Cherenkov Calibration and Analysis

Simona Malace

Cherenkov Calibration

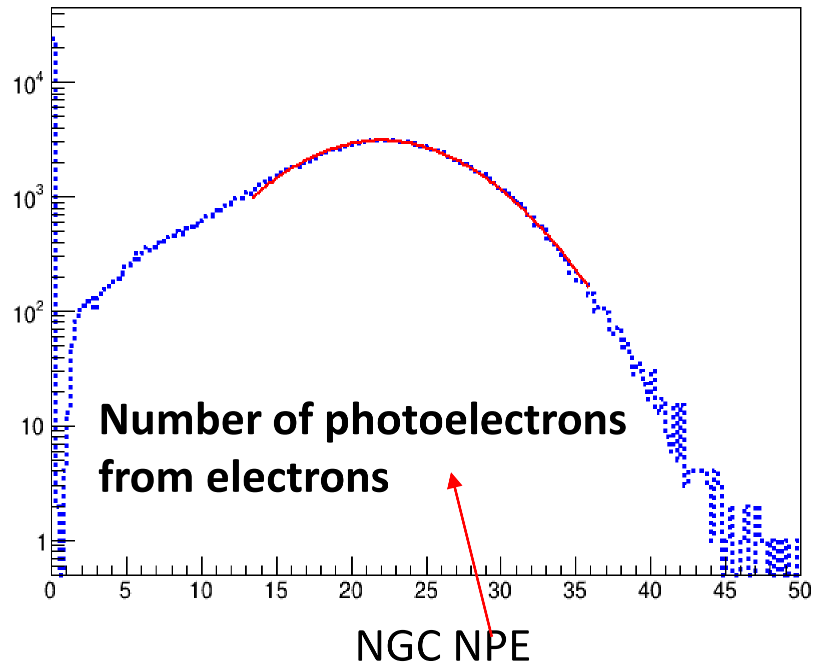
Why we do it?

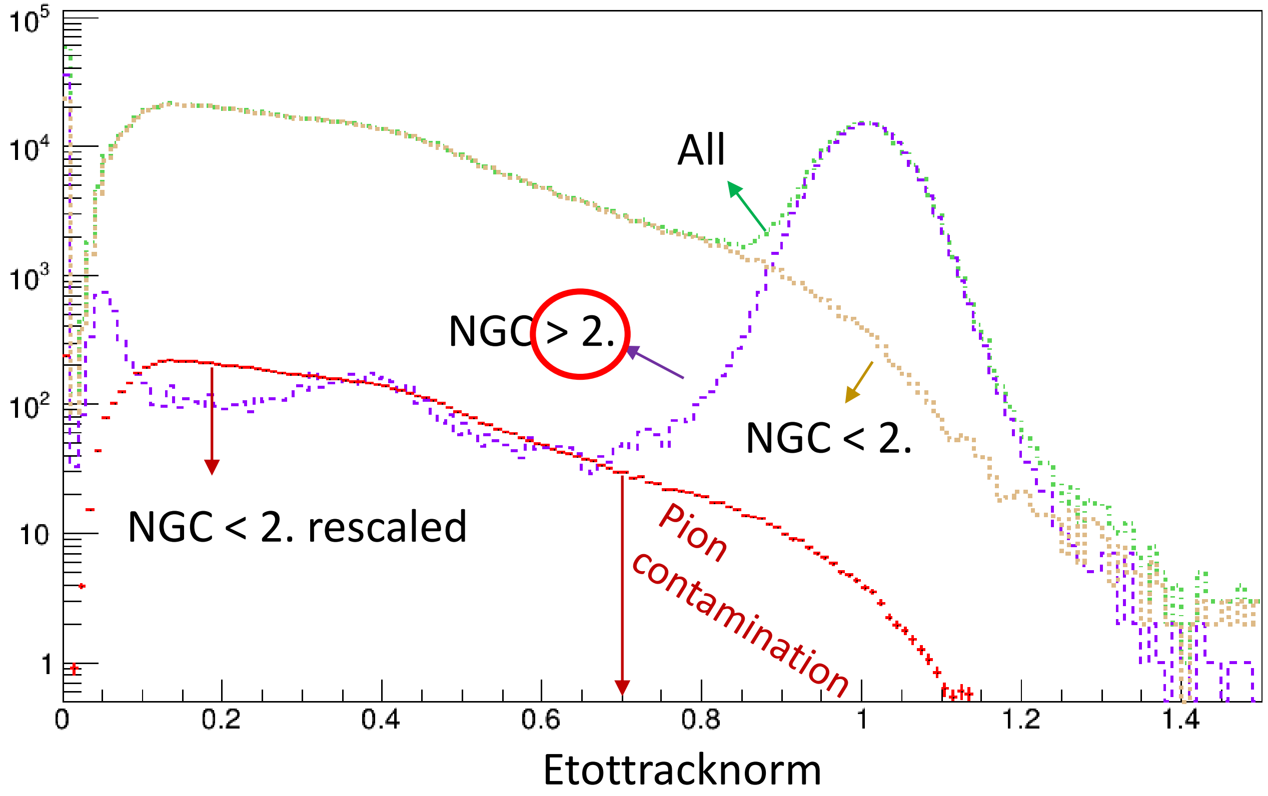
→ To quantify the Cherenkov detector response to passage of various particles

HGC: C_4F_8O at 1 atm

NGC: CO_2 at 1 atm

Pions will not produce Cherenkov light in neither gas for run 2788

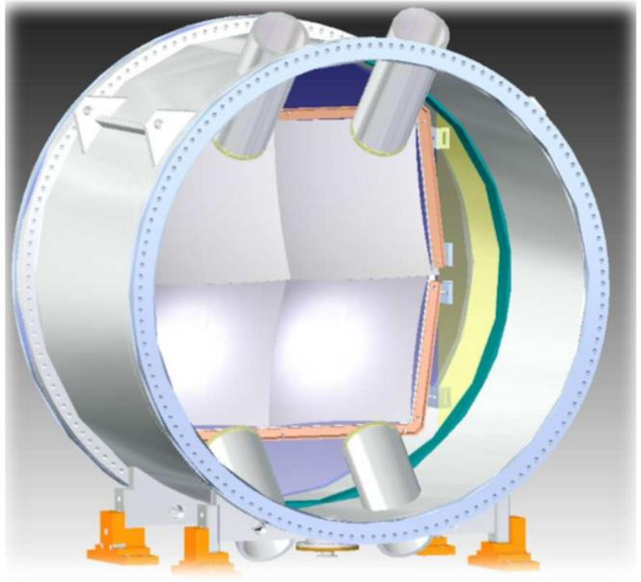




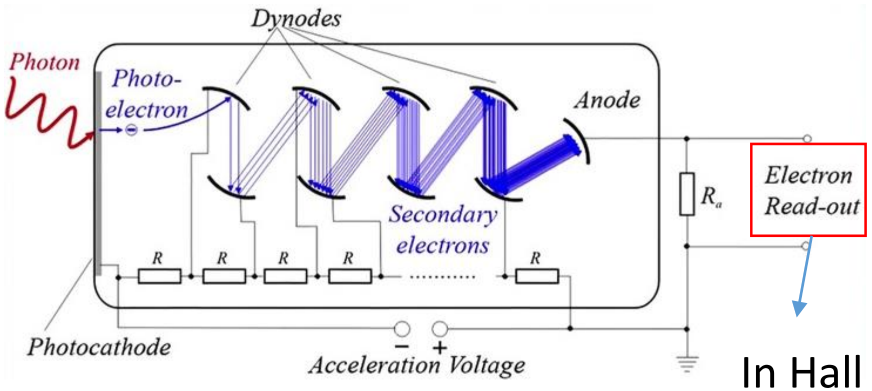
Cherenkov Calibration

<u>What is it</u>?

→ It tells you how much charge (in pC) the PMT will output when one single photoelectron is extracted from the photocathode



Photomultiplier tube (PMT)



If you change the PMTs High Voltage you will have to recalibrate!

In Hall C detected with FADC250

https://halldweb.jlab.org/DocDB/0030/003064/001/Pooser_Dissertation_DocDB.pdf

```
CALIB  CUTS  GEOM
[git:master*]
ifarm1401.jlab.org> cd CALIB/
ls
more [git:master*]
ifarm1401.jlab.org> ls
pngcer_calib_jpsi.param  pngcer_calib.param
[git:master*]
ifarm1401.jlab.org> more pngcer_calib.param
; all stat from run 2549
pngcer_adc_to_npe = 0.529763, 0.554375, 0.413854, 0.43999
```

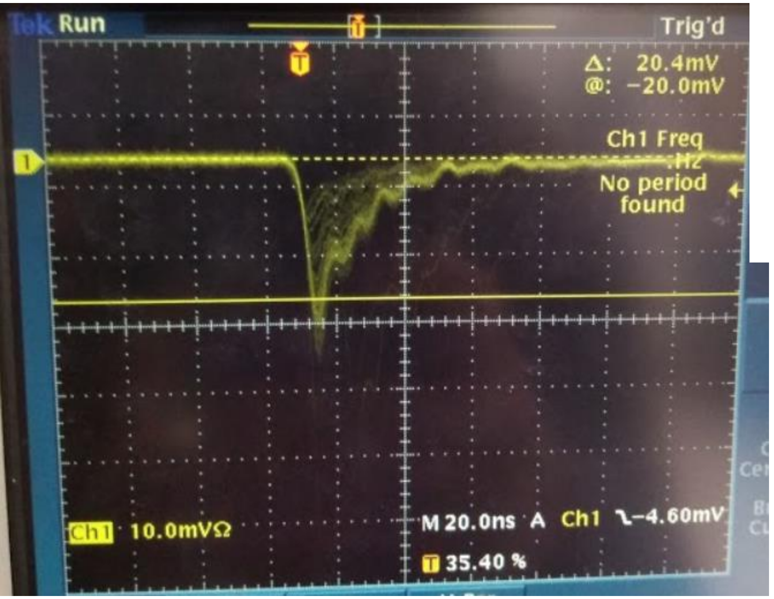
→ You are in hallc-replay/PARAM/SHMS/NGCER

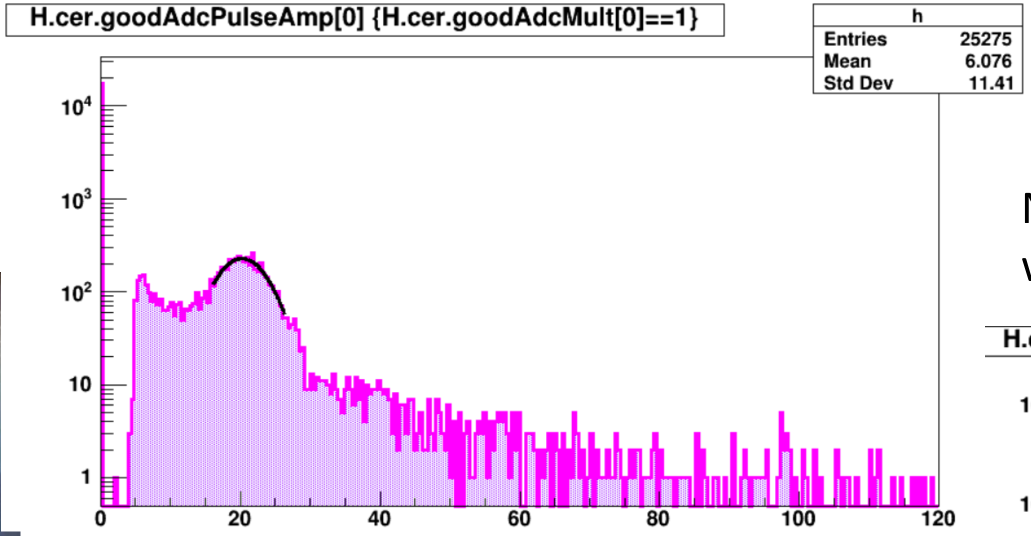
→ 1/charge per one photolelectron for NGC PMT4

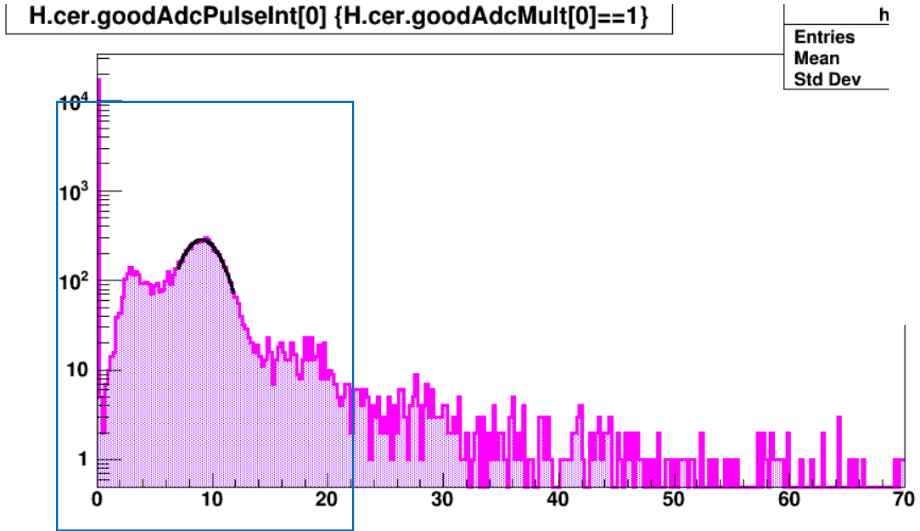
Cherenkov Calibration

<u>How do you do it</u>?

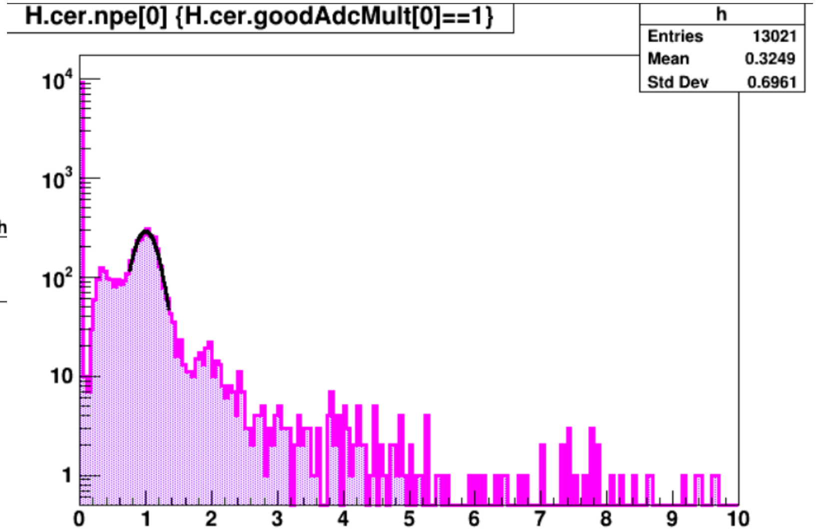
→ Method 1 applicable when you can unambiguously identify the Single PhotoElectron (SPE): fit the pulse integral from SPE and put 1/fit result in the PARAM file







NPEs after you replayed your run with the new calibration coefficients



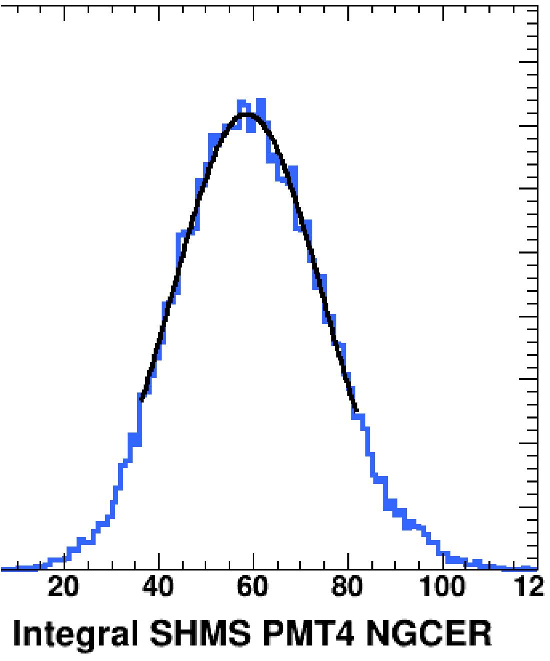
Cherenkov Calibration

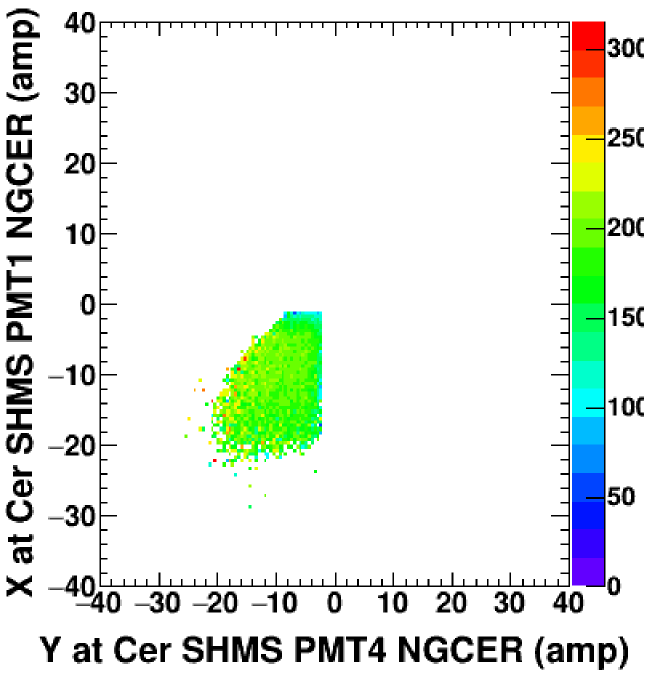
How do you do it?

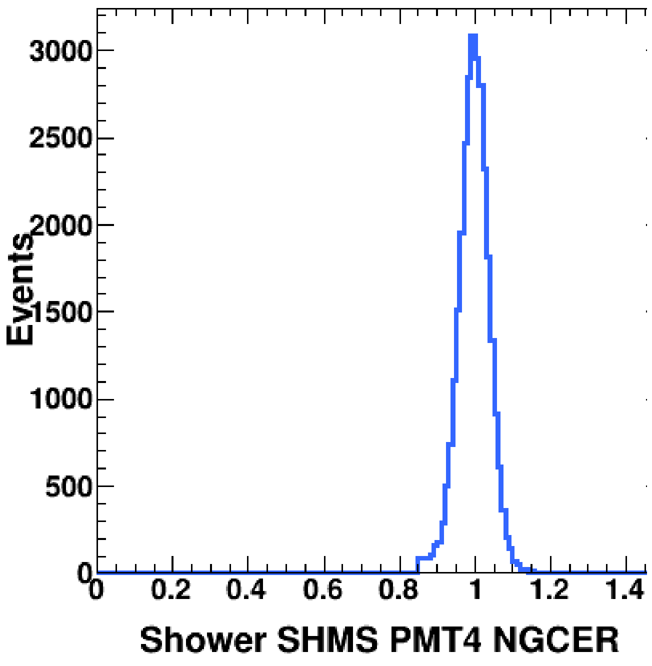
→ Method 2 applicable when you **cannot** unambiguously identify the Single PhotoElectron (SPE): fit the charge integral from electrons only and (mean/sigma)^2 will be the NPE; then take your fit results for the charge integral, divide it by NPE and out 1/that number in the PARAM file

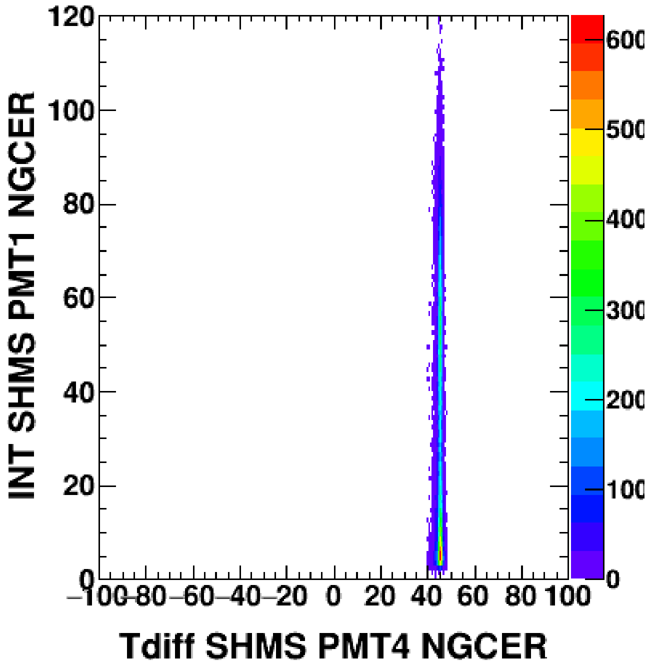
This method is not as nice and you should spend some time doing systematic studies

https://logbooks.jlab.org/entry/3650663









Cherenkov Analysis

Before you do anything:

→ You have to make sure you select the right hit per counter, per event (reference time cuts and timing cuts)

Say what?

Detector data structure in Hall C: we make hit lists per detector, per event

Cherenkov hit list for ev. 98462

```
ev, hit, counter, pulsetime, starttime, timediff: 98462 0 1 27.625 12.7414 -14.8836
ev, hit, counter, pulsetime, starttime, timediff: 98462 1 1 109.125 12.7414 -96.3836
ev, hit, counter, pulsetime, starttime, timediff: 98462 2 1 260.062 12.7414 -247.321
ev, hit, counter, pulsetime, starttime, timediff: 98462 3 2 45.1875 12.7414 -32.4461
ev, hit, counter, pulsetime, starttime, timediff: 98463 0 2 311.812 42.3889 -269.424
ev, hit, counter, pulsetime, starttime, timediff: 98466 0 3 46.0625 32.4551 -13.6074
ev, hit, counter, pulsetime, starttime, timediff: 98467 0 1 50.875 31.5731 -19.3019
ev, hit, counter, pulsetime, starttime, timediff: 98467 1 2 43.8125 31.5731 -12.2394
ev, hit, counter, pulsetime, starttime, timediff: 98468 0 1 50.75 32.1592 -18.5908
ev, hit, counter, pulsetime, starttime, timediff: 98468 1 3 46.875 32.1592 -14.7158
ev, hit, counter, pulsetime, starttime, timediff: 98468 2 4 47.125 32.1592 -14.9658
```

Event associated with a trigger

One pulse or hit resolved by the FADC250 for that event

The time of that pulse or hit

The time that the hodoscope is reporting

The time difference between the hodoscope time and the Cherenkov pulse time for that hit – it is the “thing” you put timing cuts on

Cherenkov Analysis

THcCherenkov.cxx in /hcana/src/

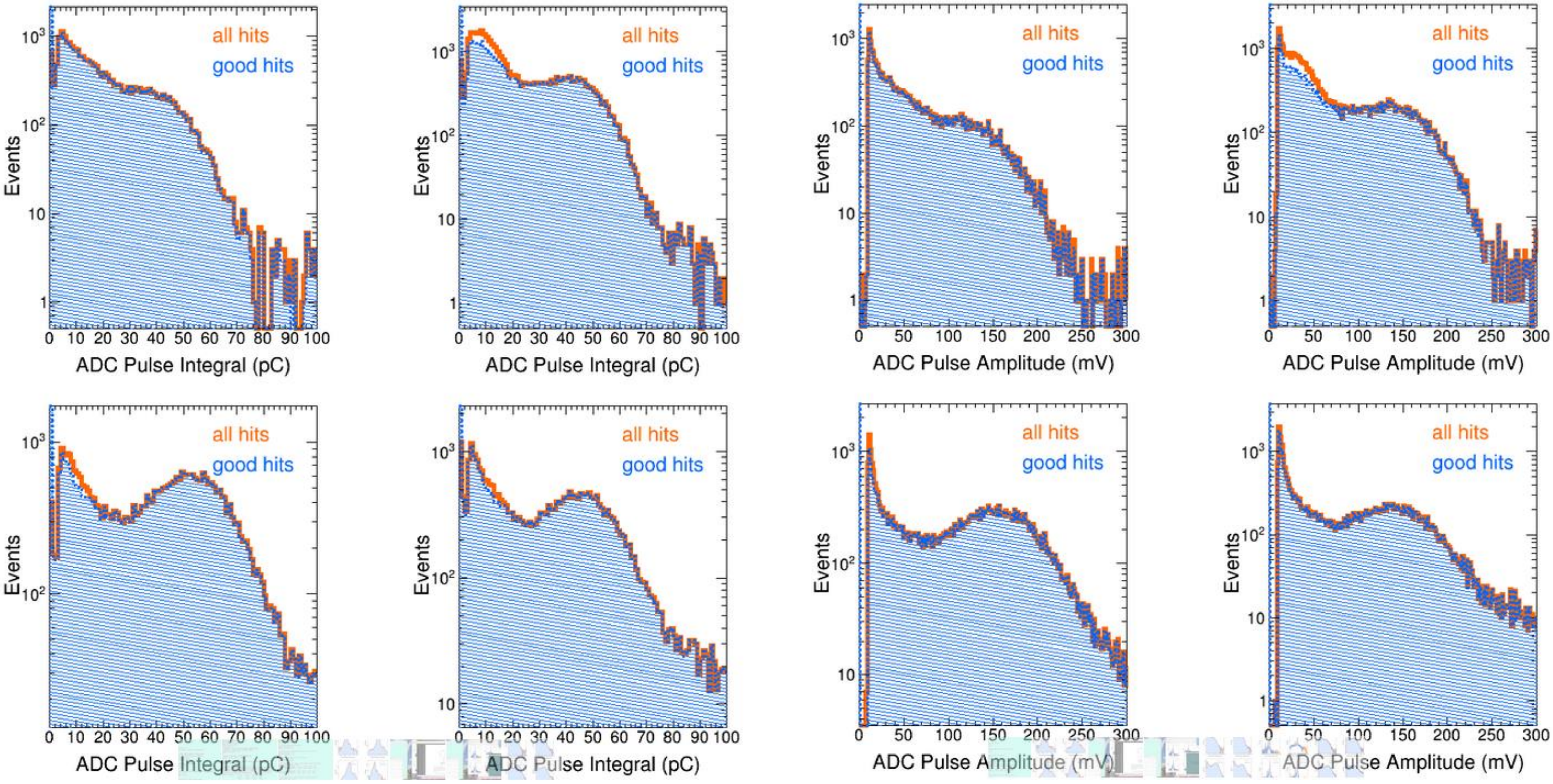
```
 Bool_t    pulseTimeCut = adctdcdiffTime > fAdcTimeWindowMin[npmt] && adctdcdiffTime < fAdcTimeWindowMax[
t];
 if (!errorFlag)
   {
     fGoodAdcMult.at(npmt) += 1;
   }
 if (!errorFlag && pulseTimeCut && pulseAmp > fAdcPulseAmpTest[npmt]) {
    fAdcGoodElem[npmt]=ielem;
    fAdcPulseAmpTest[npmt] = pulseAmp;
 }

/ Loop over the npmt
or(Int_t npmt = 0; npmt < fNelem; npmt++) {
 Int_t ielem = fAdcGoodElem[npmt];
 if (ielem != -1) {
 Double_t pulsePed     = ((THcSignalHit*) frAdcPed->ConstructedAt(ielem))->GetData();
 Double_t pulseInt     = ((THcSignalHit*) frAdcPulseInt->ConstructedAt(ielem))->GetData();
 Double_t pulseIntRaw  = ((THcSignalHit*) frAdcPulseIntRaw->ConstructedAt(ielem))->GetData();
 Double_t pulseAmp     = ((THcSignalHit*) frAdcPulseAmp->ConstructedAt(ielem))->GetData();
 Double_t pulseTime    = ((THcSignalHit*) frAdcPulseTime->ConstructedAt(ielem))->GetData();
Double_t adctdcdiffTime = StartTime-pulseTime;
 // By default, the last hit within the timing cut will be considered "good"
   fGoodAdcPed.at(npmt)          = pulsePed;
   fGoodAdcHitUsed.at(npmt)          = ielem+1;
   fGoodAdcPulseInt.at(npmt)     = pulseInt;
   fGoodAdcPulseIntRaw.at(npmt)  = pulseIntRaw;
   fGoodAdcPulseAmp.at(npmt)     = pulseAmp;
   fGoodAdcPulseTime.at(npmt)    = pulseTime;
   fGoodAdcTdcDiffTime.at(npmt)    = adctdcdiffTime;
```

Cherenkov Analysis

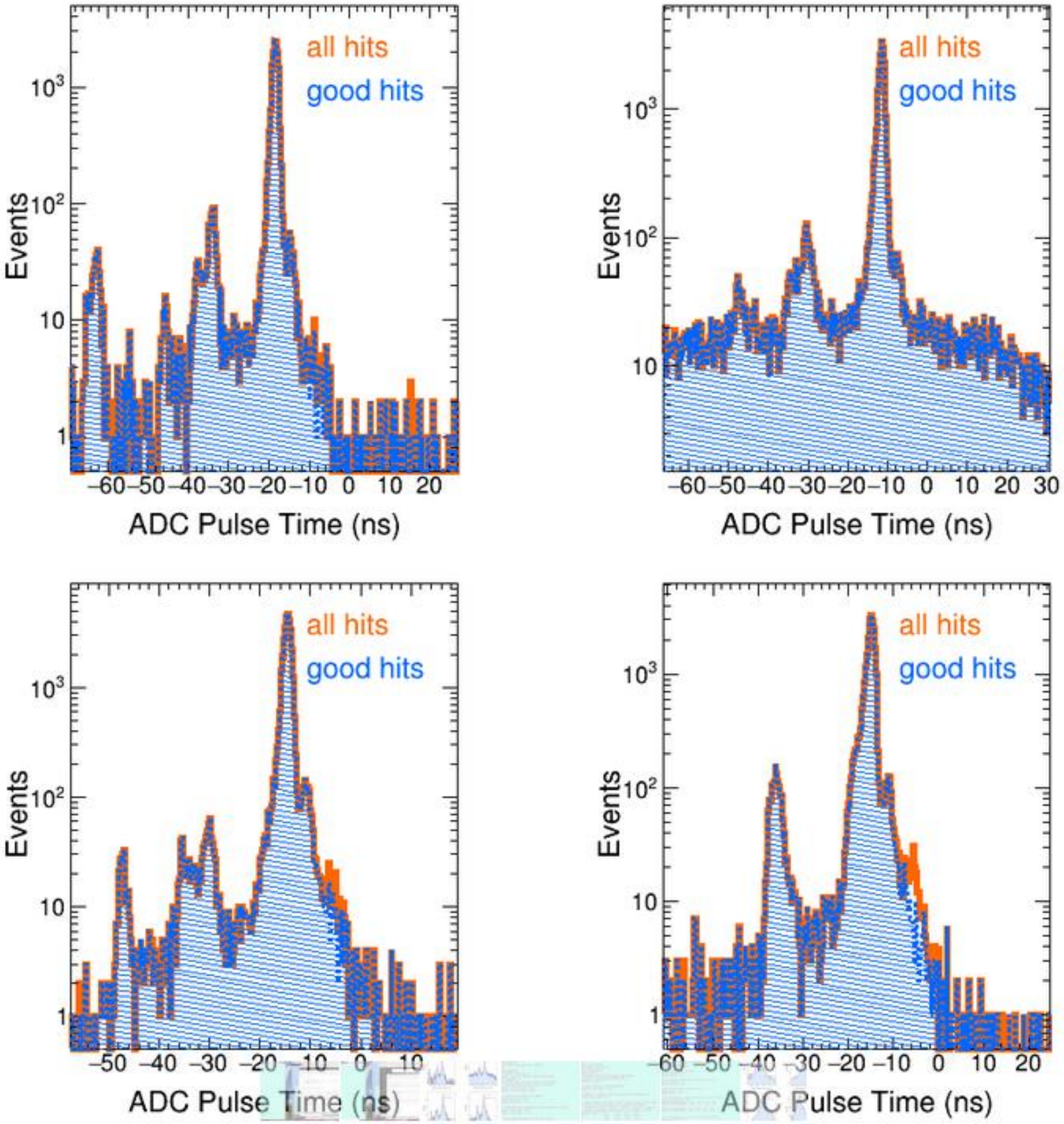
My code that makes timing (and other) distributions for the Cherenkov detector – will give to Melanie

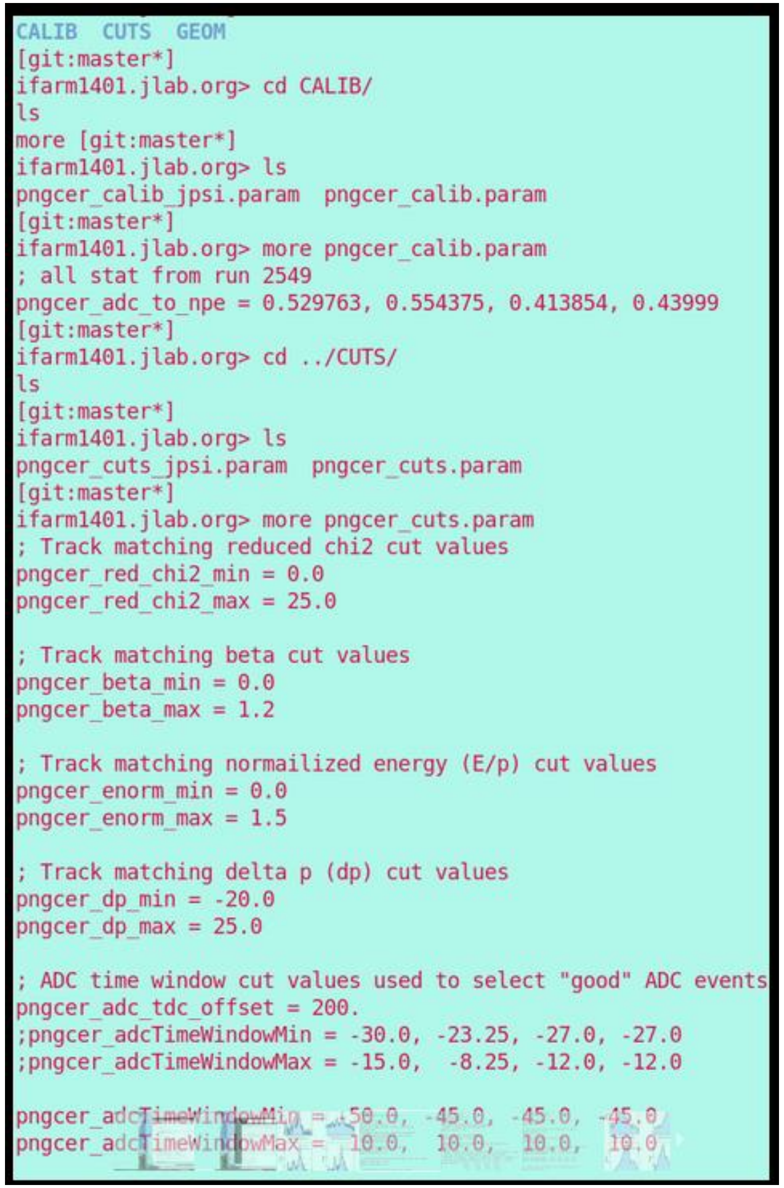
```
chain.SetBranchAddress("Ndata.P.ngcer.adcPulseInt", &ndata_ngcer_pulseint);

chain.SetBranchAddress("P.ngcer.adcCounter", &ngcer_counter[0]);
chain.SetBranchAddress("P.ngcer.adcPulseInt", &ngcer_pulseint[0]);
chain.SetBranchAddress("P.ngcer.adcPulseAmp", &ngcer_pulseamp[0]);
chain.SetBranchAddress("P.ngcer.adcPulseTime", &ngcer_pulsetime[0]);


chain.SetBranchAddress("P.ngcer.goodAdcPulseInt", &ngcer_goodpulseint[0]);
chain.SetBranchAddress("P.ngcer.goodAdcPulseAmp", &ngcer_goodpulseamp[0]);
chain.SetBranchAddress("P.ngcer.goodAdcTdcDiffTime", &ngcer_gooddifftime[0]);

chain.SetBranchAddress("P.ngcer.adcErrorFlag", &ngcer_errflag[0]);


chain.SetBranchAddress("P.cal.etottracknorm", &etotnorm);
chain.SetBranchAddress("P.ngcer.npeSum", &ngcer);
chain.SetBranchAddress("P.hod.betanotrack", &beta);
chain.SetBranchAddress("P.dc.InsideDipoleExit", &inside);

chain.SetBranchAddress("P.hod.starttime", &starttime);
```

Cherenkov Analysis



Cherenkov Analysis





```
CALIB  CUTS  GEOM
[git:master*]
ifarm1401.jlab.org> cd CALIB/
ls
more [git:master*]
ifarm1401.jlab.org> ls
pngcer_calib_jpsi.param  pngcer_calib.param
[git:master*]
ifarm1401.jlab.org> more pngcer_calib.param
; all stat from run 2549
pngcer_adc_to_npe = 0.529763, 0.554375, 0.413854, 0.43999
[git:master*]
ifarm1401.jlab.org> cd ../CUTS/
ls
[git:master*]
ifarm1401.jlab.org> ls
pngcer_cuts_jpsi.param  pngcer_cuts.param
[git:master*]
ifarm1401.jlab.org> more pngcer_cuts.param
; Track matching reduced chi2 cut values
pngcer_red_chi2_min = 0.0
pngcer_red_chi2_max = 25.0

; Track matching beta cut values
pngcer_beta_min = 0.0
pngcer_beta_max = 1.2

; Track matching normailized energy (E/p) cut values
pngcer_enorm_min = 0.0
pngcer_enorm_max = 1.5

; Track matching delta p (dp) cut values
pngcer_dp_min = -20.0
pngcer_dp_max = 25.0

; ADC time window cut values used to select "good" ADC events
pngcer_adc_tdc_offset = 200.
;pngcer_adcTimeWindowMin = -30.0, -23.25, -27.0, -27.0
;pngcer_adcTimeWindowMax = -15.0,  -8.25, -12.0, -12.0

pngcer_adcTimeWindowMin = -50.0, -45.0, -45.0, -45.0
pngcer_adcTimeWindowMax =  10.0,  10.0,  10.0,  10.0
```