thetaC from full stack

june 6, 2026
w.j. llope

Outline:

An overview slide presented as plan to recoWG on May 4

How I generated LUTs using npsim.

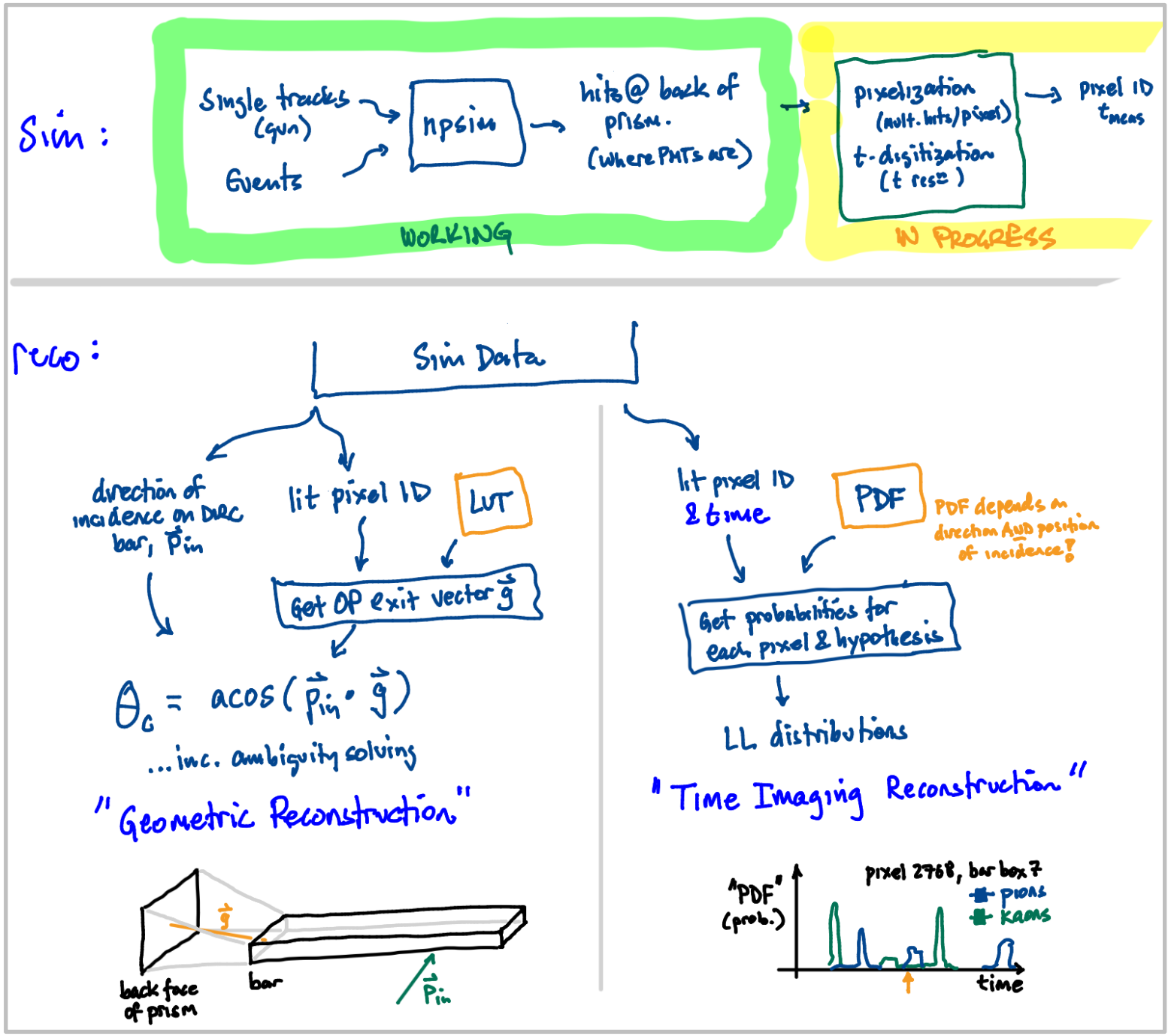
thetaC ingredients:
- geant4 hits (OPs at back face of prism), fake-pixelized
- track incidence positions and angles at DIRC bars
- OP bar-exit angles and weights for each "pixel" in the readout plane

Goal:

Calculate thetaC for 1000 π, K, and p, each w/ ptot = 3 GeV, and theta = 30 deg

but using only npsim and C code in root, no standalone sim, PrtTools, etc...

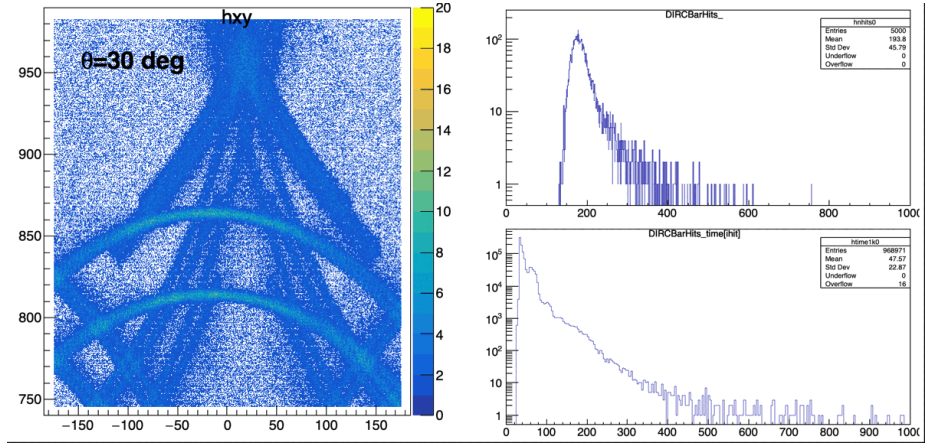
@end, some questions from Wouter



← simplified schematic view of hpdirc code

Strategy:

- Left first, right next
- Macro first, eicrecon next

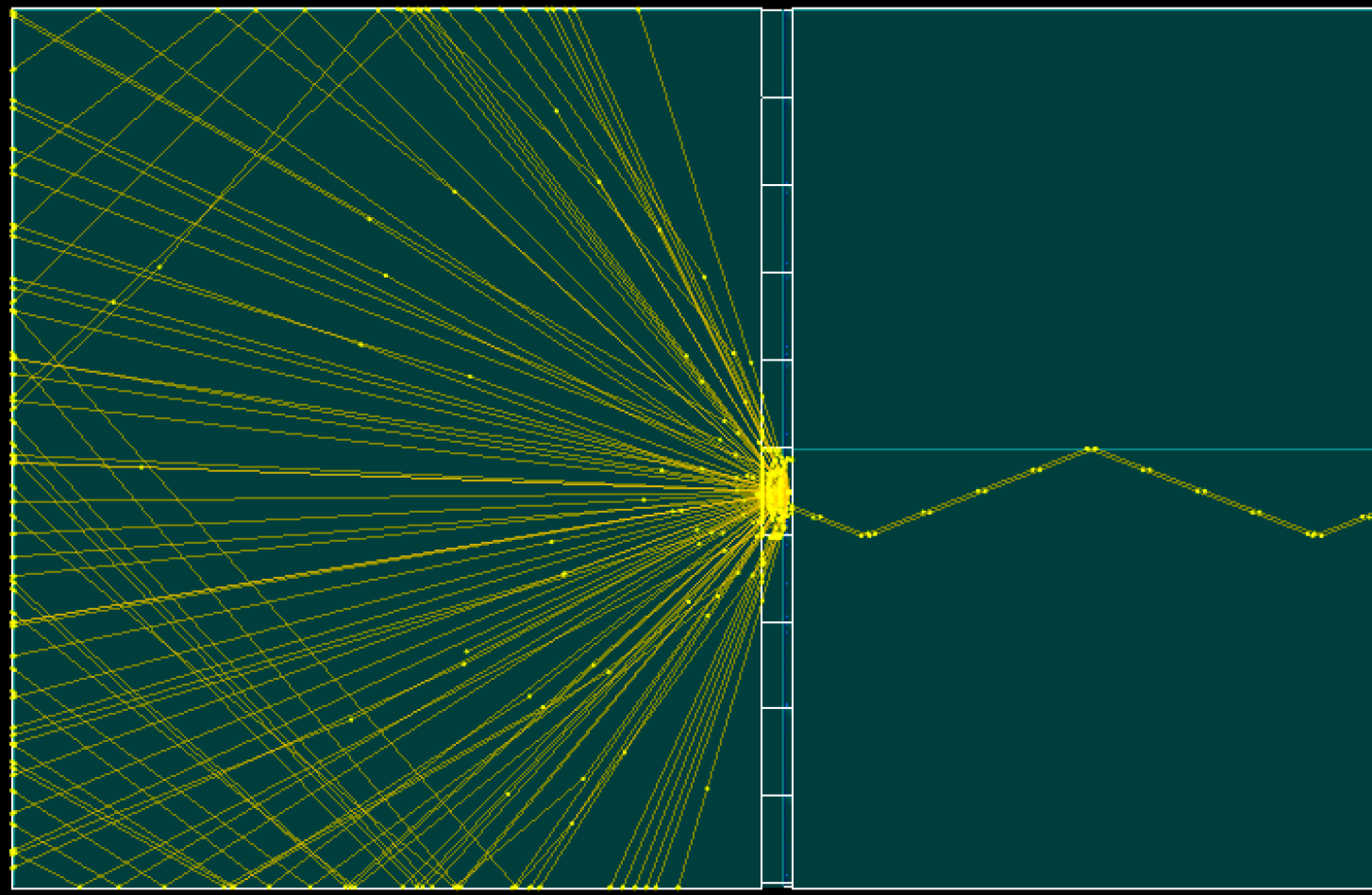


OPs bouncing through lenses+prism for each of 10 bars were simulated in npsim.
20M OPs takes 6 minutes on 2026 M5.

Next steps (all macro, not EICrecon):

- fake-pixelize (th2d)
- implement pixel→angles map
- calculate theta_C for π,K,p
- implement ambiguity sorting
- show expected GR performance

then move it all to EICrecon.



LUT simulation

- pick bar
- throw 20M OP into 2π from bar end

map all OP paths from bar to pixel

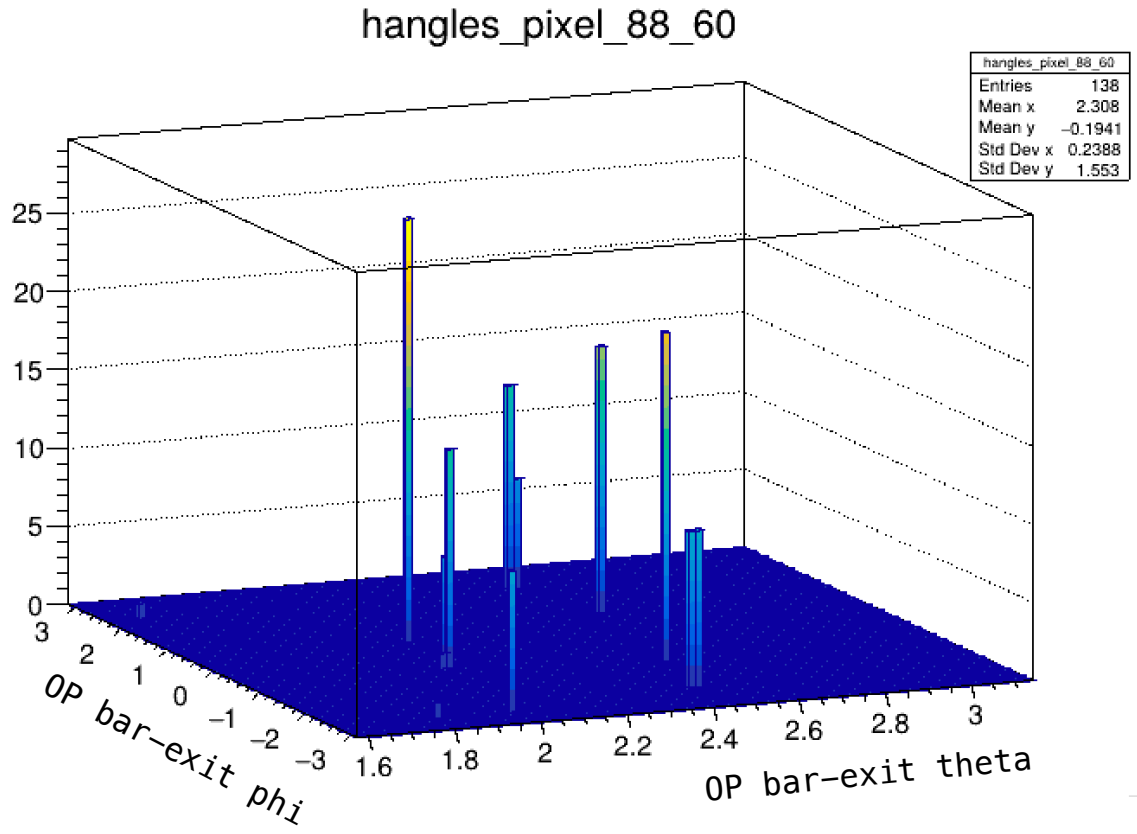
Variables saved in standard .edm4hep.root file:

```
Int_t           DIRCBarHits_;
Float_t         DIRCBarHits_time[kMaxDIRCBarHits];         //[DIRCBarHits_]
Double_t        DIRCBarHits_position_x[kMaxDIRCBarHits];   //[DIRCBarHits_]
Double_t        DIRCBarHits_position_y[kMaxDIRCBarHits];   //[DIRCBarHits_]
Double_t        DIRCBarHits_position_z[kMaxDIRCBarHits];   //[DIRCBarHits_]
Float_t         DIRCBarHits_momentum_x[kMaxDIRCBarHits];   //[DIRCBarHits_]
Float_t         DIRCBarHits_momentum_y[kMaxDIRCBarHits];   //[DIRCBarHits_]
Float_t         DIRCBarHits_momentum_z[kMaxDIRCBarHits];   //[DIRCBarHits_]
```

global->local hit positions
y'=x, x'=-y

Then hits from -175 < x' < 175 mm roughly
and 740 < y' < 980 mm
divide into 175 x 120 2mmx2mm bins (TH2D)

Then plot the OP bar-exit angles for all OPs that hit a specific pixel.



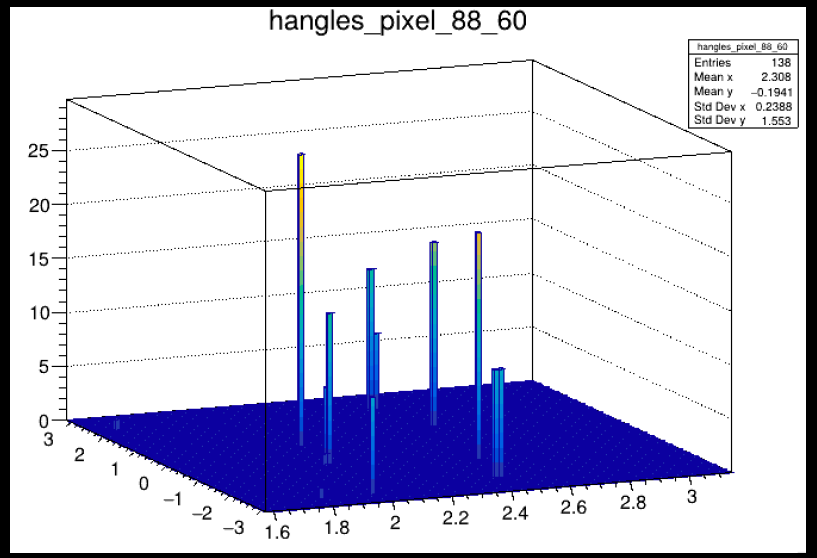
Each of these spikes is a "path".
Each pixel has 2-20 paths.
LUTs calculated with 1000s of OPs/path/pixel.

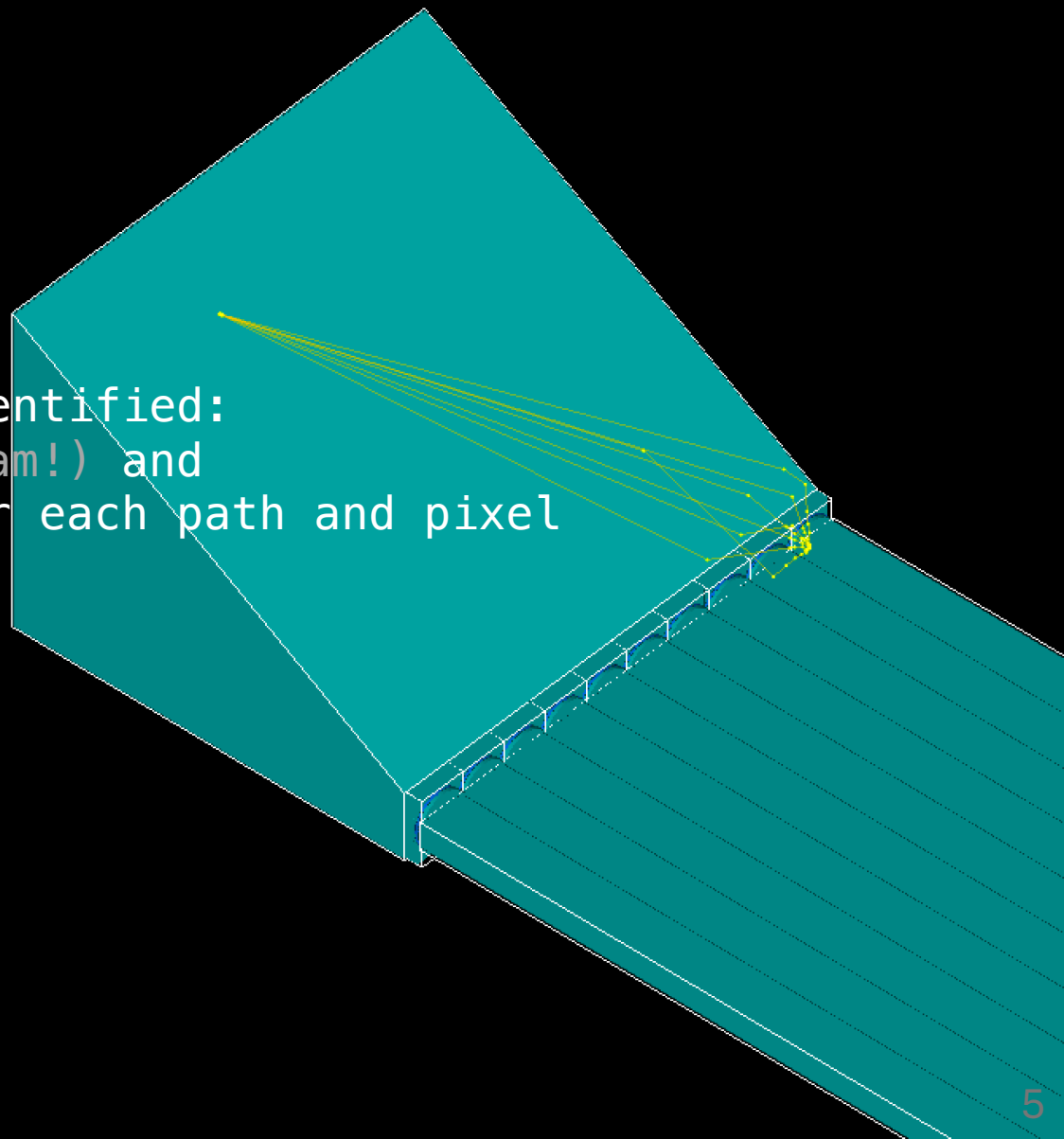
There is a separate incidence angles histogram for each pixel

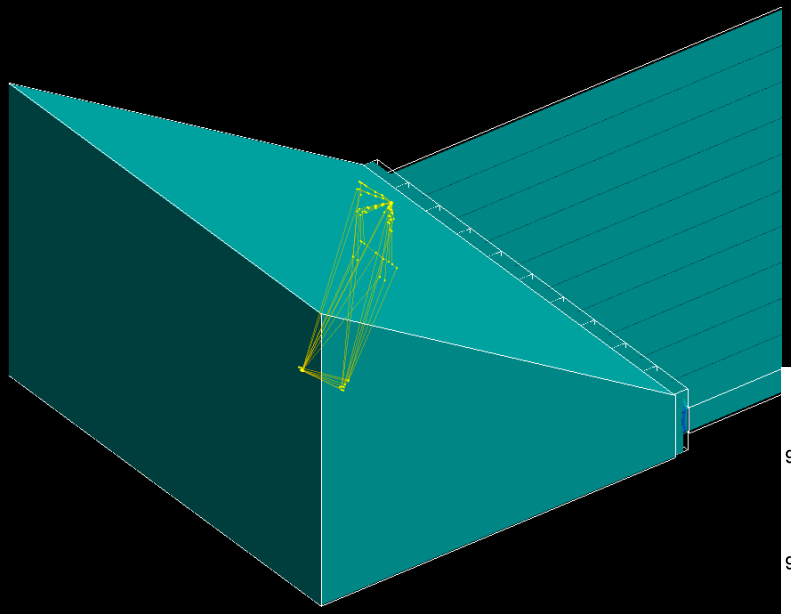
Histograms used only as a simple pattern recognition.

0. Find highest bin.
1. Add any OPs in 8 neighbor bins
2. This is a new path to this pixel!
3. Zero out this bin(s).
4. goto 0, until you run out of non-zero bins.

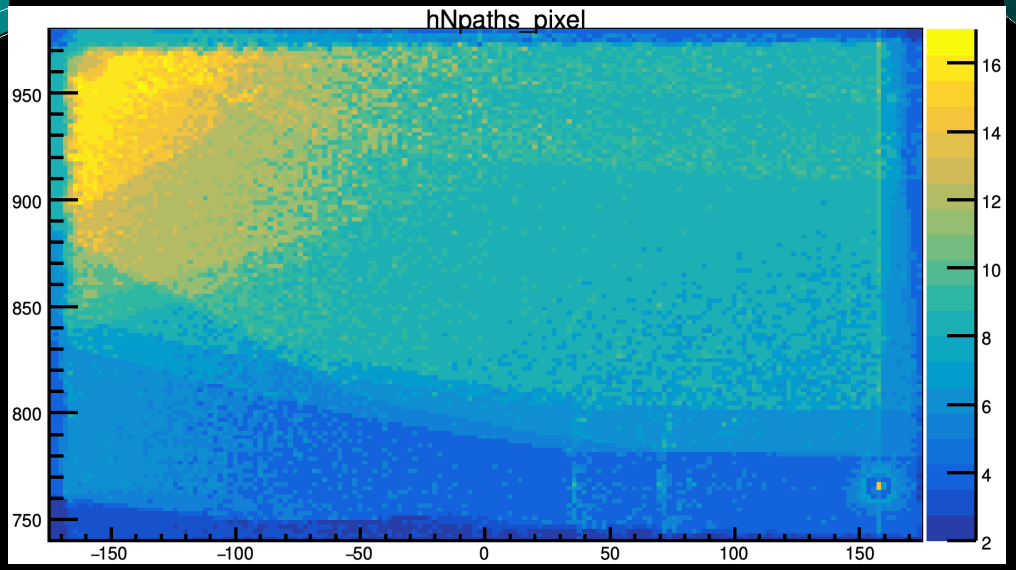
Once the bins for a given path and pixel are identified:
loop through the original data (not the histogram!) and
calculate the mean OP bar-exit theta and phi for each path and pixel





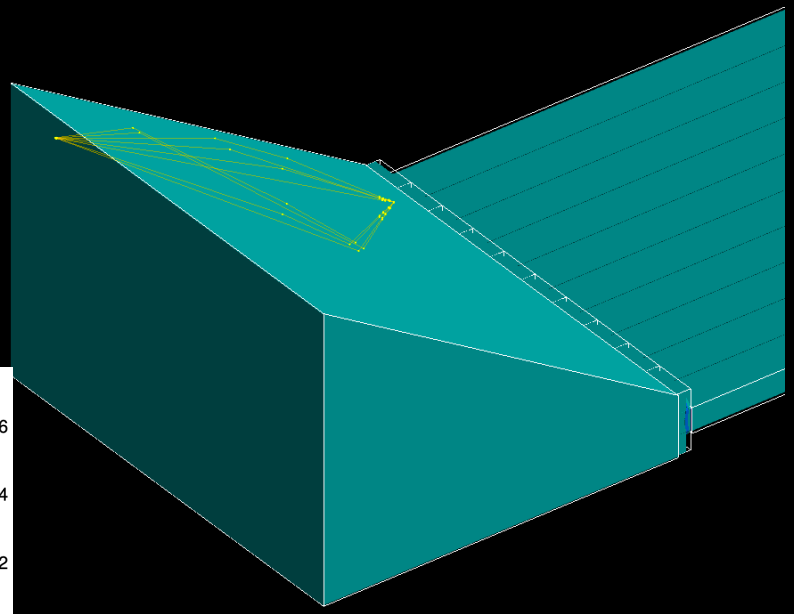


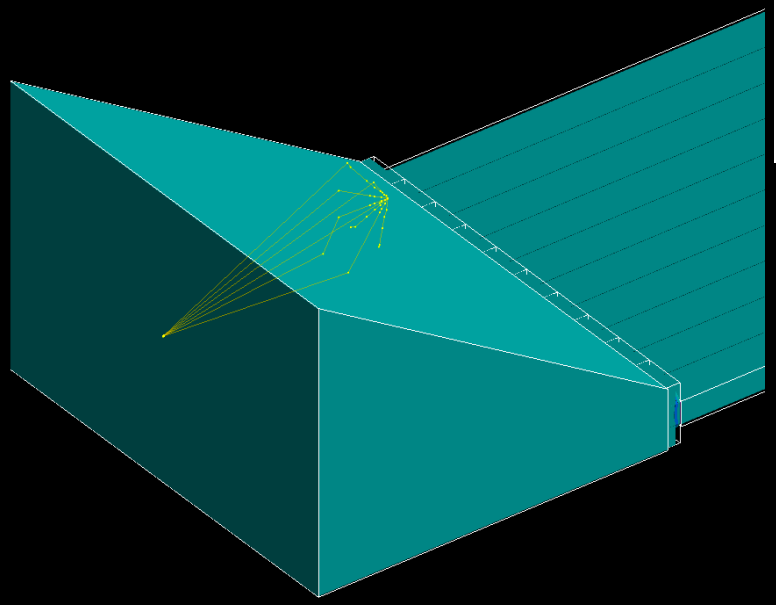
makemac.C: read LUT TTree
and write a vis.mac file that
uses G4 GPS to throw one OP/path
for any user-defined pixel.

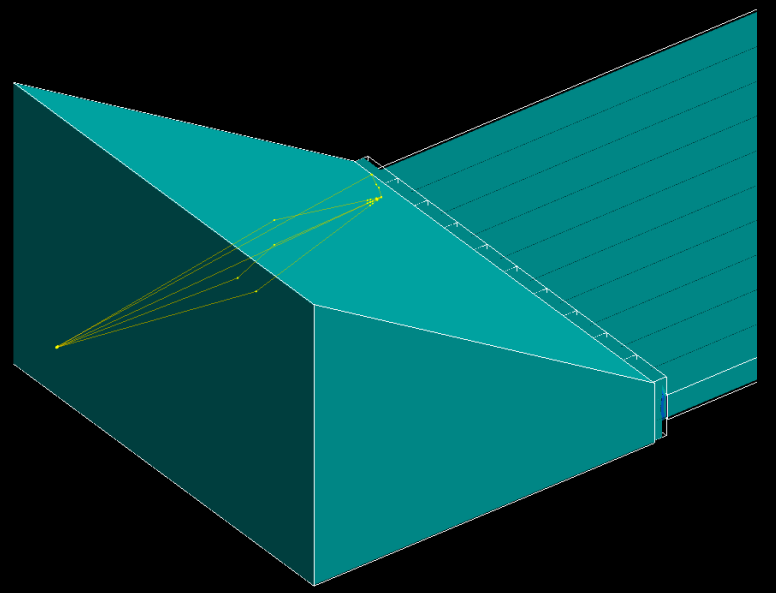


The pixel plane, Z-axis is
number of paths per pixel

(here showing paths from bar9)

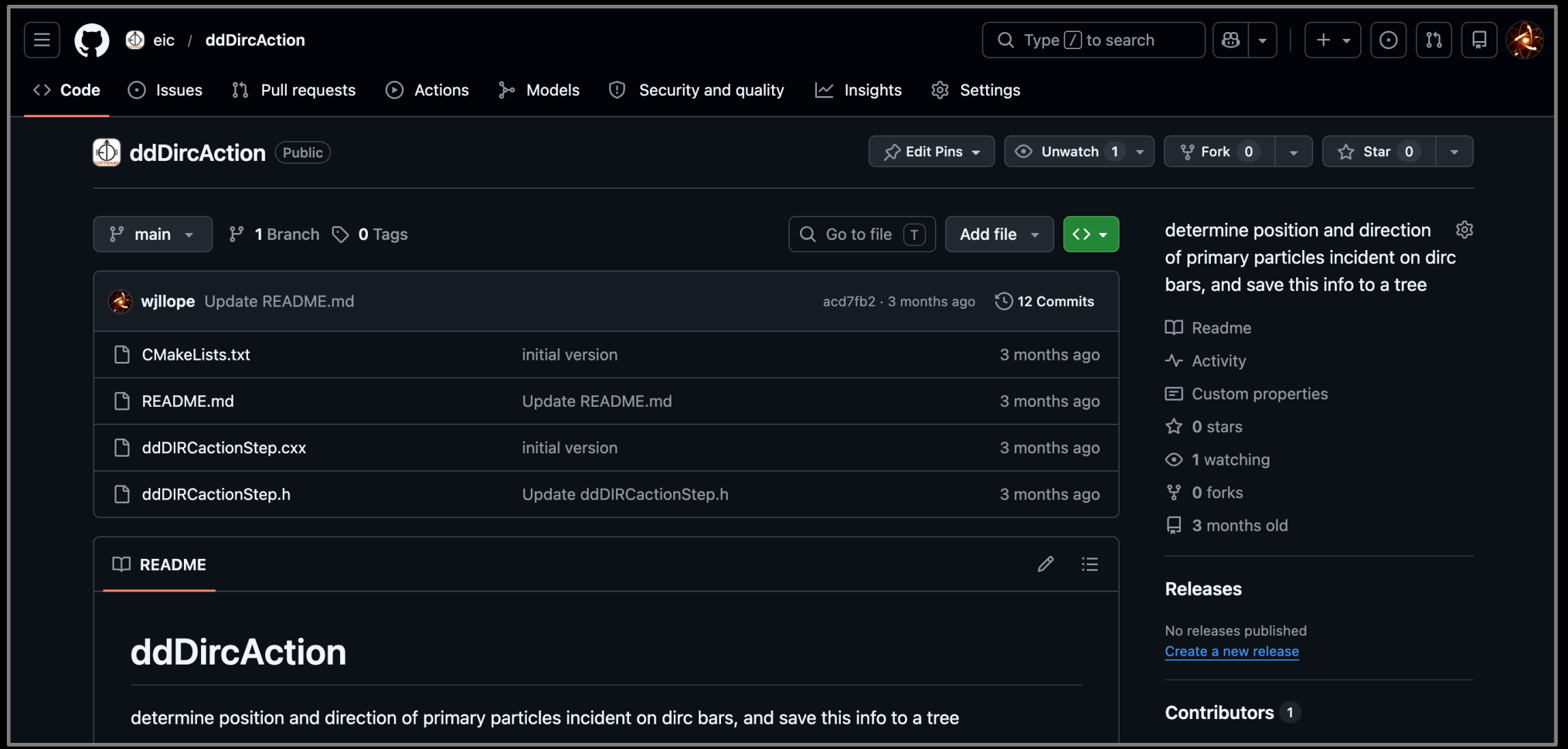






eic/ddDircAction

Writes the incidence tree: x,y,z,px,py,pz when primaries enter DIRC bars
A stepping action, called with an arcane npsim command line interface
Instructions in readme on github



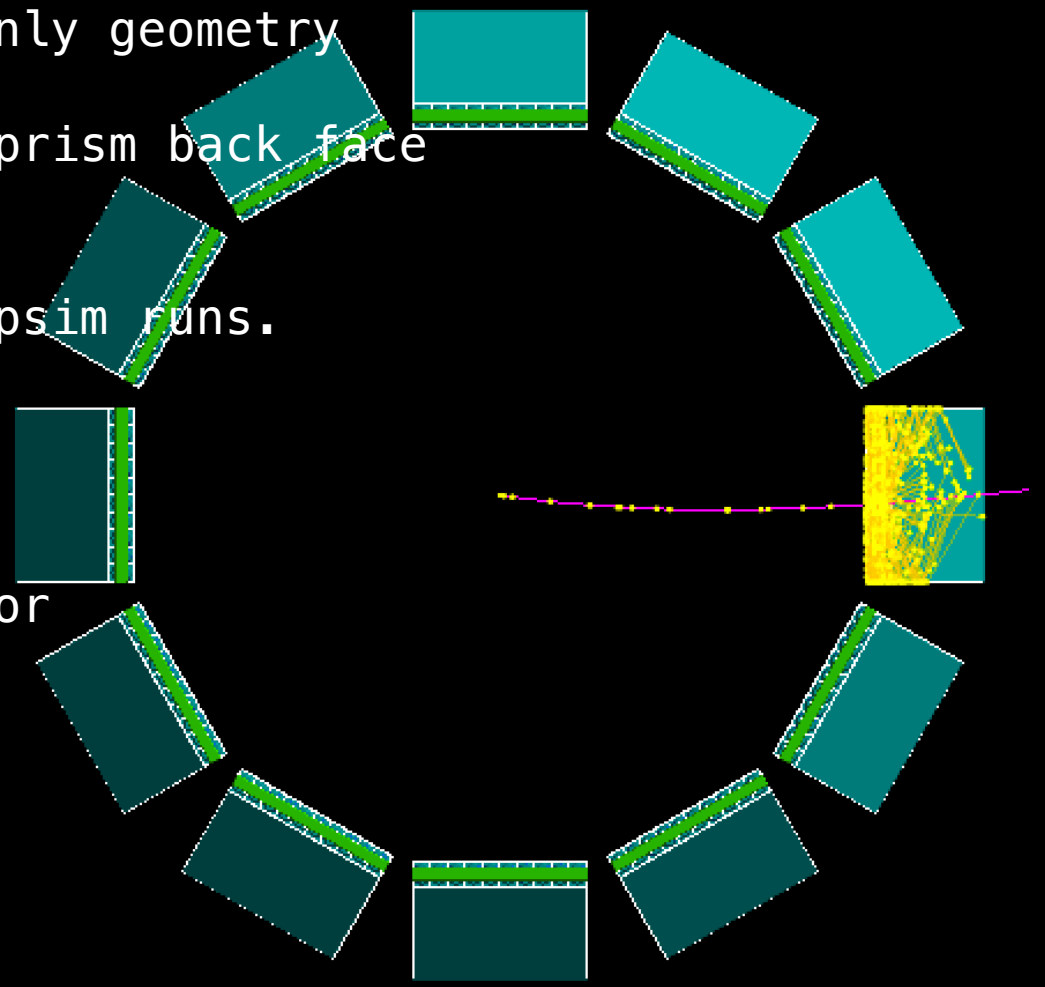
With LUTs in hand, now generate events with DIRC-only geometry

npsim writes standard "events" TTree with hits at prism back face
ddDircActionStep writes the incidence tree.

Generate 1000k pi+, kaon+, proton+ in 3 separate npsim runs.
   theta = 30 deg, ptot = 3 GeV.
   set primary azimuth to hit center of bar4

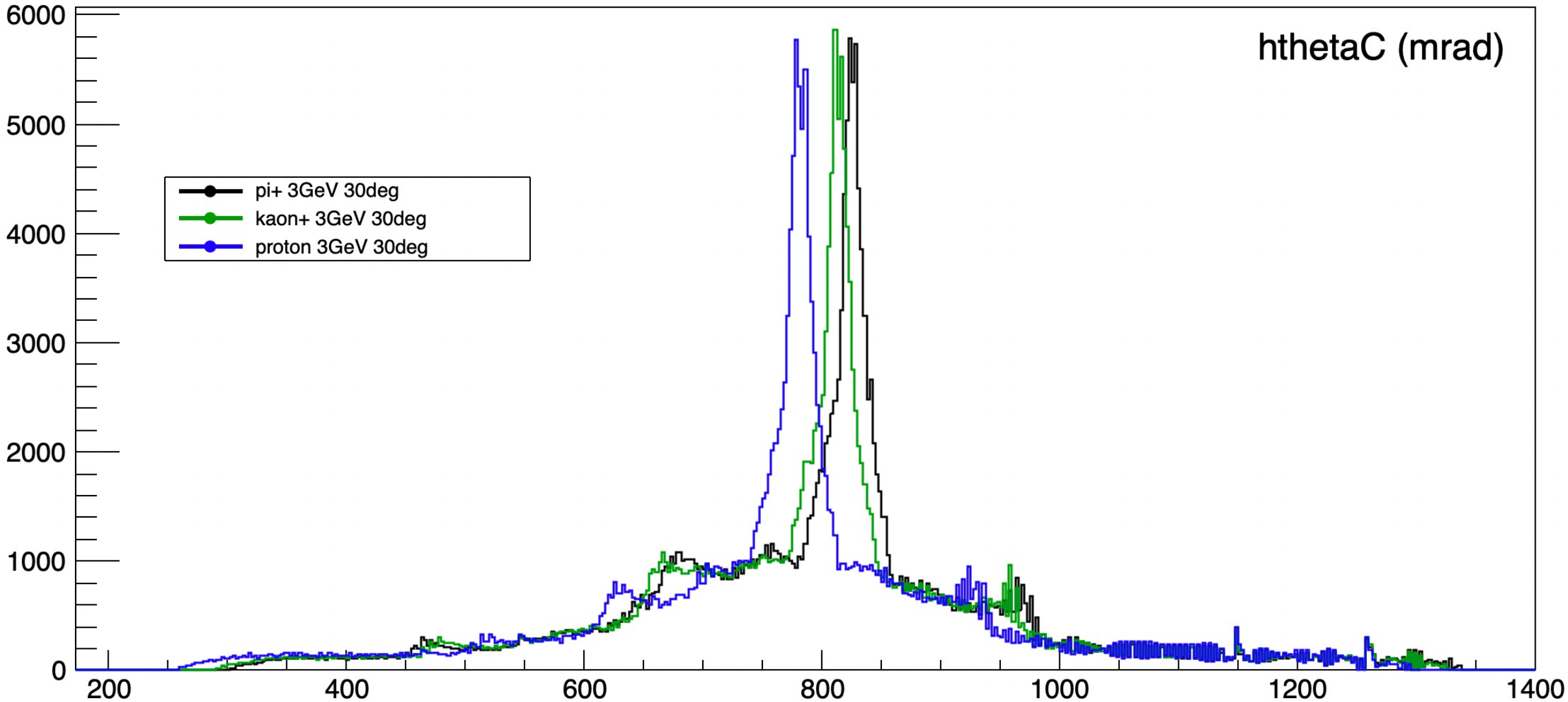
Calculate thetaC as dot-product of OP exit angle
vectors from pixel->LUT and primary incidence vector
over all paths leading to each lit pixel.

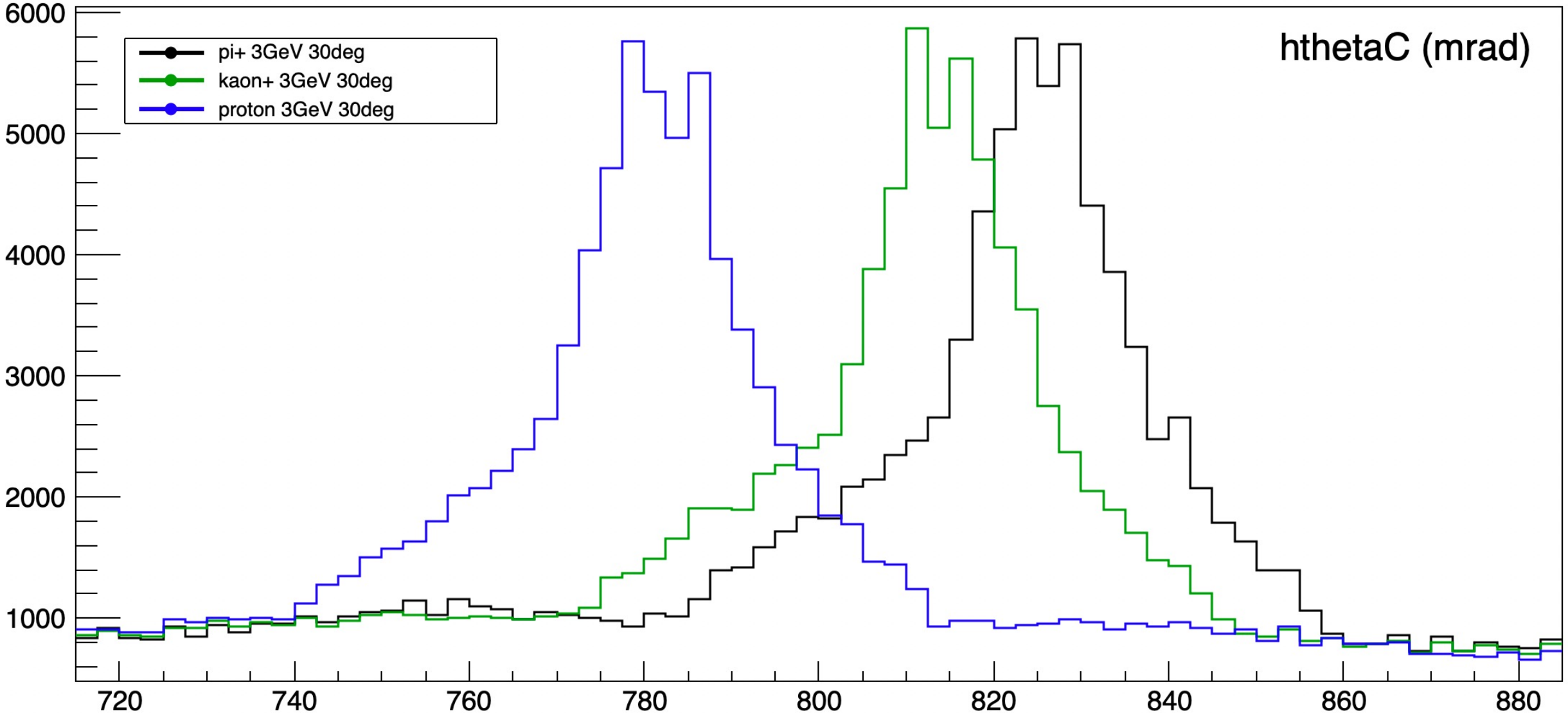
Check each of 8 ambiguity sign flips.
      Keep best flip for each hypothesis.
      expected: $\theta_C = \cos^{-1}\frac{1}{1.47\beta}$



thetaC histogram filled with weight = probability for this path to this pixel.
      (weights for each pixel sum to unity)

Not doing any time cuts, or any path sanity cuts, and really haven't optimized
any binning or anything.


hthetaC (mrad)
pi+ 3GeV 30deg
kaon+ 3GeV 30deg
proton 3GeV 30deg
6000
5000
4000
3000
2000
1000
0
200
400
600
800
1000
1200
1400


pi+ 3GeV 30deg
kaon+ 3GeV 30deg
proton 3GeV 30deg
hthetaC (mrad)
6000
5000
4000
3000
2000
1000
720
740
760
780
800
820
840
860
880

Wouter Deconinck (he/him) 12:26 PM

@bill llope Do you have info on the DIRC bar surface roughness that we can include in the geometry? Asking for performance reasons more than physics (unphysically long lived photons are a peformance concern).

And related to this: do you have any plans to enable Mie or Rayleigh scattering (or WLS physics) in the DIRC bars in the near future? Another performance concern is just unused processes that are evaluated at each step without ever actually doing anything due to the absence of relevant optical parameter tables.

4 replies

Wouter Deconinck (he/him) 12:51 PM

https://github.com/eic/epic/pull/1113 enables roughness (0.2 rad)

https://github.com/eic/npsim/pull/67 disables Mie, WLS, WLS2 (keeps Rayleigh) *Edited*

Wouter Deconinck (he/him) last week

And related to this: do you have any plans to enable Mie or Rayleigh scattering (or WLS physics) in the DIRC bars in the near future? Another performance concern is just unused processes that are evaluated at each step without ever actually doing anything due to the absence of relevant optical parameter tables.

4 replies

bill llope last week

will look into it and get back Note we have our dirc annual meeting next week and some of this we may want to discuss then. how urgently is this needed? cheers

Wouter Deconinck (he/him) last week

If it's not used right now we might just go ahead and disable the processes. We can always enable again. Just want to make sure it's clear to people doing studies that the process might be disabled regardless of what parameters they add.

But we've learned not to wait for promised studies for these things (this is not specific to the DIRC)

The part that's more urgent is the long photon tails. They are potentially dominant contributions to the already dominant DIRC simulation compute cost.

github.com/eic/epic/pull/1113

eic / epic ☆ 29

Type / to search

Code | Issues 73 | Pull requests 52 | Actions | Projects | Models | Security and quality | Insights | Settings

wdconinc requested your review on this pull request. Add your review

DIRC: add bar lateral surface with UNIFIED model micro-facet roughness #1113

Awaiting approval | Code

Open | wdconinc wants to merge 1 commit into `main` from `dirc-bar-surface-roughness`

Conversation 5 | Commits 1 | Checks 111 | Files changed 2 | +16

wdconinc (Wouter Deconinck) last week — Member

Summary

Add an explicit `G4LogicalSkinSurface` on `bar_vol` (the quartz bars) using the UNIFIED optical model with `ground` finish and `sigma_alpha = 0.02 rad`.

Physics motivation

Currently the bar lateral faces have **no explicit optical surface**, defaulting to perfect `glisur/polished` total internal reflection. This causes an unphysical simulation pathology: optical photons can bounce indefinitely (up to 690 m / 113,000 steps in profiling runs), limited only by the bulk absorption length.

In a real polished quartz bar, surface micro-roughness causes photons near the TIR critical angle to be deflected onto micro-facets at slightly different angles. A fraction of these exceed the critical angle and **refract out of the bar** instead of reflecting. The UNIFIED model with `ground` finish and `sigma_alpha` (Gaussian std-dev of micro-facet normal angle) correctly models this physics.

`sigma_alpha = 0.02 rad` (≈ 1.1°) is the standard value for polished quartz used in hpDIRC simulations (BaBar DIRC, Belle II iTOP papers).

Reviewers – review now – approve now

Copilot

wjllope

At least 1 approving review is required to merge this pull request.

Still in progress? Convert to draft

Assignees – assign yourself

Labels

topic: materials × | topic: PID ×

Projects

Milestone

github.com/eic/npsim/pull/67

eic / npsim

Type / to search

Code | Issues 10 | Pull requests 15 | Actions | Models | Security and quality 14 | Insights | Settings

npsim: disable unused optical processes (OpMieHG, OpWLS, OpWLS2) #67

Awaiting approval | Code

Open wdconinc wants to merge 1 commit into `main` from `disable-unused-optical-processes-v2`

Conversation 1 | Commits 1 | Checks 45 | Files changed 1 | +15

wdconinc (Wouter Deconinck) last week • edited — Member

Problem

Optical processes OpMieHG, OpWLS, and OpWLS2 have no material property tables in the EIC epic detector description and always return DBL_MAX (infinite mean free path). Evaluating them on every optical photon step wastes GPIL time without contributing to the physics.

Solution

Add standard Geant4 UI commands in `commandsConfigure` to deactivate these three processes before the physics list is finalized:

```
/process/optical/processActivation OpMieHG false
/process/optical/processActivation OpWLS false
/process/optical/processActivation OpWLS2 false
```

OpRayleigh is kept active — aerogel in dRICH/pfRICH defines RAYLEIGH tables.

Reviewers – review now – approve now

Copilot

At least 1 approving review is required to merge this pull request.

Still in progress? Convert to draft

Assignees – assign yourself

Labels

Projects

Milestone

Development

Yesterday

Wouter Deconinck (he/him) 4:27 PM

Do you have short term plans to add the HRPPDs to the sensor box in ePIC for the DIRC? I would like to implement this in relatively short term to get better QA on the changes we are making.

b a c k u p

✓	2025	OP kill at creation according to photodetector QE (W. Deconinck) presently the efficiency table is defined in npsim.py itself (should be user stacking action instead?)	not too many OPs
✓	2025	hpDIRC CADD geometry pdf released (K. Cleveland)	geometry known
✓	Feb 2026	dd4hep/npsim geometry modified to match (WJL) issue: https://github.com/eic/epic/issues/1053 PR: https://github.com/eic/epic/pull/1054	geometries agree
✓	Mar 2026	stepping action to record dircbar entrance vector (WJL) MM: https://chat.epic-eic.org/main/pl/86hcat7zi3g7uereeok8swi8na code: https://github.com/eic/ddDircAction	$\vec{p}_{inc}$ for GeoReco
✓	Mar 2026	randomize non-zero polarization in optical photon gun (W. Deconinck) MM: https://chat.epic-eic.org/main/pl/34f6bnjhy7n5tfeirfhe8t11jw PR: https://github.com/AIDASoft/DD4hep/pull/1598	LUT gen. physical
	Mar 2026	npsim crash when generating optical photons MM: https://chat.epic-eic.org/main/pl/34f6bnjhy7n5tfeirfhe8t11jw	Fixed (wouter)
✓	Apr 2026	compile standalone (eicdirc) under shell (WJL) required removal of TSpectrum, and cxx20	backup LUT approach

-> try to debug npsim crash?
-> make standalone geometry in eic-shell same as ePIC CADD
-> implement LUT format from npsim LUT generation
-> **Make root macro to read npsim output, and LUTs, and make theta-C.**