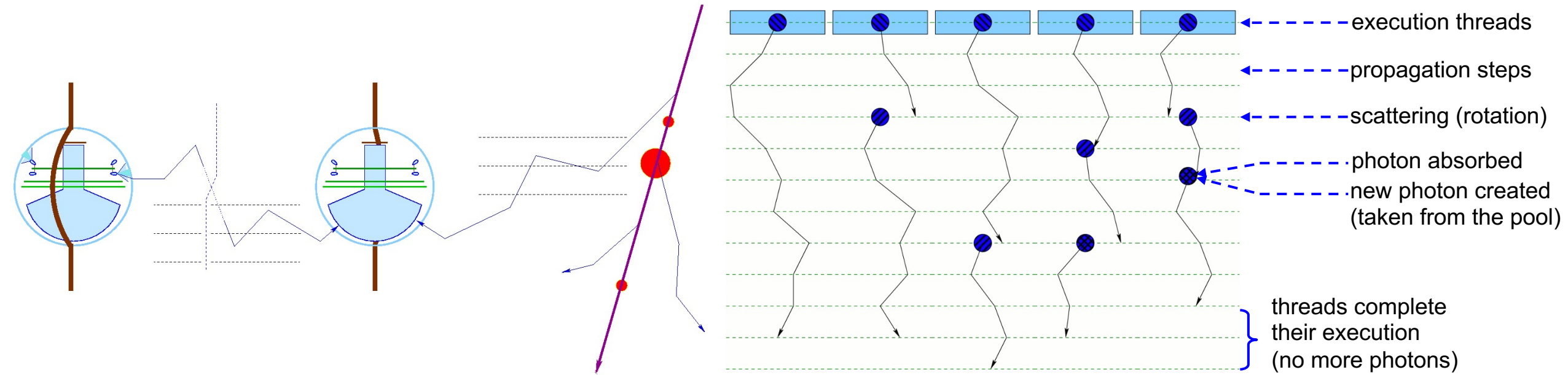


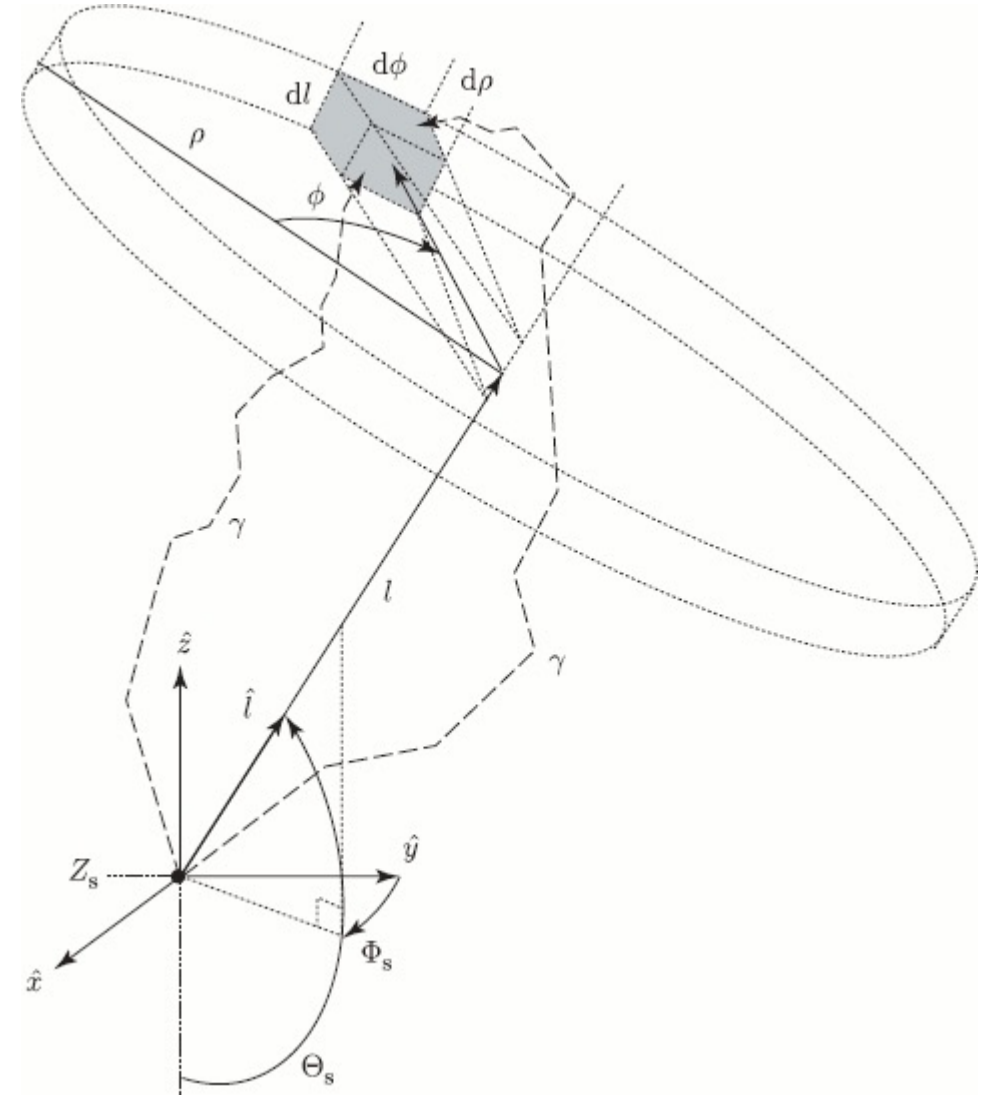
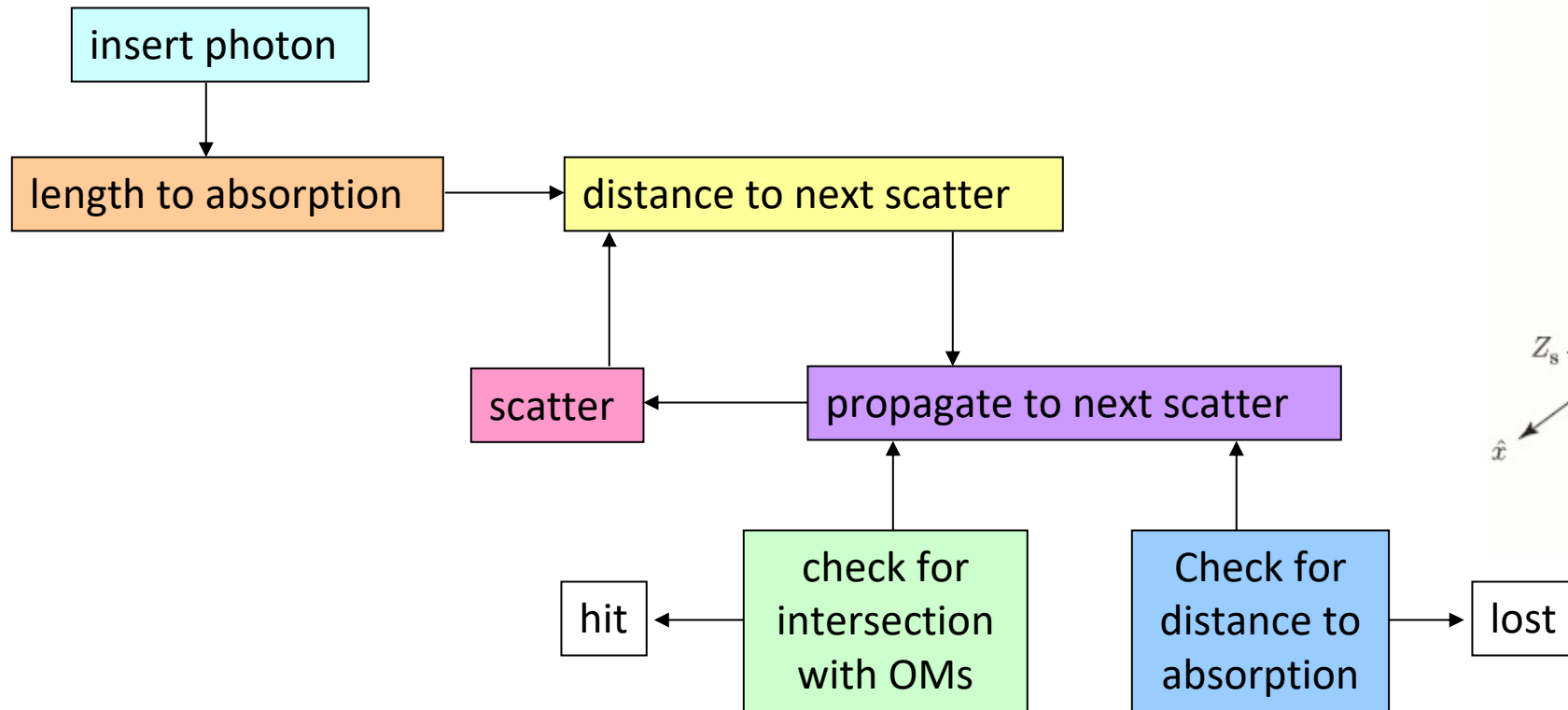
# Photon Propagation Code (PPC): overview



*D. Chirkin, UW-Madison*

# PPC vs. Photonics

propagate photons directly when needed



The code lives on main branch on IceCube/icetray git.

Both OPenCL (ppc/private/ppc/ocl/) and CUDA (ppc/private/ppc/gpu/) versions of code are available, presenting exactly the same functionality

- The CUDA version is 15-35% faster than the OpenCL version.
- The CUDA version can also be compiled to run on single CPU core
- The OpenCL version can be run on CPUs taking over all available CPU cores
- By default OpenCL version is compiled. Need to bring back cmake functionality that allowed to choose OpenCL or CUDA

Only one ice model is maintained in resources/ice. This is usually the latest “bleeding-edge” model.

- As the README file says, do not use for production as this ice model is often updated
- Pick an ice model from ice-models project instead

Full documentation is at <https://github.com/icecube/icetray/blob/main/ppc/resources/docs/index.rst>.

- Recently updated (as of 09/2021)

PPC project also hosts llh/DirectFit (flasher-fitting and cascade/track reconstruction software) and BFR (detailed photon propagation through birefringent crystals)

# Configuration and ice model support

- All icecal-approved ice models and many others are supported. This includes BFR-v1 and BFR-v2 (birefringence-based anisotropy description), EMRM (absorption-based), all SPICE 3.X (scattering-based) and other legacy models (Lea, Mie, 2.X, 1, AHA, WHAM, etc.) It is possible to configure and combine multiple anisotropy effects simultaneously.
- Split configuration directory is possible (feature added by Juan Carlos): PPCTABLESDIR, ICEMODELDIR, TILTMODELDIR directories, and PPCHOLEICE file. Falls back to PPCTABLESDIR (or ICEMODELDIR for TILTMODELDIR) if unset.
- Precise simulation of IceCube flasher LEDs is available (includes lab-measured angular and timing distributions, positioning inside DOMs, photon blocking by internal structure (some), harness belt and cable).
- Can be run outside icetray. This is used for ice model fitting and llh/DirectFit event reconstruction

# Performance

PPC employs a very efficient method for photon packing across multiple events, copying to GPU, and unpacking there. Best possible efficiency is achieved for low, high, or mixed-energy events

- uses photon-emitting segments, which include a “number of photons”
- uses very little bandwidth to transmit the photon data to the GPU
- this is faster than transmitting an array of individual photons
- merges flasher/particle interface, flasher configuration is passed in the “photon segment”

Can be run on multiple GPUs, homogeneous or not. Balances load proportionally to the amount of resources reported by the driver, can be re-balanced/tuned at run time. CUDA version supports disabling of individual MultiProcessors on older GPUs (in case an MP is causing errors)

Excellent performance in the GPU kernel.

- GPU loading (the indicator of thread divergence) is 24.7/32.
- cf. clsim: 13.7/32 before the fall of 2020 optimization, and 17.0/32 after.

# Workarounds and quirks

Have my own wait code that avoids spinning CPU when waiting on GPU

- OpenCL functionality provided by *clFinish(cq)* is unreliable.

OpenCL kernel is configured with driver-recommended work-group size, and then incrementally tests larger work-group size to determine the largest possible (which is most efficient). For some reason driver recommendation often falls short (e.g., 256 when 1024 will work).

- OpenCL functionality provided by *clGetKernelWorkGroupInfo()* is unreliable.

A number of GCD parameters (geometry, RDEs, OMType etc.) can be overridden with external files

- A warning will be printed on stderr

PPC is configured with a combination of module parameters, environmental variables, and tables

- Some parameters are set in multiple ways
- Consult documentation when making changes

# Supported features

All ice models are supported including optional depth-dependent anisotropy of SPICE 3.2.2 and SPICE BFR

DOM axis tilt, rotation, and cable position can be configured

Direct hole ice column simulation with radius,  $\lambda_e$ ,  $g=\langle\cos(\theta)\rangle$  and  $f_{SL}$  as configurable parameters

Geant interface (via CLSteps) is fully supported

Snowstorm systematics simulation is supported

Ice layer thinning/thickening treatment due to ice layer tilt can be enabled

New 2d ice layer tilt parameterization, including hexagon boundary extrapolation

Photons landing on sensor surface can be saved into both the MCTree and/or I3PhotonSeriesMap

Upgrade/Gen2 elongated/multi-PMT sensors are also supported natively

Monopole propagation including delayed ice luminescence is maintained by BSM group

# Running ppc

```
#muongun
time python $I3_SRC/simprod-scripts/resources/scripts/muongun.py --UseGSLRNG --emin 1e3 --emax 1\
e6 --nevents 100000 --outputfile muons.i3 --gcdfile /cvmfs/icecube.opensciencegrid.org/data/GCD/\
GeoCalibDetectorStatus_AVG_55697-57531_PASS2_SPE_withScaledNoise.i3.gz --no-propagate-photons
```

```
#clsim
time python $I3_SRC/simprod-scripts/resources/scripts/clsim.py --UseGSLRNG --inputfile muons.i3\
--outputfile mcpe-clsim.i3 --gcdfile /cvmfs/icecube.opensciencegrid.org/data/GCD/GeoCalibDetec\
torStatus_AVG_55697-57531_PASS2_SPE_withScaledNoise.i3.gz --no-PropagateMuons --UseGPUs --no-Run\
MPHitFilter
```

```
real    54m42.949s
user    40m56.495s
sys     13m43.625s
```

CLsim

```
#ppc OpenCL
time python $I3_SRC/simprod-scripts/resources/scripts/ppc.py --UseGSLRNG --inputfile muons.i3 -\
-outputfile mcpe-ppc-opencl.i3 --gcdfile /cvmfs/icecube.opensciencegrid.org/data/GCD/GeoCalibDe\
tectorStatus_AVG_55697-57531_PASS2_SPE_withScaledNoise.i3.gz --no-PropagateMuons --no-volume\
cyl
```

```
real    36m58.612s
user    8m5.803s
sys     0m32.680s
```

PPC OpenCL

```
#ppc CUDA
time python $I3_SRC/simprod-scripts/resources/scripts/ppc.py --UseGSLRNG --inputfile muons.i3 -\
-outputfile mcpe-ppc-cuda.i3 --gcdfile /cvmfs/icecube.opensciencegrid.org/data/GCD/GeoCalibDete\
ctorStatus_AVG_55697-57531_PASS2_SPE_withScaledNoise.i3.gz --no-PropagateMuons --no-volume\
cyl
```

```
real    29m23.894s
user    8m24.614s
sys     0m30.172s
```

PPC CUDA

25% faster than OpenCL version



- 05/2009 First version of ppc (CPU-only) available for IceCube event simulation<sup>1</sup>
- 08/2009 Added SSE2-accelerated version written in Assembly<sup>2</sup>
- 09/2009 Added GPU-accelerated version (CUDA)
- 10/2011 Added OpenCL version<sup>3</sup>
- 10/2014 Added an option to run on accelerator chips (e.g., Intel Phi)
- 08/2017 Removed CUDA version<sup>3</sup>
- 10/2020 Added CUDA version back<sup>3</sup>
- 11/2020 Added support for the Upgrade geometry
- 02/2021 Moved to git<sup>4</sup>
- 06/2021 Added support for the Gen-2 geometry
- 10/2021 Added support for elongated sensors with multiple PMTs

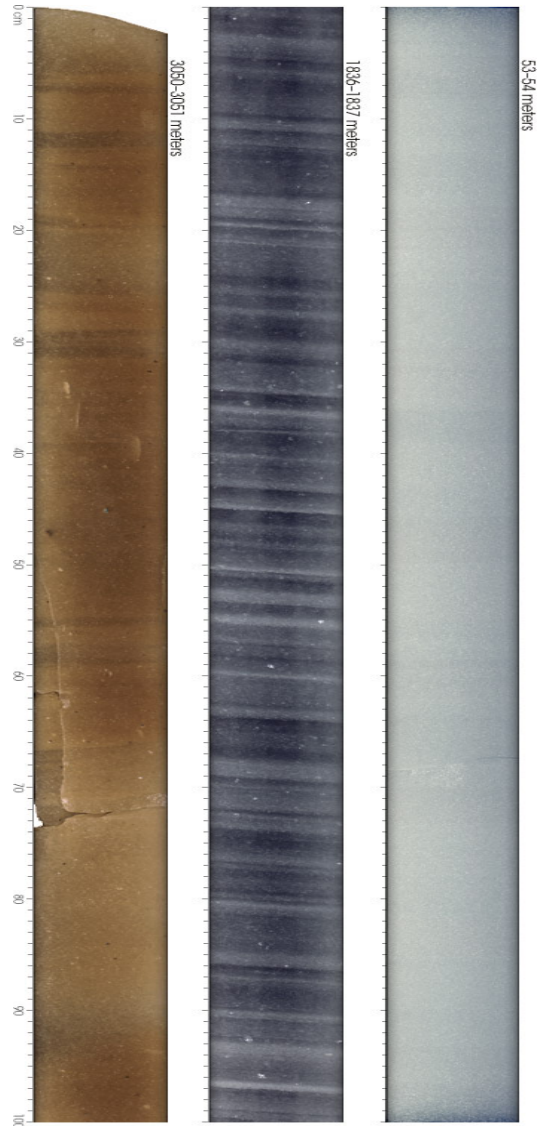
1) <http://code.icecube.wisc.edu/svn/meta-projects/combo/trunk/ppc/> (moved to git)

2) <http://code.icecube.wisc.edu/svn/sandbox/dima/asm/> (not maintained)

3) (at the request of the simulation group)

4) <https://github.com/icecube/icetray/tree/main/ppc> (current repository)

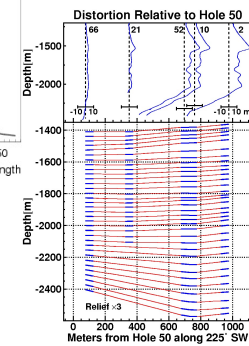
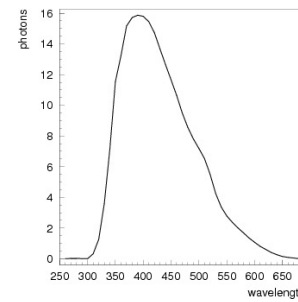
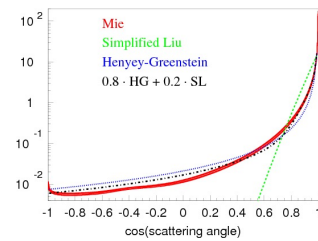
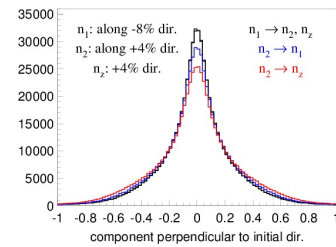
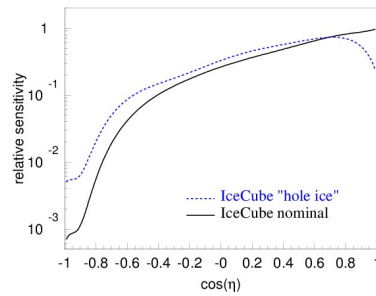
# Ice Models



AMANDA ice models:		model error
bulk, f125, mam, mamint, stdkurt, sudkurt, kgm, ...		
millennium (published 2006) → AHA (2007)		55%
IceCube ice models:		
WHAM	(2011)	42%
SPICE 1	(2009)	29%
SPICE 2, 2+, 2x, 2y	(2010)	added ice layer tilt
SPICE Mie	(2011)	fit to scattering function
SPICE Lea	(2012)	fit to scattering anisotropy
SPICE (Munich)	(2013)	7-string, LED unfolding
SPICE <sup>3</sup> (CUBE)	(2014)	lh fixes, DOM sensitivity fits
SPICE 3.0	(2015)	improved RDE, ang. sens. fits
SPICE 3.1, 3.2	(2016)	85-string, correlated model fit
SPICE HD, 3.2.2	(2017)	direct HI and DOM sens., cable, DOM tilt
SPICE EMRM	(2018)	absorption-based anisotropy
SPICE BFR	(2020)	birefringence-based anisotropy
SPICE BFRv2	(2021)	bfr+abs+sca anisotropy, 2d tilt

Model error (precision in charge prediction): <10%  
 Extrapolation uncertainty: 13% (sca) / 15% (abs)  
 Linearity: < 2% in range 0.1 ... 500 p.e.

# Overview of inputs to PPC



# Ice Configuration

Ice is configured in a directory, its location passed to the module via a parameter PPCTABLESDIR (set with os.putenv).

Ice models are kept in ice-models project. A recent model is maintained in ppc.

## SPICE 3.2:

as.dat	angular sensitivity curve: flasher-based with $p=0.3$
cfg.txt	main configuration file
icemodel.dat	ice model table (scattering, absorption vs. depth)
icemodel.par	wavelength dependence coefficients
lcmode.tbl	air bubble parametrization above 1350 m depth
rnd.txt	random number multipliers
tilt.dat	table of ice layer relief values at tabulated points
tilt.par	locations of tabulated points
wv.dat	normalized cherenkov+transmission convolution
cx.dat	table of DOM tilts
dx.dat	table of cable positions

# Ice configuration

If running ppc outside icetray (as a stand-alone command-line program or in concert with DirectFit (distributed with ppc in private/ppc/llh directory) the following files are required:

geo-f2k	detector geometry
---------	-------------------

Optional files:

eff-f2k	table of RDE values
wv.rde	tabulated difference of high-QE/nominal response vs. wavelength
hvs-f2k	table of high voltages (zero voltage disables DOM)

There are also multiple additional input files to llh/DirectFit, these will be described in documentation

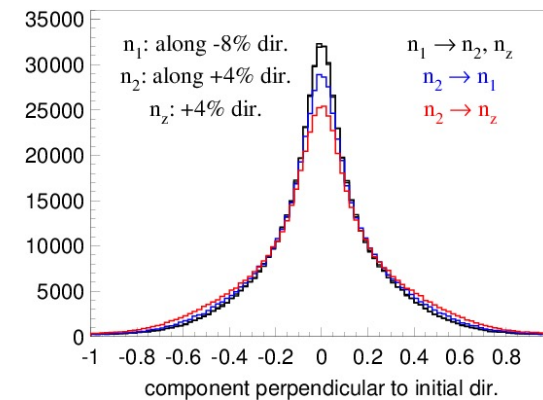
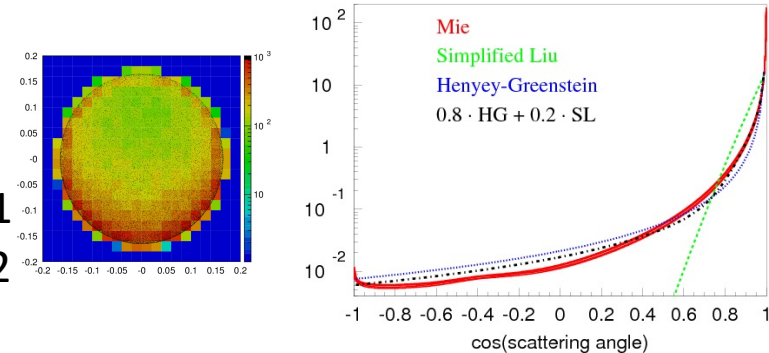
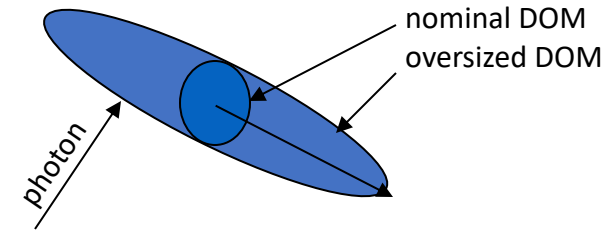
# Main configuration file: cfg.txt

cfg.txt:

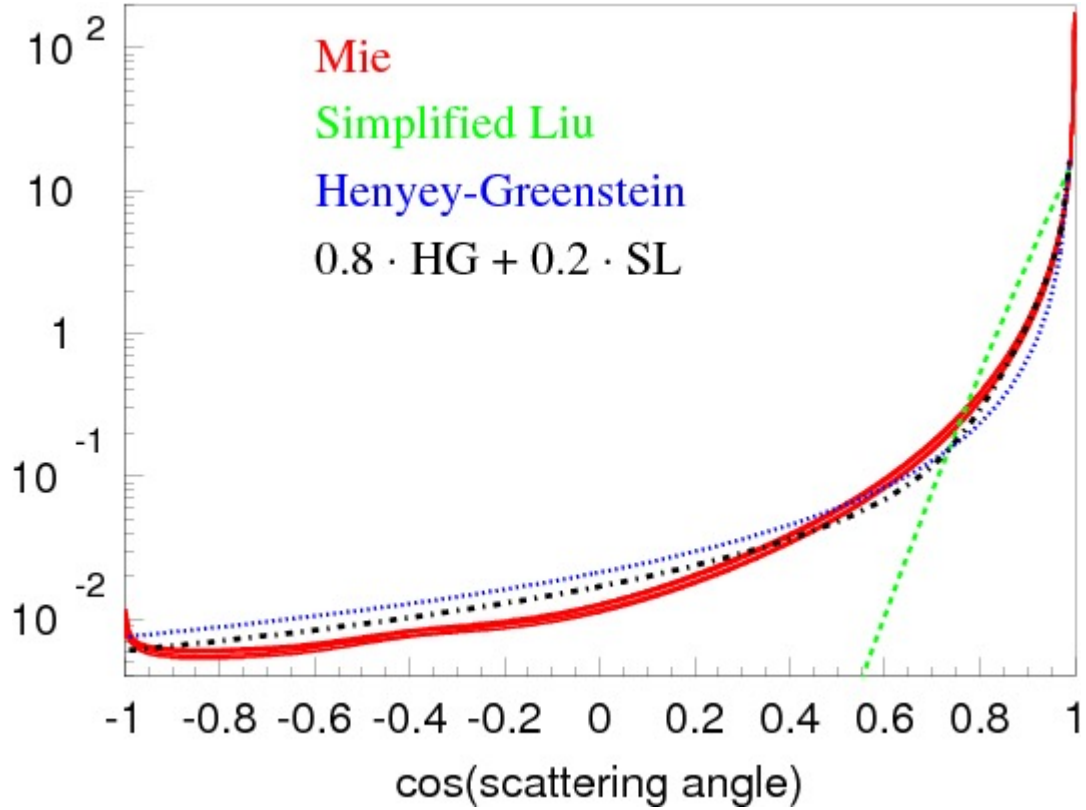
```
# ppc configuration file: follow strict order below
5      # over-R: DOM radius "oversize" scaling factor
1.0    # overall DOM efficiency correction
0.35   # 0=HG; 1=SAM
0.9    #  $g = \langle \cos(\theta) \rangle$ 

130    # direction of ice tilt (perp. to flow)
-0.106 # magnitude of major anisotropy coefficient k1
0.053  # magnitude of minor anisotropy coefficient k2

0.5    # hole ice radius in units of [DOM radius]
0.5    # hole ice effective scattering length [m]
100    # hole ice absorption length [m]
0.35   # hole ice 0=HG; 1=SAM
0.9    # hole ice  $g = \langle \cos(\theta) \rangle$ 
```



cfg.txt



We approximate and fit data with a mixture of:

**Simplified Liu:**  $p(\cos \theta) \sim (1 + \cos \theta)^\alpha$ , with  $\alpha = \frac{2g}{1-g}$

**Henyey-Greenstein:**  $p(\cos \theta) = \frac{1}{2} \frac{1-g^2}{[1+g^2-2g \cdot \cos \theta]^{3/2}}$

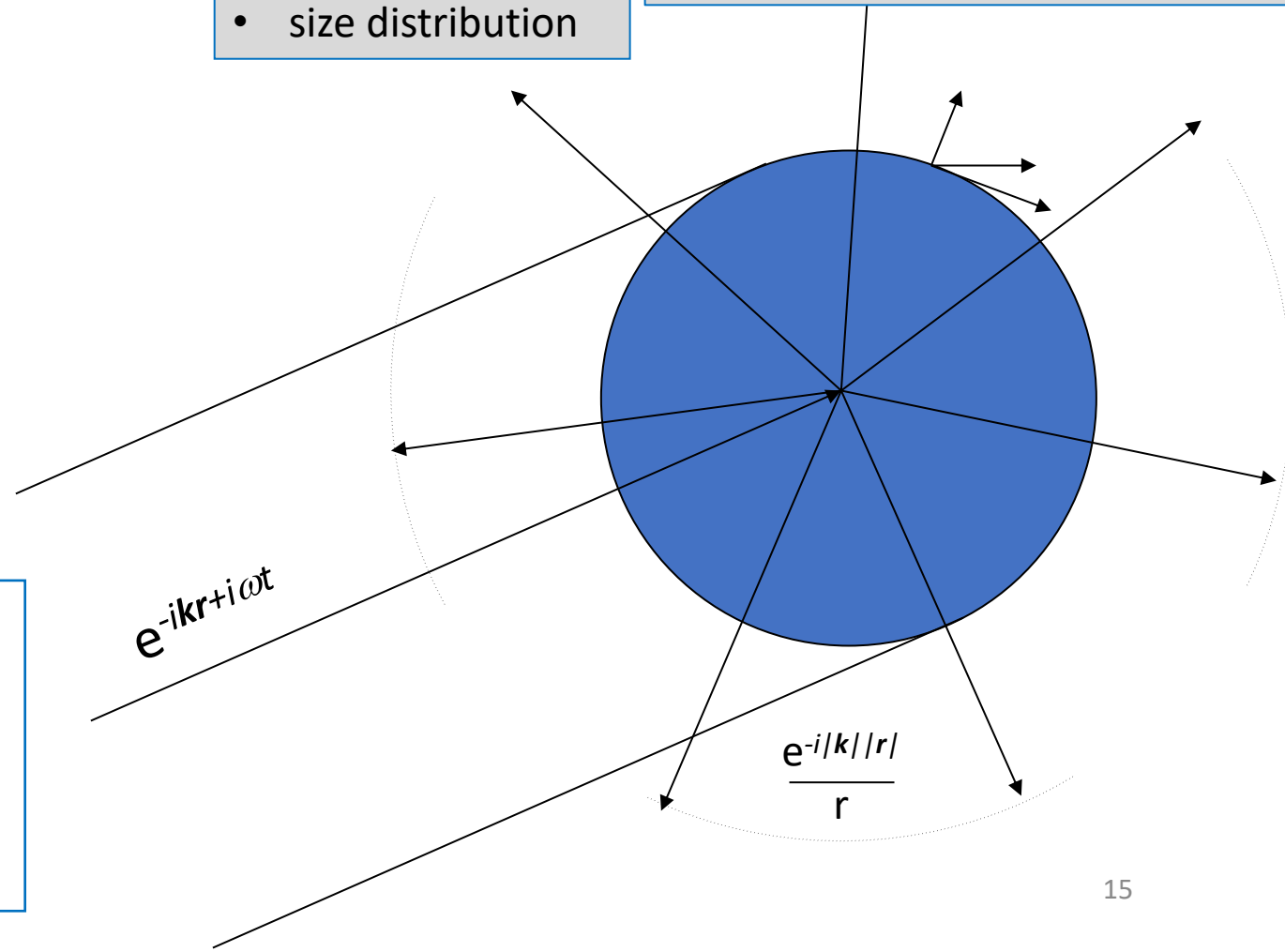
# Mie scattering theory

Can be solved for spherically symmetric particles

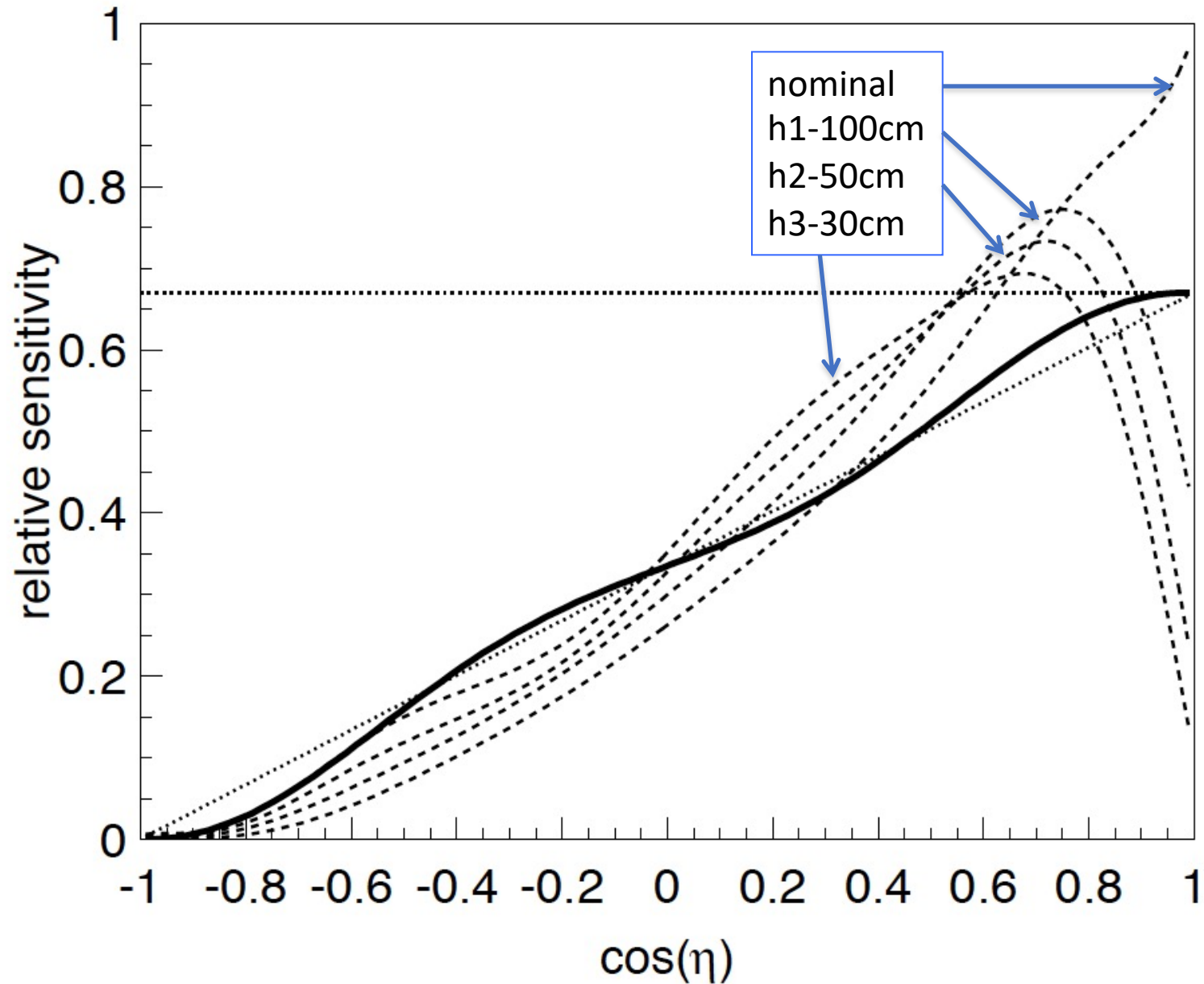
Need to know:

- refractive index
- size distribution

Continuity in E, H: boundary conditions in Maxwell equations



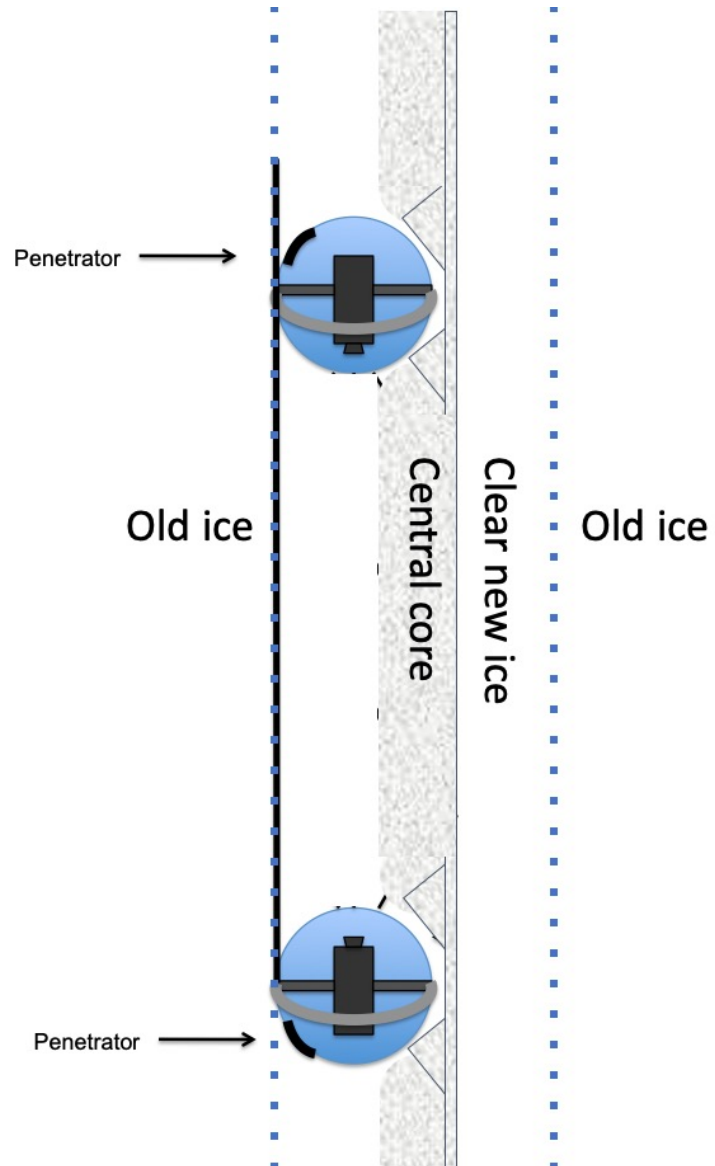
# Traditional “hole ice” angular sensitivity



as.dat



# Direct Hole Ice simulation



cfg.txt  
as.dat

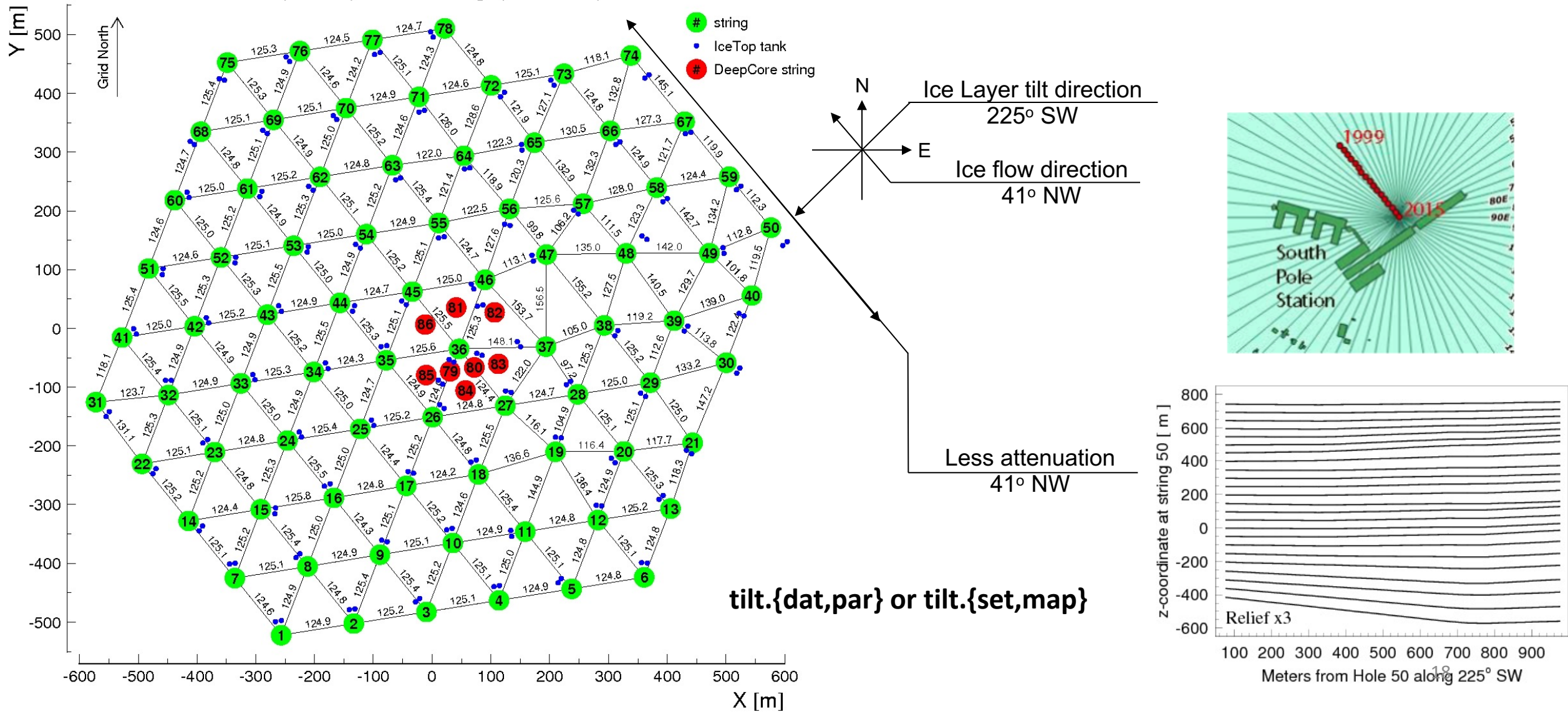


DOM touches the hole wall, is  $\frac{2}{3}$  of the hole diameter

Most of the HI is transparent, except for the milky central column centered in the hole and  $\frac{1}{3}$  of hole diameter (referred to as HI in the following, starting with the next line)

HI diameter is  $\frac{1}{2}$  of DOM diameter

# Glacial ice flow, ice layer tilt at the South Pole



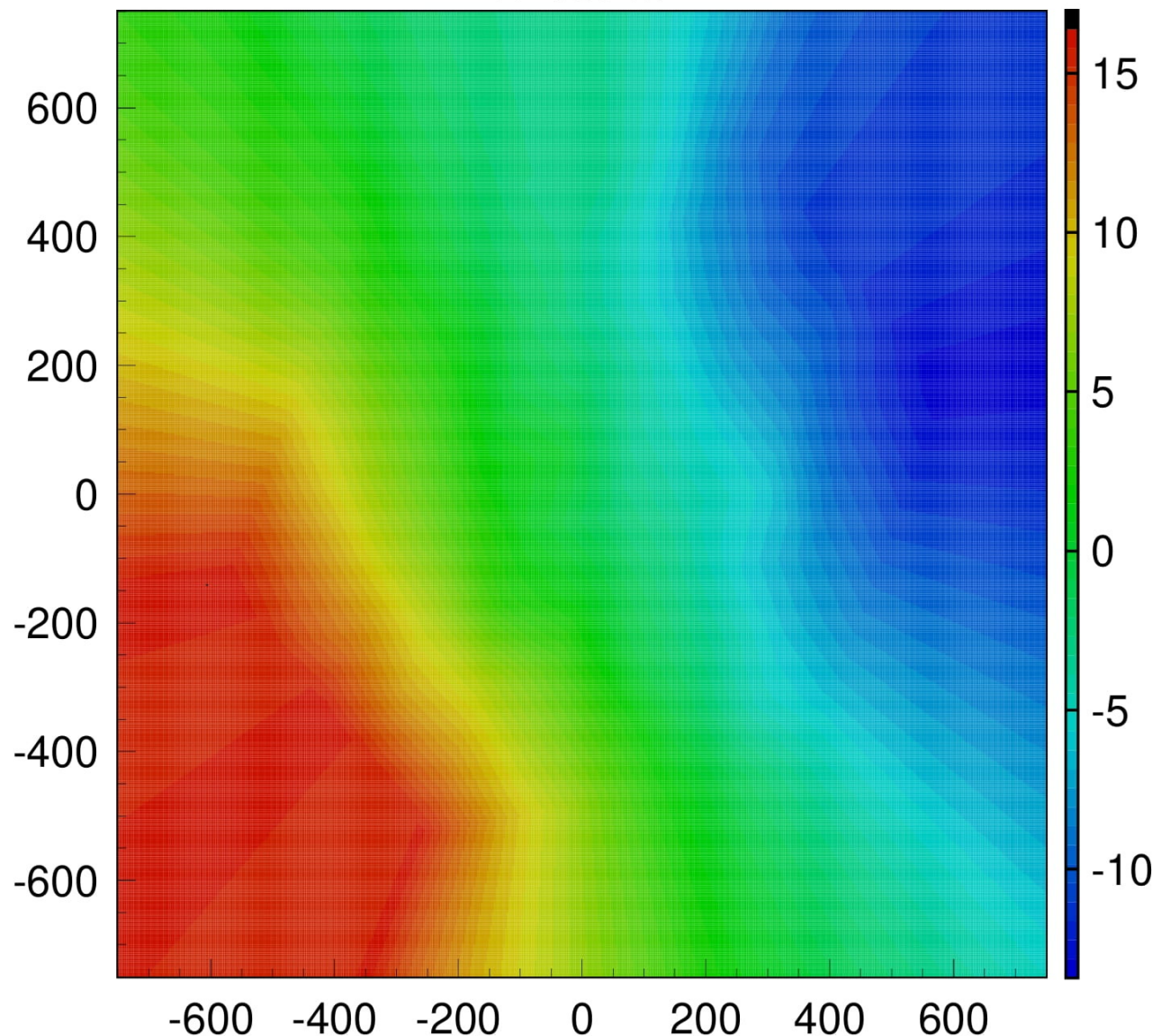
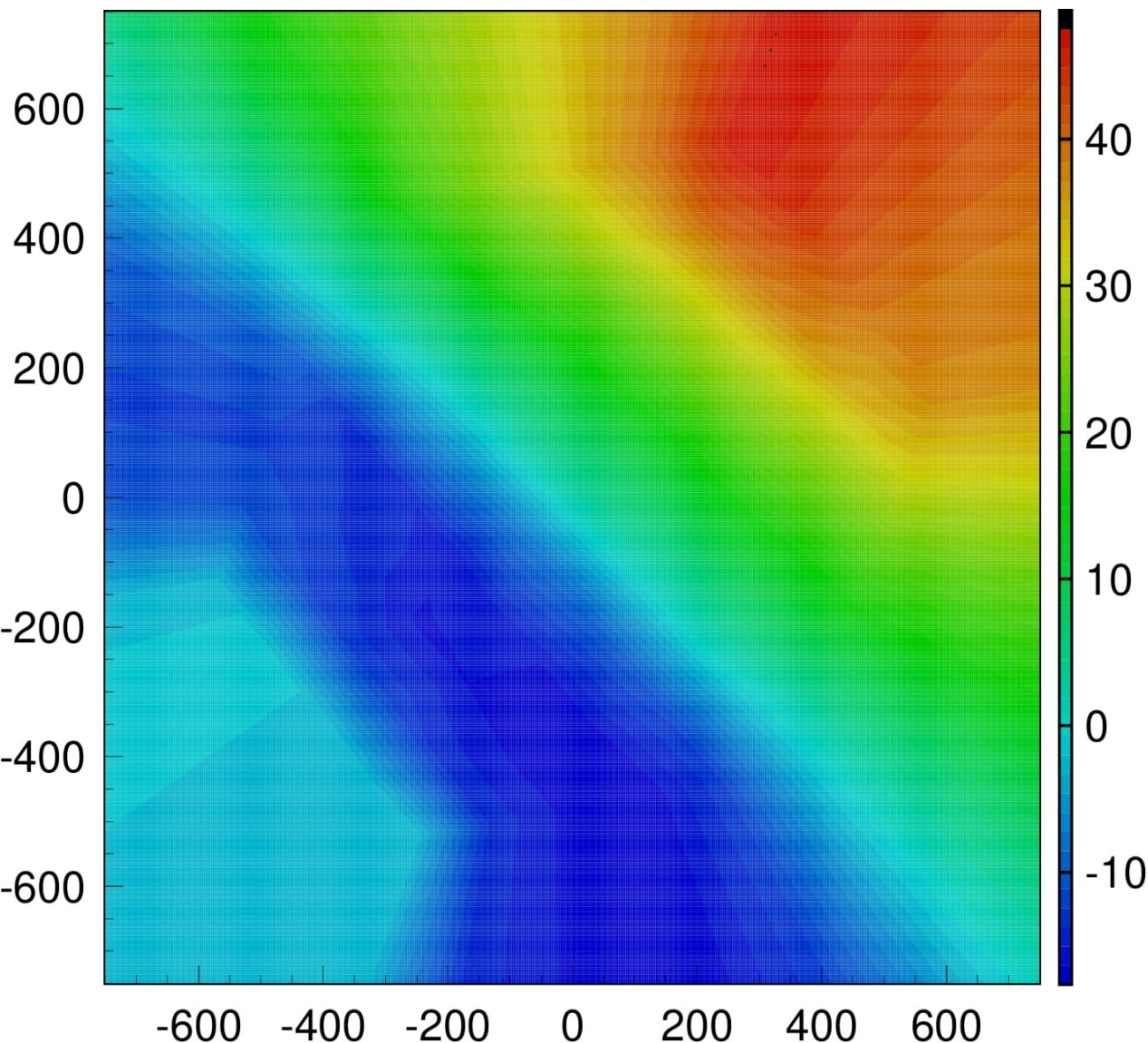


# Parameterized tilt maps

PMC *reduced maps* – fixed (*string 78 corner*)

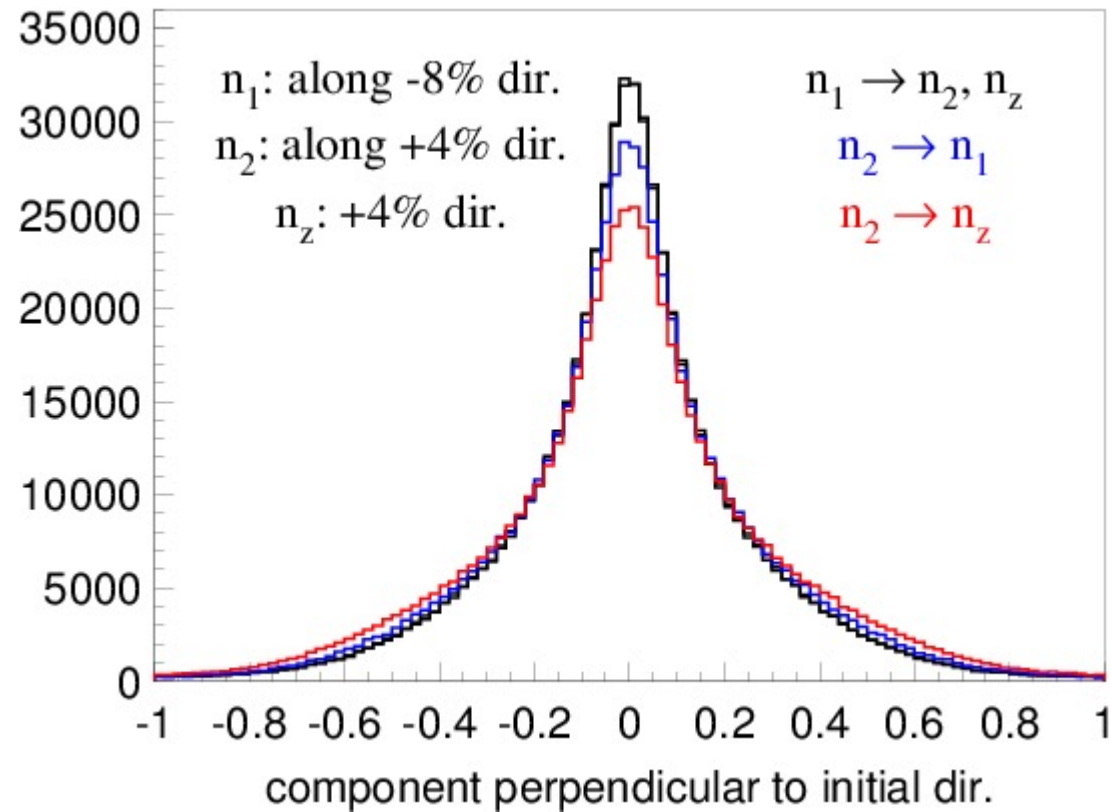
$z=-450$  m

$z=450$  m



# Scattering anisotropy

$$f(\vec{n}_i \cdot \vec{n}_o) \rightarrow f(\vec{k}_i \cdot \vec{k}_o), \quad \vec{k}_{i,o} = \frac{A\vec{n}_{i,o}}{|A\vec{n}_{i,o}|}$$



cfg.txt and icemodel.dat



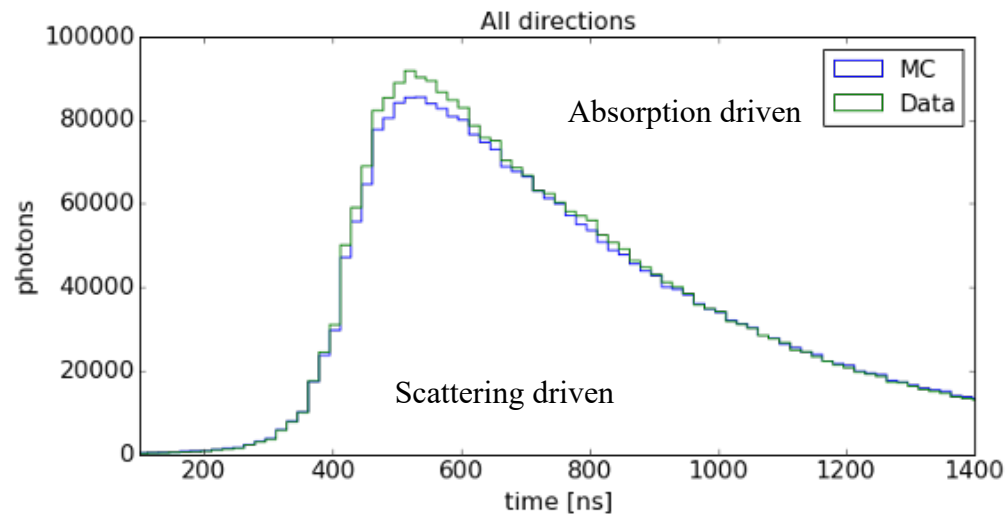
# Models of optical ice anisotropy in IceCube

1. Scattering (mainly): direction dependent scattering function (ICRC 2013)
2. Absorption (mainly): direction dependent absorption (studied in 2018)

Introduced depth-dependence (2017)

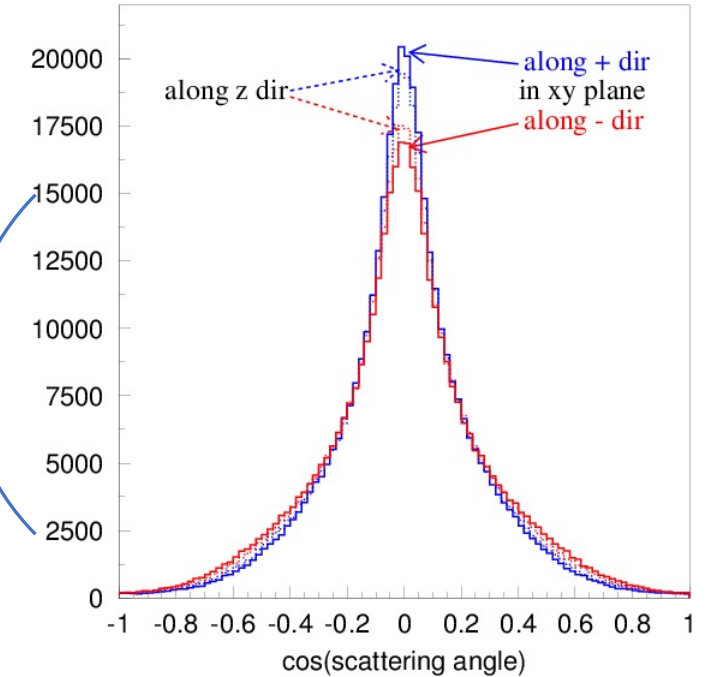
Discrepancies between data and simulation remain

Cannot simultaneously fit total charge and arrival time distribution to statistical precision



SPICE Lea, 3.2.x

scattering-based

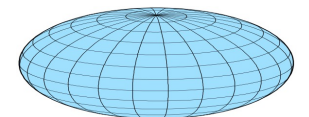
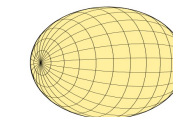


SPICE EMRM

absorption-based

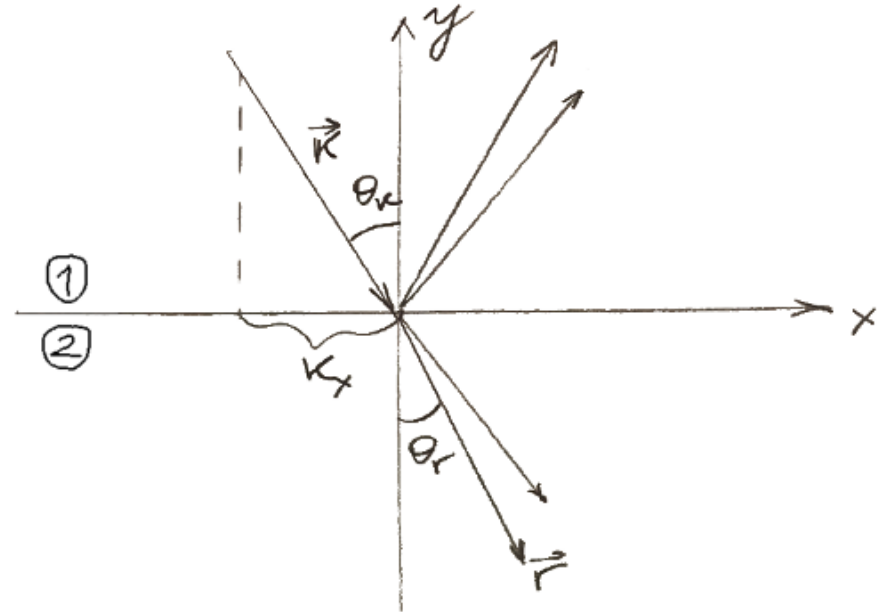
prolate

oblate



# Birefringence

- Ice is a birefringent material with  $n_e - n_o = 0.0015$ . This tiny difference builds to a macroscopic effect due to 1000s of ice crystal boundaries crossed per meter of traveled distance
- At each grain boundary every ray is split into two reflected and two refracted rays, one ordinary and one extraordinary ray each
- Wave vector component parallel to surface is conserved, norm is proportional to the refractive index
- Poynting vectors are derived from wave vectors and boundary conditions
- Outgoing ray is randomly sampled from Poynting vectors according to Poynting theorem (Poynting vector component through the plane is conserved)

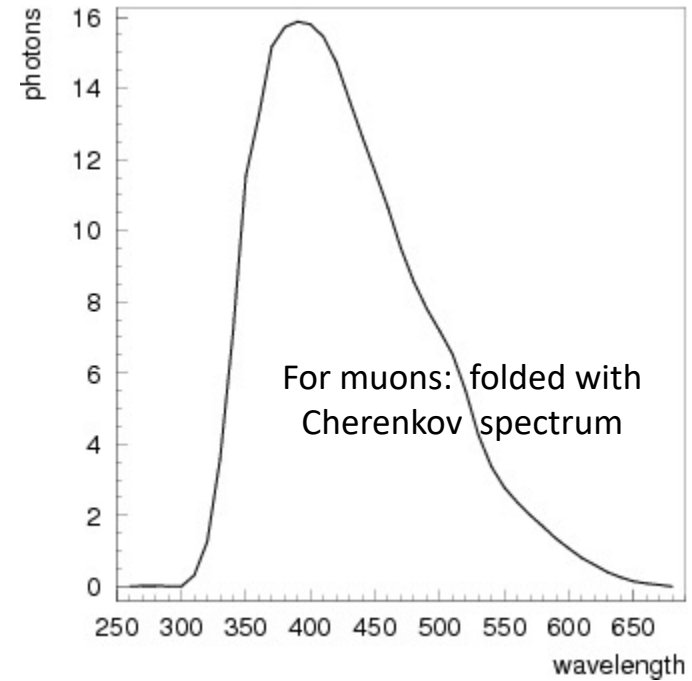
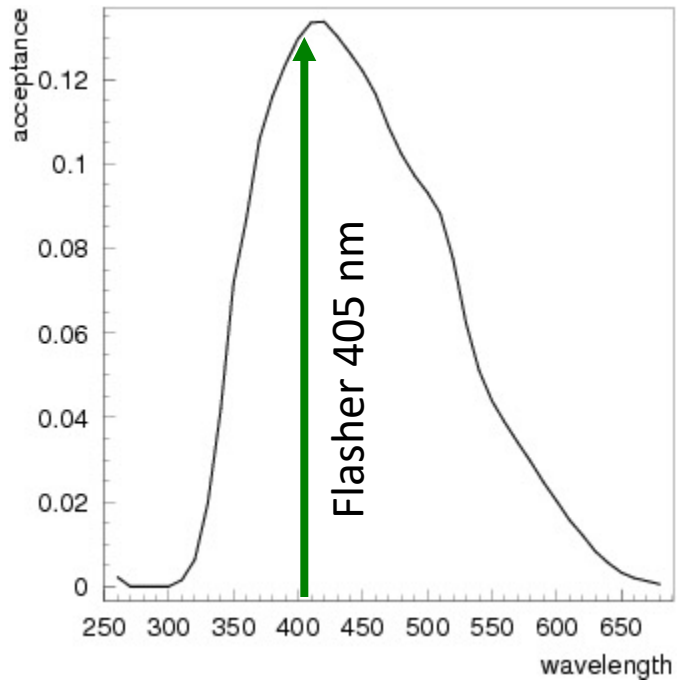


$$\begin{aligned}\hat{n} \cdot D_2 &= \hat{n} \cdot D_1 \\ \hat{n} \cdot B_2 &= \hat{n} \cdot B_1 \\ \hat{n} \times E_2 &= \hat{n} \times E_1 \\ \hat{n} \times H_2 &= \hat{n} \times H_1.\end{aligned}$$

Hence we can make the following observations:

1. Normal components of  $D$  and  $B$  are continuous across a dielectric interface
2. Tangential components of  $E$ ,  $H$  are continuous across a dielectric surface

# Photon wavelength sampling



# Light yield parametrization

Calculated 2450.08 of Cherenkov photons per meter of bare muon track when convolved with transmission properties of glass, gel, and detection efficiency of PMT for the nominal (non-high-QE) DOM.

Additional factor: efficiency, or “cable shadowing” is applied to this number.

Via compile-time configuration choose:

Light yield:

- ~~C.W. parametrization of bare muon/em/hadr. cascades~~
- ~~M.K. updated parametrization for em/hadr. cascades~~
- Leif Raedel updated parametrization of muon/em/hadr. cascades

Longitudinal profile and Cherenkov cone:

- ~~C.W. parametrization (simplified fit)~~
- Leif Raedel updated parametrization



# ppc example script run.py (cont.)

```
os.putenv("PPCTABLESDIR",  
expandvars("$I3_BUILD/ppc/resources/ice/lea"))
```

```
load("libxppc")  
load("libppc")
```

```
tray.AddModule("i3ppc", "ppc")(  
    ("gpu", gpu),  
)
```