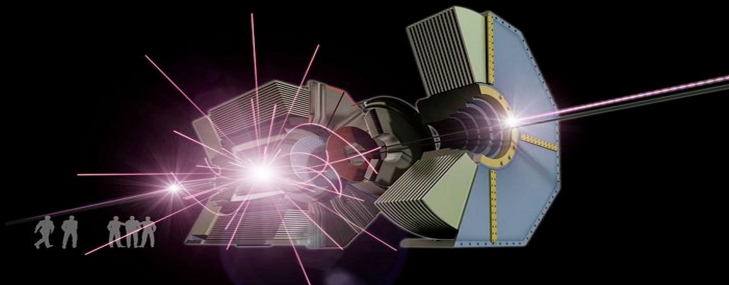


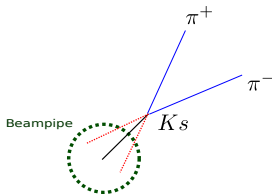
V0Finder status report

Paul Jaeger | 20.08.2015

INSTITUT FÜR EXPERIMENTELLE KERNPHYSIK (IEKP)



V0Finder



- **reconDecay** (current method): tracks pions back onto IP. Vertex fit only at analysis level (kFit). Shift to higher K_s masses due to wrong pion Energy: $E_W + \frac{dE}{dx}$
- **V0Finder**: promises unbiased K_s mass, since pions are not tracked back until IP: E_W . Thus we expect better reconstruction efficiency and vertex resolution.(performs **RAVEVertexFit** during reconstruction. Tracks are updated by vertex-fit, but results themselves are not saved! later - at analysis level - additional **KFit** when creating the particle list.)
- **inside beampipe**: the two results should be the same.

Overview

- how to call the two reconstruction methods:

reconstructDecay

post-mdst reconstruction of Ks from tracked pions

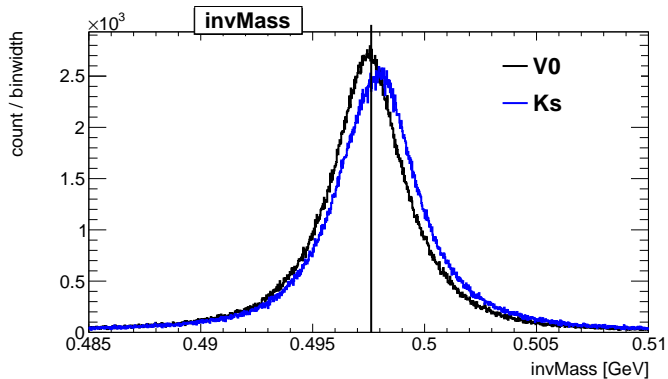
```
from stdLooseFSParticles import stdVeryLoosePi
stdVeryLoosePi()
reconstructDecay('K_S0:ks -> pi-:all pi+:all', '0.4 < M < 0.6')
vertexKFit('K_S0:ks', 0.0)
applyCuts('K_S0:ks', '0.477614<M<0.517614')
```

v0finder

In time v0 recognition at reconstruction stage. Better handling of v0's outside beampipe.

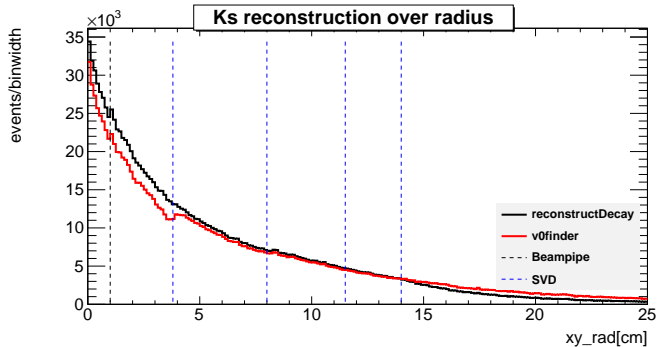
```
from stdV0s import stdKshorts
stdKshorts()
fillParticleList('K_S0:all', '0.3 < M < 0.7', True)
vertexKFit('K_S0:all', 0.0)
applyCuts('K_S0:all', '0.477614<M<0.517614')
```

Energy correction works...



- Ks Mass reconstruction without bias possible via V0Finder.

The Problem



- $\approx 2\%$ of K get lost mainly inside beampipe and until first SVD layer.
(1 million $B \rightarrow K_s K_s K_s$ with MCMatching)

Potential error sources

- 1 decay too complex. V0 has problems with 6π per event...
 - generated 1mil single Ks with **particleGun** → no solution
- 2 inhomogeneous cuts.
- 3 RAVEVertexFit fails.

Cuts

inside V0-module:

- $\chi^2 < 50$ cut on vertex quality
- cut on Extrapolation to cylinder: *"This is intended to reject tracks that curl away before meeting"*
- cut on V0 mass inside beampipe $r < 1\text{ cm}$: 60 MeV mass window

at analysis stage:

- 400 MeV mass window at FillParticleList (before KFit)
- Confidence Level of Kfit (default 0.0 still cuts Ks out!)
- 40 MeV mass window after KFit

Took out all cuts, now efficiency loss at $\approx 0.7\%$

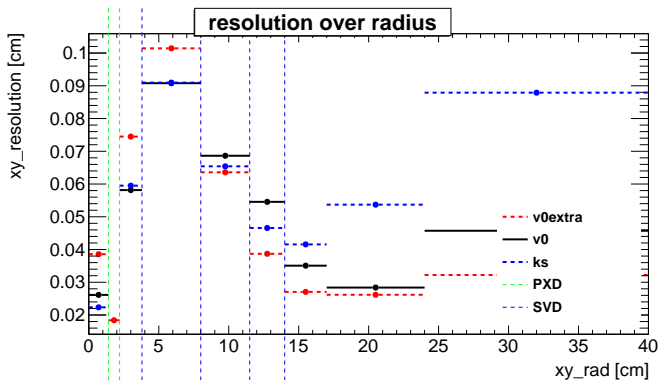
RAVE VertexFitter

- Isolated RAVE information (directly from V0Finder module → "V0extra") that has not been saved before, in order to check if it causes the Ks-loss:
almost all ($\approx 0.69\%$) remaining Ks losses due to a RAVE-Fail!
- Implemented missing error message in V0Finder module.
- What information do we have about Failed events?

- B2Display: Tracks do not touch in XY-plane. (really far apart?)

Resolution plots

- resolution: $Rad - MCRad$.
- the width of the histogram is defined as:
 $quantile(0.75) - quantile(0.25)$ ("Inter quantile range")



Resolution plots

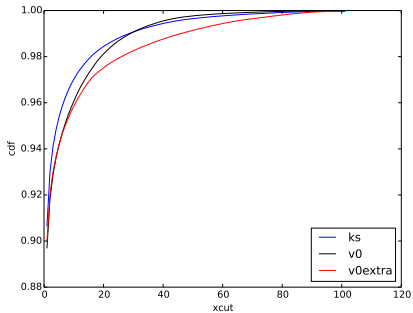
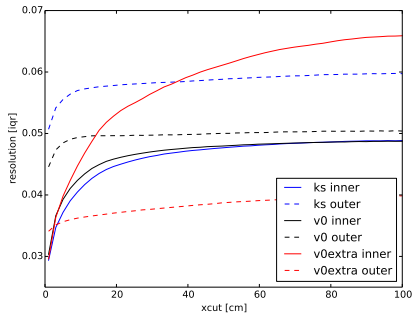


Figure : **left:** reso-variation over outlier cut.
(*inner = rad < 8cm, outer = rad > 8cm*);
right: cdf over outlier cut

Resolution plots

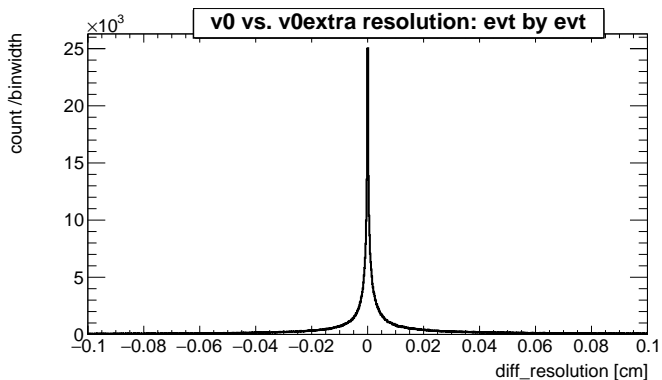


Figure : event by event difference of res_{V0} and $res_{V0extra}$

MCMatching - Redundance

- 2 MCMatching methods for V0s encountered in basf2:
 - `analysis/modules/MCMatching/src/MCMatchingModule.cc`
 - `tracking/modules/mcMatcher/src/MCV0Matchermodule.cc`
- analysis MCMatching seems more elaborate (includes more cases like pion decaying instantly into a muon)
- take MCMatching at reconstruction level out?

How to use V0Finder?

- Idea: Modified Particle Combiner uses V0 info if provided. Otherwise standard Ks Reconstruction. Maximal information exploitation.
- Implement Rave-Info in V0 object?
- develop concept for standard particle lists. Apply cuts in one visible spot only. Create several lists with different chi2 cuts?