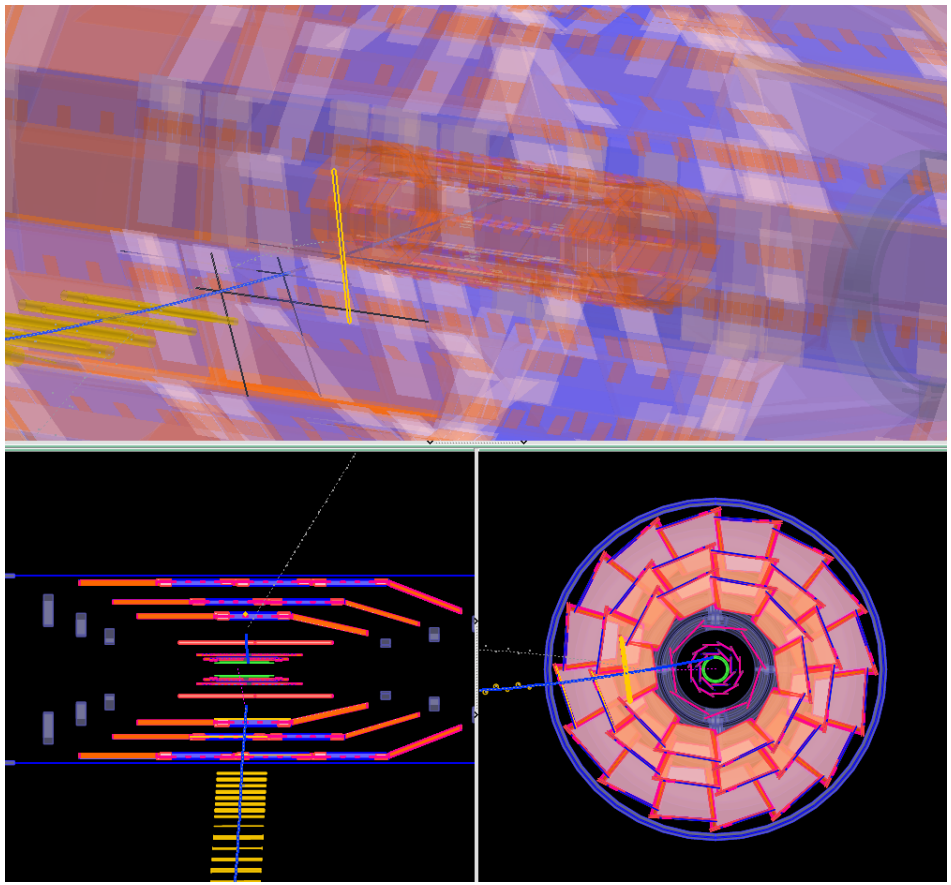
SVD Hits for CDC Tracks

Ian J. Watson

University of Tokyo

Belle 2 F2F Tracking Meeting
April 21-22, 2015

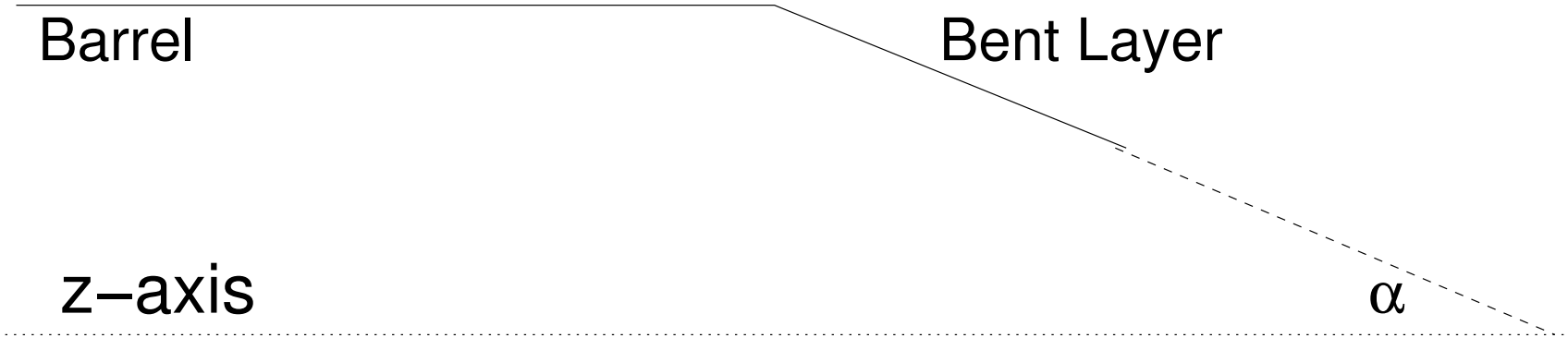
Project Goals



Example: pion from K_S decays between layer 3 & 4, not found by VXD track finder

- Current tracking requires particle can be found stand-alone in VXD for VXD hits to be used
- But, e.g., K_S can decay inside VXD, leaving some VXD hits without enough to find as a standalone track
- Idea for this project is to create a module to take CDC-only tracks and extrapolate back into VXD, create new track adding compatible VXD hits

Cone Algorithm

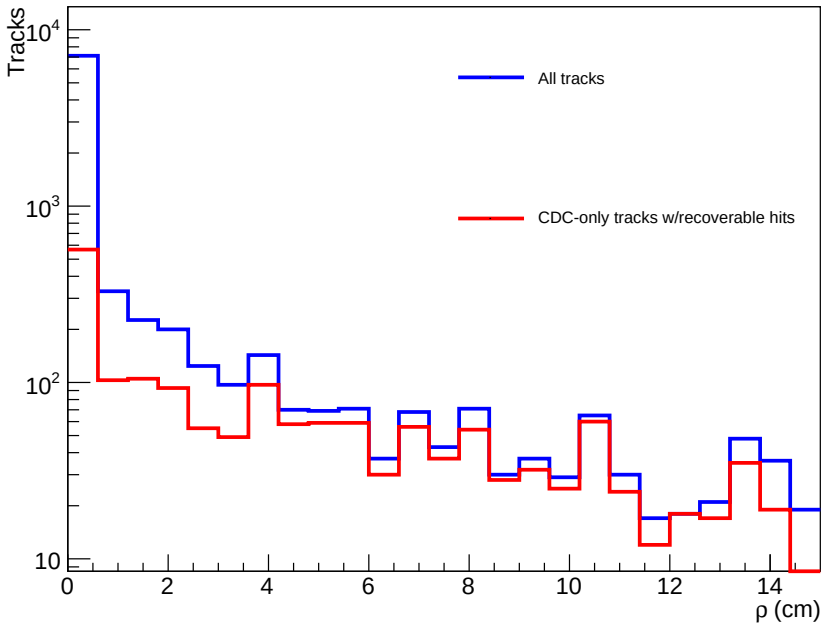


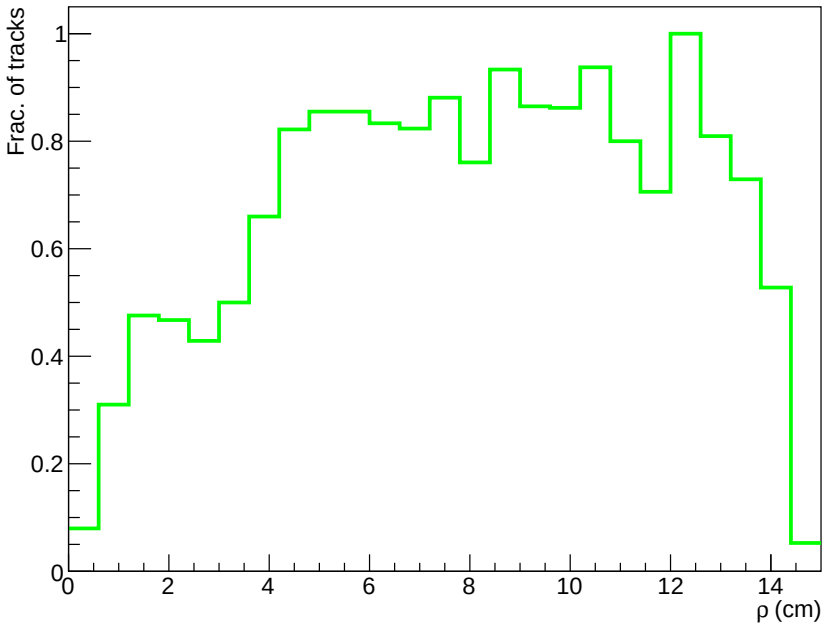
- For method I started attempting (next page), needed to extrapolate to the bent layer, far from effective cylinder
- Added extrapolateToCone method to genfit
  - So, as a warning, adding the code as it stands to BASF2 requires also an (small) addition to genfit
- Used `extrapolateToCylinder` as base (also, used for barrel part)
  - Gets the straightline distance to the surface, extrapolates the track by that much, loops until the distance to the surface is within a tolerance
- Defined cone by a point of the tip, conical axis, and opening angle
  - Defined bent surface by figure above, with angle and relevant radii input by hand from SVD-Components.xml
- Implemented as `extrapolateToCylinder` with cone-line intersection

Basic Algorithm

- Take tracks from genfit stage, look for ones with hits only in CDC
- Extrapolate from first hit (0th hit in the associated CDCHit container) to last layer of SVD
  - Extrapolate to the effective cylinder of layer, if beyond barrel region (in z), extrapolate again to effective cone
- Search through clusters on layer, accept closest to extrapolated track
  - First, look for z cluster (acceptance measure is Δ z)
  - Then, find a r-phi cluster (Δ ϕ)
  - Only accept clusters if they are within a module width (Δ x $<$ SensorInfo.Width()/Length() / 2)
- Continue searching for clusters layer by layer, end if no cluster pair found for current layer
  - Currently, only searching in SVD layers
- If hit(s) were found, do another round of genfit (here, with DAF)
- Add tracks to a new Track output collection (if no hits added, just add the old track to the new container)
  - Needed to uncomment some code in GenFitter to allow redirecting the track collection ouput

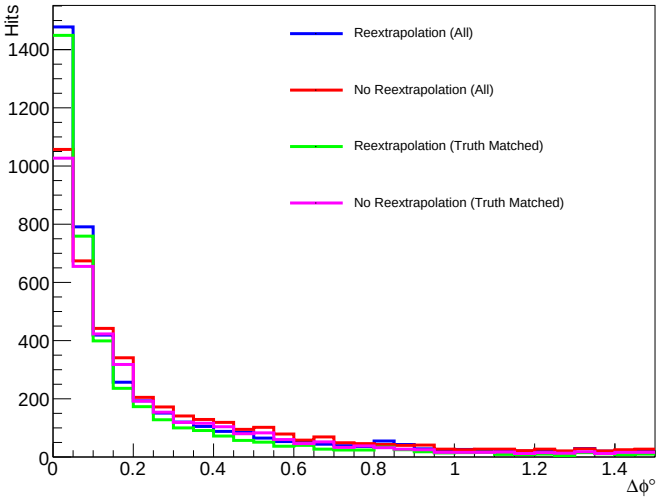
of CDC-only reconstructed tracks with VXD hits

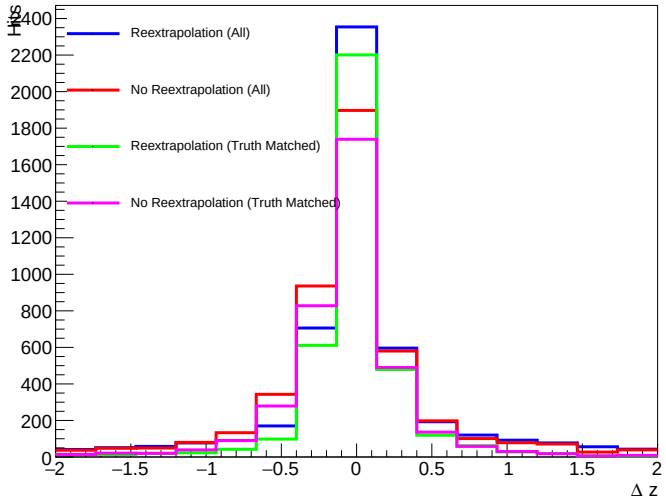




- 1000 event $\Upsilon(4S)$ sample with a $B \to [K_S\pi\pi]_D K$
- Using standard reconstruction tracking
- For these events, currently around 10% of tracks need to recover hits
- Large dependence on ρ, production radius

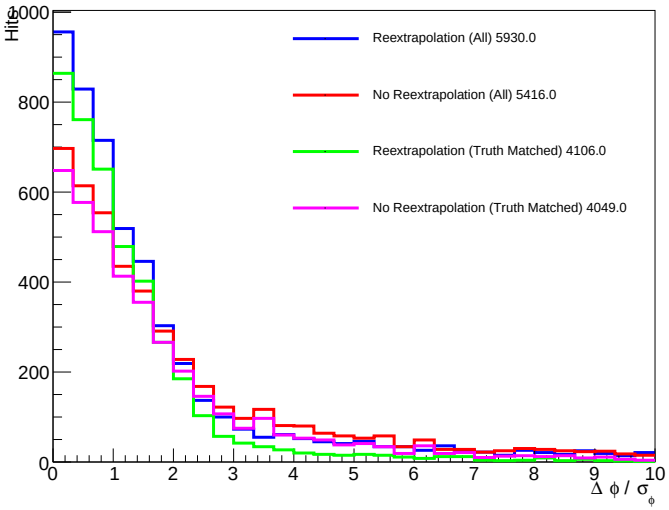
Adding extrapolation to sensor

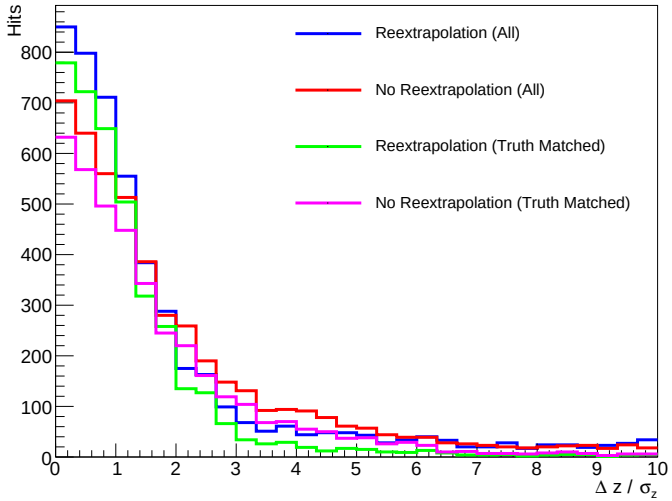




- Tested re-extrapolating to the sensor of the hit (for each hit)
  - `(SVDRecoHit rhit(cluster); genfit::SharedPlanePtr plane = rhit.constructPlane(mop); mop.extrapolateToPlane(plane);)`
  - So extrapolate to the basic geometry, search for hits, and for each compatible hit, do the extrapolation again to the surface of the hit detector
  - Do each time because my broad "compatible" definition means there could be several detectors
- Significantly more gaussian Δ distributions

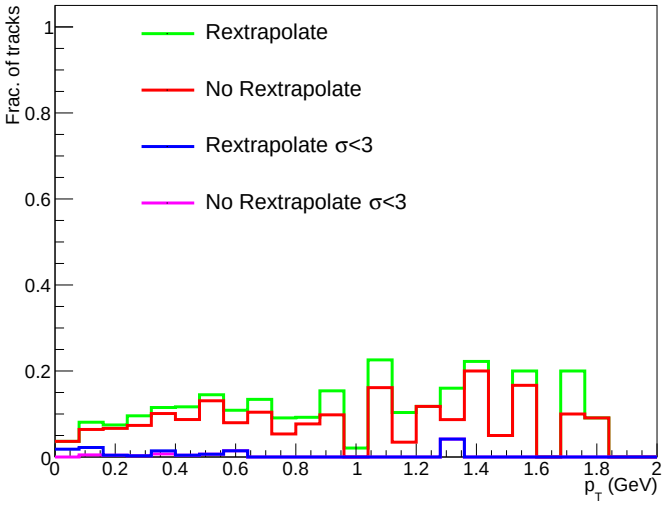
Pulls

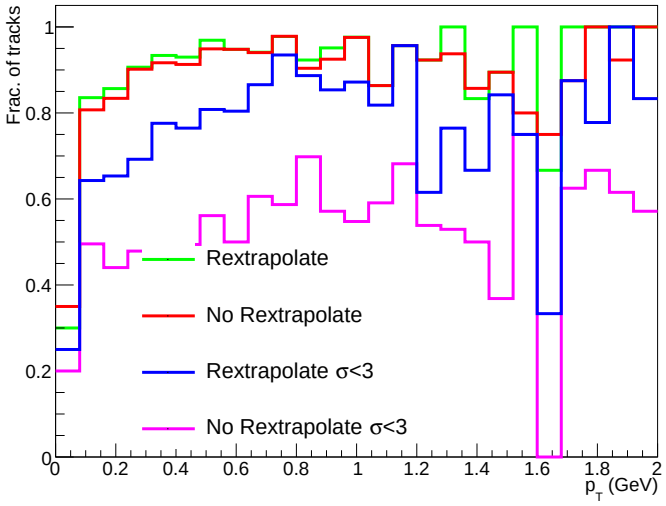




- Can see that the problem that the track errors alone give an underestimate of the errors (with sensor extrap. pull widths ~ 1.3)
- From the geometry of the situation: i.e. detectors not on the effective cone/cylinder
- Timing problems extrapolating every hit (100ms vs 1s/evt vs several hundred ms/evt for all other tracking code)
  - To use the cylinder alone with a χ^2-like cutoff, would need to incorporate the geometry error
  - Will first try to cache hits / detector, so at least not rerunning extrapolation every hit

Cutoff Investigation

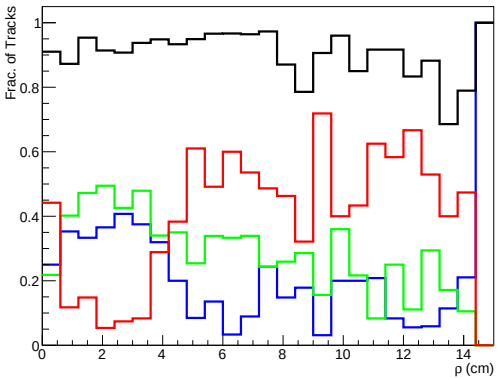




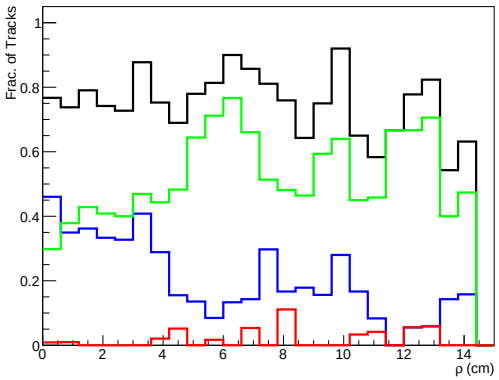
- Looked also at requiring that the best hit be within N=3σ_{track}
- Left shows the fraction of tracks recovered without truth VXD hits
  - # tracks with added VXD without truth VXD hits / Total tracks with added VXD hits
- Right shows fraction of recoverable tracks that get recovered
  - # tracks with truth VXD hits / # truth VXD tracks with added VXD hits
- Cut gets rid of bad tracks, but impacts efficiency
- Have also run N=5σ, but need to remake plots

Cutoff Investigation (cont'd)

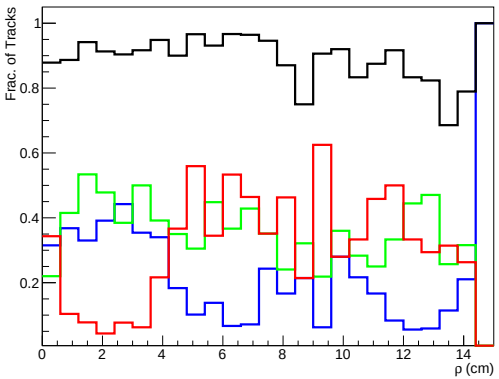
without σ cutoff
Rextrapolating to sensor

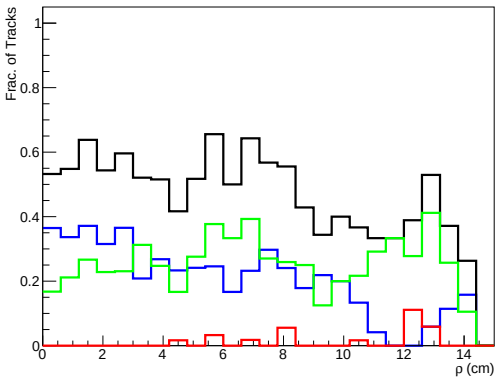


with σ cutoff



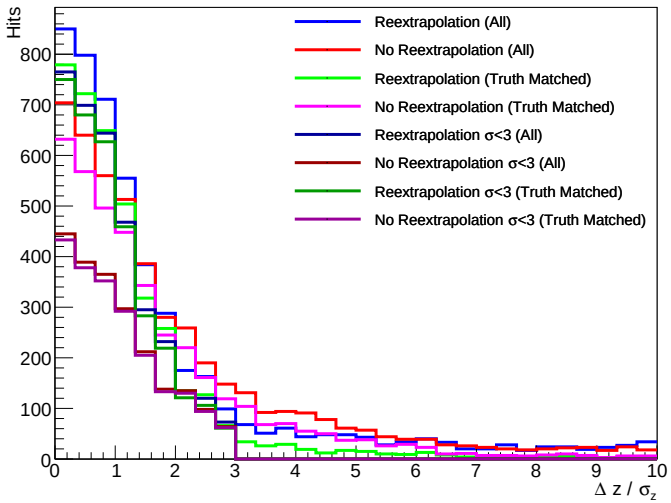
Only to cylinder/cone

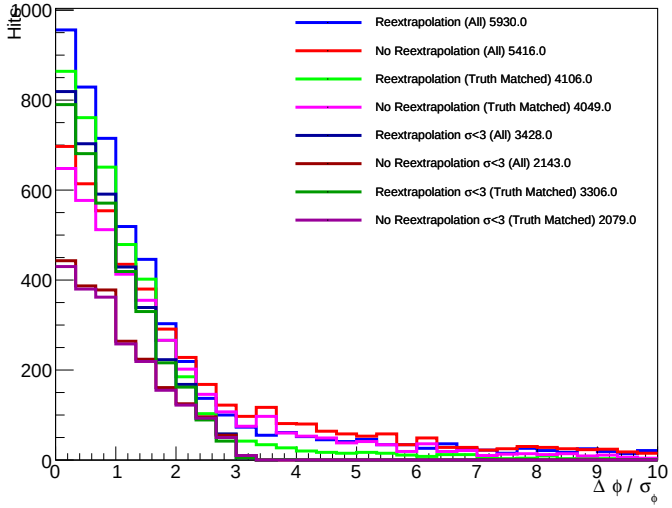




- Showing fraction of good, recovered tracks (per prev. page) # hits added same as truth MC hits fewer hits added more hits added
- Cutoff destroys the more hits component, improves exact hits in the case of extrapolation to sensor
- (Note though, not truth matching hits)

True Hits





- Also ask if hit also associated to the track's MCParticle and plot against $\Delta x/\sigma_x$ and compare against all
- Extrapolating to sensor and use a cutoff, greatly reduces the fake hits, but with some reduction in truth hits
- With Extrapolation to sensor and truth match. Add cutoff and truth match
- Pulls worse for non-sensor-extrapolated tracks, so not suprising theres a large eff. drop, want to scan cutoff

Summary

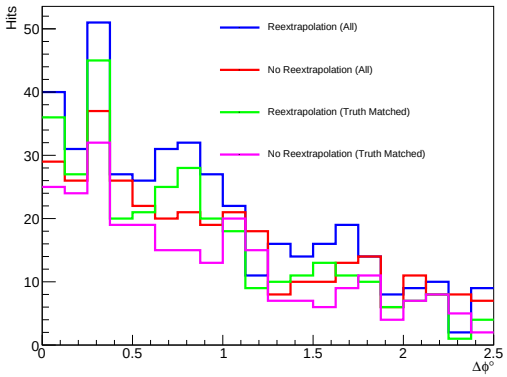
- With the error-based cutoff, can kill most of the fake hits
- Needs to extrapolate onto sensor in order to keep efficiency up, though
  - For the same σ cut-off
- Started some basic physics studies, see improvement in K_S mass and vertex resolutions (not yet quantified)

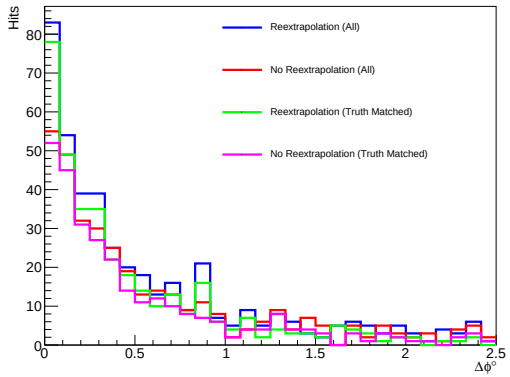
List of things I want to look at:

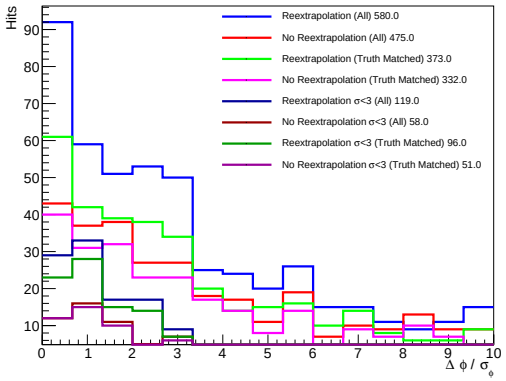
- Extrapolate to layer once, maybe twice at most (the "re-extrapolation" gives 1s/evt)
  - Might try first solving this by "caching" the hits by saving to a map (Sensor->Extrapolated Position)
  - If timing is an issue later, need to investigate if we can stop extrapolation if a sensitive layer is hit?
- Add pixel layer extrapolation
- Tuning of the σ cutoff
- Code cleanup

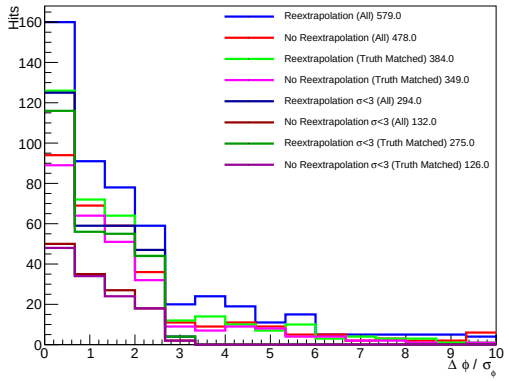
BACKUP

Bent layer





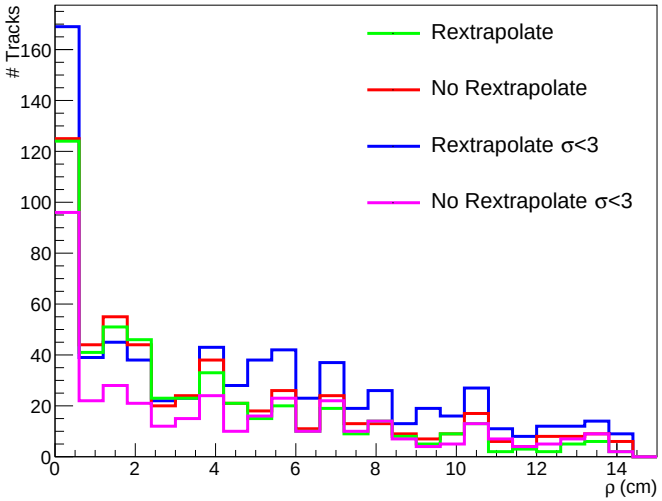




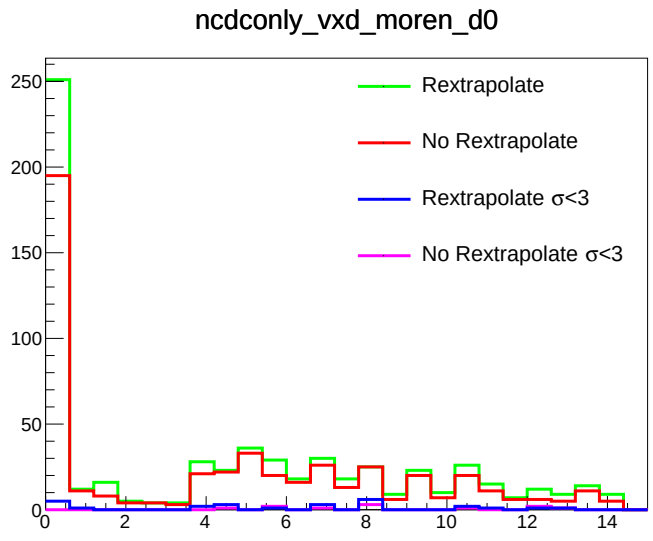
- Top left shows $\Delta\phi$ without the u-coord wedge shape correction (previous presentation results), right with the correction
- Bottom row is the pull distributions on the bent layer
- Without wedge correction, severe drop in efficiency with σ cutoff

Cutoff Investigation

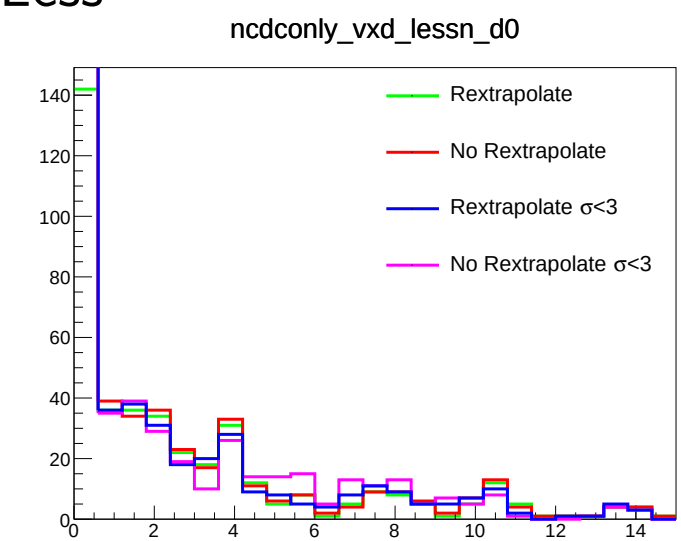
Hits Exact



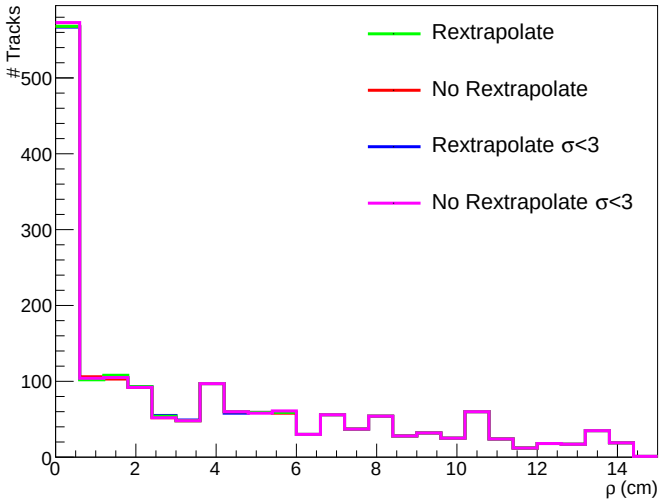
More



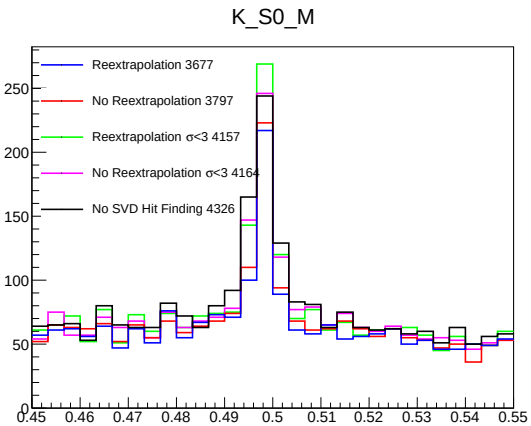
Less

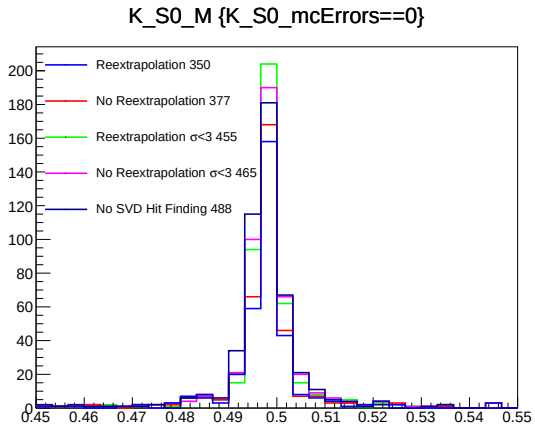


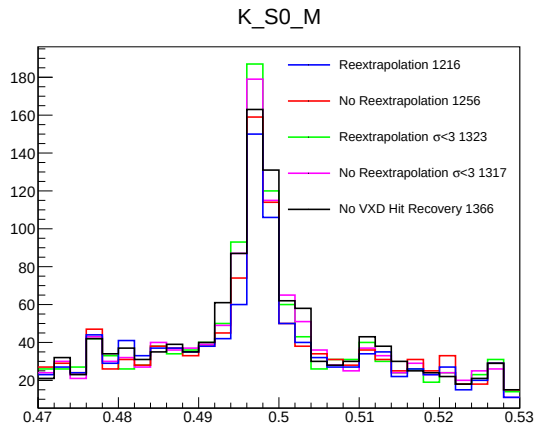
Total tracks with added hits



Impact on K_S mass

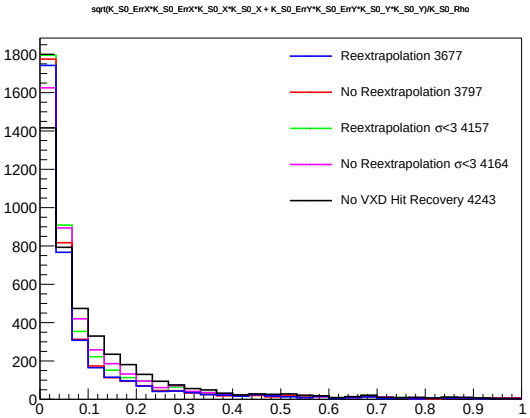


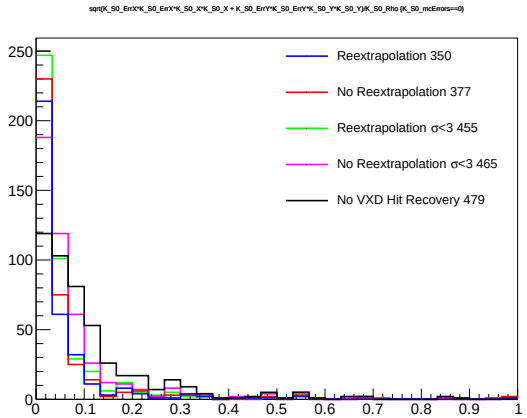


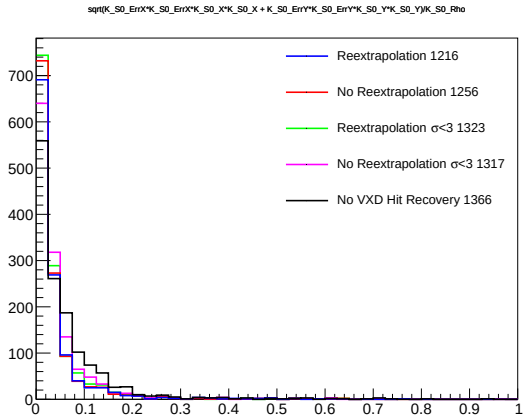


- Events have been $B \to [K_S\pi\pi]_D K$
- Tried reconstructing K_S in those events
  - Both by vertexing $\pi\pi$, no cuts, and `stdKshort()`
- Plots are all $\pi\pi$, truth matched $\pi\pi$, and `stdKshort()`
  - Unfortunately, haven't gotten truth matching information in `stdKshort()` to work
- Some losses when using the added VXD information, but with the σ cutoff and sensor extrapolation, this is minimal
- Improves mass resolution

Impact on K_S vertex







- Plots as for prev. page but with the error of the vertex radius
- Again, see that the inclusion of VXD hits improves errors