nuPRISM Monte Carlo and Reconstruction

Carl Rethmeier | TRIUMF

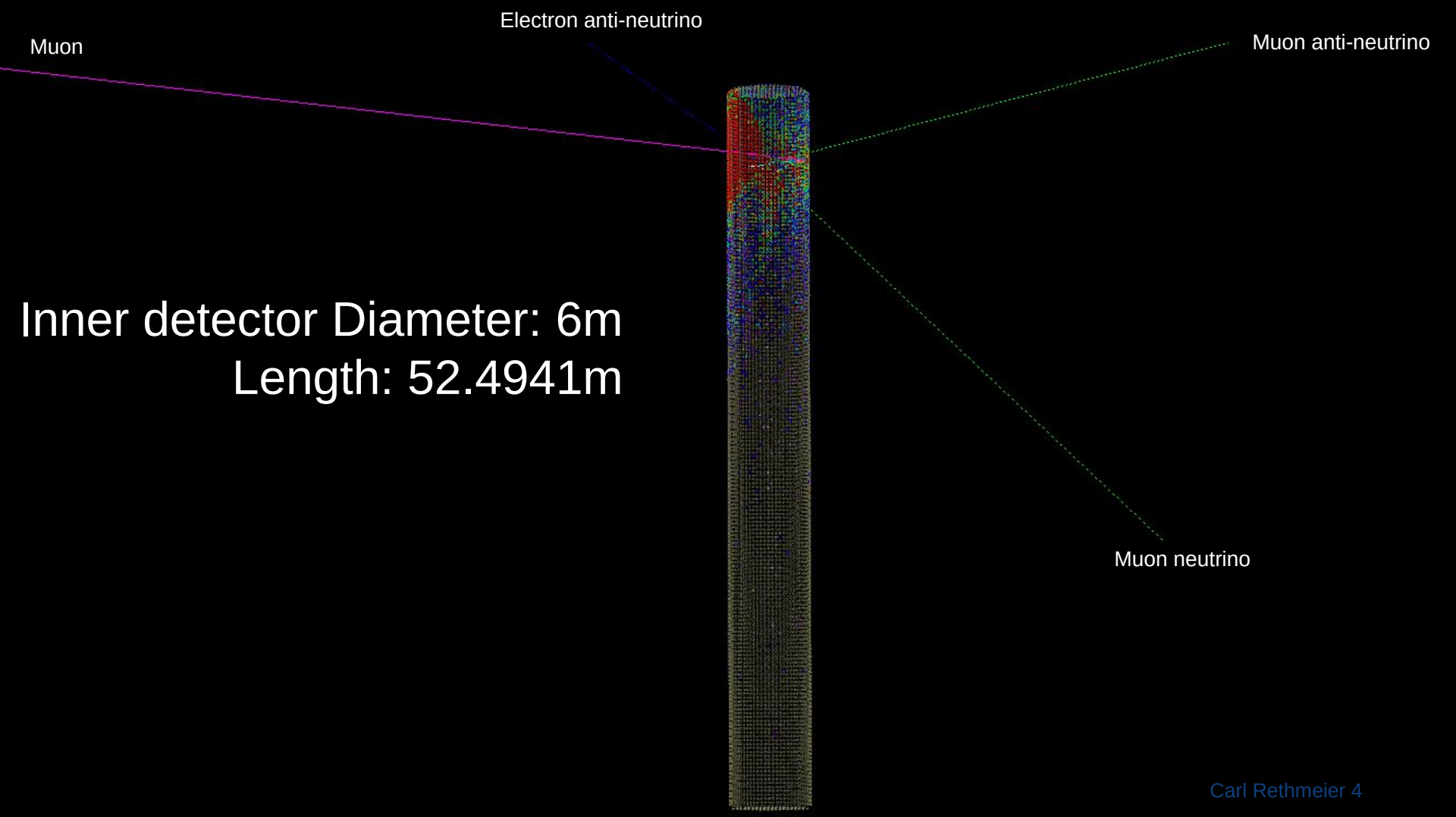
Introduction to WCSim

- WCSim is a water Cherenkov detector Monte Carlo built using GEANT4.
- WCSim was used for LBNE and is currently being used for HyperK
- It has been adapted for nuPRISM R&D.
- Users can control the detector geometry through a text file which is loaded at runtime. This text file uses GEANT4 messenger commands to configure things like the detector dimensions, the PMT spacing, etc.
- The geometry information is saved in the output ROOT file.
- For more information about WCSim, please see Chris Walter's presentation at: http://indico.ipmu.jp/indico/getFile.py/access?contribId=51&sessionId=9&resId=0&materialId=slides&confId=7

PMT Placement in nuPRISM

- PMTs are arranged in a cylinder inside a volume of water.
- The cylinder of PMTs does not completely fill the water volume.
- Particles are produced somewhere in this volume, and WCSim records the electronic response of the PMTs to the light produced by particles as they travel through the water.

nuPRISM Geometry



Inner detector Diameter: 6m
Length: 52.4941m

NEUT and nRooTracker

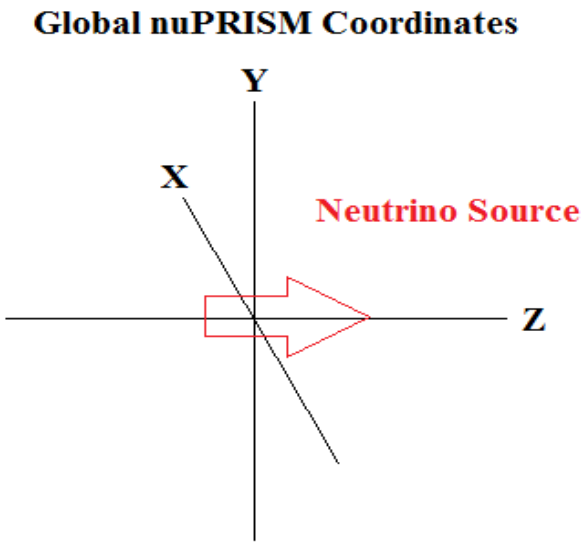
- Mark Hartz used the NEUT neutrino interaction generator to simulate neutrino interactions in nuPRISM.
- NEUT records information about each particle produced in one of these interaction events.
- The nRooTracker format is used to store a very large amount of information about each event.
- All the information about these events and the particles produced in them is stored in arrays in the RooTracker tree.

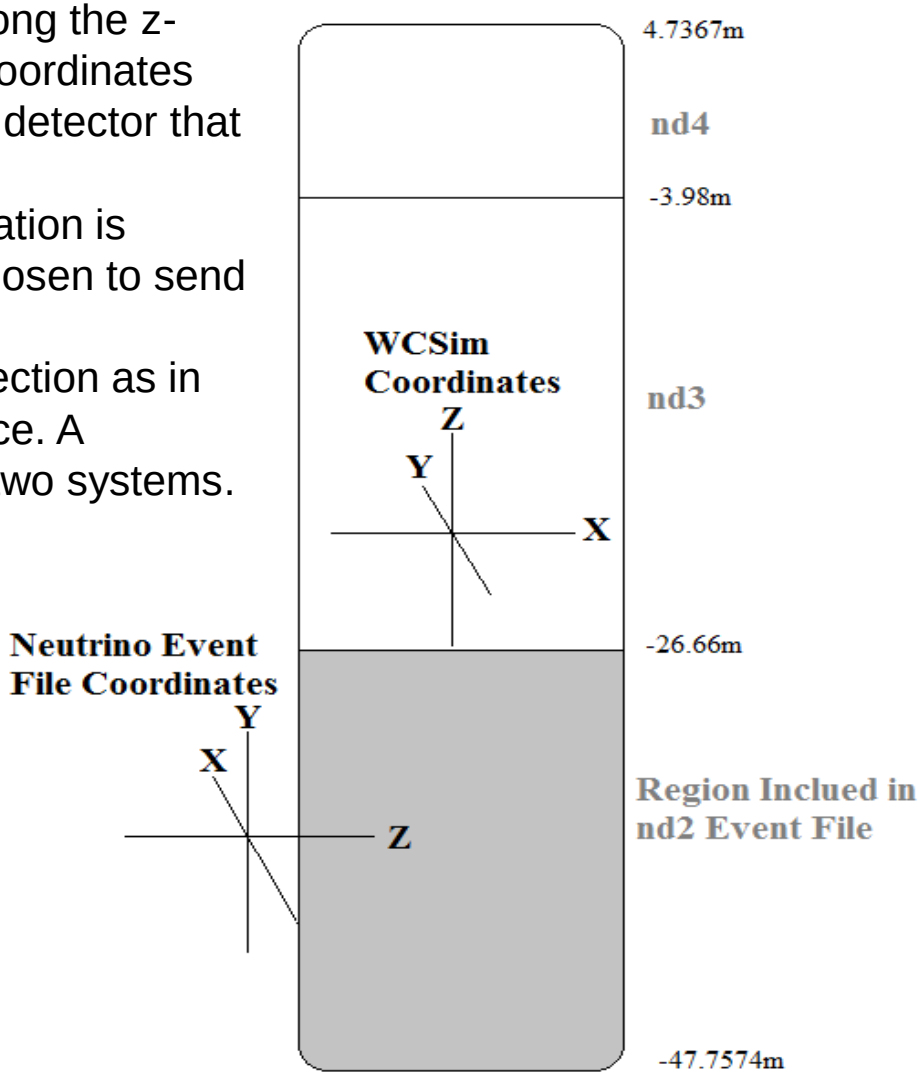
Generation of Events in WCSim

- WCSim can now accept RooTracker input files.
  - I modified WCSim to have the ability to copy the input file to the output when given the command in the configuration text file loaded at runtime.
- The results of the neutrino interaction simulation for nuPRISM are stored in “genev” files in the RooTracker format.
- The coordinate system in the neutrino event files is not the same as the coordinate system used in WCSim.
- This requires a transformation…

Coordinate Systems

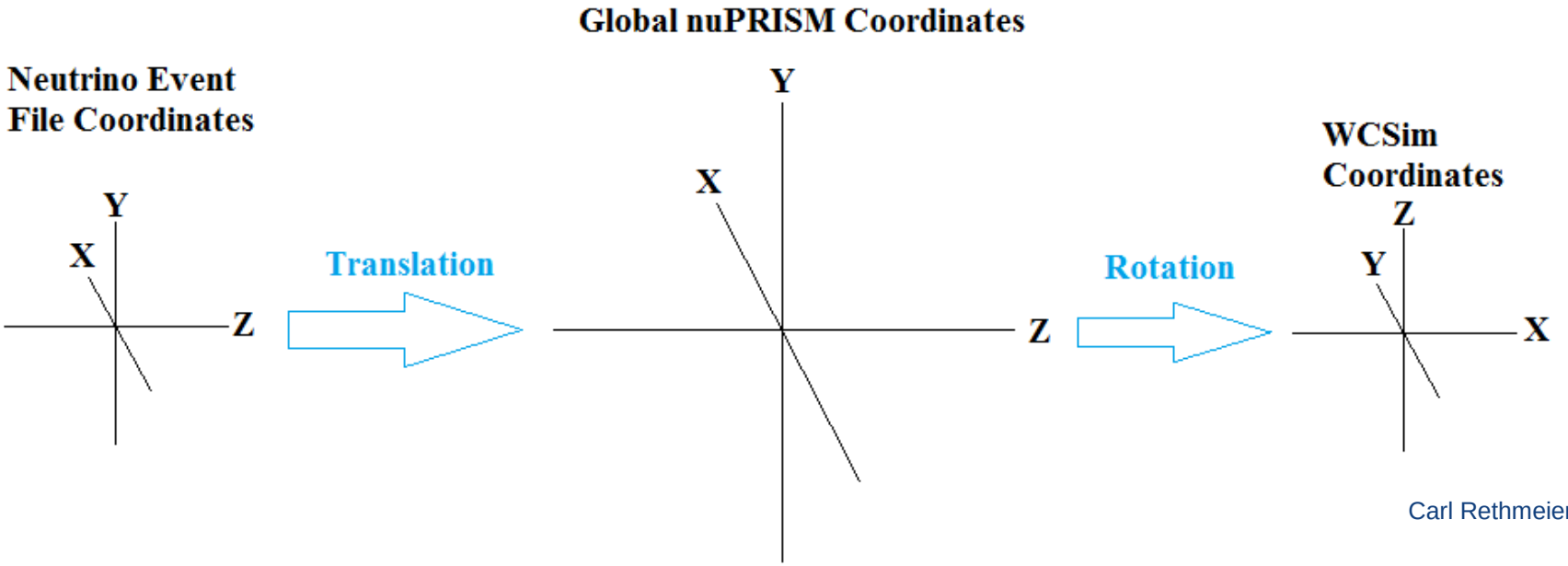
- In the neutrino event files, the beam travels along the z-axis, while the y-axis is the vertical axis. The coordinates are centered near the side of the region of the detector that the specific file includes events for.
- In WCSim, the z-axis is vertical, and the simulation is symmetric with respect to x and y. We have chosen to send the beam along the x-axis.
- The global nuPRISM axes run in the same direction as in the event files. It is centered at the beam source. A translation is required to convert between the two systems.

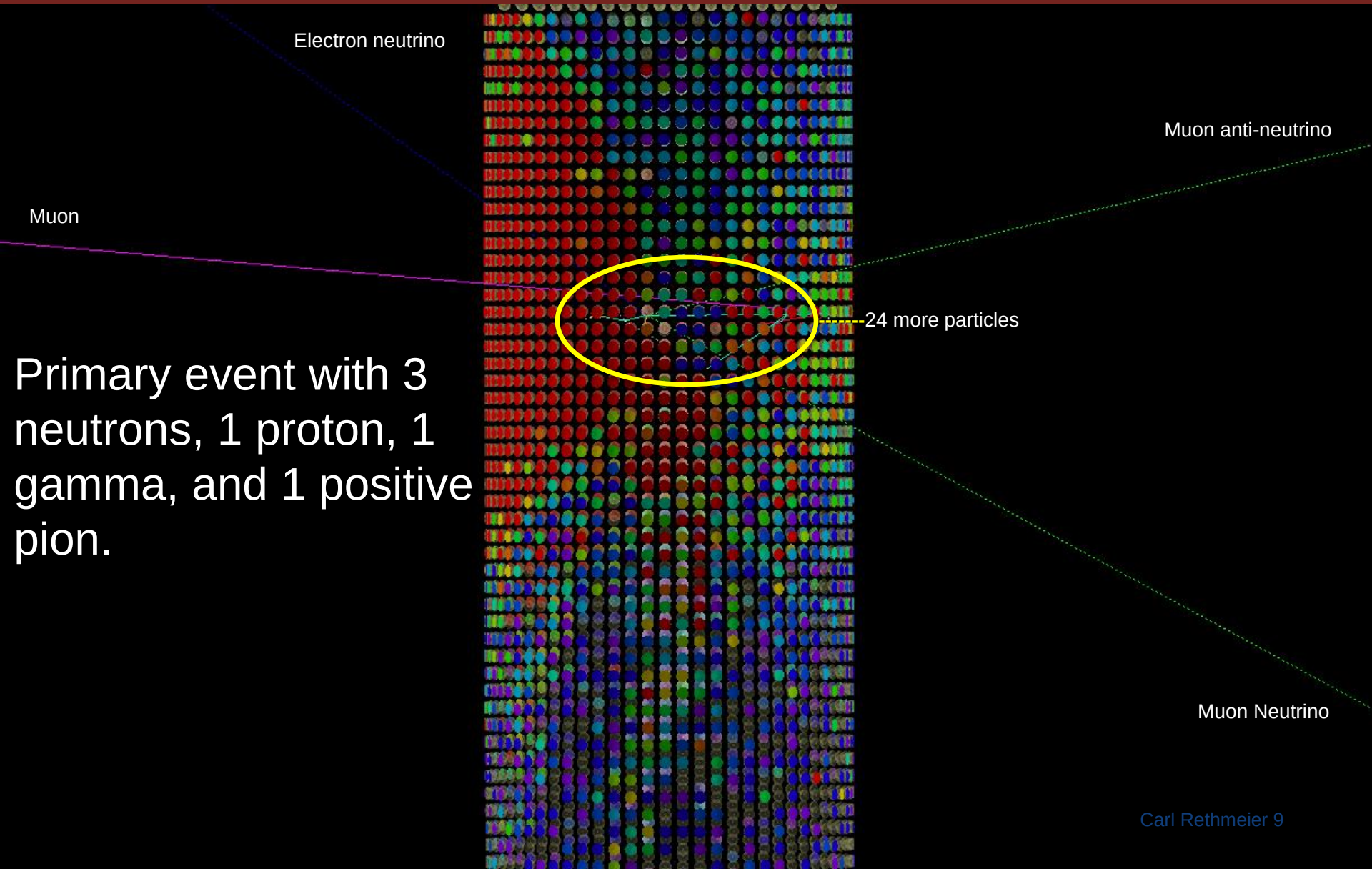




Conversion in WCSim

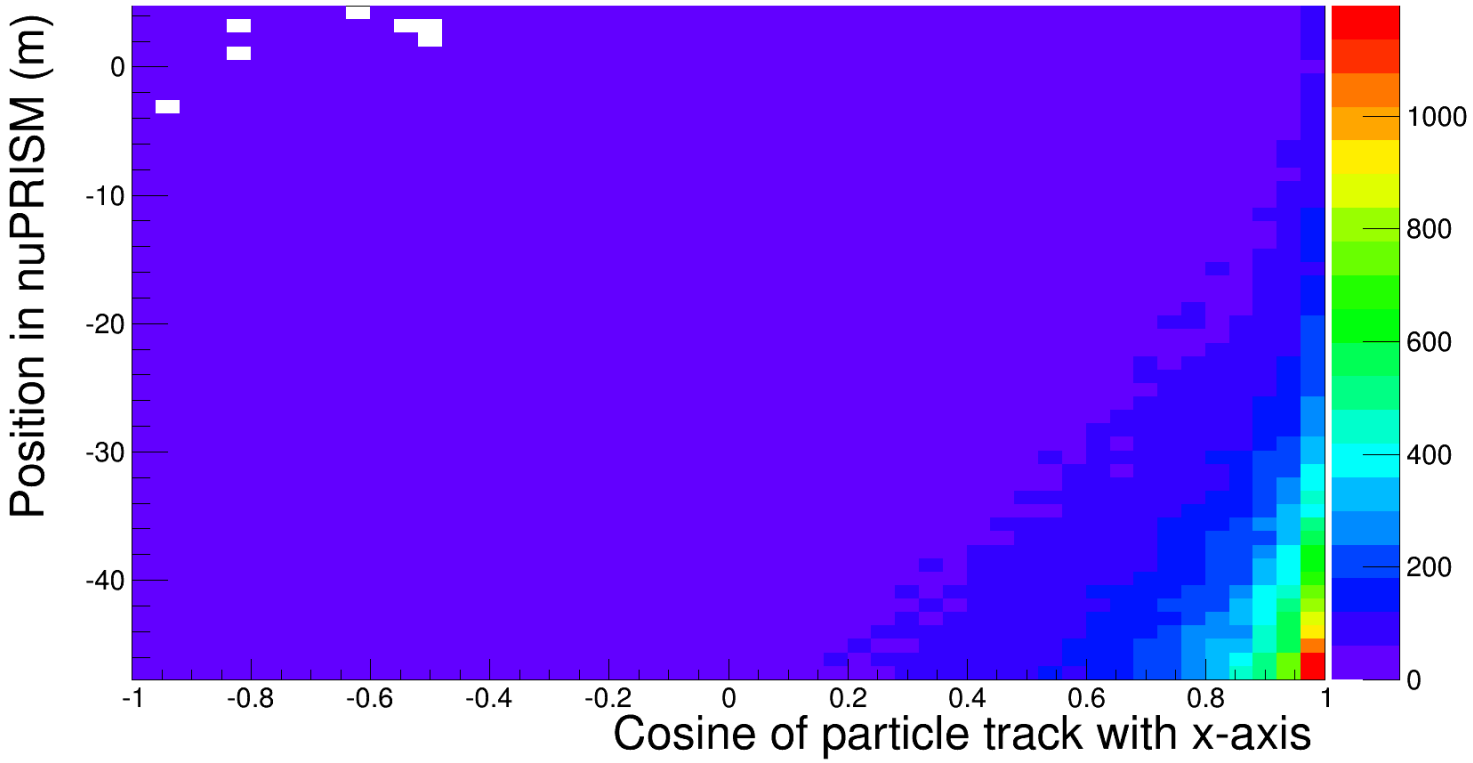
- In WCSim, a simple translation converts the event file coordinates into the global nuPRISM coordinates. This is done so that anyone needing to change the way the input is read in the future does not have to know the WCSim coordinate system. He or she would only need knowledge of the global nuPRISM coordinate system.
- The conversion to WCSim coordinates then takes place. The particle type and energy is also passed to WCSim using GEANT4's "ParticleGun".





Primary event with 3 neutrons, 1 proton, 1 gamma, and 1 positive pion.

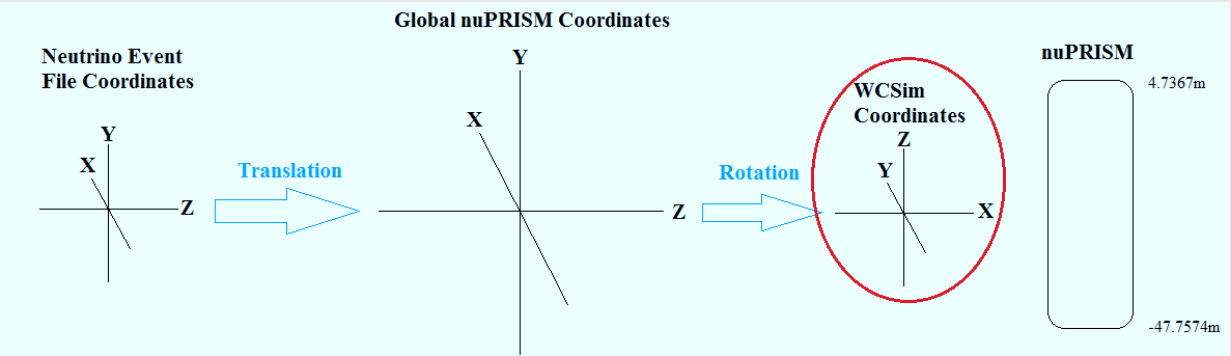
Tests: Interaction Vertex vs Muon Direction X-Component



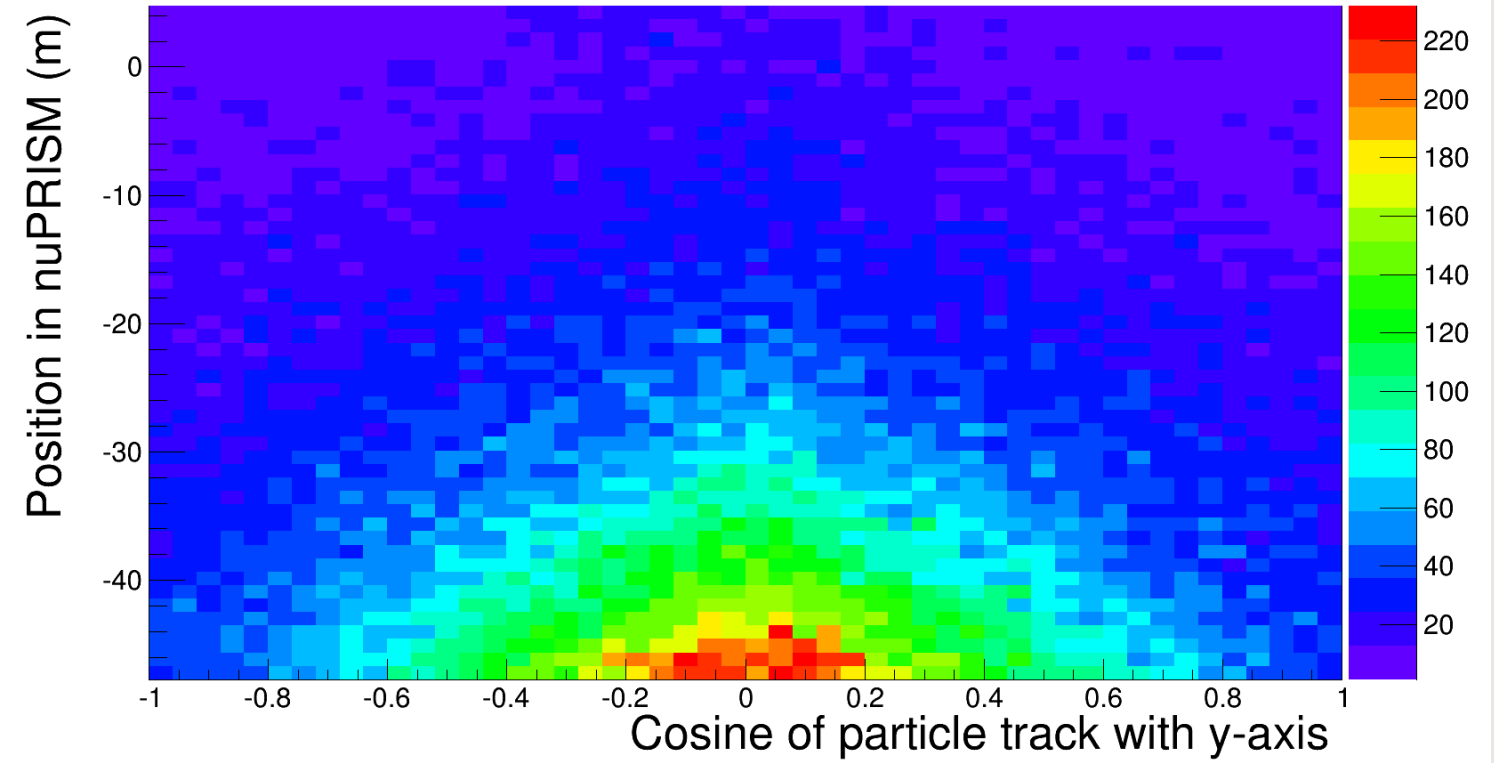
WCSim was run to make sure that the coordinate transformation worked correctly.

The muons' momentum vectors point in the x-direction (WCSim coordinates) on average, with many more events near the bottom of the detector

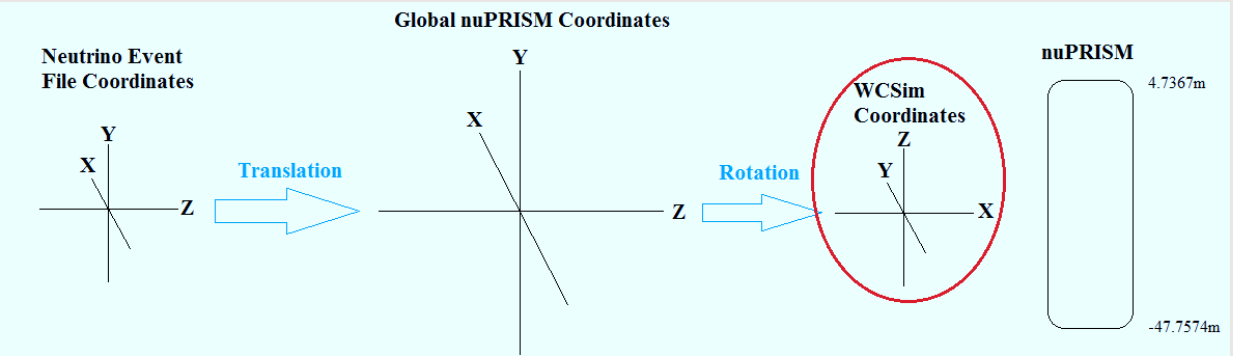
Note: All particles in this and the following plots are muons.



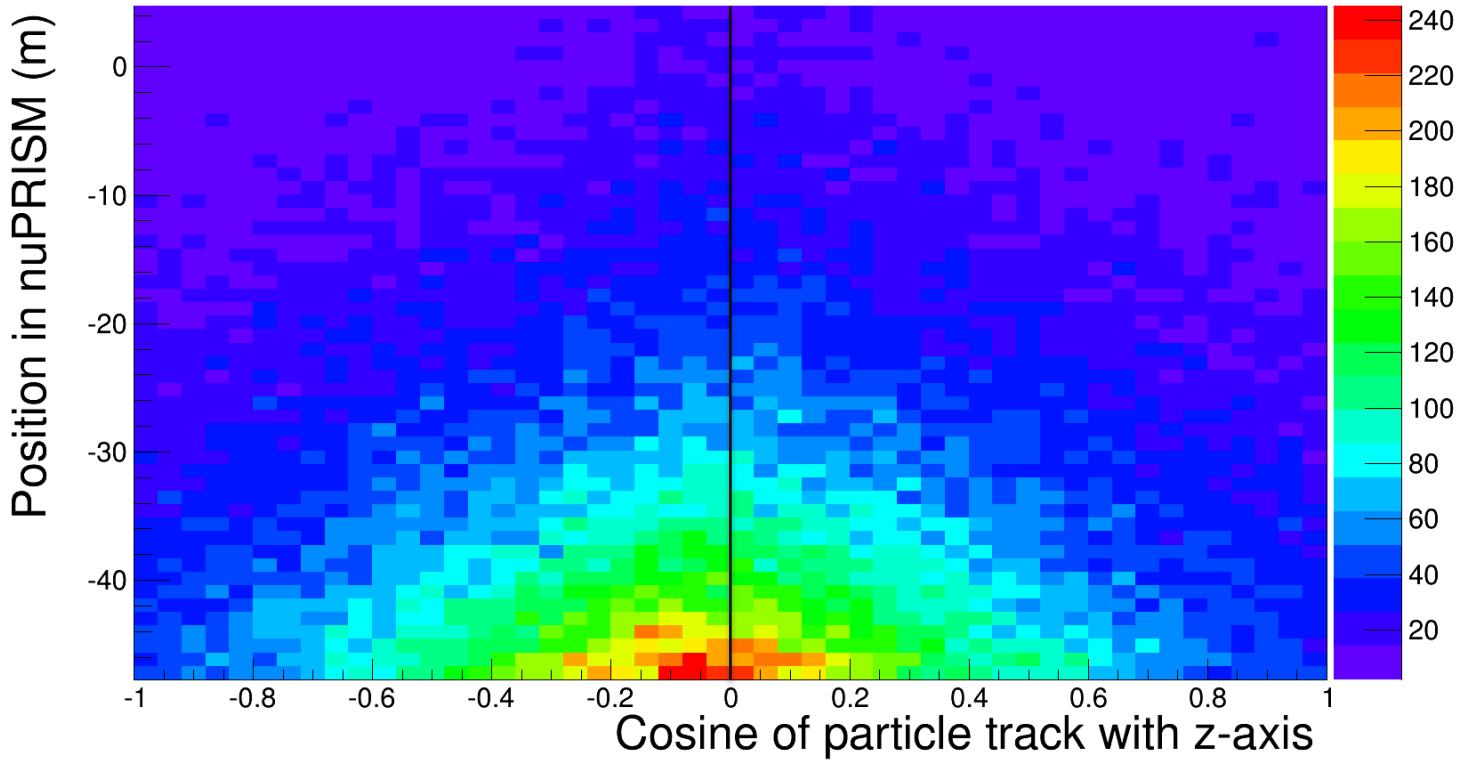
Tests: Interaction Vertex vs Muon Direction Y-Component



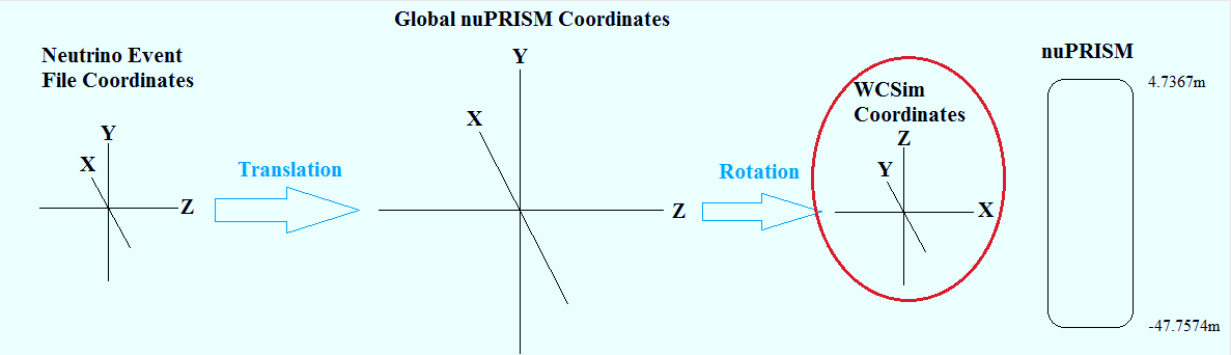
The muon momentum direction is rotationally symmetric along the y-axis in WCSim (the axis perpendicular to both the beam and to the detector axis).



Tests: Interaction Vertex vs Muon Direction Z-Component

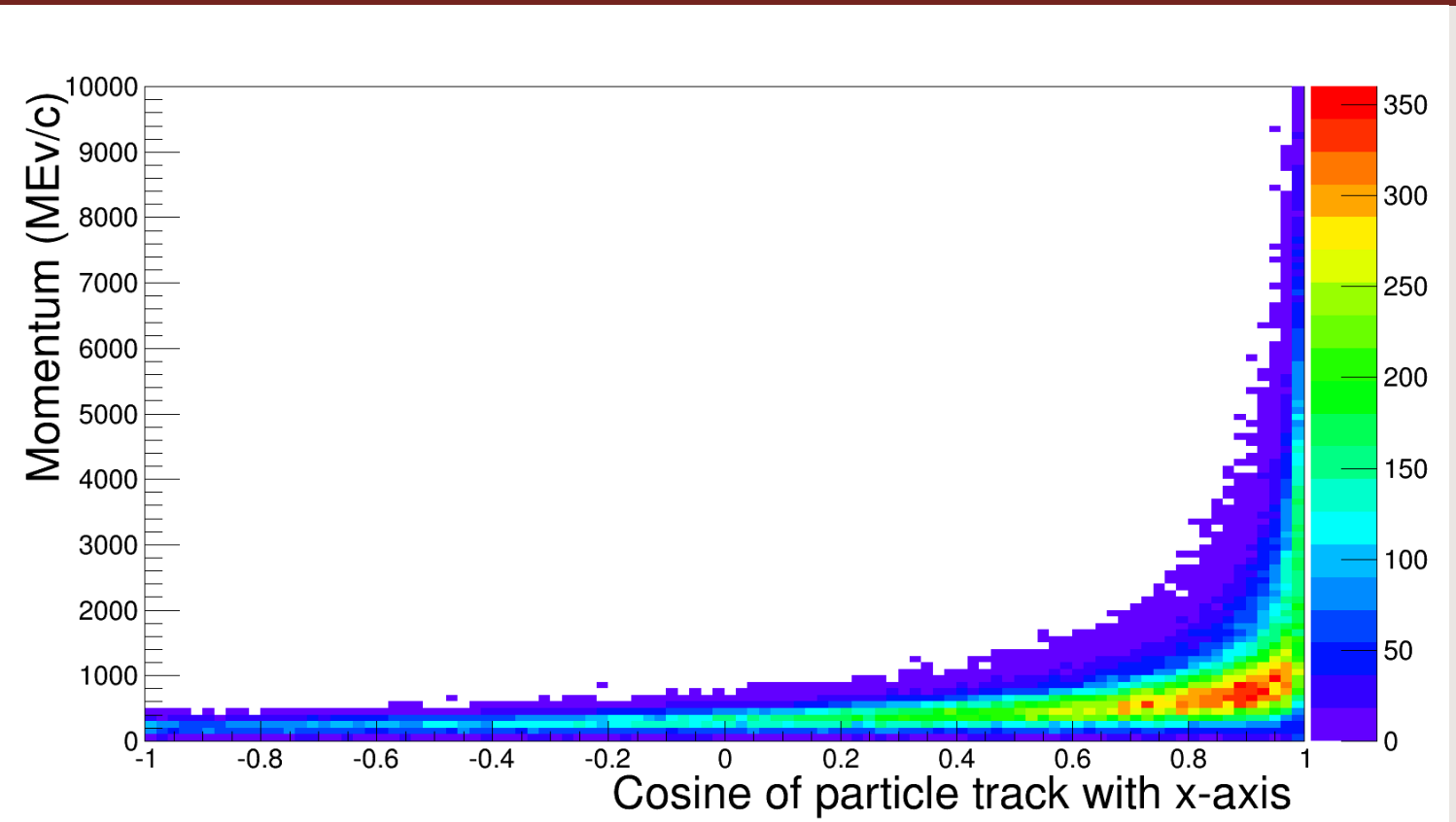


The muon direction is not symmetric along the z-axis. The average muon momentum has a small negative z component. In addition, this effect only exists near the bottom of the detector. This is expected,

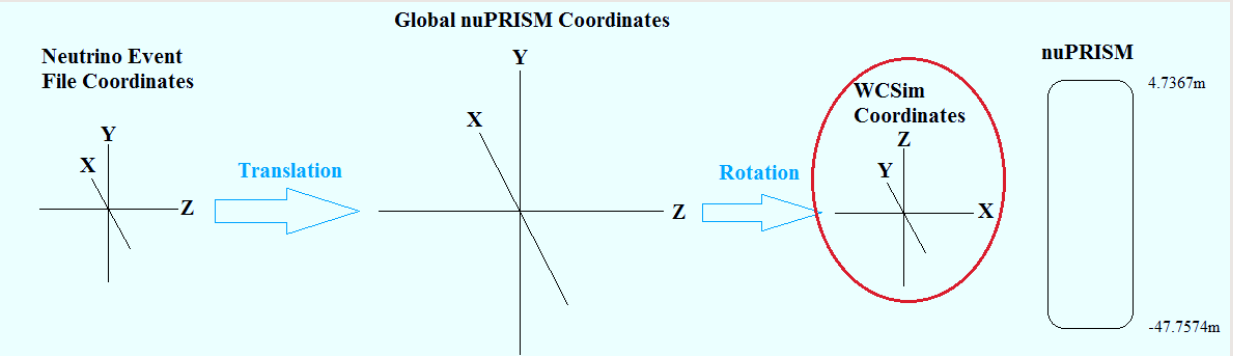


as the beam is located near the origin of the global coordinates and is pointed down 3.6 degrees.

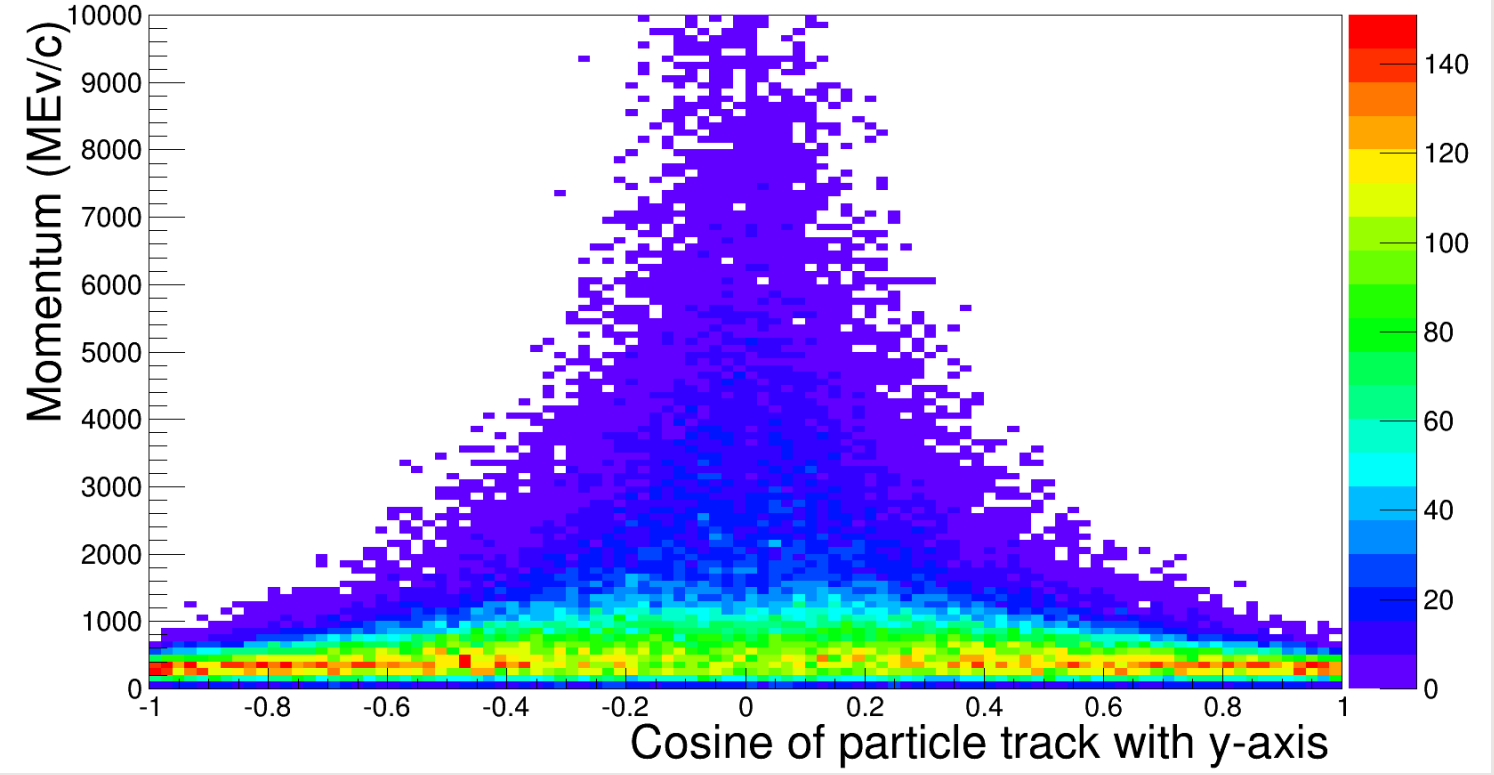
Tests: Total Momentum vs Muon Direction X-Component



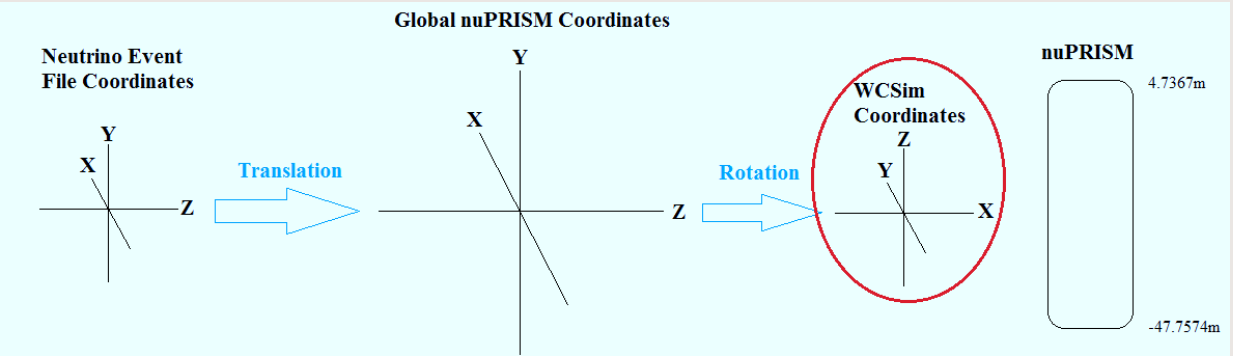
When the muon's momentum is plotted against the x-component of its direction, it is seen that high energy muons always travel in the x direction in WCSim. This is expected, as the highest energy muons carry away most of the momentum of the incident neutrino, which was initially travelling in the x direction.



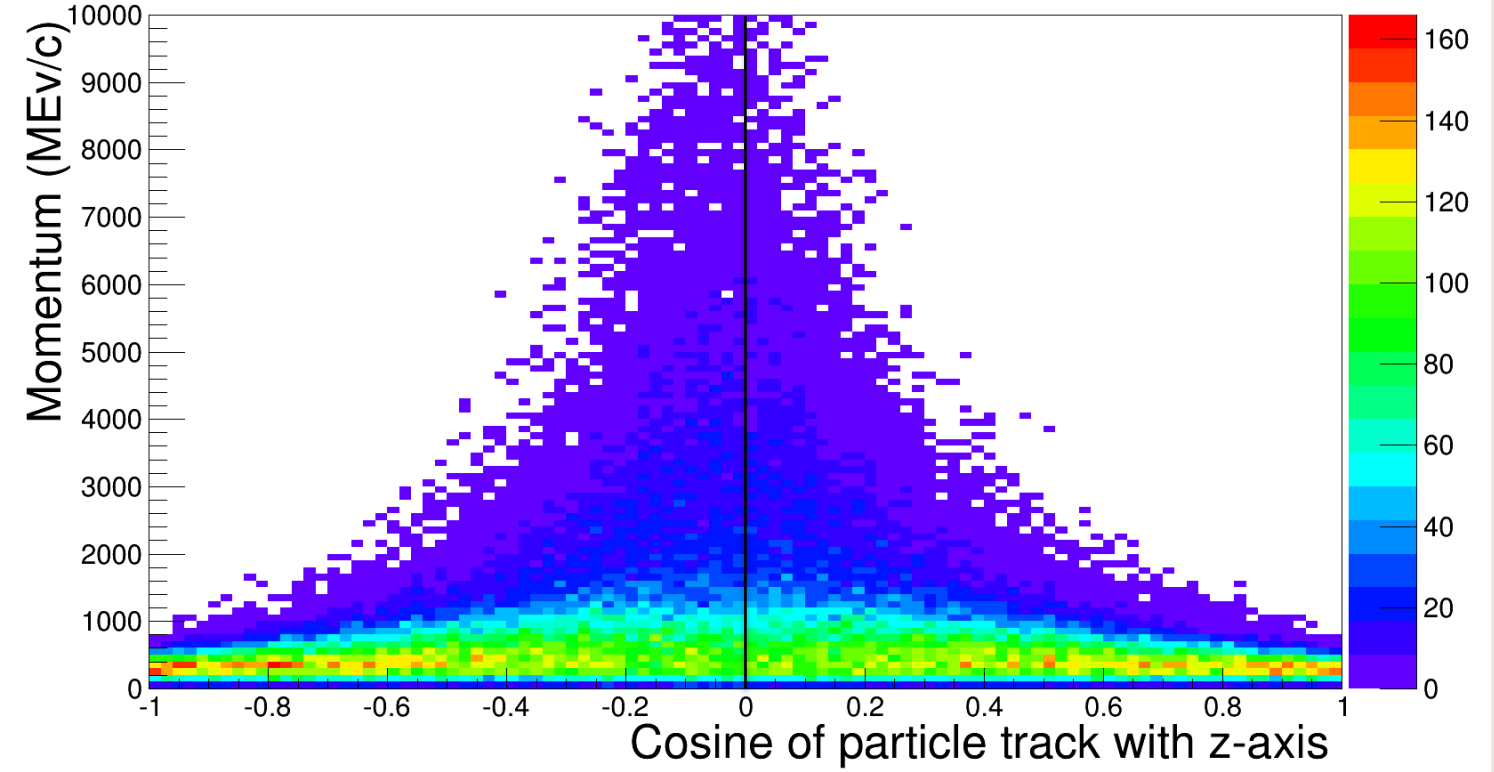
Tests: Total Momentum vs Muon Direction Y-Component



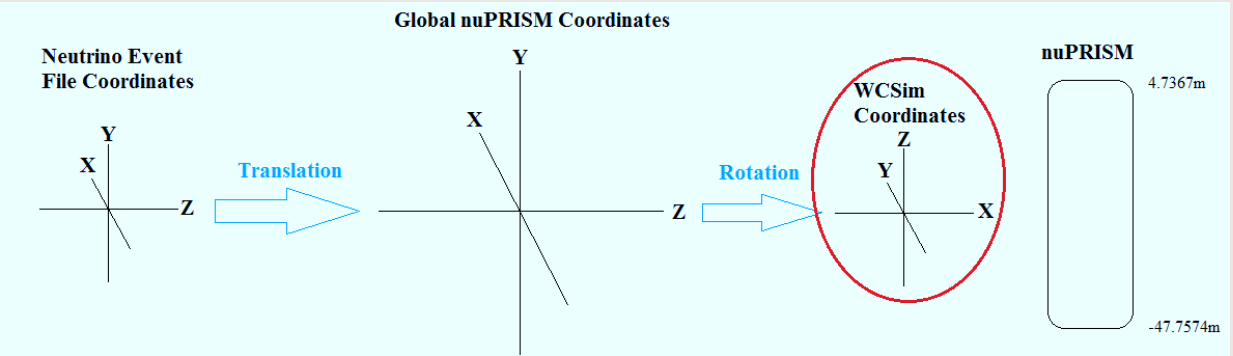
The high energy muons have almost no y-component. This is expected rom the fact that the incoming neutrino beam is rotationally symmetric along the y-axis.



Tests: Total Momentum vs Muon Direction Z-Component



The higher energy muons have a slight negative average z-component, as they should since the highest energy particles carry away most of the incoming neutrino energy.



WCSim truth flat tree program

- The results produced by WCSim are output into a WCSim custom ROOT format.
- In order to view the results in ROOT, appropriate dictionaries must be loaded prior to loading the output file. Get commands then have to be used on each event to access the information.
- Since this makes quick analysis in ROOT's interactive mode nearly impossible, I wrote a small application to convert both the truth information and the results into a flat ROOT Ttree.

Tree structure

WCSim Tree

WCSimTree
- WCSimRootEvent
  - WCSimRootTrigger
    - WCSimRootTrack
      - TrackInfo (accessed through function calls)

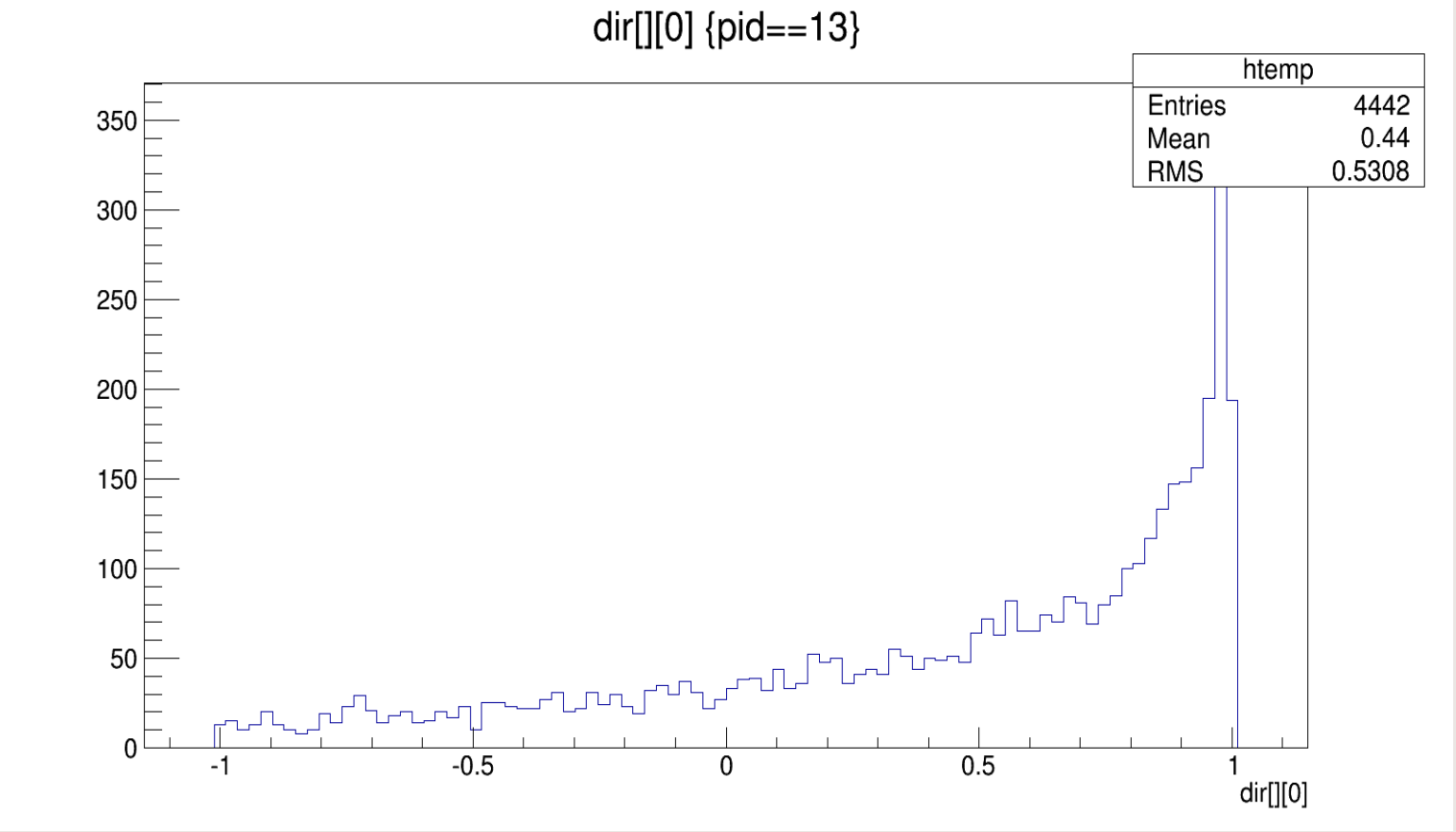
Flat Tree

WCSimExtractedTree
- ntrack (number of tracks in event)
- parent[ntrack] (parent pid)
- pid[ntrack] (particle pid)
- dir[ntrack][4] (particle direction)
- start[ntrack][4] (starting position)
- stop[ntrack][4] (end position)
- mom[ntrack] (total momentum)
- mass[ntrack] (particle mass)
- energy[ntrack] (particle energy)
- startvol[ntrack] (starting volume id)
- stopvol[ntrack] (ending volume id)
- time[ntrack] (creation time of particle)

Description of flat tree program

- The program loads the input TFile and creates an output TFile.
- The "wcsimT" output tree is loaded. The event branch, of type "WCSimRootEvent", is loaded.
- Branches to hold the desired track information to be saved are created.
- Loop through events:
  - The "WCSimRootTrigger" is loaded to determine the total number of tracks in the event.
  - Loop through tracks:
    - Load each track and extract information from it.
  - Save information in output tree and move on to next event.

Example: Plot Muon Momentum Spectrum in X-Direction




```
> root -l nd4aEx.root
root [0]
Attaching file nd4aEx.root as _file0...
root [1] wcsimEx->Draw("dir[][0]","pid==13")
```

fiTQun

- fiTQun is an event reconstruction program.
- It is used on the WCSim output to try to determine information about the particle that produced the Cherenkov radiation seen by the detector.
- fiTQun has not yet been tuned for nuPRISM. Problems persist with momentum biasing, and the light scattering table has not been generated for the nuPRISM geometry.
- Once tuned, fiTQun, WCSim, and NEUT will enable us to simulate nuPRISM from the time the neutrinos enter the detector to the reconstruction of the events seen by the detector.

Summary

- WCSim can now generate events based on the results of the NEUT neutrino interaction simulation.
- WCSim can determine based on the NEUT event file name (or a settings tree within the file) where in the detector the events should occur.
- I have confirmed that these WCSim is accurately reproducing these events.
- WCSim (and my copy of fiTQun) can copy the contents of the input file into the output file.
- I wrote a small program to extract the results out of the WCSim output file into ROOT Ttrees for easy analysis.

To Do

- Allow for multiple detector regions within nuPRISM, i.e., multiple cylindrical regions of PMTs.
- Add the ability to read from multiple input RooTracker files.
- Simulate event spills by generating multi-vertex events distributed in time according to beam structure and interaction probability.
- Test fiTQun with WCSim output once fiTQun has been tuned.
- Include particles from events that occur in the rock outside the detector (from separate simulation).

Acknowledgements

I would like to thank Mark Scott, Mike Wilking, Mark Hartz. They have done much to help as I learn the concept behind nuPRISM and as I work on the simulation software.

I would also like to thank Stan Yen for giving me the opportunity to work on the T2K experiment.