Stony Brook University

FiTQun tuning for Hyper-K and nuPRISM

Cristóvão Vilela

FiTQun! Workshop
TRIUMF

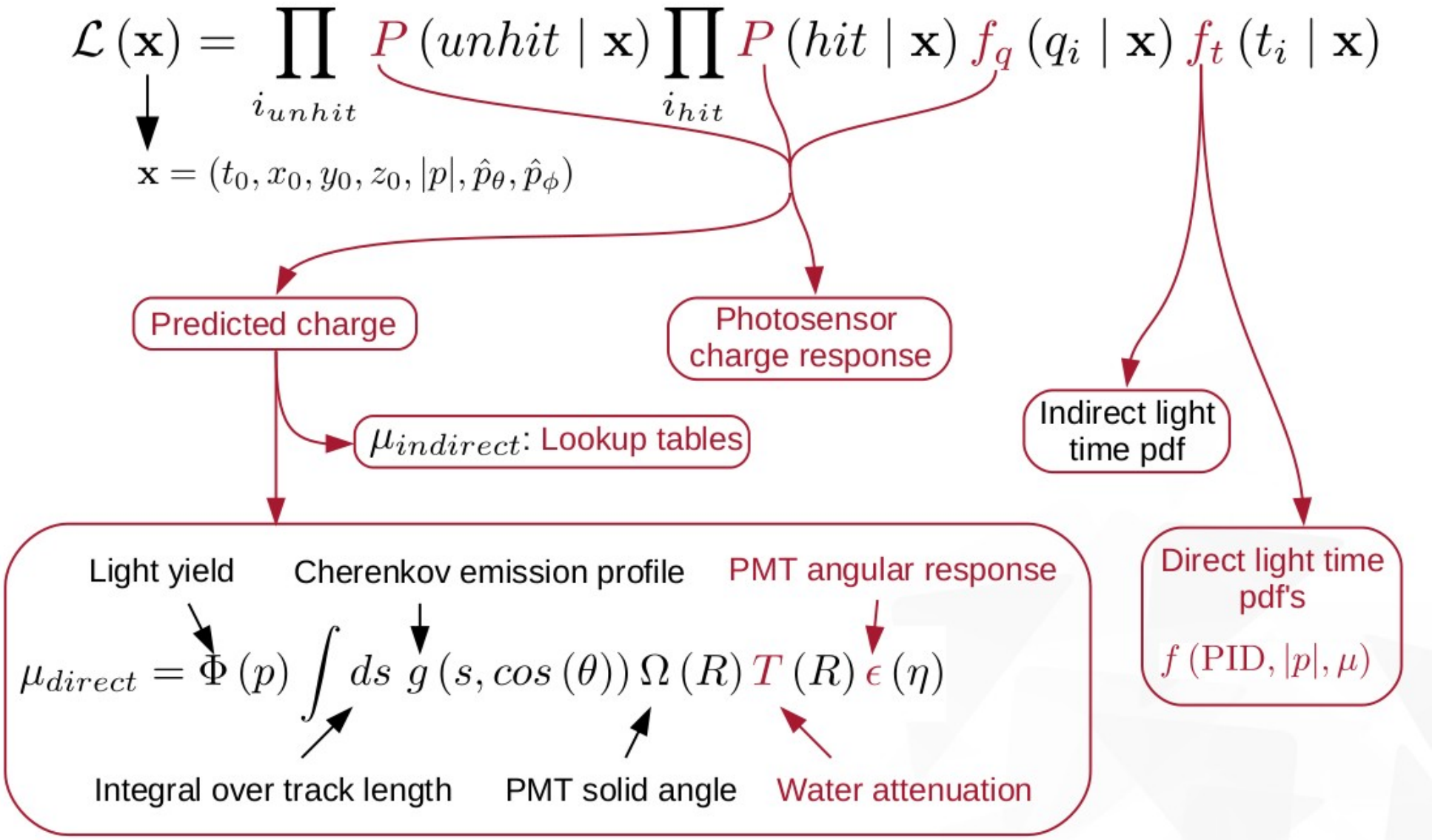
August 20 2015

Introduction

- We are using fiTQun reconstruction on Hyper-K and nuPRISM
  - FiTQun v4r2 was able to read in WCSim MC (WCSimWrapper)
    - However, the SK1 tune was used – had to overwrite SK1 tune to properly reconstruct WCSim geometries
  - Since then we have started producing tunes for multiple WCSim configurations
    - A version of WCSim for fiTQun tuning is available at
      - https://github.com/cvilelasbu/WCSim/tree/feature/fiTQunTuningNuPRISM
  - We have also adapted fiTQun so that the different WCSim tunes can be easily used

- In this talk
  - Overview of the tuning procedure
  - Changes to fiTQun for integration with WCSim

What needs to be tuned?



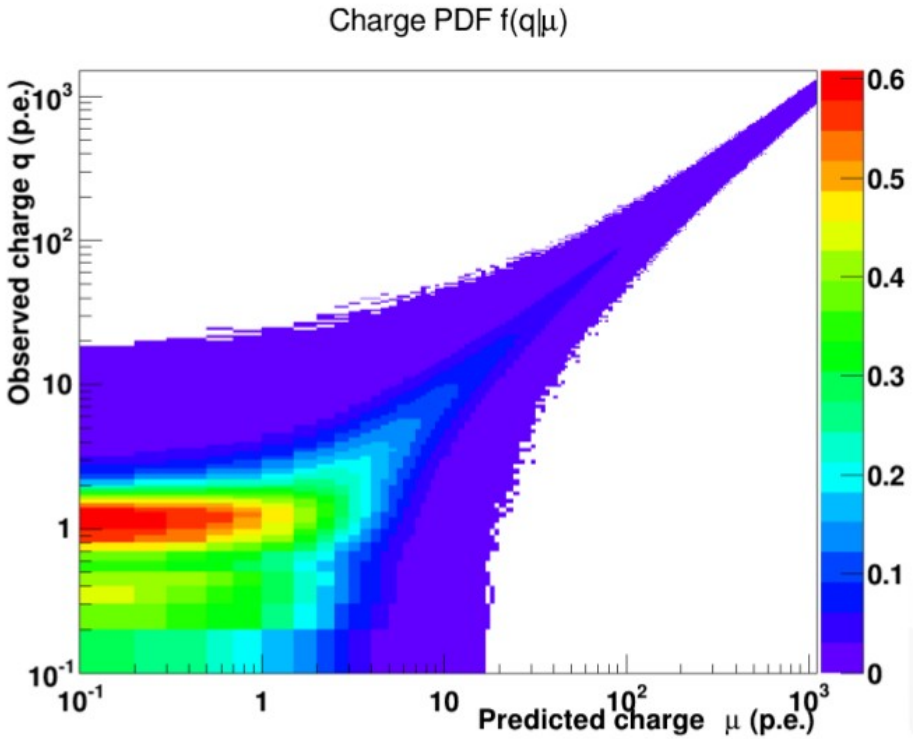
Photosensor charge response

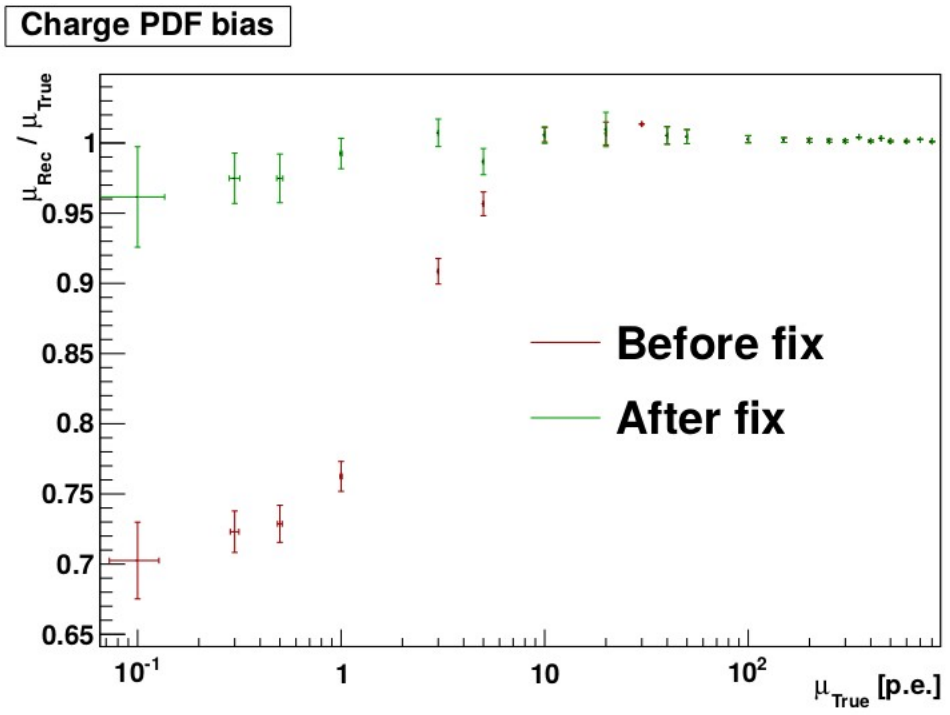
- Want to know, for a given true number of predicted p.e. (**μ**), what is the distribution of observed charge

- Generate dedicated MC sample
  - No particles / Cherenkov photons generated
  - Zero dark rate
  - For each PMT directly add N p.e. to the raw hit collection
    - N = Poisson (**μ**)
  - Plot distribution of digitised charge
  - Fit with polynomials and pass coefficients to fiTQun
  - Check result by using fiTQun to fit a uniform charge distribution over the detector
    - Should get back **μ**

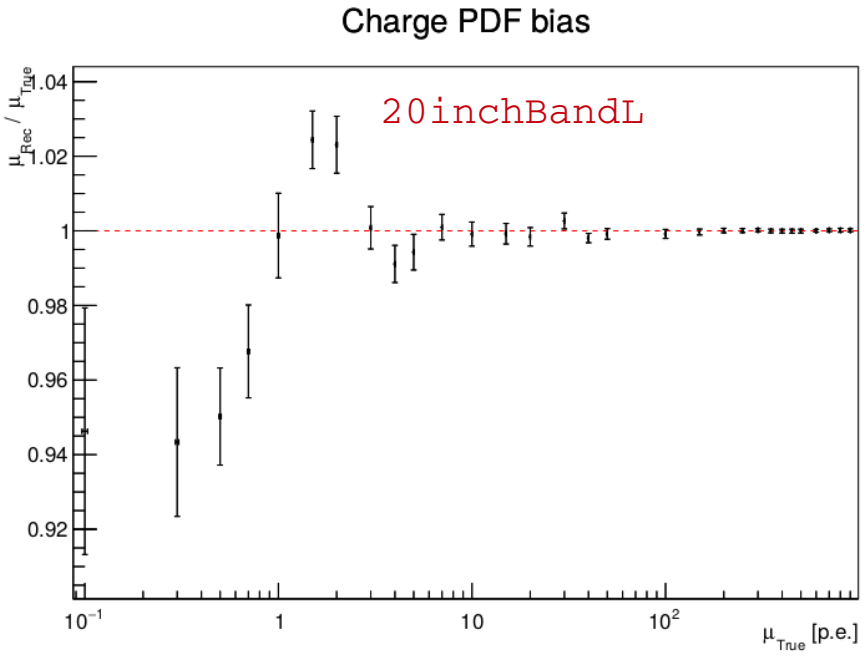
Photosensor charge response

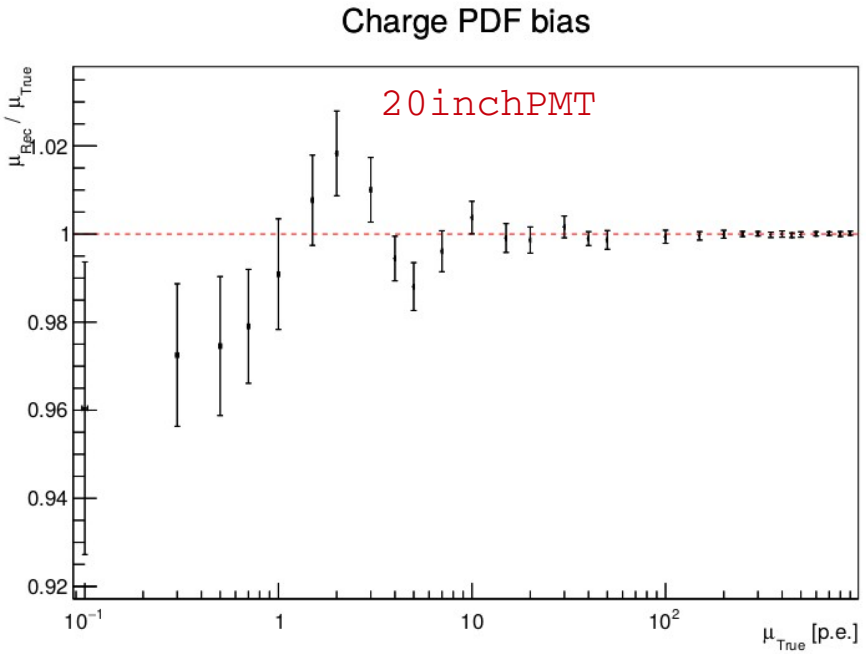
Get charge response from WCSim → Test with fiTQun

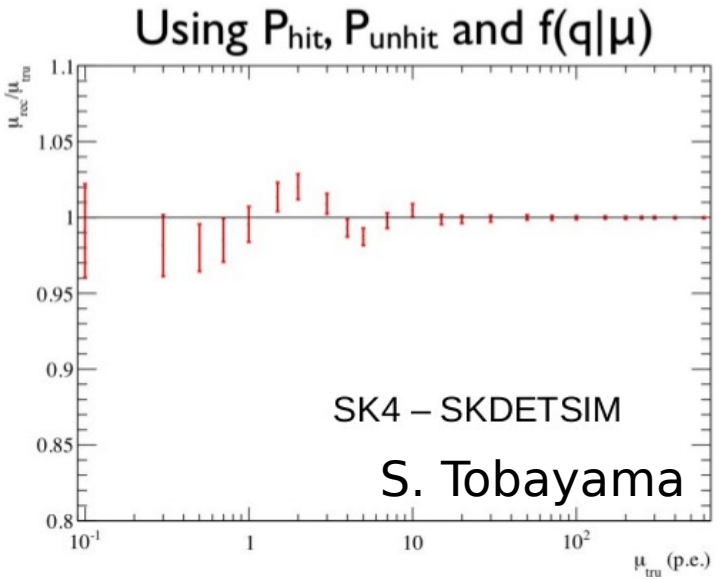




Photosensor charge response

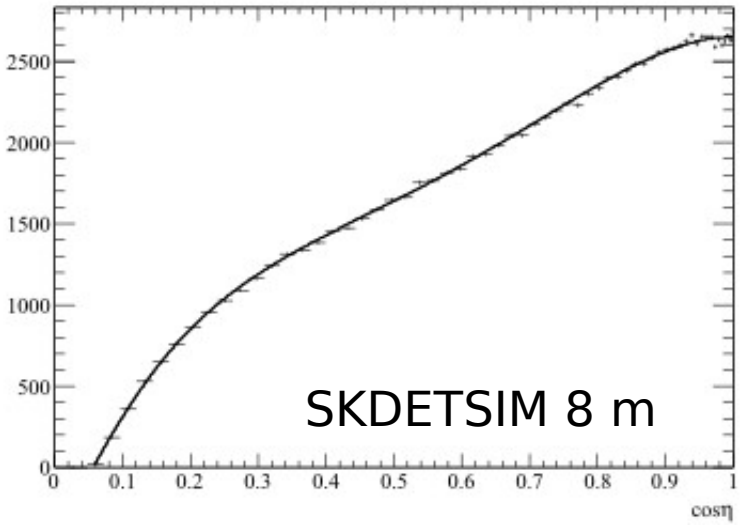


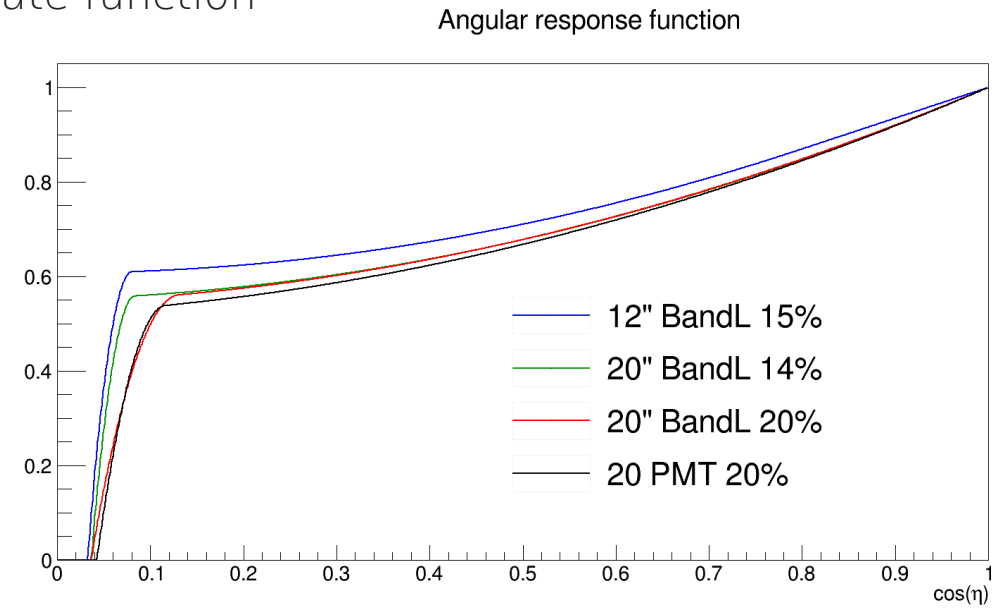




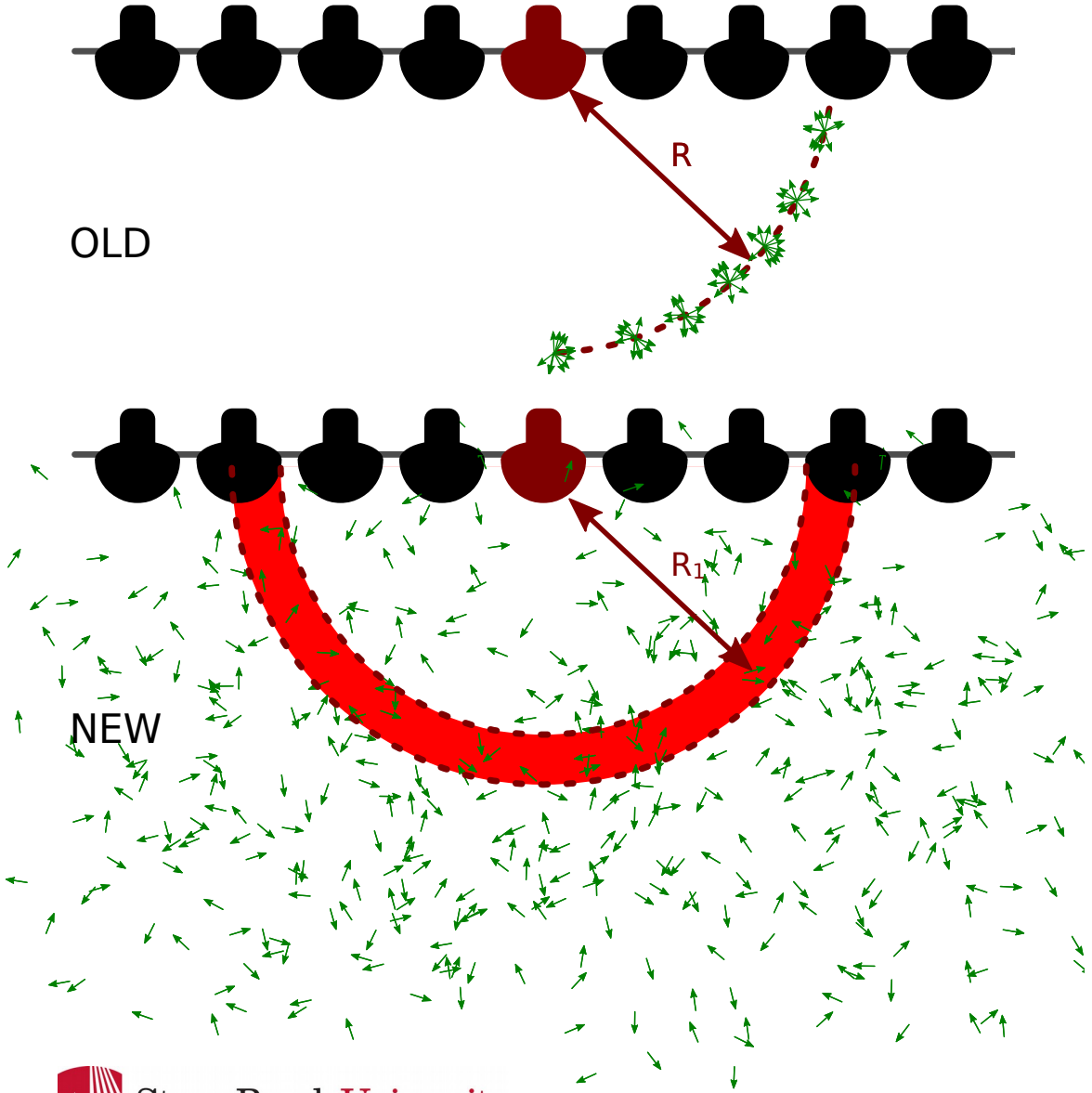
Angular response function

- This function provides a normalisation of the light yield as a function of the angle between the position of the charged particle and the axis of the PMT
  - Captures shape and material properties of PMT (reflectivity, etc) as well as shadowing effects due to PMT spacing
  - Initially found a few differences between SKDETSIM and WCSim
    - Identified a bug in how the number of p.e.'s was being extracted from WCSim
    - Developed new method to calculate function





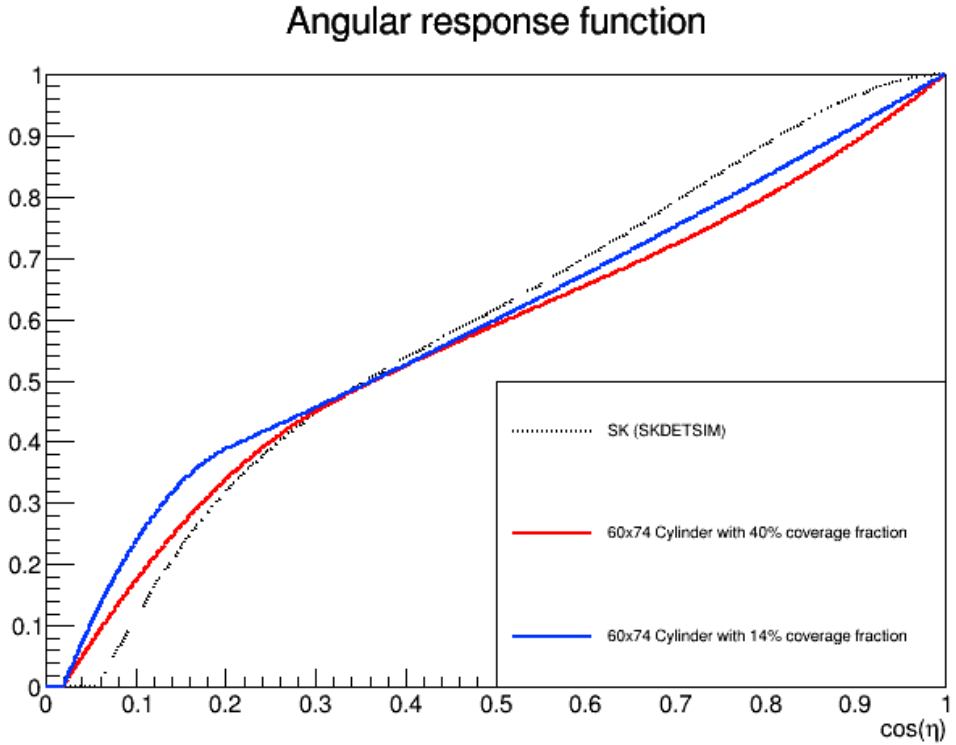
Angular response function

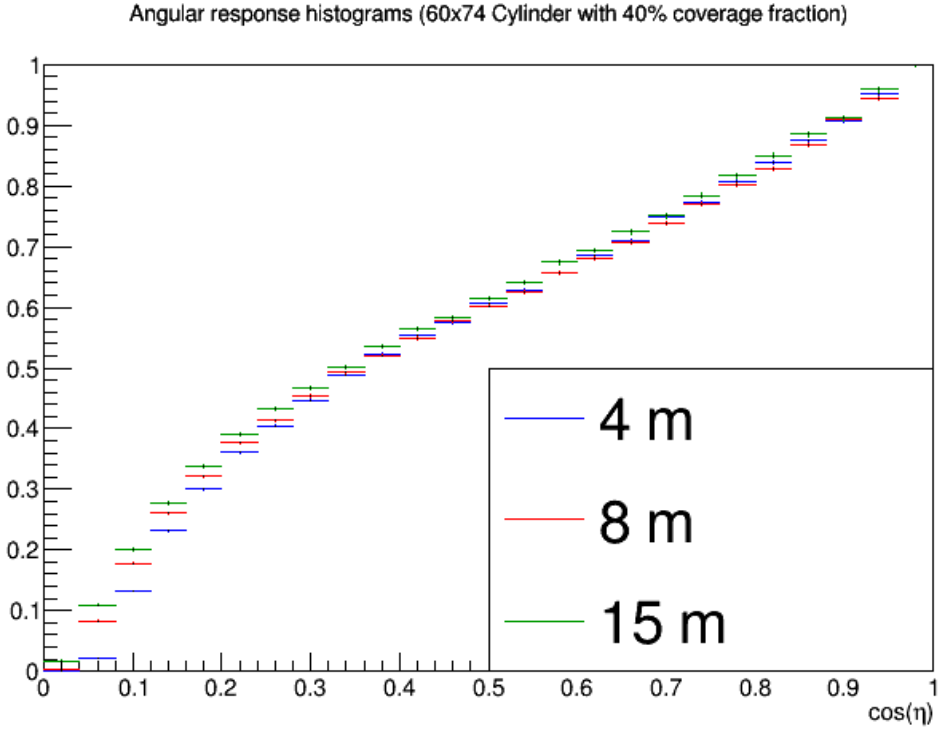


- Before we were sampling angles discreetly by generating isotropic 3MeV electron "bombs" at several points of constant distance from a PMT
  - Chose a particular PMT (middle of endcap)
  - θ was randomised for SKDETSIM but not for first iteration of WCSim
- Now use sample generated all over the detector and select events where particle vertex is in spherical shell around PMTs
  - Average over all PMTs
    - Or select barrel only, endcap only
  - Can sample several radii easily

Angular response function

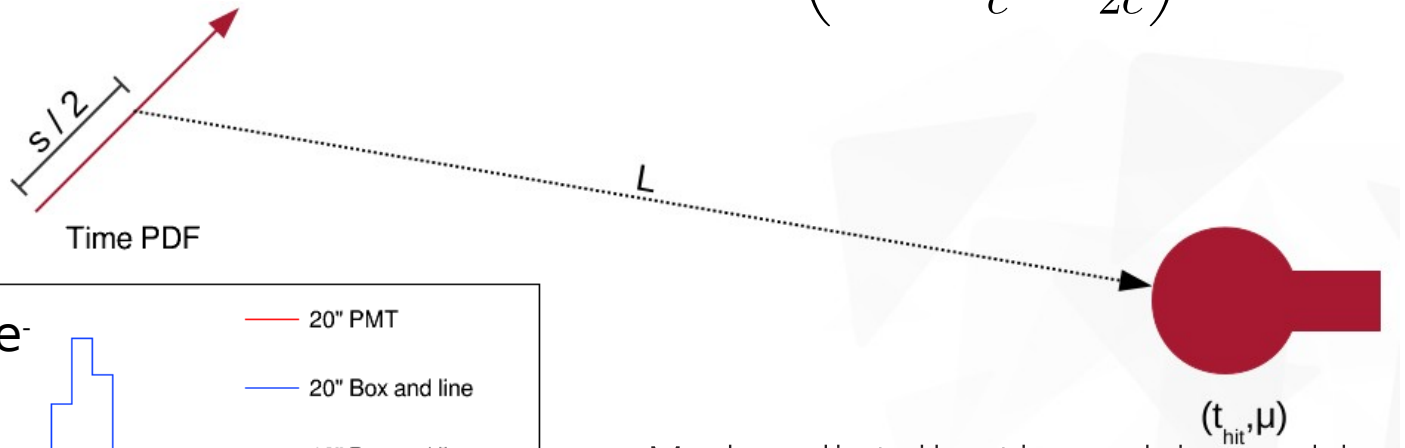
- New angular response functions are closer to SKDETSIM
  - No shoulder, in particular for the 40% coverage fraction configuration
- There is no significant dependence with radius
  - Except at shallow angles

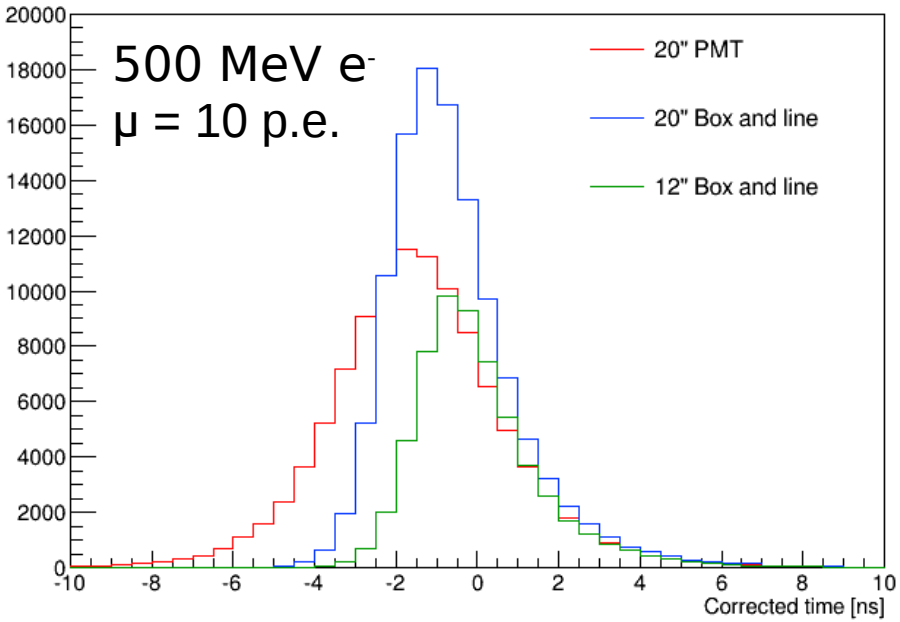




Photosensor time response

- Generate particle gun MC without scattering, reflections, decays or hadronic interactions
- Calculate corrected time $t_{corr} = t_0 - \left(t_{hit} - \frac{Ln}{c} - \frac{s}{2c} \right)$



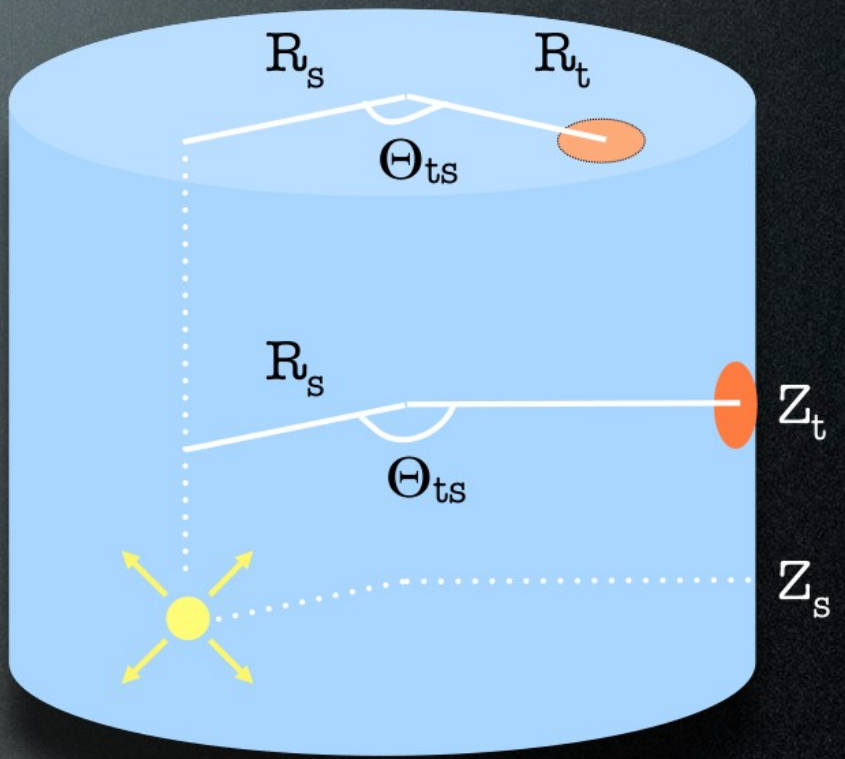


- Make distributions binned in momentum and predicted charge at the PMT
- Fit with Gaussian curves and give to fiTQun
  - New WCSim PMTs have non-Gaussian response

Scattered and reflected light

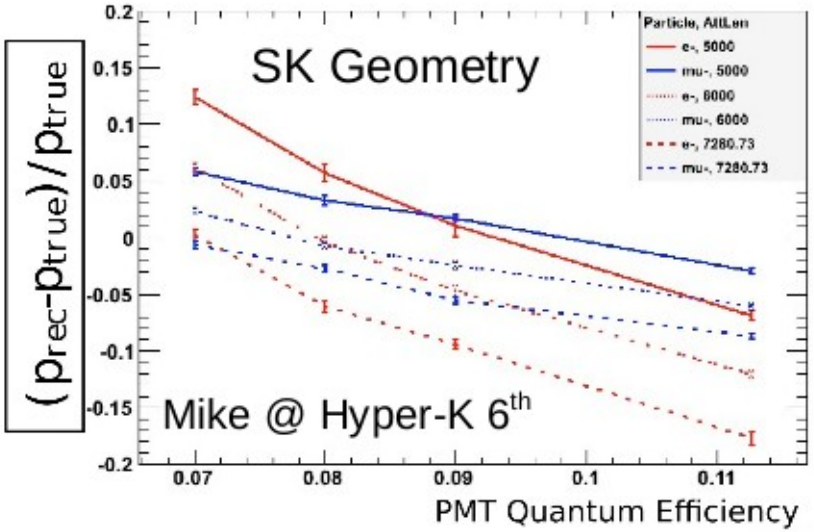
- Currently assume a **cylindrical geometry**
- Tabulate $\frac{d\mu^{indirect}}{d\mu^{direct,iso}}$ (= A_{scat})
  - Source direction (θ_s, ϕ_s)
  - Source position (Θ_{ts}, R_s, Z_s)
  - Z_t for PMTs on the sides
    - $A_{side}(\theta_s, \phi_s, \Theta_{ts}, R_s, Z_s, Z_t)$
  - R_t for PMTs on the ends
    - $A_{end}(\theta_s, \phi_s, \Theta_{ts}, R_s, Z_s, R_t)$
- Must tabulate 6-dimensional scattering tables every time the geometry or optical properties are changed

Mike @ Hyper-K 6th



“Scalar” parameters

- Find which combination of parameters gives the least biased reconstruction



- For example:
  - Water attenuation length
    - SK4: 7496 cm
    - WCSim: 5900 cm
  - Effective quantum efficiency
    - SK4: 0.1082
    - WCSim 12inch_BandL: 0.156

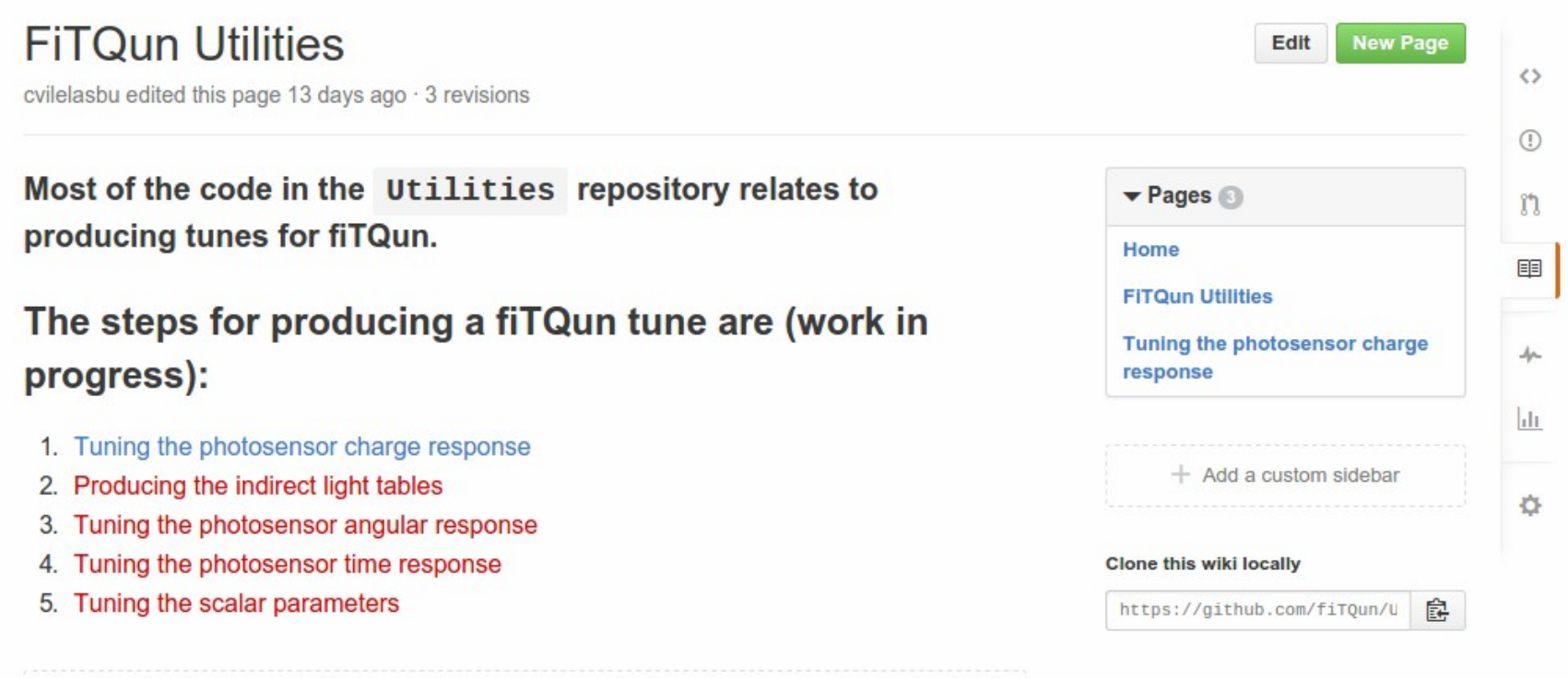
- For the SKDETSIM tune, these parameters were obtained from “first principles” (e.g., measuring light loss vs r for attenuation length, etc)
- Will develop equivalent tools for WCSim

Tuning procedure summary

- The SKDETSIM tuning procedure has been translated to WCSim
  - New developments with angular response function will hopefully fix some of the biases we have seen
    - ~6% e- momentum bias at 7 m to the wall (in old Hyper-K test configuration)
  - Some improvements can be done
    - Mainly developing the tools for extracting the water attenuation length, effective quantum efficiency
    - Angular response function: f(r) ? Barrel and Endcap separately?
    - PMT time response: Non-gaussian time response for new WCSim tubes
- At the moment we are finishing tunes for the two "new" Hyper-K test configurations
  - 60 x 74 m Cylinder with 20" Box & Line PMTs at 40% and 14% coverage fractions
- Also completing a re-tune of nuPRISM with new angular response functions
- Some of the code for this is up on fiTQun/Utilities and instructions are being written in wiki

Tuning instructions

https://github.com/fiTQun/Utilities/wiki/FiTQun-Utilities



More coming soon ...

Multi-configuration fiTQun

- With multiple WCSim tunes available for fiTQun, we need a way to let the user choose what configuration to use
  - With further development of WCSim/fiTQun this might become automatic
- The idea right now is to provide tunes which have the WCSim configuration or PMT name as a suffix:
  - cPDFpar_20inchBandL.root
  - 211_tpdfpar_nuPRISM.root
- Then have parameter override files where two parameters are specified:
  - `WCSimConfig`
  - `WCSimPMT`
  - And any additional parameter that might need to be adjusted: `QEEffWCSim`, `DarkRateWCSim`, etc
- One core fiTQun change was required
  - The angular response function used to be hard-coded, and the same for all SK eras

New treatment of angular response

```
-double fiTQun::epsilon(double coseta){//make sure coseta can be negative!!
-  //  return 1.;
-
-//  -0.183124  +  3.4469 *x   +  -4.6833 *x^2  (for 0.05 < x < 0.196688)
-//  -0.207593  +  4.03995 *x   +  -8.99093 *x^2   +  10.6753 *x^3   +  -4.5175 *x^4  (for 0.196688 < x < 0.986394)
-//  0.879163  +  0.120837 *x  (for 0.986394 < x < 1)
-
-  if (coseta < 0.196688){
-    double val = -0.183124+coseta*(3.4469-coseta*4.6833);
-    if (val > 0.) {
-      return val;
-    } else {
-      return 0.;
-    }
-  } else if (coseta < 0.986394) {
-    return -0.207593+coseta*(4.03995+coseta*(-8.99093+coseta*(10.6753-coseta*4.5175)));
-  } else {
-    return 0.879163+coseta*0.120837;
-  }
-
-}
-
 double fiTQun::J(double R, double coseta, double Zsrc, double Zpmt){//divided by 2pi
-  return Omega(R)*T(R,Zsrc,Zpmt)*epsilon(coseta);
 }
```

```
  double fiTQun::J(double R, double coseta, double Zsrc, double Zpmt){//divided by 2pi
+  return Omega(R)*T(R,Zsrc,Zpmt)*fiTQun_shared::epsilon(coseta);
  }
```

Instead of hard-coded function, TF1 objects are read in and sampled to populate an array.

This array is interpolated whenever the angular response function is called

If no angular response file is provided the old function will be used to populate the angular response array.

```
+double fiTQun_shared::epsilon(double coseta) {
+  double dix=coseta*epsilon_binwrecp;
+
+  if (!(dix >= 0. && dix < nEpsilonBins)) {
+    return 0;
+  }
+
+  int ix=(int)dix;
+  double xd=(coseta-epsilon_bins[ix])*epsilon_binwrecp;
+  return epsilon_Tbl[ix]*(1.-xd)+epsilon_Tbl[ix+1]*xd;
+}
```

Multi-configuration fiTQun

```
1  ********************************************************************************************
2  fiTQun parameters for SuperK_12inchBandL_15perCent
3  ********************************************************************************************
4  < fiTQun.WCSimConfig = SuperK_12inchBandL_15perCent >
5  < fiTQun.WCSimPMTType = 12inchBandL >
6
7  From Mike's initial tune
8  < fiTQun.WaterAttenuationLengthWCSim = 5900. >
9
10 Turn off proton fit by default. The Time PDF files for protons haven't been produced yet
11 < fiTQun.DoProton1RFit = 0 >
12
13 For now these are the same as for SK1
14 < fiTQun.PeakThrWCSim = 9. >
15
16 < fiTQun.DarkRateWCSim = 4.2 >
17
18 < fiTQun.QEEffWCSim = 0.156 >
19
20 < fiTQun.TotQFactWCSim = 0.153 >
21
22 < fiTQun.PIDCutMuEWCSim = 0.25 >
23 < fiTQun.PIDCutMuPip0WCSim = 20. >
24 < fiTQun.PIDCutMuPip1WCSim = 0.2 >
25
26 < fiTQun.RCCuta0WCSim = 120. >
27 < fiTQun.RCCuta1WCSim = -0.6 >
28
29
30 < fiTQun.DoBiasCorrection = 0 > Set to 1 if you want to apply empirical bias corrections to fit results
31 < fiTQun.UseScatteredLight = 4 > Flag for scattered light predicted charges (0:off, 1:6D, 2:3D, 4:6D for 1Rfit 3D othe
32 < fiTQun.TotalChargeConstraintWeight = 10 > Weight of the total charge constraint term in the likelihood function.
33 < fiTQun.UseTimeLikelihood = 1 > Set to 1 to use time likelihoods
34 < fiTQun.QECorrElectron1RFit = 0.973 >
35 < fiTQun.QECorrMuon1RFit = 0.953 >
```

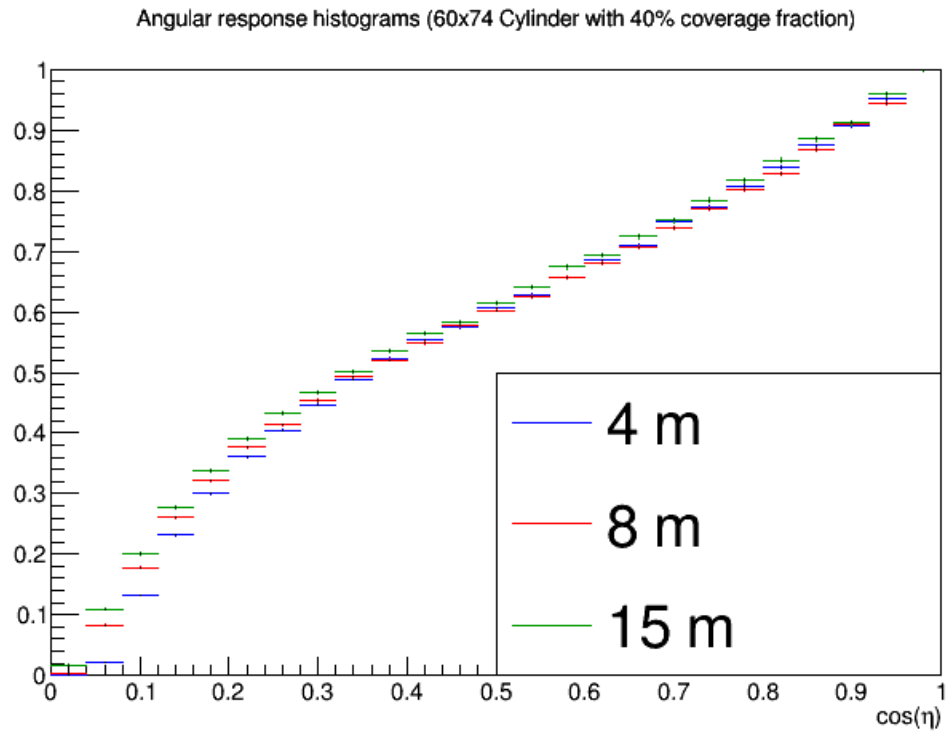
```
>./runfiTQunWC -p ParametersOverride/myWCSimConfig.parameters.dat myInput.root -r myOutput.root
```

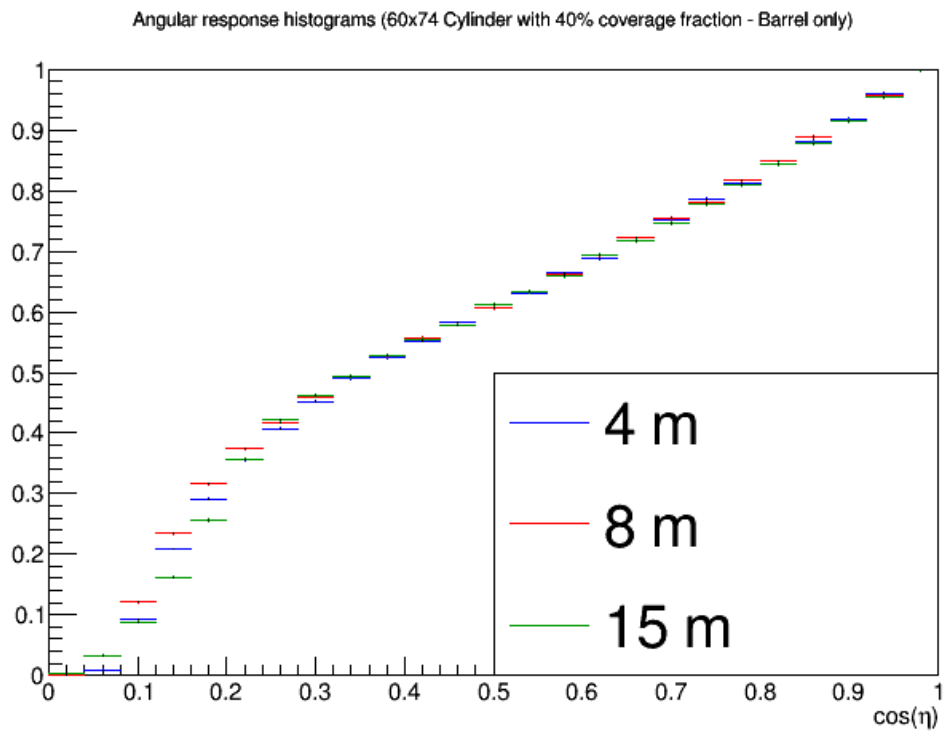
Summary

- The fiTQun tuning procedure for WCSim configurations is now at a mature stage
  - Instructions are being written up on the Wiki
  - Due to the large amount of MC generation necessary, it takes about 1 week to produce a tune
- FiTQun has been adapted so that tunes can be easily chosen at runtime
  - No effect on behaviour when running on SKDETSIM MC
  - Latest release was shipped with Hyper-K tunes, next one will have new Hyper-K configurations and nuPRISM

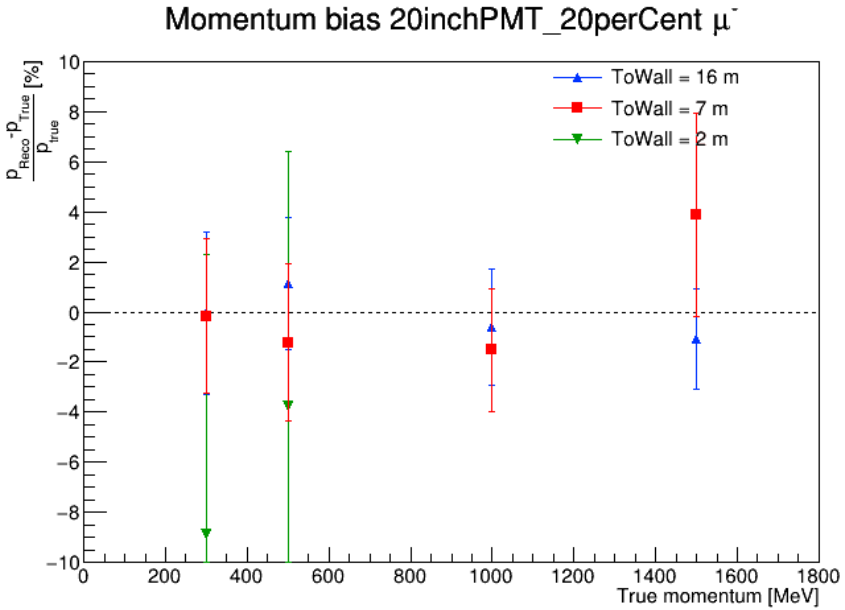
Extras

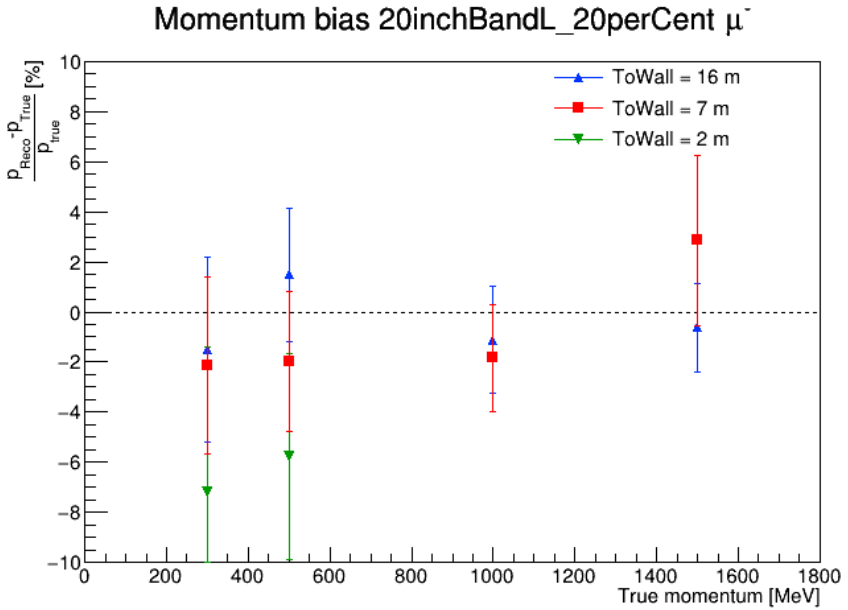
Endcap vs Barrel angular response

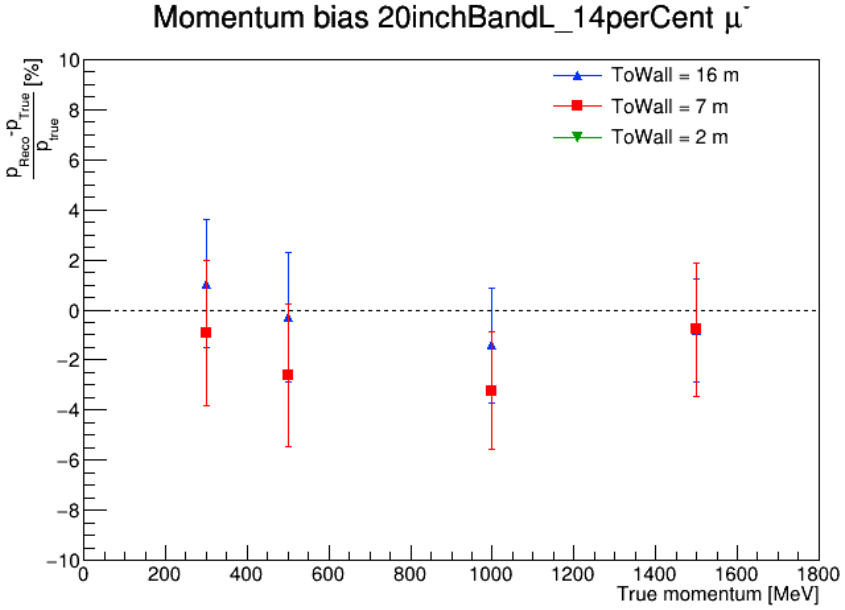


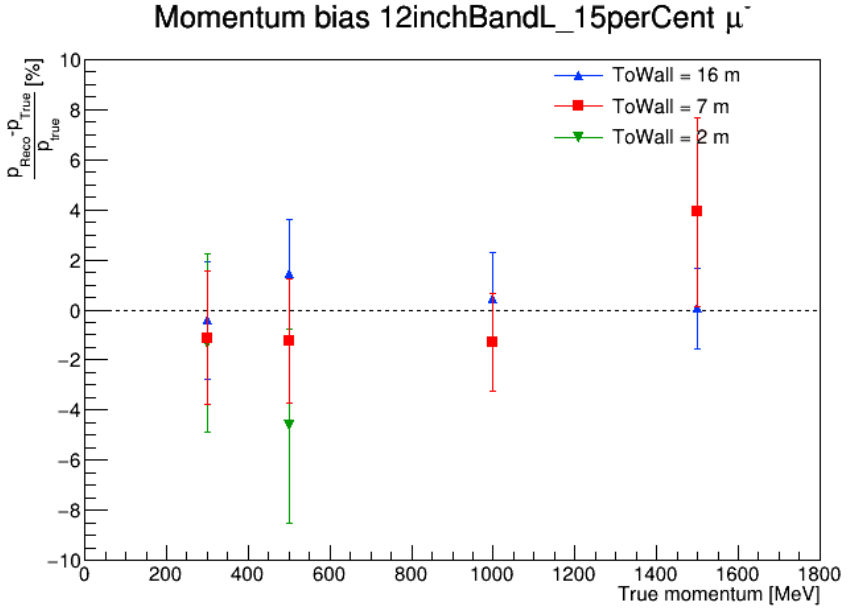


“Old” Hyper-K tunes









“Old” Hyper-K tunes