = I3CLSimMediumCylinder()
bubble_column_cylinder.x = -256.14 + (0.30 - 0.16510) # m
bubble_column_cylinder.y = -521.08 # m
bubble_column_cylinder.radius = 0.10 # m
bubble_column_cylinder.scattering_length = 0.001 # m

cable_cylinder = I3CLSimMediumCylinder()
cable_cylinder.x = -256.14 + (0.16510 + 0.03) # m
cable_cylinder.y = -521.08 # m
cable_cylinder.top = 496.03 + 0.5 # m
cable_cylinder.bottom = 496.03 - 0.5 # m
cable_cylinder.radius = 0.03 # m
cable_cylinder.absorption_length = 0.0 # m

cylinders =
↪  I3CLSimMediumCylinderSeries([drill_hole_cylinder,
↪  bubble_column_cylinder, cable_cylinder])
geometry_frame.Put("I3CLSimMediumCylinders", cylinders)
```

README: `https://github.com/fiedl/hole-ice-scripts`

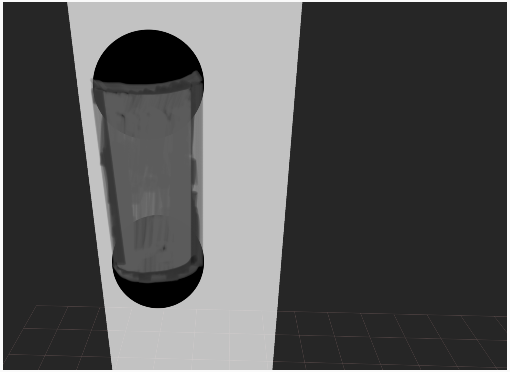
How can I configure the simulation?



README: `https://github.com/fiedl/hole-ice-scripts`

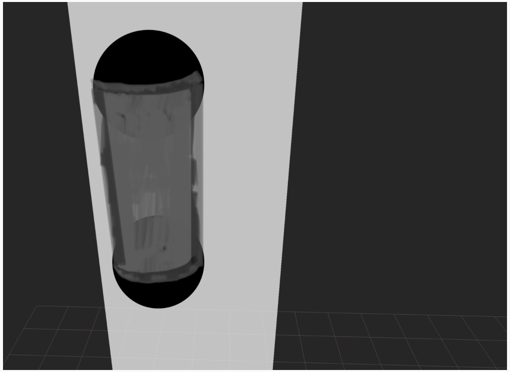
```python
from icecube.simclasses import I3CLSimMediumCylinder,
↪  I3CLSimMediumCylinderSeries

drill_hole_cylinder = I3CLSimMediumCylinder()
drill_hole_cylinder.x = -256.14 + (0.30 - 0.16510) # m
drill_hole_cylinder.y = -521.08 # m
drill_hole_cylinder.radius = 0.30 # m
drill_hole_cylinder.scattering_length = 0.004 # m

bubble_column_cylinder = I3CLSimMediumCylinder()
bubble_column_cylinder.x = -256.14 + (0.30 - 0.16510) # m
bubble_column_cylinder.y = -521.08 # m
bubble_column_cylinder.radius = 0.10 # m
bubble_column_cylinder.scattering_length = 0.001 # m

cable_cylinder = I3CLSimMediumCylinder()
cable_cylinder.x = -256.14 + (0.16510 + 0.03) # m
cable_cylinder.y = -521.08 # m
cable_cylinder.top = 496.03 + 0.5 # m
cable_cylinder.bottom = 496.03 - 0.5 # m
cable_cylinder.radius = 0.03 # m
cable_cylinder.absorption_length = 0.0 # m

cylinders =
↪  I3CLSimMediumCylinderSeries([drill_hole_cylinder,
↪  bubble_column_cylinder, cable_cylinder])
geometry_frame.Put("I3CLSimMediumCylinders", cylinders)
```

How can I configure the simulation? [WIP]



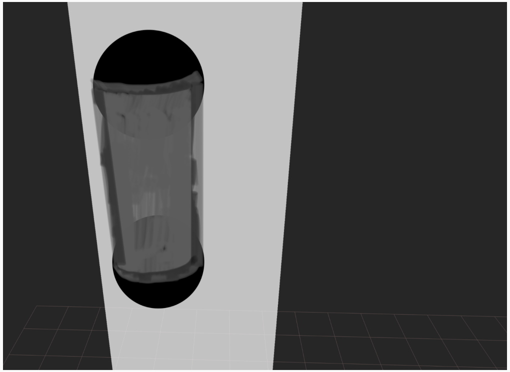
```
# work in progress:

upper_pmt = I3CLSimMediumSphere()
upper_pmt.absorption_length = 0 # m
upper_pmt.detection_probability = 0.2
# ...

lower_pmt = I3CLSimMediumSphere()
lower_pmt.absorption_length = 0 # m
lower_pmt.detection_probability = 0.2
# ...

cylinder = I3CLSimMediumCylinder()
cylinder..absorption_length = 0 # m
# ...

shapes = I3CLSimMediumShapeSeries([upper_pmt, lower_pmt,
↪  cylinder])
geometry_frame.Put("I3CLSimMediumShapes", shapes)
```

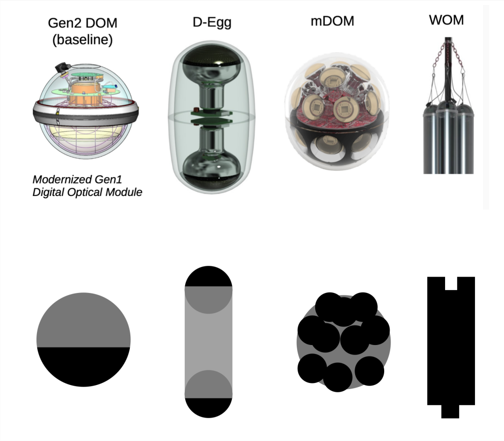
How can I configure the simulation? [WIP]



```
# work in progress:

upper_pmt = I3CLSimMediumSphere()
upper_pmt.absorption_length = 0 # m
upper_pmt.detection_probability = 0.2
# ...

lower_pmt = I3CLSimMediumSphere()
lower_pmt.absorption_length = 0 # m
lower_pmt.detection_probability = 0.2
# ...

cylinder = I3CLSimMediumCylinder()
cylinder..absorption_length = 0 # m
# ...

shapes = I3CLSimMediumShapeSeries([upper_pmt, lower_pmt,
↪  cylinder])
geometry_frame.Put("I3CLSimMediumShapes", shapes)
```

How can I configure the simulation? [WIP]






```
# work in progress:

upper_pmt = I3CLSimMediumSphere()
upper_pmt.absorption_length = 0 # m
upper_pmt.detection_probability = 0.2
# ...

lower_pmt = I3CLSimMediumSphere()
lower_pmt.absorption_length = 0 # m
lower_pmt.detection_probability = 0.2
# ...

cylinder = I3CLSimMediumCylinder()
cylinder..absorption_length = 0 # m
# ...

shapes = I3CLSimMediumShapeSeries([upper_pmt, lower_pmt,
↪  cylinder])
geometry_frame.Put("I3CLSimMediumShapes", shapes)
```

Performance

Performance

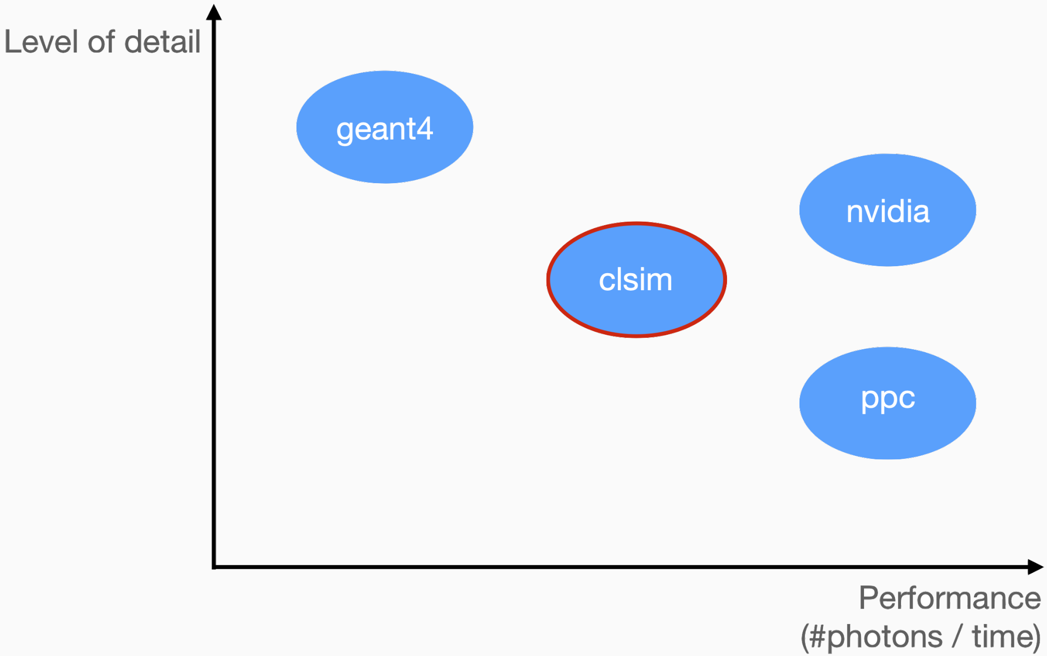
- Performance per scattering step is the same as with standard clsim
- However, for smaller scattering lengths, the number of scattering steps increases

$\rightarrow$ Longer simulation time with hole ice

Source: `https://github.com/fiedl/hole-ice-study/issues/69`

Where does this tool fit in?

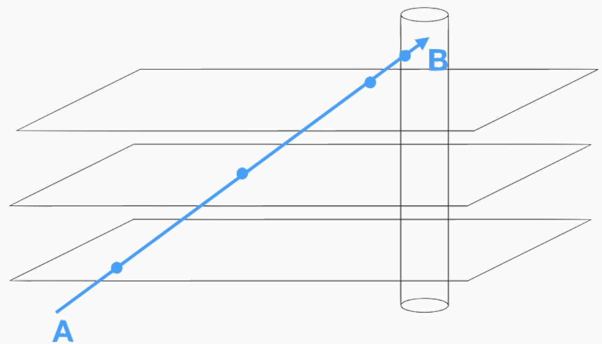
Where does this tool fit in?



Status

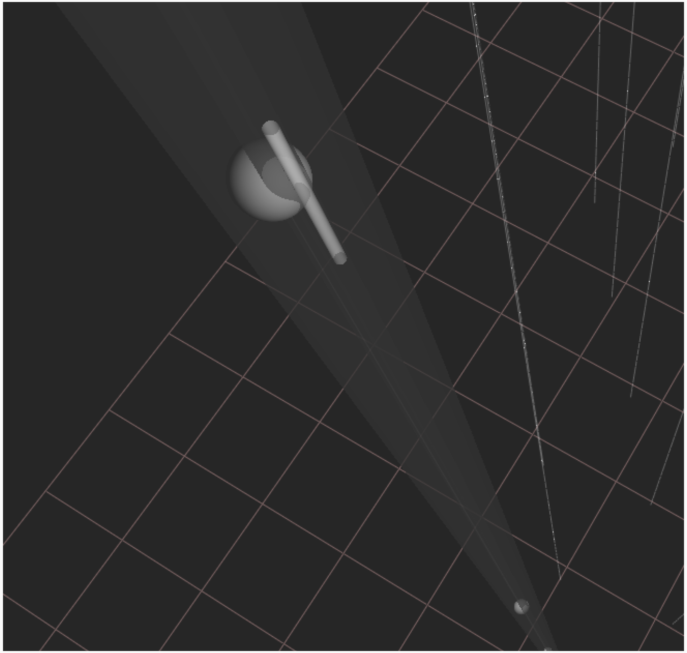
Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property



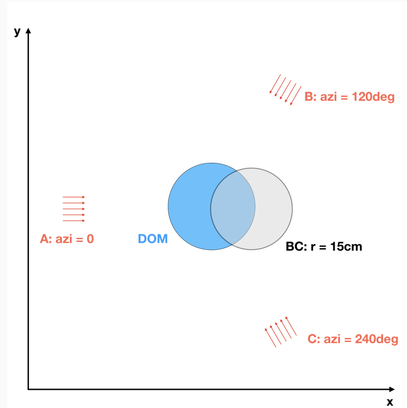
Project status

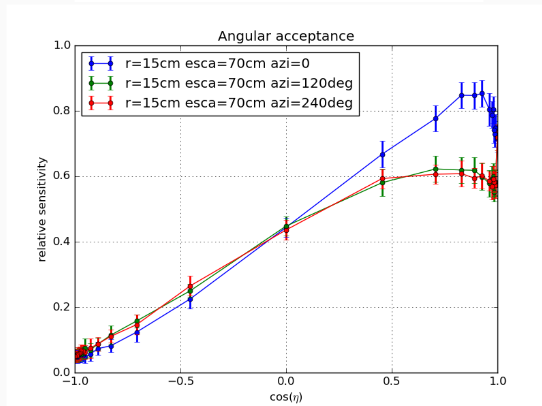
- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property



Project status

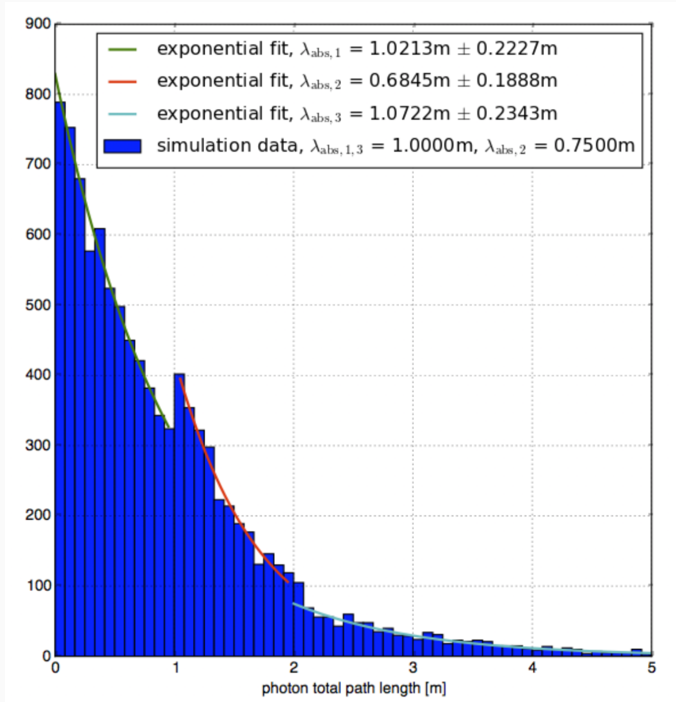
- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
    - ☐ ice tilt, ice anisotropy
    - ☐ bfr
    - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property





Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property



See also: `https://github.com/fiedl/hole-ice-study/issues/66`

Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property

Ported to icetray main branch of September 2022.

Pull request draft

`https://github.com/icecube/icetray/pull/2957`

Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property

Example scripts for new interface:

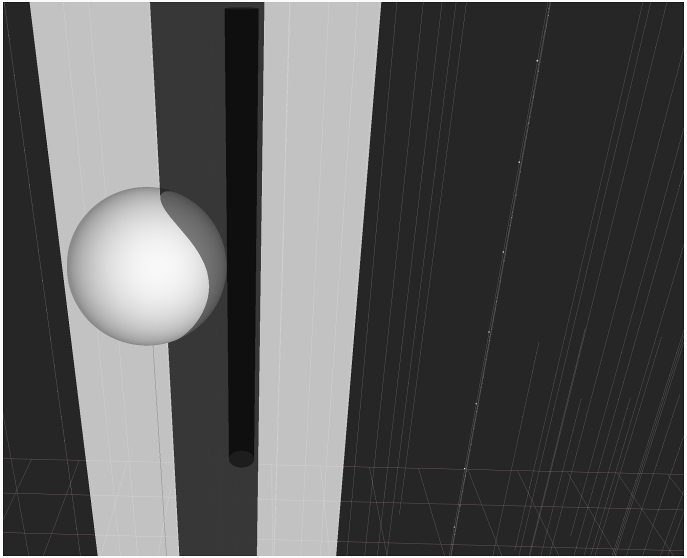
Repository

`https://github.com/fiedl/hole-ice-scripts`

Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property

Ready-to-use steamshovel artists included in pull request.

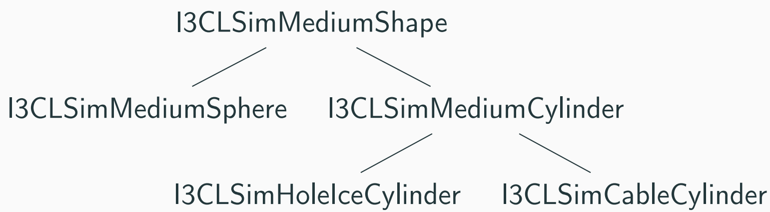


Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property

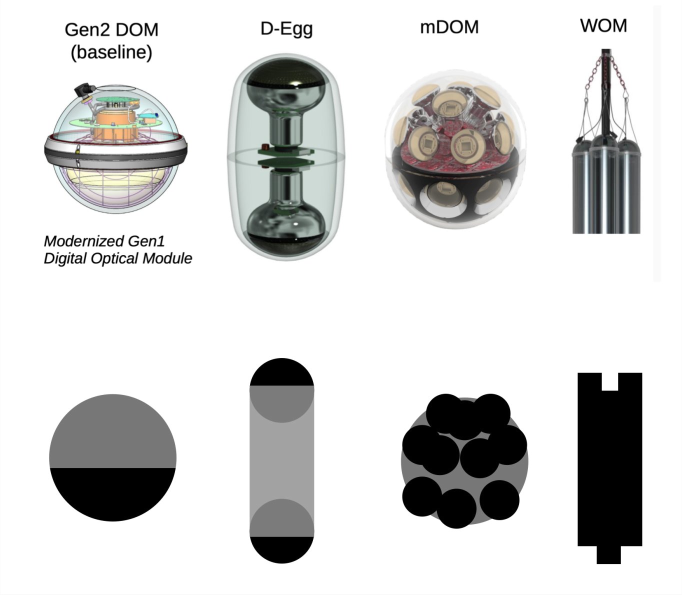
Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property



Project status

- ✓ Implement new ray-tracing algorithm
- ✓ Implement hole ice and cables
- ✓ Verify plausibility with examples
- ✓ Verify with statistical cross checks
- ✓ Bring interface up to date with current icetray
- ✓ Provide python-3 example scripts & documentation
- ✓ Provide steamshovel artists
- ⇒ Fix compatibility issues with
  - ☐ ice tilt, ice anisotropy
  - ☐ bfr
  - ☐ direct detection
- ☐ Convenient subclassing
- ☐ Implement detection-probability property



Thank you!

Thank you!

Tianlu Yuan & Jakob van Santen

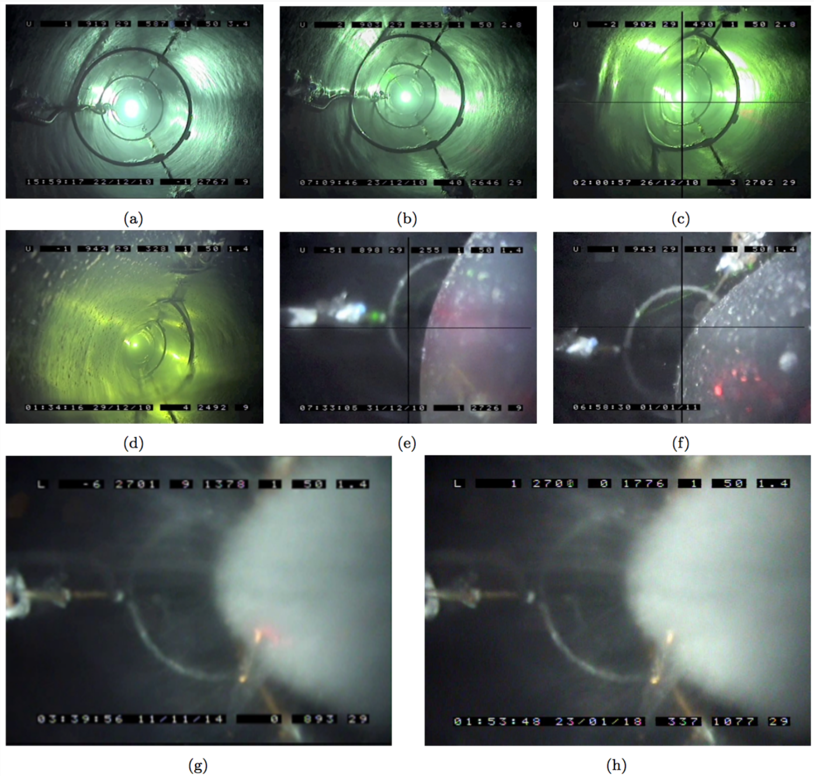
Follow or review pull request:

`https://github.com/icecube/icetray/pull/2957`

Slack: `@fiedl` sebastian.fiedlschuster@fau.de

Backup slides

Hole ice

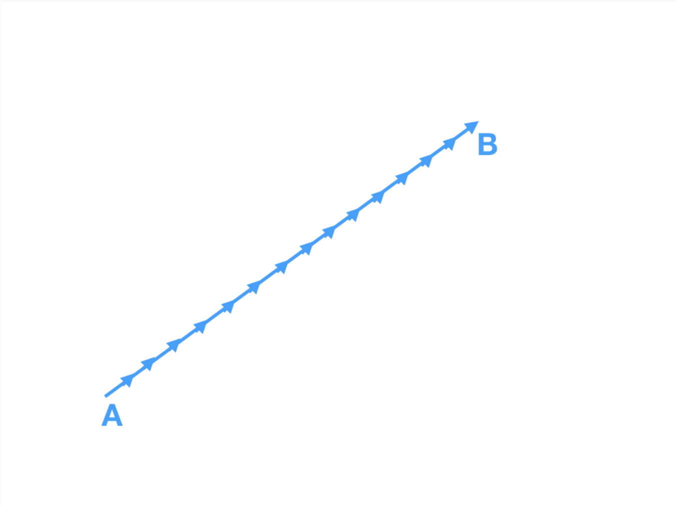


- Hole ice is the refrozen water in the drill holes around the detector modules
- possibly different optical properties than surrounding bulk ice
- special kinds:
  drill-hole ice
  bubble column

Images (a) to (g) show a time series of the freeze-in process. Image (h) shows has been taken several years after the freeze-in process. Image sources: Resconi, Rongen, Krings: The Precision Optical CAlibration Module for IceCube-Gen2: First Prototype, 2017. Finley et al.: Freezing in the IceCube camera in string 80, 22 Dec - 1st Jan. 2011. Rongen: The 2018 Sweden Camera run — light at the end of the ice, 2018.

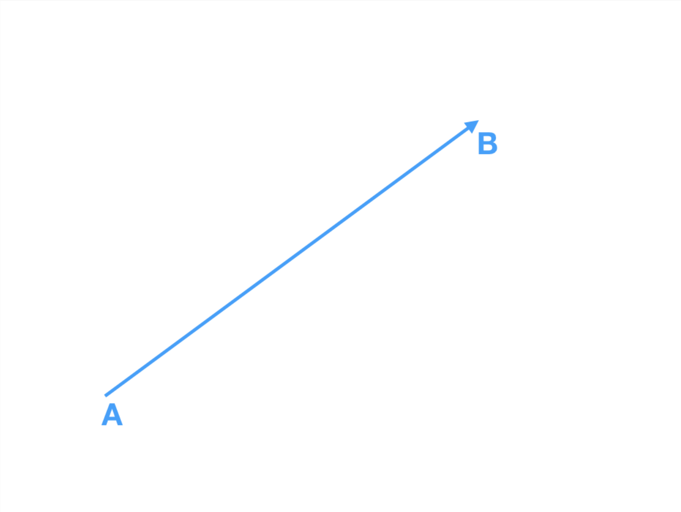
Photon-Propagation Algorithm

How does it work?



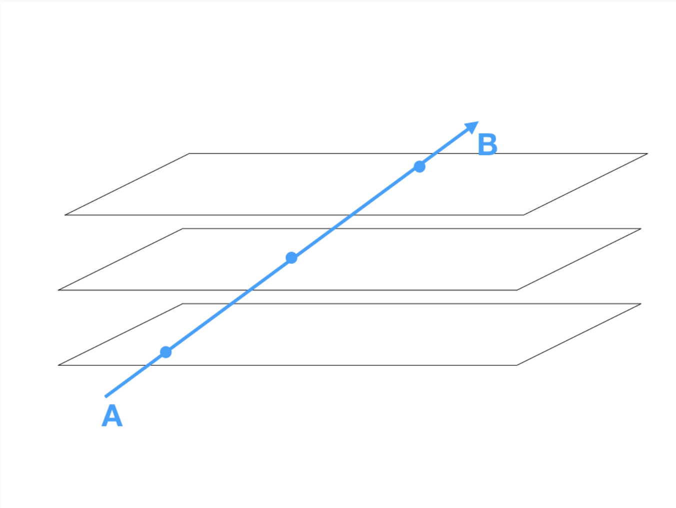
- Photon scattering points A and B
- Naive algorithm: Propagate photon small distance δx in each simulation step and randomize whether the photon will scatter in this step (easy to implement local properties)
- Faster algorithm: Randomize geometric distance to next scattering point and propagate from A to B in one simulation step
- Ice layers with different optical properties: Randomize number of scattering lengths between A and B as budget and calculate geometric distance by spending the budget over the ice layers
- New: Generalize budget algorithm to support cylinders and possibly other shapes with distinct scattering and absorption lengths and detection probabilities.

How does it work?



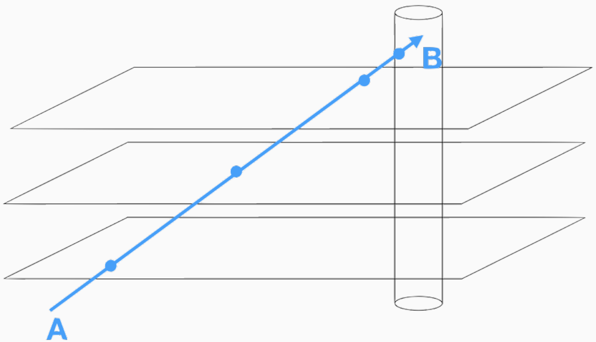
- Photon scattering points A and B
- Naive algorithm: Propagate photon small distance δx in each simulation step and randomize whether the photon will scatter in this step (easy to implement local properties)
- Faster algorithm: Randomize geometric distance to next scattering point and propagate from A to B in one simulation step
- Ice layers with different optical properties: Randomize number of scattering lengths between A and B as budget and calculate geometric distance by spending the budget over the ice layers
- New: Generalize budget algorithm to support cylinders and possibly other shapes with distinct scattering and absorption lengths and detection probabilities.

How does it work?



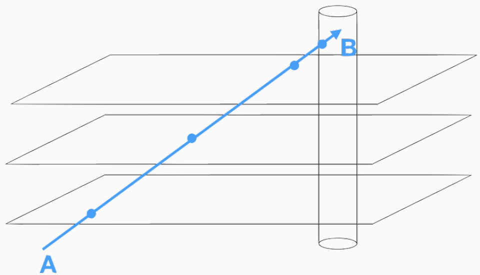
- Photon scattering points A and B
- Naive algorithm: Propagate photon small distance δx in each simulation step and randomize whether the photon will scatter in this step (easy to implement local properties)
- Faster algorithm: Randomize geometric distance to next scattering point and propagate from A to B in one simulation step
- Ice layers with different optical properties: Randomize number of scattering lengths between A and B as budget and calculate geometric distance by spending the budget over the ice layers
- New: Generalize budget algorithm to support cylinders and possibly other shapes with distinct scattering and absorption lengths and detection probabilities.

How does it work?



- Photon scattering points A and B
- Naive algorithm: Propagate photon small distance δx in each simulation step and randomize whether the photon will scatter in this step (easy to implement local properties)
- Faster algorithm: Randomize geometric distance to next scattering point and propagate from A to B in one simulation step
- Ice layers with different optical properties: Randomize number of scattering lengths between A and B as budget and calculate geometric distance by spending the budget over the ice layers
- New: Generalize budget algorithm to support cylinders and possibly other shapes with distinct scattering and absorption lengths and detection probabilities.

How approach B work?



Within standard clsim kernel

- Take current photon position
- and properties of ice layers
- Loop over layers
- Calculate physical distance to next interaction

$\Downarrow$

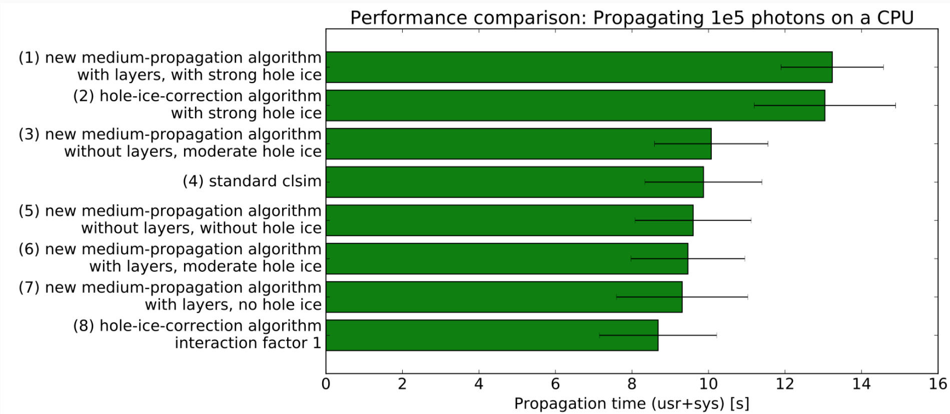
Replace with: `apply_propagation_through_media`

- Take current photon position
- Define arrays:
  `distances_to_medium_changes`
  `local_scattering_lengths`
  `local_absorption_lengths`
- Add ice layers to these arrays
- Add hole-ice and cable cylinders to these arrays
- Sort arrays by ascending distance from photon
- Loop over arrays
- and calculate physical distance to next interaction

Source:
https://github.com/fiedl/clsim/tree/sf/hole-ice-2018/resources/kernels/lib

Performance

Time measurement: Propagating 10^5 photons on CPU



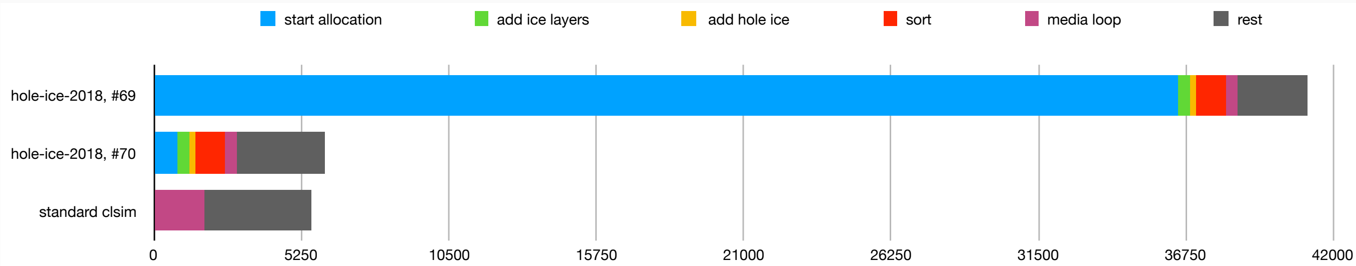

```
$ICESIM/env-shell.sh
cd $HOLE_ICE_STUDY/scripts/AngularAcceptance
time ./run.rb --distance=1.0 --number-of-runs=1 --number-of-parallel-runs=1 --cpu --angle=45 --plane-wave
↪  --number-of-photons=1e5
```

- Medium propagation features (hole ice, layers) have no measurable performance impact for scattering lengths comparable to bulk-ice scattering ($\lambda_s = 20\,\mathrm{m}$).
- Performance drop can be seen when lowering the scattering length, i.e. increasing the number of simulation steps ($\lambda_s = 3\,\mathrm{mm}$).

Source: https://github.com/fiedl/hole-ice-study/issues/49

Performance on GPU

Performance of one simulation step depends on optimizations:



Total performance depends on number of scatters:

Standard clsim with hole-ice approximation: 11 mins

New algorithm, no hole ice: 10 mins

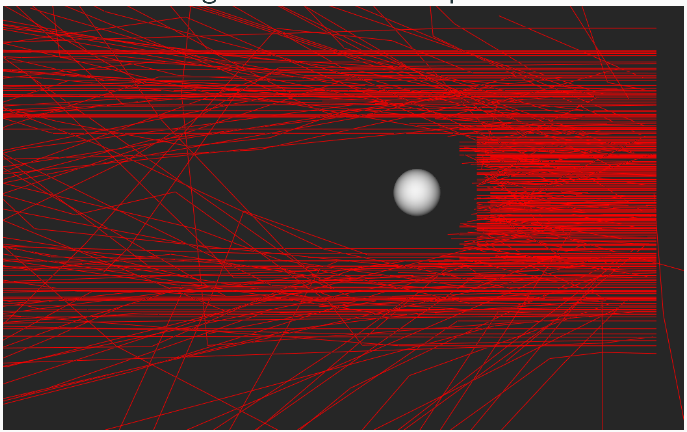
New algorithm, about H2 hole ice: 15 mins

Source: `https://github.com/fiedl/hole-ice-study/issues/69`

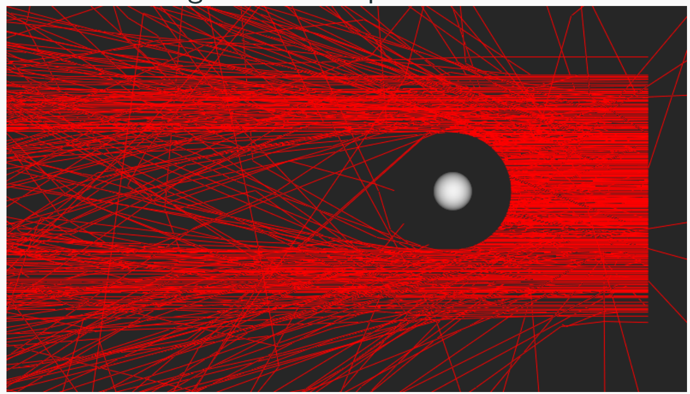
Coordinates-vs-vectors bug

Scenario: Instant absorption. Top view. Mathematics of intersection calculations and starting conditions are the same in both figures.

Before: Treating coordinates as separate variables

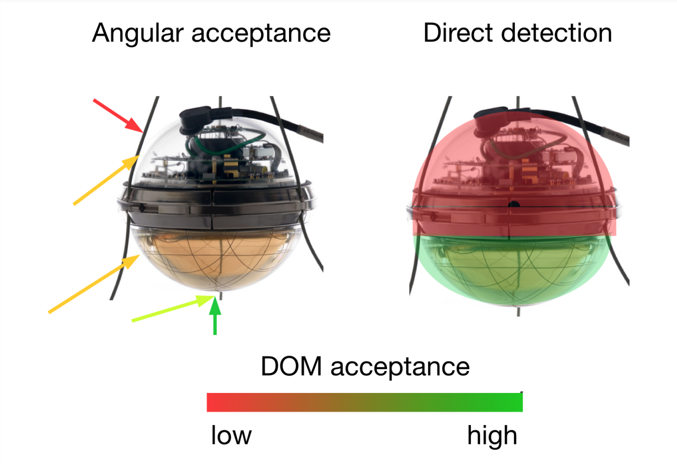


After: Treating vectors as opencl-native vectors



Source: `https://github.com/fiedl/hole-ice-study/issues/28`

Direct detection



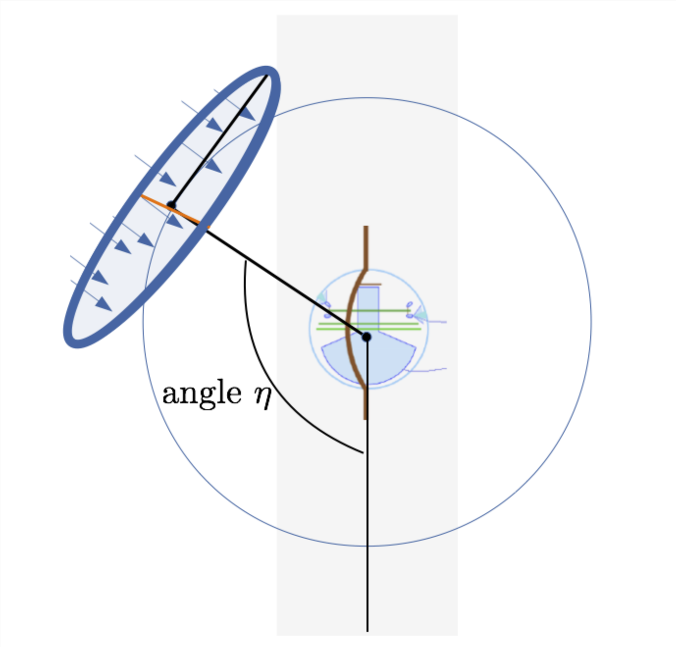
Source: Image: Martin Rongen, *Status and future of SpiceHD DARD*, 2017, Slide 17,
See also: `https://github.com/fiedl/hole-ice-study/issues/32`

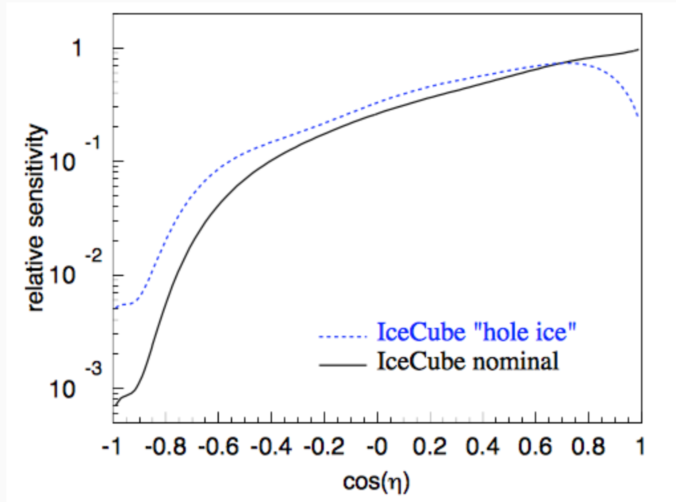
- The DOM looks downwards by design
- Currently, the hit position is not used when determining DOM acceptance, just the photon direction when hitting the DOM (*DOM angular acceptance*)
- Direct detection: Accept all hits below the waist band, reject all others
- Direct detection is easy with clsim
  - Hit position is known and guaranteed to be on the DOM sphere
  - Idea: Accept hits depending on z of the hit position
  - Patch is a couple of lines: fiedl/clsim@96a2e3f
- Still work to be done:
  - Implement a switch for direct detection vs. DOM angular acceptance

`https://github.com/fiedl/hole-ice-study/issues/32`

Angular acceptance

For each angle η, shoot photons onto the DOM and count hits.



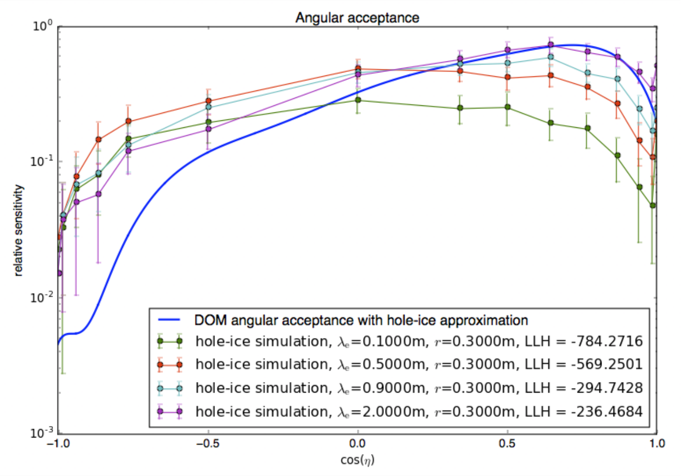


Angular acceptance *reference curves*. The nominal model is based on lab measurement, the hole ice curve on previous simulations.

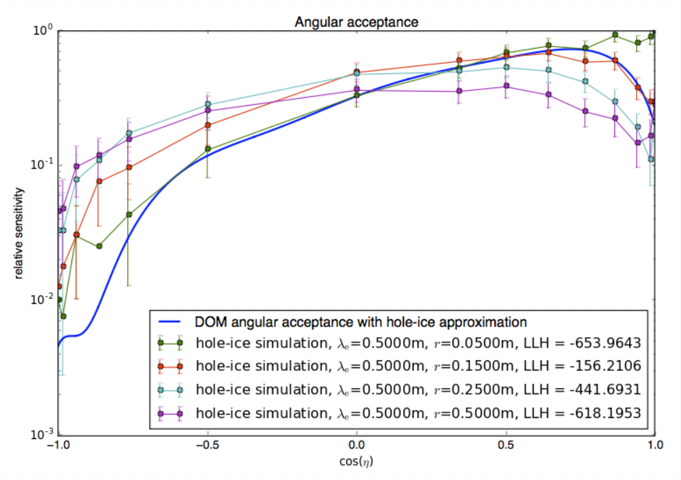
Source: Image: Martin Rongen, Calibration Call 2015-11-06, DARD Update, Slide 9
Plot: Measurement of South Pole ice transparency with the IceCube LED calibration system, 2013, figure 7. See also: `https://github.com/fiedl/hole-ice-study/issues/10`

Angular acceptance for different hole-ice parameters

Vary hole-ice scattering length:



Vary hole-ice radius:



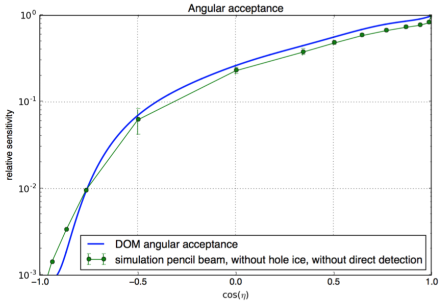
Systematics:

For direct detection + plane waves, increased number of photons for $\cos\eta < 0$.

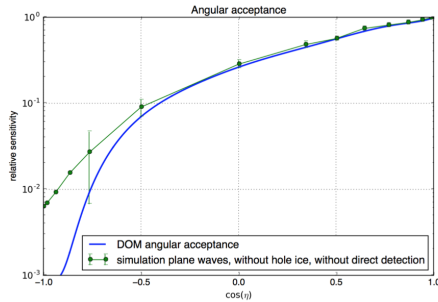
plane extent 1 m, starting distance 1 m

non-perfect bulk-ice properties

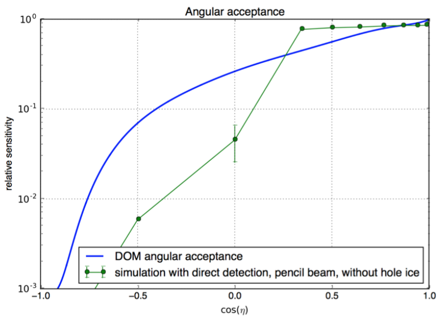
Angular acceptance: Sources and acceptance criteria



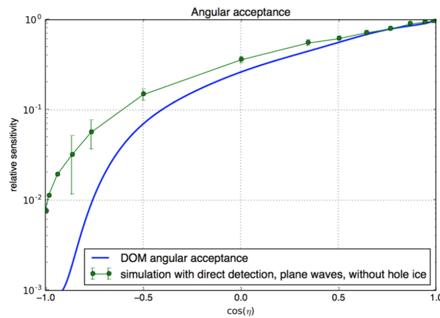
(a) pencil beams, a priori angular acceptance



(b) plane waves, a priori angular acceptance



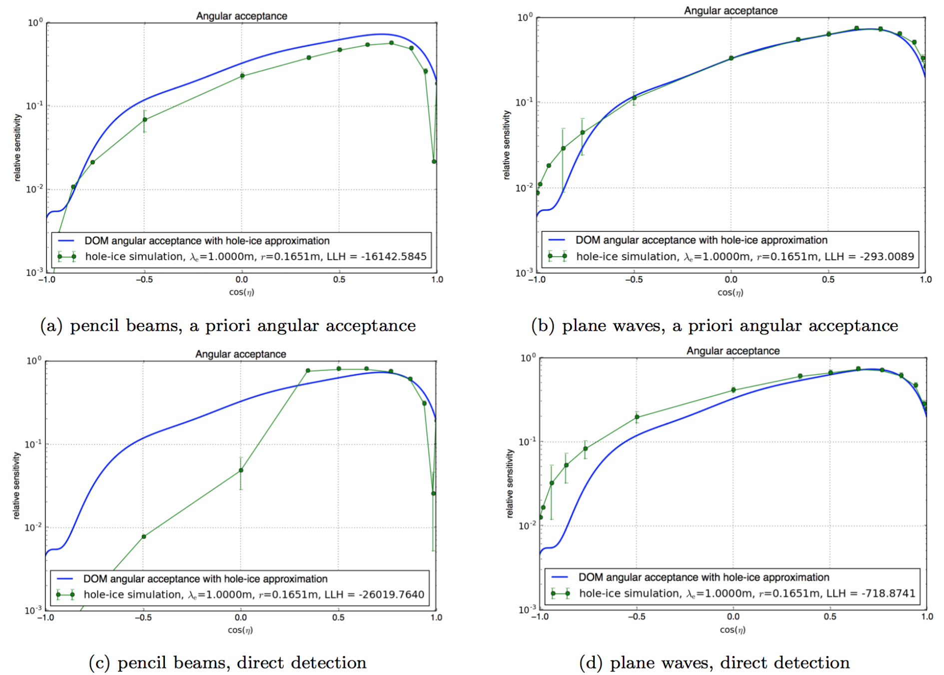
(c) pencil beams, direct detection



(d) plane waves, direct detection

Source: `https://github.com/fiedl/hole-ice-study/issues/98` and `https://github.com/fiedl/hole-ice-study/issues/99`

Angular acceptance: Sources and acceptance criteria



(a) pencil beams, a priori angular acceptance

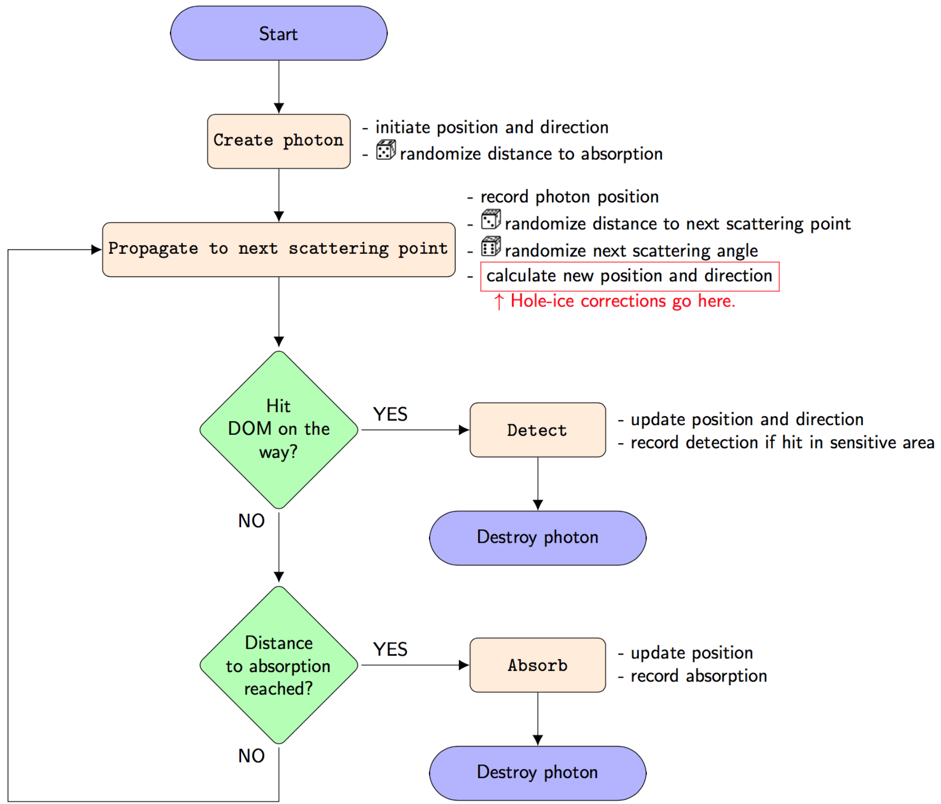
(b) plane waves, a priori angular acceptance

(c) pencil beams, direct detection

(d) plane waves, direct detection

Source: `https://github.com/fiedl/hole-ice-study/issues/98` and `https://github.com/fiedl/hole-ice-study/issues/99`

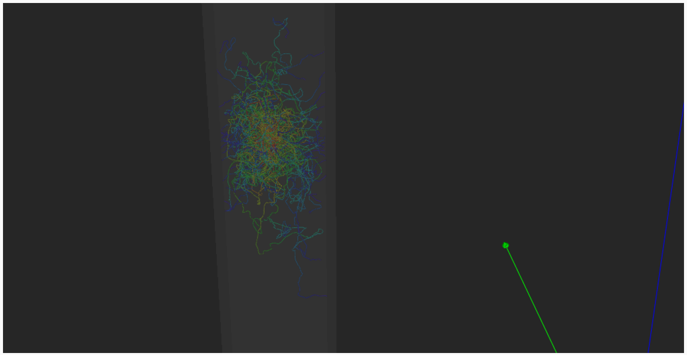
Simplified simulation-step flow chart



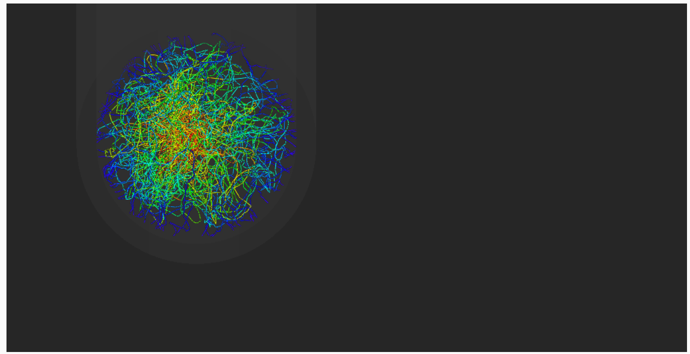
Source: `https://github.com/fiedl/hole-ice-study/issues/75`

Instant absorption with nested cylinders

The inner cylinder is configured for small scattering length, the outer cylinder for instant absorption.



With outer cylinder configured for instant absorption

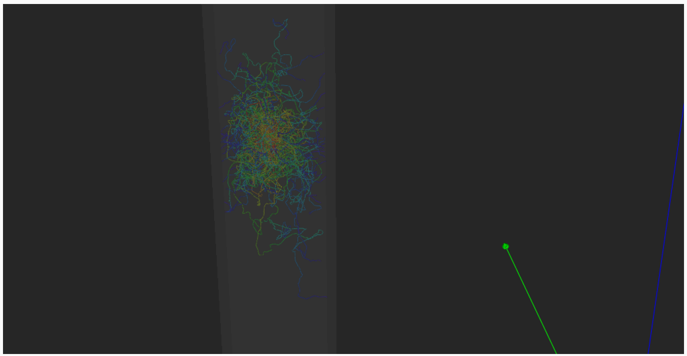


Top view

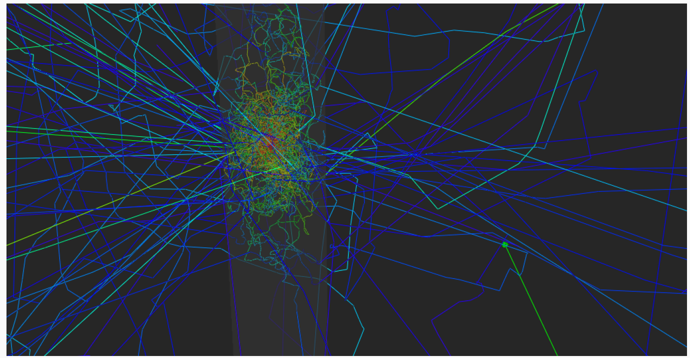
Source: `https://github.com/fiedl/hole-ice-study/issues/47`

Instant absorption with nested cylinders

The inner cylinder is configured for small scattering length, the outer cylinder for instant absorption.



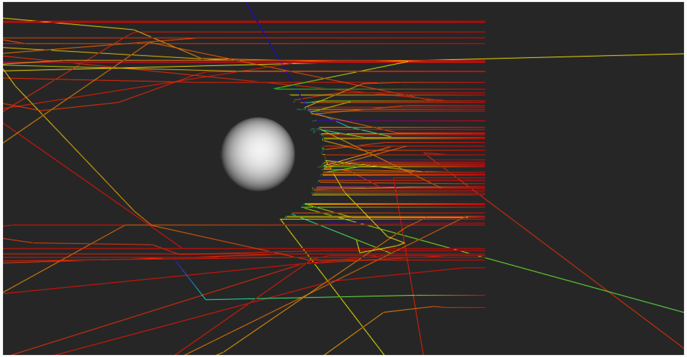
With outer cylinder configured for instant absorption



Without the outer cylinder

Source: `https://github.com/fiedl/hole-ice-study/issues/47`

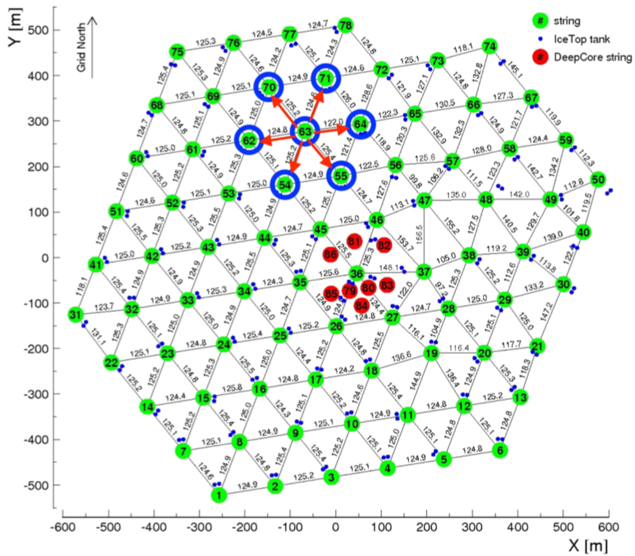
Scattering example

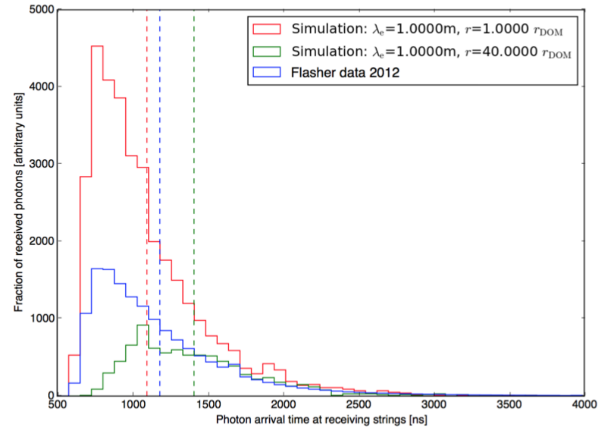


- Hole-ice absorption length: about 5 cm
- Hole-ice scattering length factor: 0.001

```
$ICESIM/env-shell.sh
cd $HOLE_ICE_STUDY/scripts/FiringRange
./run.rb --scattering-factor=0.001 --absorption-factor=0.00033 --distance=1.0
↪  --number-of-photons=100 --number-of-runs=1 --number-of-parallel-runs=1
↪  --save-photon-paths --cpu --plane-wave
steamshovel tmp/propagated_photons.i3
```

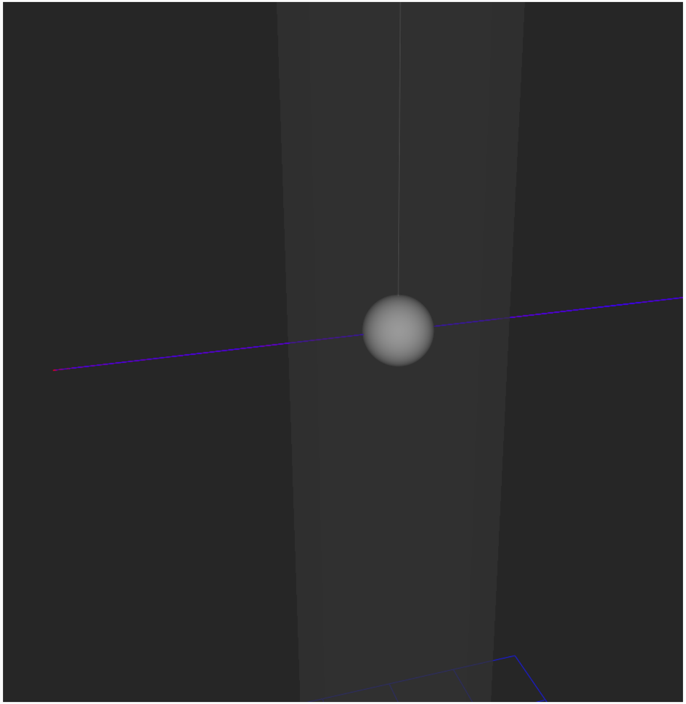
Cross checks: Arrival-time distributions

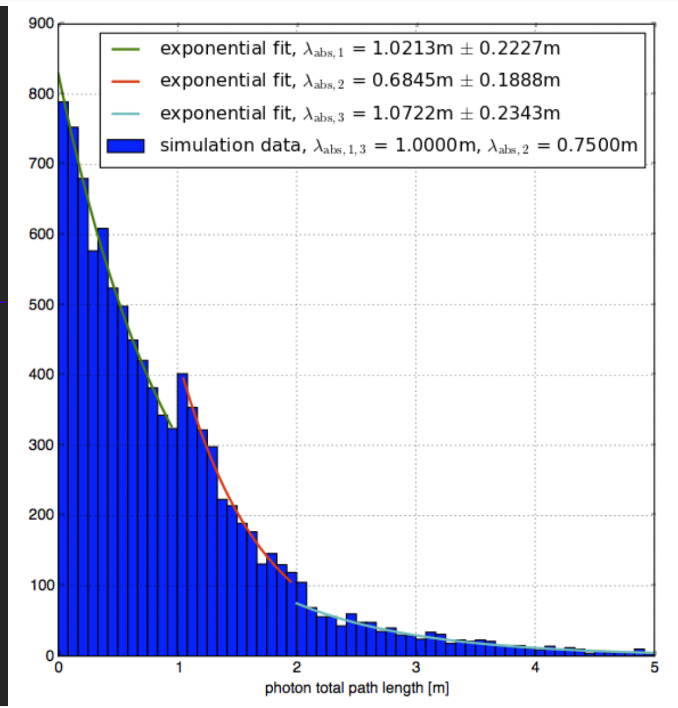




Source: `https://github.com/fiedl/hole-ice-study/issues/91`. Image based on `https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg`.

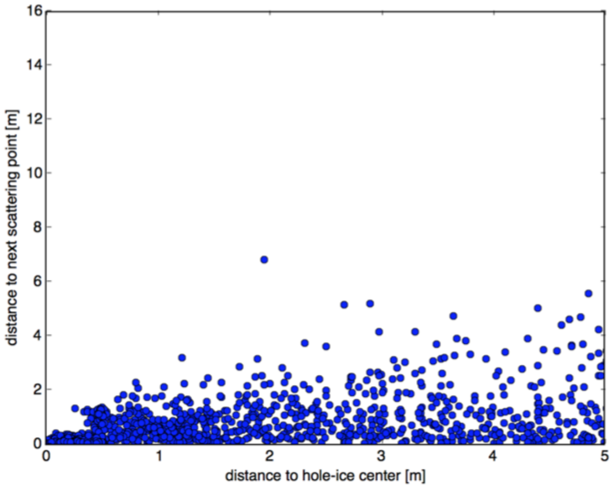
Cross checks: Path-length distributions



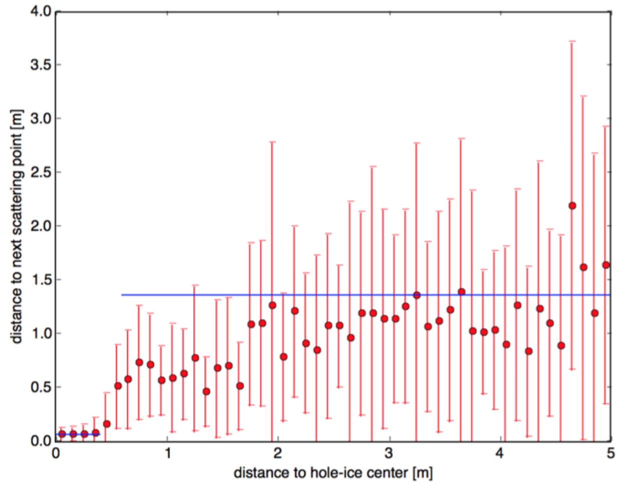


Source: `https://github.com/fiedl/hole-ice-study/issues/66`

Cross checks: Distance to next scattering point vs. dst. from hole-ice center



(a) All data points

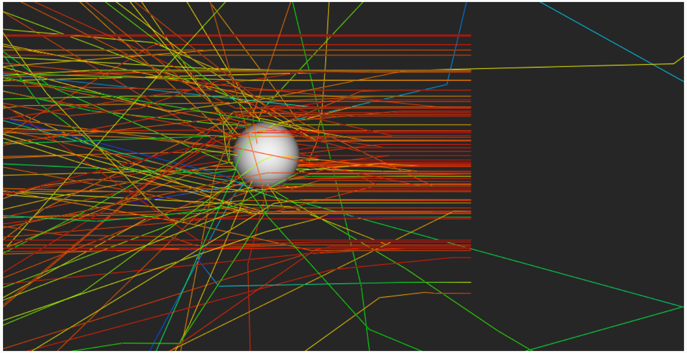


(b) Averaged for bins of a width of 10 cm

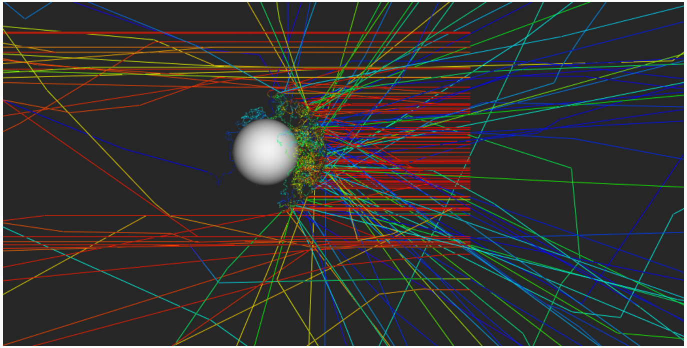
Source: `https://github.com/fiedl/hole-ice-study/issues/71`

Trying out different hole-ice scattering lengths

The exact optical properties of the hole ice are unknown. With the simulation, one can try out different properties, e.g. scattering length.



Scattering length $\lambda_{\text{sca,hole-ice}} = 10^{-1}\,\lambda_{\text{sca,bulk}}$.
Absorption length $\lambda_{\text{abs,hole-ice}} = \lambda_{\text{sca,bulk}}$.



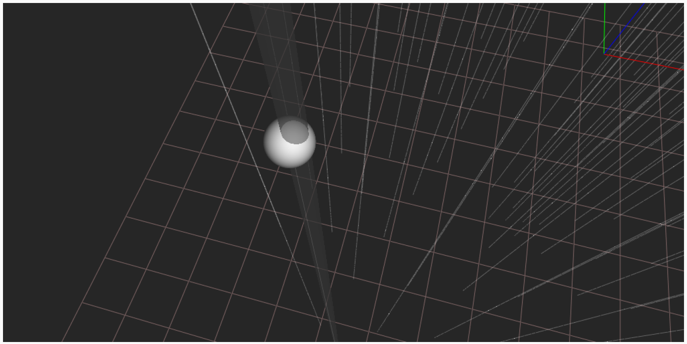
Scattering length $\lambda_{\text{sca,hole-ice}} = 10^{-3}\,\lambda_{\text{sca,bulk}}$.
Absorption length $\lambda_{\text{abs,hole-ice}} = \lambda_{\text{sca,bulk}}$.

Animation on youtube: `https://youtu.be/BhJ6F3B-I1s`

View from top onto a detector module within a hole-ice cylinder. Colors indicate simulation steps, i.e. number of scatterings relative to the total number until absorption. Red: Photon just created, blue: Photon about to be absorbed.

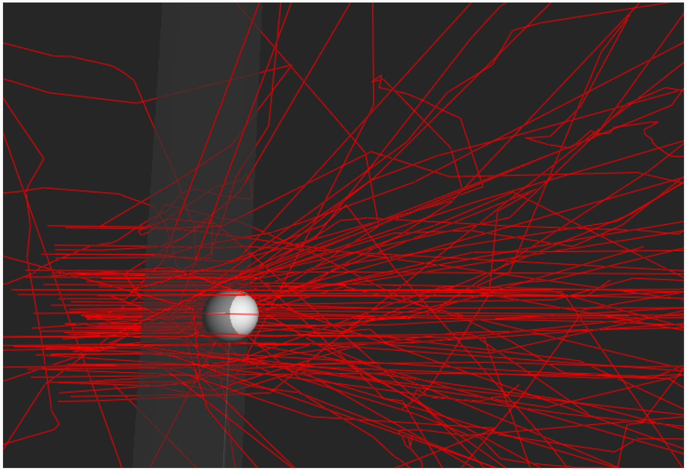
Source: `https://github.com/fiedl/hole-ice-study/issues/39`

Separate hole-ice cylinder positions



- Each string can have its own hole-ice cylinder configuration
  - cylinder position
  - cylinder radius
  - scattering length within cylinder
  - absorption length within cylinder
  - DOM positions — DOMs may not be perfectly centred relative to the hole ice

Asymmetry example



For angle $\eta = \pi/2$, shoot photons from planes onto the DOM and count hits.

Hole-ice radius: 30 cm

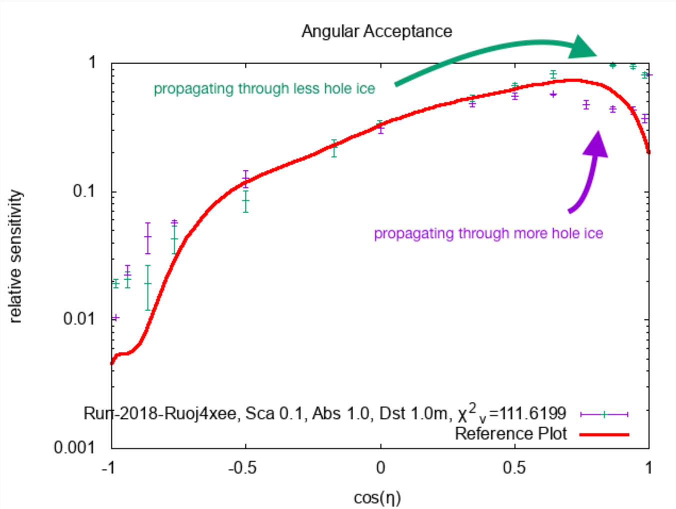
$$\lambda_{\text{sca,hole-ice}} = \frac{1}{10} \quad \lambda_{\text{sca,bulk}}$$
$$\lambda_{\text{abs,hole-ice}} = \quad \lambda_{\text{sca,bulk}}$$

The **hole-ice is shifted in x-direction against the DOM** position by 20 cm.

```
$ICESIM/env-shell.sh
cd $HOLE_ICE_STUDY/scripts/AngularAcceptance
./run.rb --scattering-factor=0.1 --absorption-factor=1.0
↪  --distance=1.0 --plane-wave --number-of-photons=1e2
↪  --cylinder-shift=0.2 --save-photon-paths --cpu
steamshovel tmp/propagated_photons.i3
```

Source: https://github.com/fiedl/hole-ice-study#asymmetry-example, https://github.com/fiedl/hole-ice-study/issues/8

Asymmetry example



For each angle $\eta \in [0; 2\pi[$, shoot photons from planes onto the DOM and count hits.

Hole-ice radius: 30 cm

$$\lambda_{\text{sca,hole-ice}} = \frac{1}{10} \quad \lambda_{\text{sca,bulk}}$$

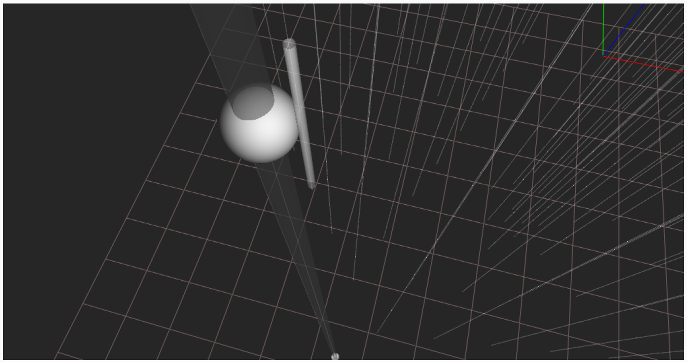
$$\lambda_{\text{abs,hole-ice}} = \quad \lambda_{\text{sca,bulk}}$$

The **hole-ice is shifted in x-direction against the DOM** position by 20 cm.

```
$ICESIM/env-shell.sh
cd $HOLE_ICE_STUDY/scripts/AngularAcceptance
./run.rb --scattering-factor=0.1 --absorption-factor=1.0 --distance=1.0
↪  --plane-wave --number-of-photons=1e5
↪  --angles=0,10,20,30,40,50,60,70,90,120,140,150,160,170,190,200,210,220,
↪  240,260,270,290,300,310,320,330,340,350 --number-of-runs=2
↪  --number-of-parallel-runs=2 --cylinder-shift=0.2
open results/current/plot_with_reference.png
```

Source: `https://github.com/fiedl/hole-ice-study#asymmetry-example`, `https://github.com/fiedl/hole-ice-study/issues/8`

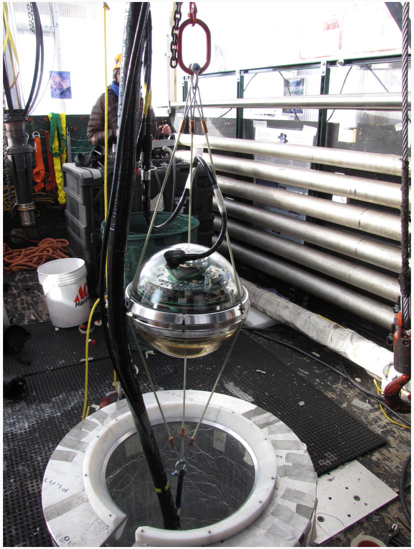
Cable shadows

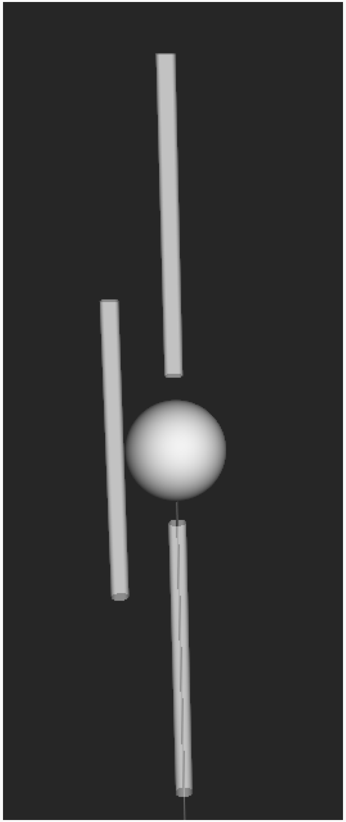


- Cables can be modelled as separate cylinders
  - for each DOM separate position
  - 1 m height
  - configured for instant absorption
- This image:
  - DOM radius: 16.5 cm
  - bubble-column radius: 8.0 cm
  - cable radius: 2.0 cm

Source: `https://github.com/fiedl/hole-ice-study/issues/35`

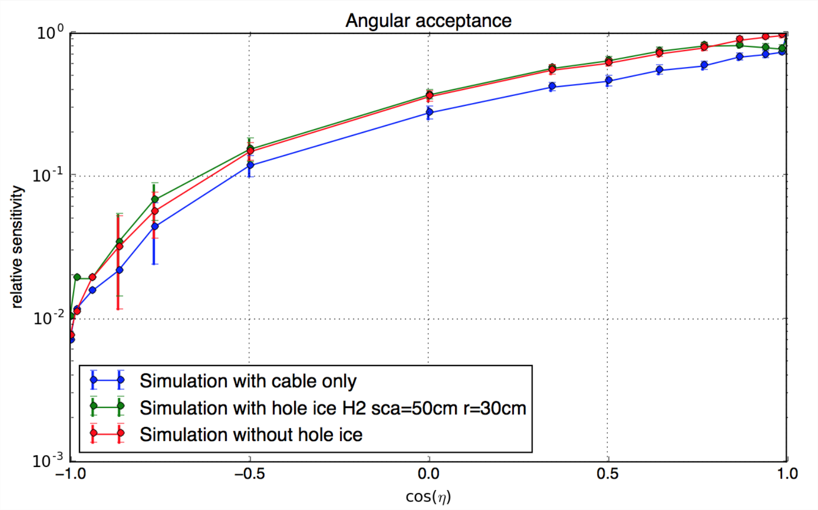
Direct cable simulation: Angular acceptance

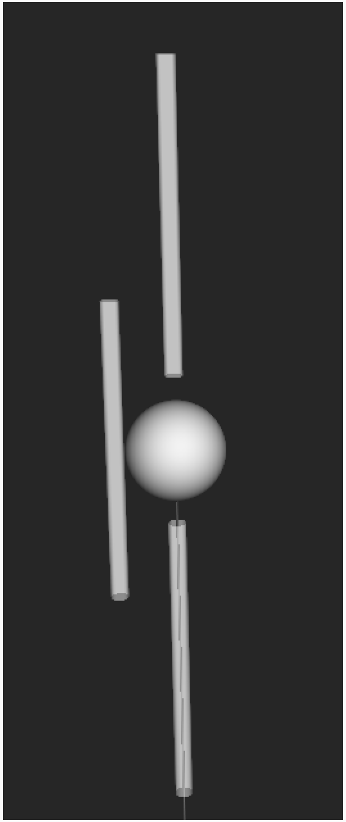




Source: `https://github.com/fiedl/hole-ice-study/issues/101`. Images: `https://icecube.wisc.edu/gallery/view/153`, `https://gallery.icecube.wisc.edu/internal/v/GraphicRe/graphics/arraygraphics2011/sketchup/DOMCloseUp.jpg.html`

Direct cable simulation: Angular acceptance

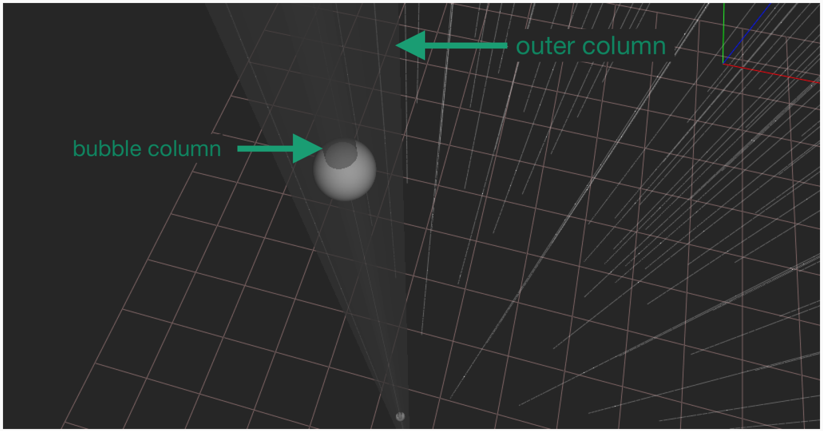




The azimuthal starting angle is such that the cable shadow is maximal.

Source: `https://github.com/fiedl/hole-ice-study/issues/101`.

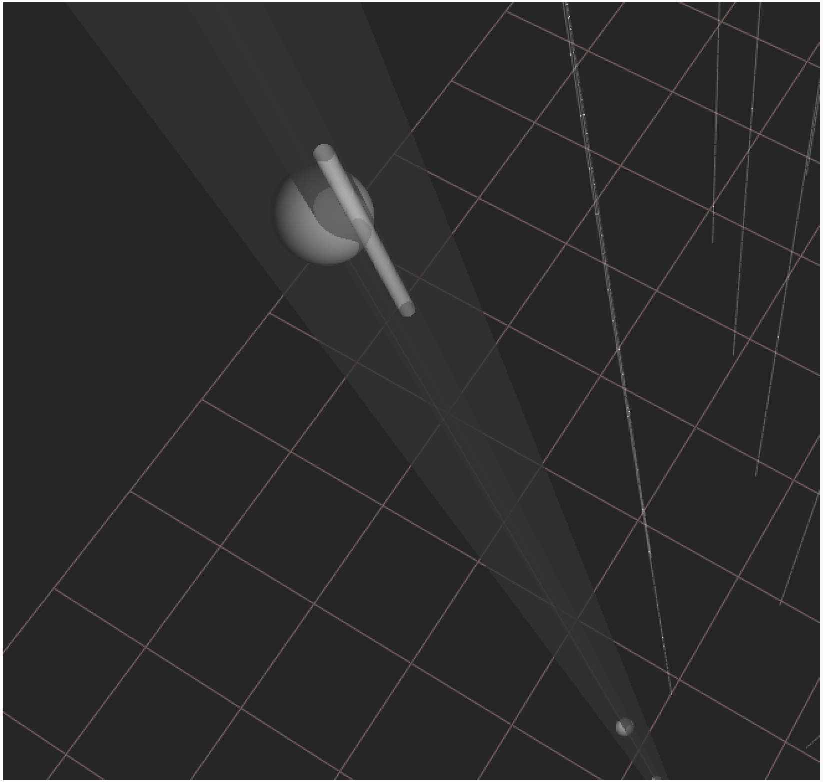
Nested hole-ice cylinders



- Hole-ice cylinders can be nested
    - for each string separate positions
    - for each string and each column separate radii
- This image:
    - DOM radius: 16.5 cm
    - bubble-column radius: 8.0 cm
    - outer-column radius: 30.0 cm

Source: `https://github.com/fiedl/hole-ice-study/issues/7`

Realistic simulation scenario

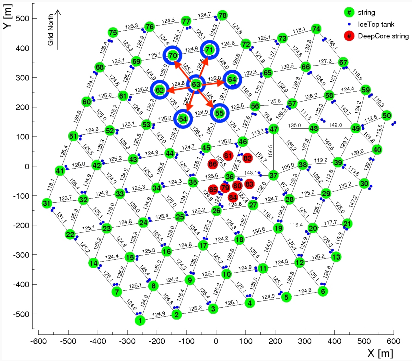


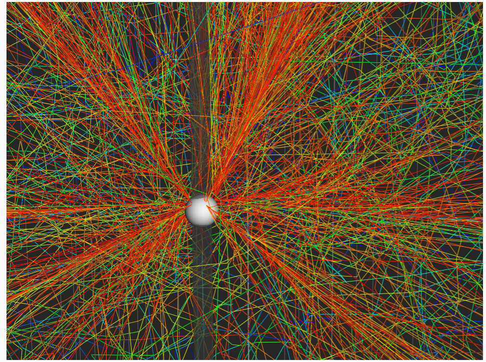
- DOM: radius 16.5 cm, shifted by 12.0 cm against the center of the bore hole
- bubble column: radius 8.0 cm
- drill-hole column: radius 30.0 cm
- cable: radius 3.0 cm, placed next to the DOM, partially within the bubble column

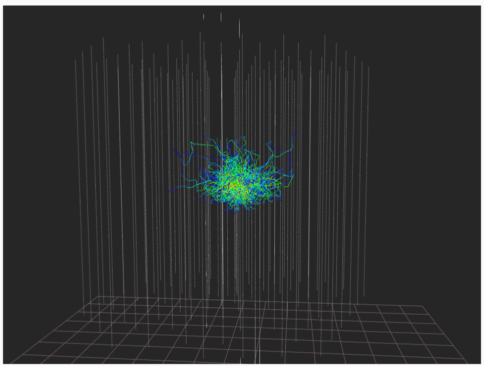
See also: `https://github.com/fiedl/hole-ice-study/issues/110`

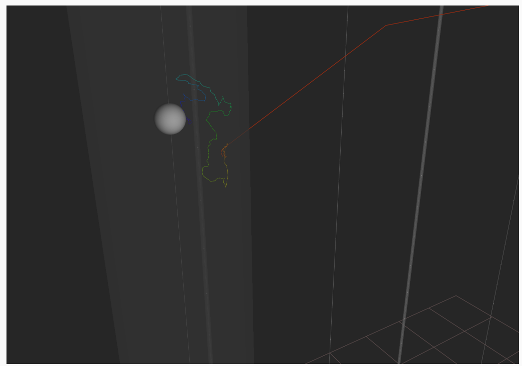
Flasher-simulation example

Calibration: Find out the properties of the hole ice by comparing simulations with differnt properties to data of IceCube's LED-flasher-calibration system.



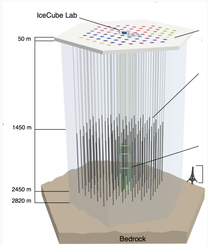


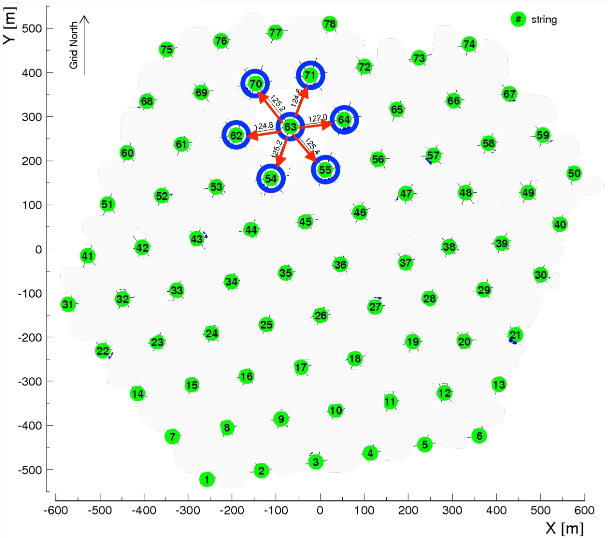


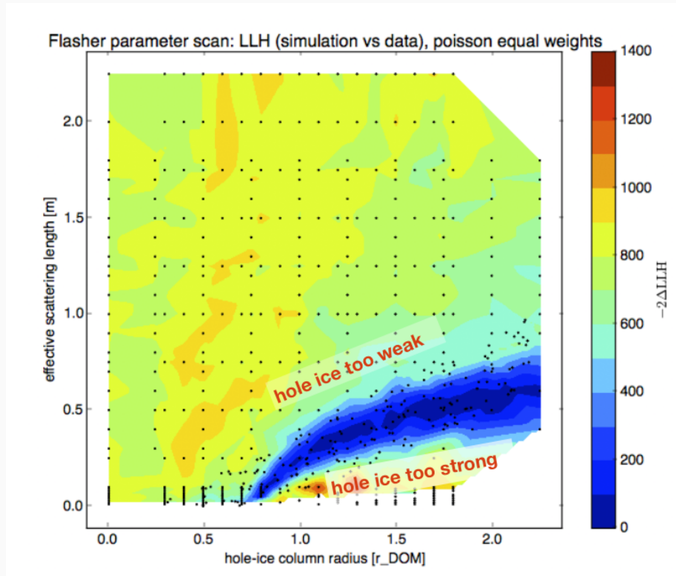


See `https://github.com/fiedl/hole-ice-study/issues/107`

Example scan for best hole-ice parameters based on calibration data

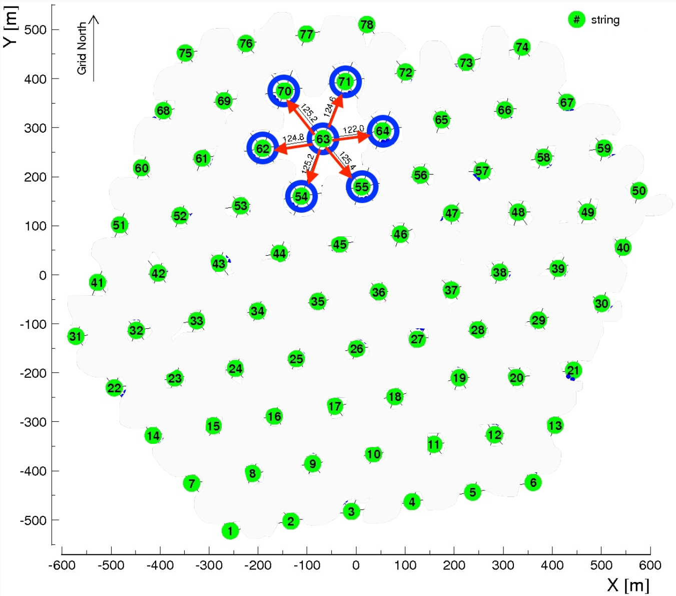






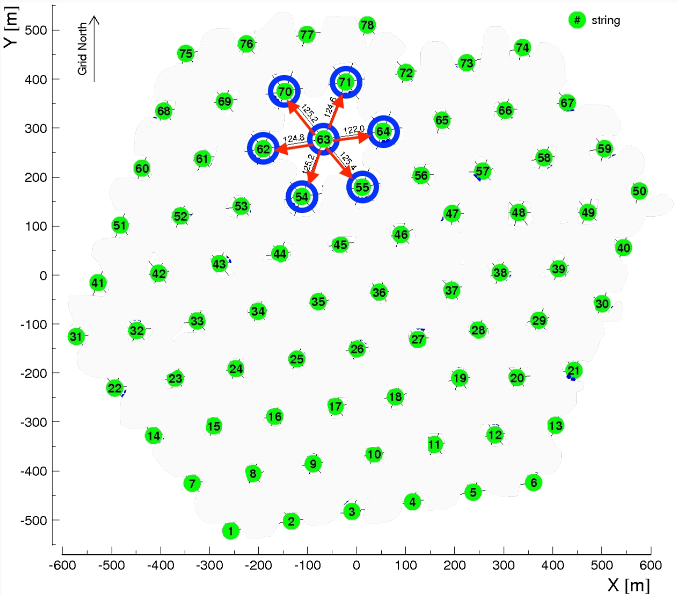
Source: `https://github.com/fiedl/hole-ice-study/issues/59`. Footprint based on `https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg`. Image: Aartsen et al., The IceCube Neutrino Observatory: Instrumentation and online systems, 2017.

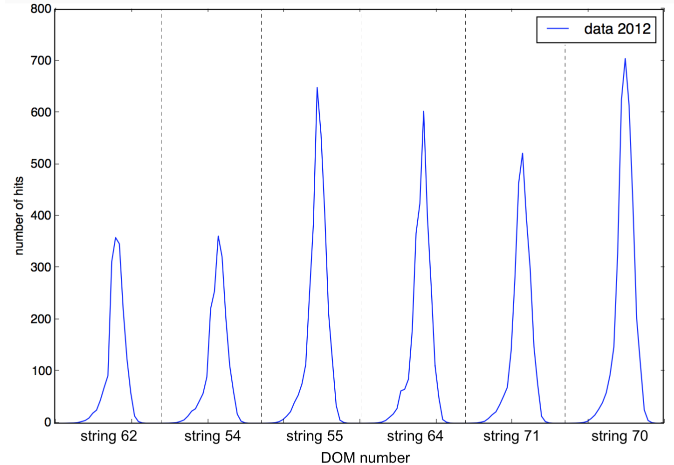
Early results: Calibration data suggest asymmetric shielding by hole ice



Source: `https://github.com/fiedl/hole-ice-study/issues/97`. Image based on `https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg`.

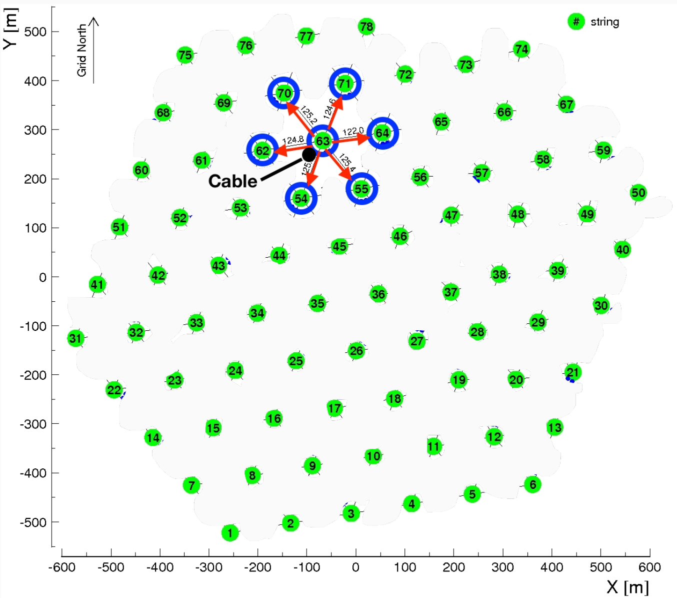
Early results: Calibration data suggest asymmetric shielding by hole ice

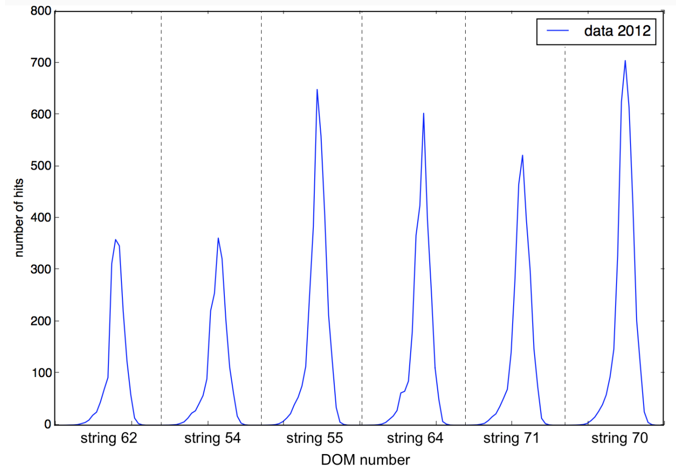




Source: `https://github.com/fiedl/hole-ice-study/issues/97`. Image based on `https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg`.

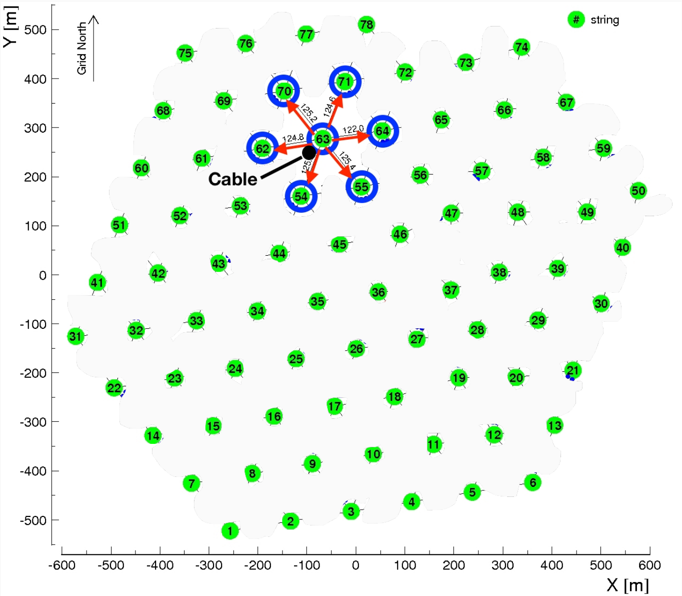
Early results: Calibration data suggest asymmetric shielding by hole ice

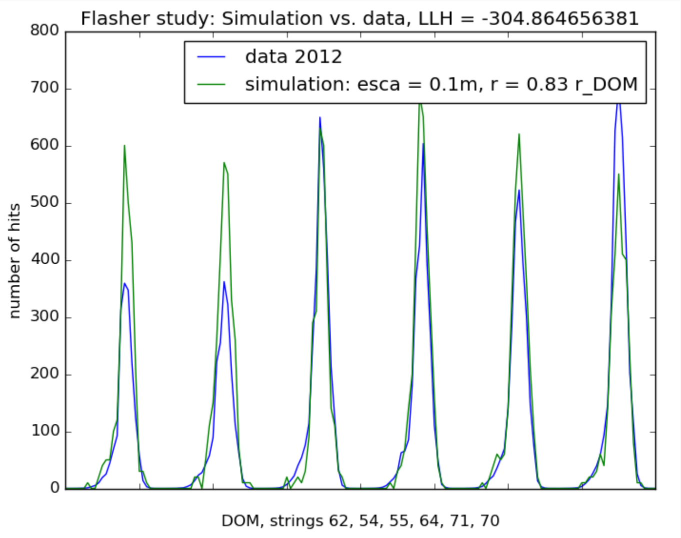




Source: `https://github.com/fiedl/hole-ice-study/issues/97`. Image based on `https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg`.

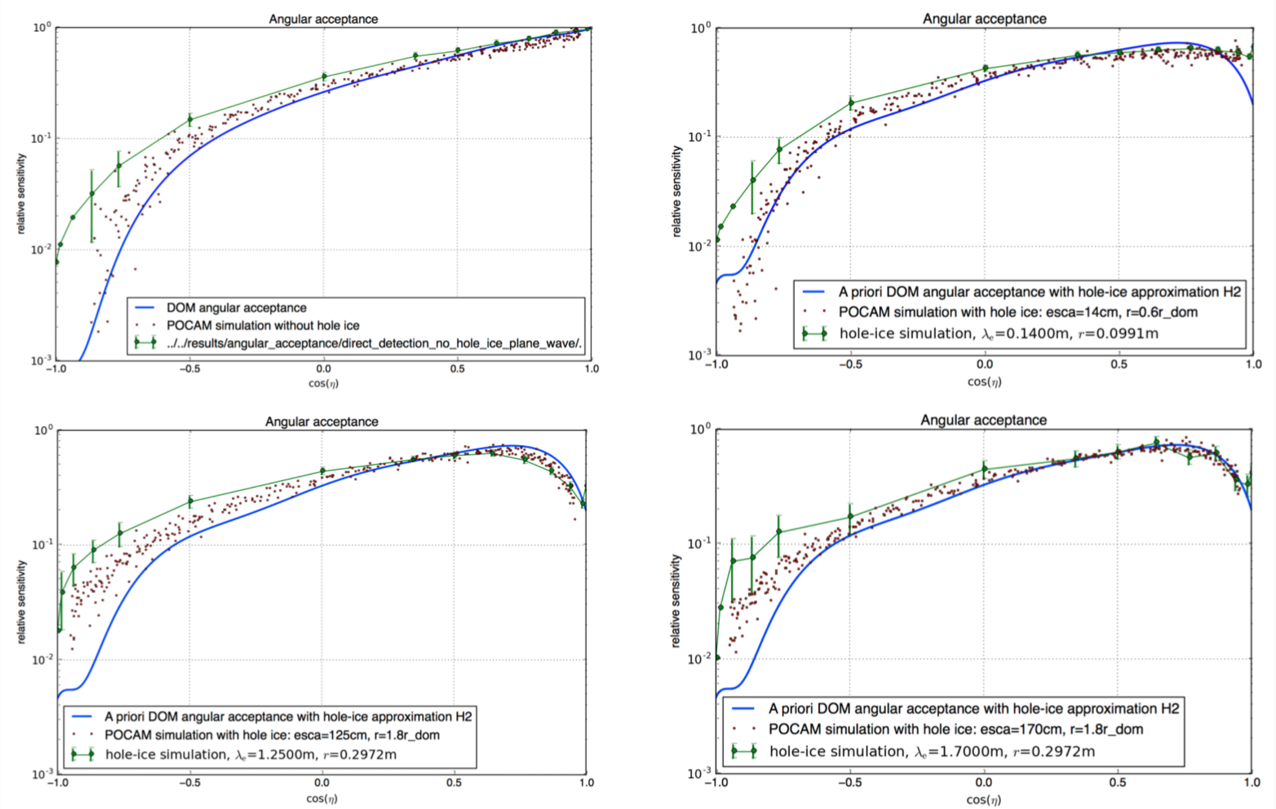
Early results: Calibration data suggest asymmetric shielding by hole ice





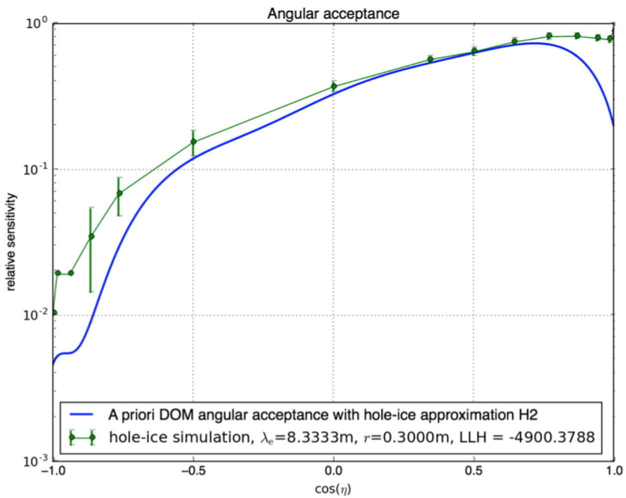
Source: `https://github.com/fiedl/hole-ice-study/issues/97`. Image based on `https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg`.

Comparison to ppc simulation

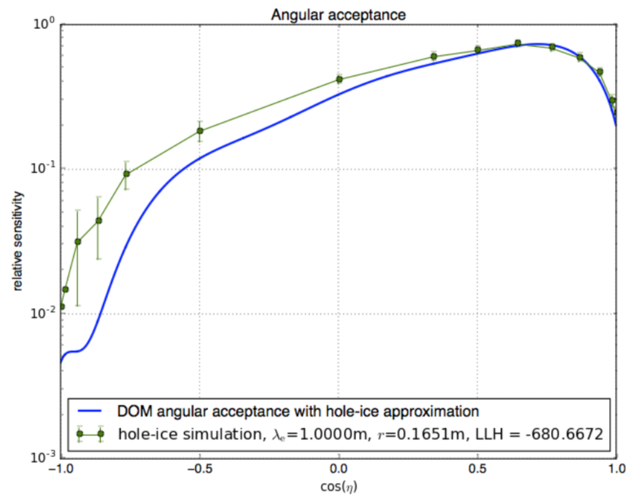


Source: `https://github.com/fiedl/hole-ice-study/issues/4`, POCAM ppc data source: Resconi, Rongen, Krings: The Precision Optical CAlibration Module for IceCube-Gen2: First Prototype, 2017.

Comparison to H2 hole-ice model



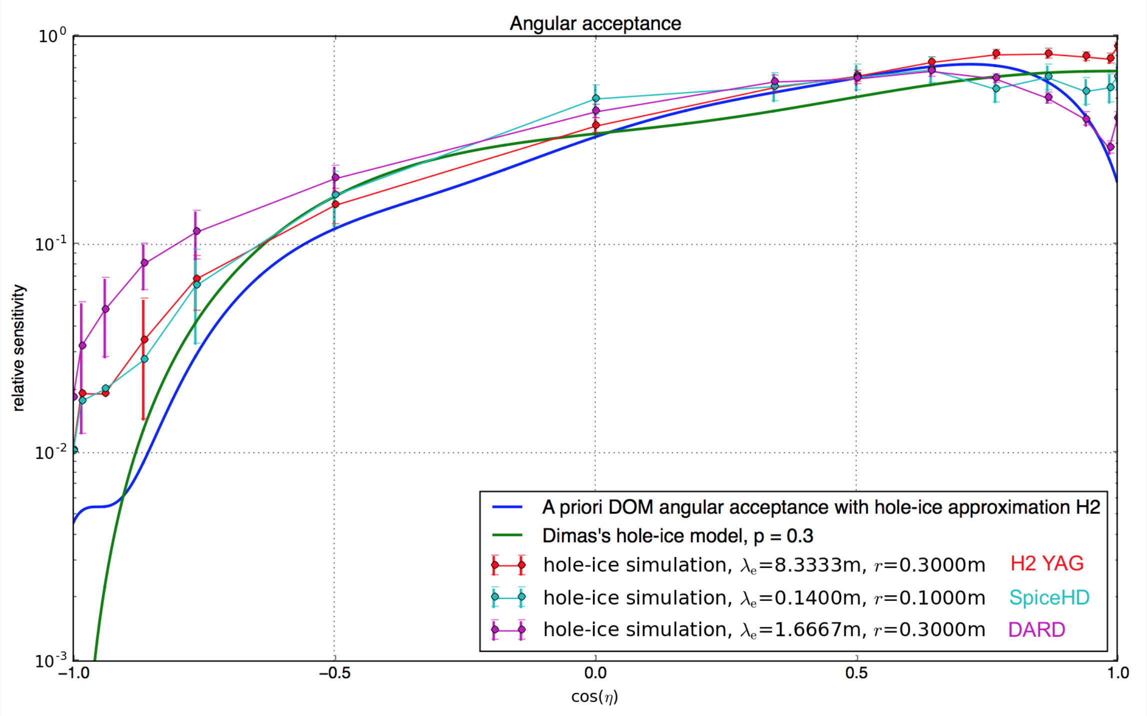
(a) CLSIM simulation with H2 hole-ice parameters: $r = 30\,\mathrm{cm}, \lambda^{\mathrm{H}}_{\mathrm{sca}} = 50\,\mathrm{cm}, \lambda^{\mathrm{H}}_{\mathrm{sca}} = 8.33\,\mathrm{m}$



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

Source: `https://github.com/fiedl/hole-ice-study/issues/80`. H2 curve source: IceCube Collaboration et al. Measurement of South Pole ice transparency with the IceCube LED calibration system. 2013. H2 parameter source: Albrecht Karle. Hole Ice Studies with YAG. `http://icecube.berkeley.edu/kurt/interstring/hole-ice/yak.html`. 1998.

Comparison of parameters from calibration measurements



Source: `https://github.com/fiedl/hole-ice-study/issues/104` H2 YAG: `https://github.com/fiedl/hole-ice-study/issues/80`. Karle, Hole Ice Studies with YAG, `http://icecube.berkeley.edu/kurt/interstring/hole-ice/yak.html`, 1998. SpiceHD: `https://github.com/fiedl/hole-ice-study/issues/87`. Rongen, Status and future of SpiceHD and DARD, Calibration Workshop August 2017. DARD: `https://github.com/fiedl/hole-ice-study/issues/105`. Rongen, Measuring the optical properties of IceCube drill holes, 2016. Rongen, DARD Update, Calibration Call 2015-11-06.