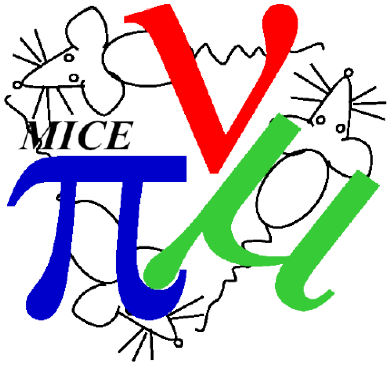
Emittance Evolution



C. Rogers, ISIS Intense Beams Group
Rutherford Appleton Laboratory

Overview

- Chris Hunt new digitizer
  - Hope for improved efficiency
- Reminder of efficiency algorithm
  - Consider MC sample
    - Must be in truth upstream sample
      - In practice this is the same as recon upstream sample
    - True track within fiducial volume
    - True track must be a muon in every tracker station
    - True momentum in 90 < TKD < 170 MeV/c
  - Consider Reco sample
    - Must be in recon upstream sample
    - Exactly one track in TKD
    - Recon track within fiducial volume
    - Chi2 less than **4** (new recon) **8** (old recon)
- Try running on
  - hybrid MC
  - 4-140 beam
  - lH2 empty

Recon file

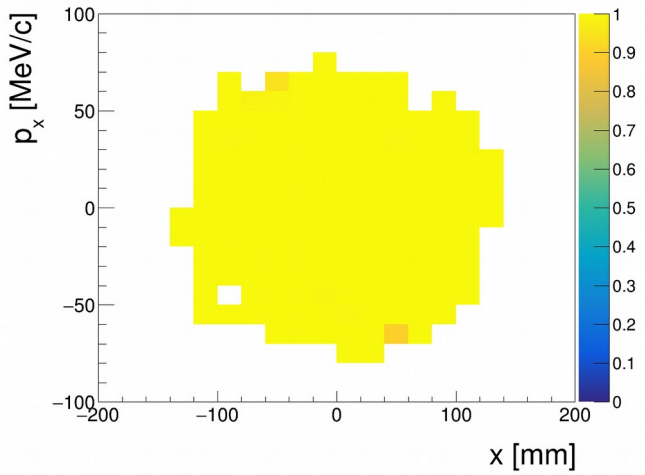
```
#!/usr/bin/env python

"""
Simulate the MICE experiment

This will simulate MICE spills through the entirety of MICE using Geant4, then
digitize and reconstruct TOF and tracker hits to space points.
"""

import io   #  generic python library for I/O

import MAUS # MAUS libraries

def run():
    """ Run the macro
    """

    # This input generates empty spills, to be filled by the beam maker later on
    my_input = MAUS.InputCppRootData()

    # Create an empty array of mappers, then populate it
    # with the functionality you want to use.
    my_map = MAUS.MapPyGroup()

    # Pre detector set up
    #my_map.append(MAUS.MapCppMCReconSetup())  #  geant4 simulation
    # SciFi
    my_map.append(MAUS.MapCppTrackerMCDigitization()) # SciFi electronics model
    my_map.append(MAUS.MapCppTrackerClusterRecon()) # SciFi channel clustering
    my_map.append(MAUS.MapCppTrackerSpacePointRecon()) # SciFi spacepoint recon
    my_map.append(MAUS.MapCppTrackerPatternRecognition()) # SciFi track finding
    my_map.append(MAUS.MapCppTrackerPRSeed()) # Set the Seed from PR
    my_map.append(MAUS.MapCppTrackerTrackFit()) # SciFi track fit
    #my_map.append(MAUS.MapCppTrackerTOFCombinedFit())
    #my_map.append(MAUS.MapCppTrackerDownstreamReFit())

    # Then construct a MAUS output component - filename comes from datacards
    my_output = MAUS.OutputCppRoot()

    # can specify datacards here or by using appropriate command line calls
    datacards = io.StringIO(u"")

    # The Go() drives all the components you pass in, then check the file
    # (default simulation.out) for output
    MAUS.Go(my_input, my_map, MAUS.ReducePyDoNothing(), my_output, datacards)

if __name__ == '__main__':
    run()
```

Samples

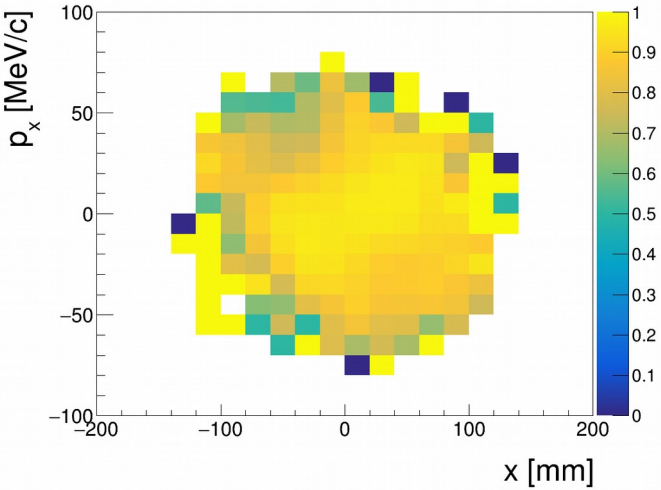
New recon			**MC**		
upstream_cut	408209	1.0000	mc_true_us_cut	408207	1.0000
scifi_tracks_ds	397206	0.9730	mc_stations_ds	396454	0.9712
chi2_ds	353184	0.8652	mc_scifi_fiducial_ds	385909	0.9454
scifi_fiducial_ds	345225	0.8457	mc_p_ds	385543	0.9445
downstream_cut	345225	0.8457	mc_true_ds_cut	385543	0.9445
Old recon			**MC**		
upstream_cut	767763	1.0000	mc_true_us_cut	767761	1.0000
scifi_tracks_ds	734954	0.9573	mc_stations_ds	743935	0.9690
chi2_ds	724396	0.9435	mc_scifi_fiducial_ds	724416	0.9435
scifi_fiducial_ds	707376	0.9213	mc_p_ds	723671	0.9426
downstream_cut	707376	0.9213	mc_true_ds_cut	723671	0.9426

X px efficiency

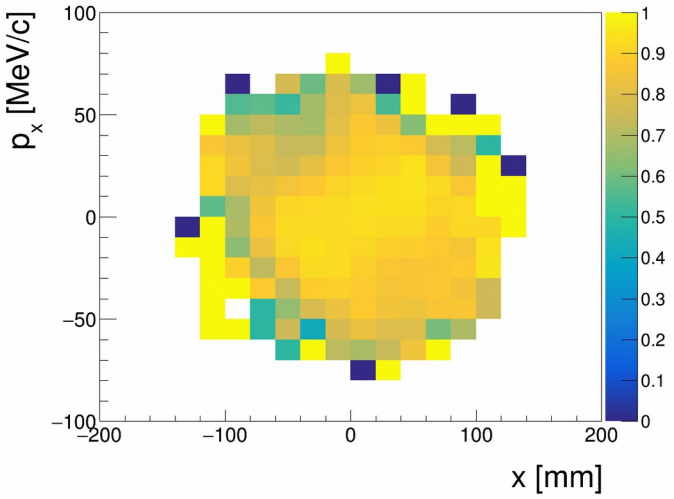
Scifi tracks



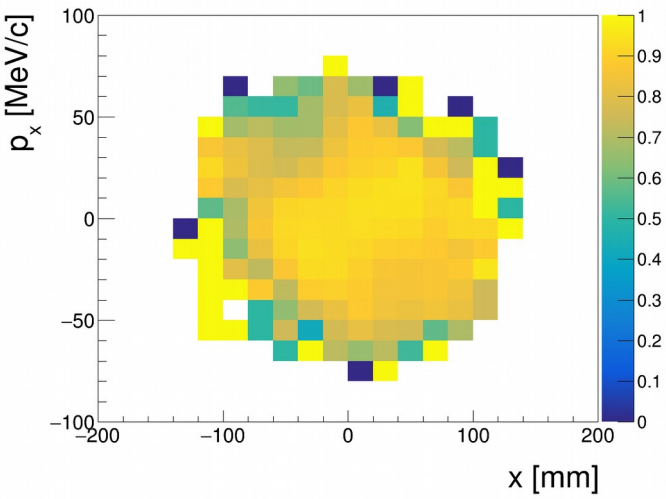
Scifi fiducial



Scifi chi2

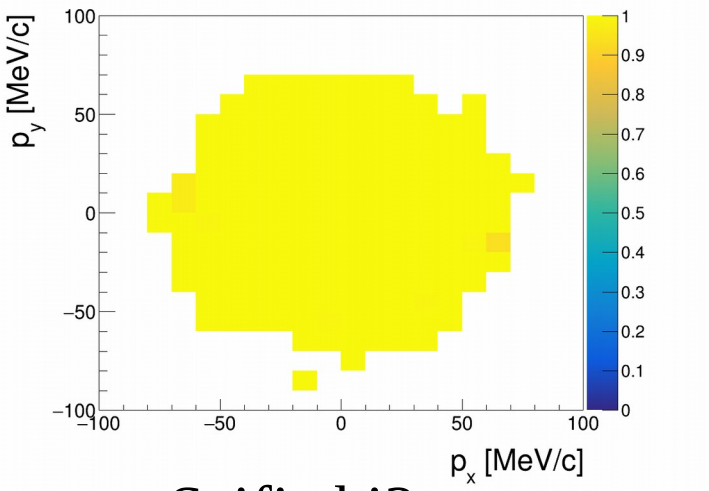


All

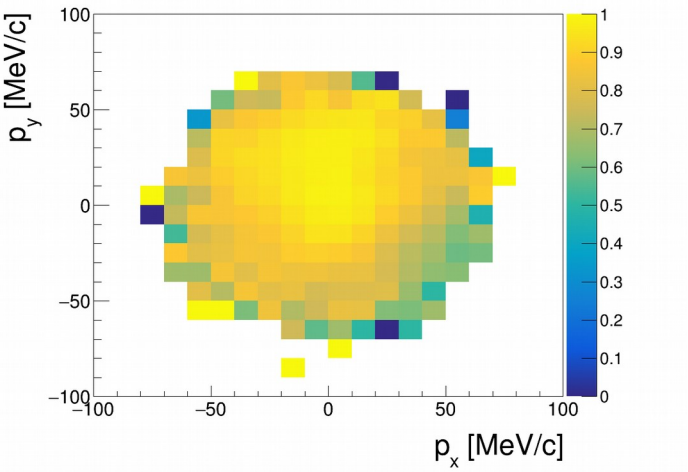


Px py efficiency

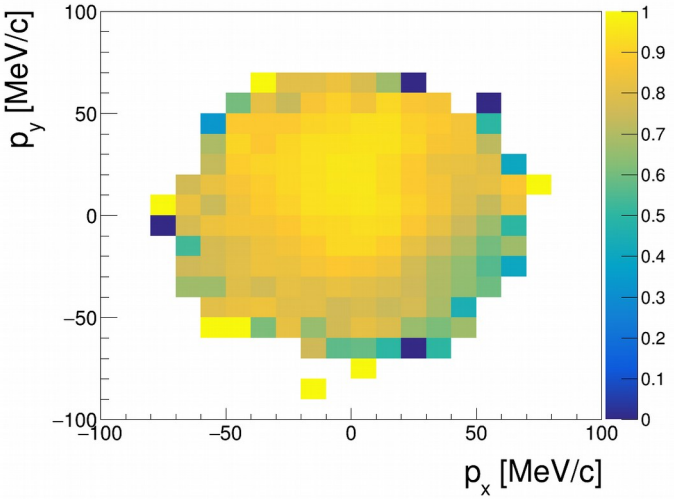
Scifi tracks



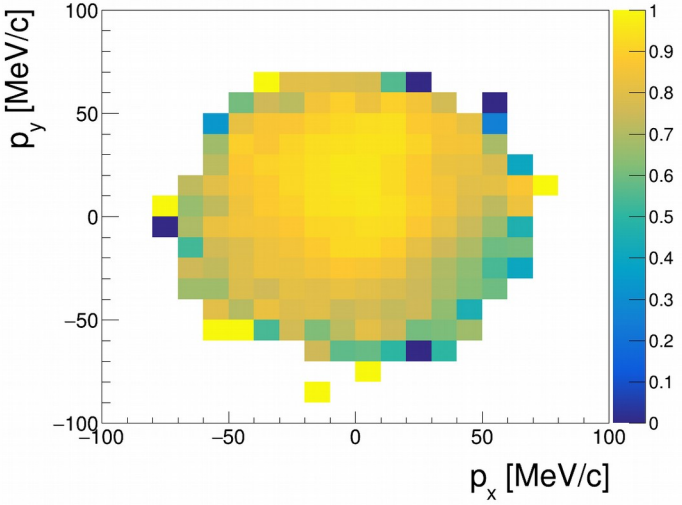
Scifi fiducial



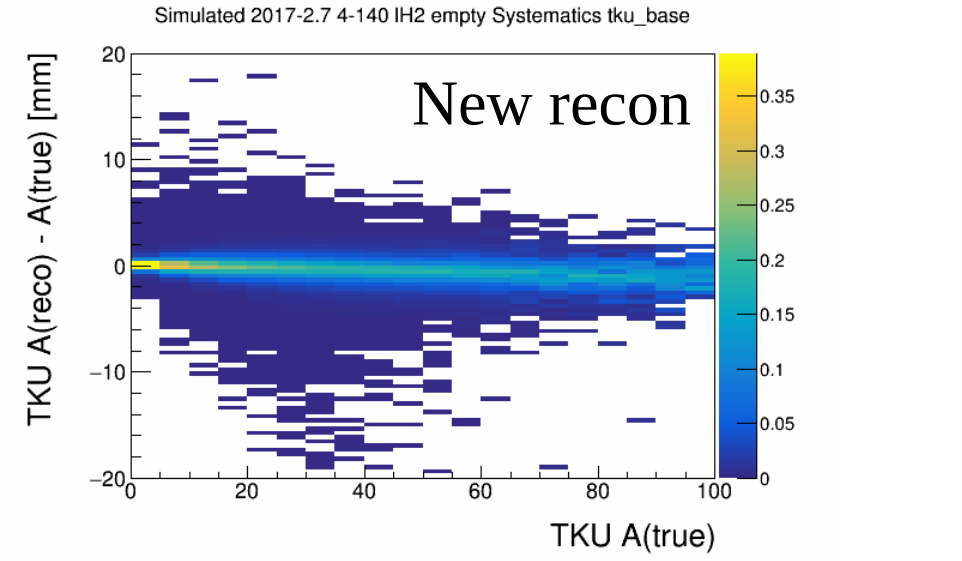
Scifi chi2

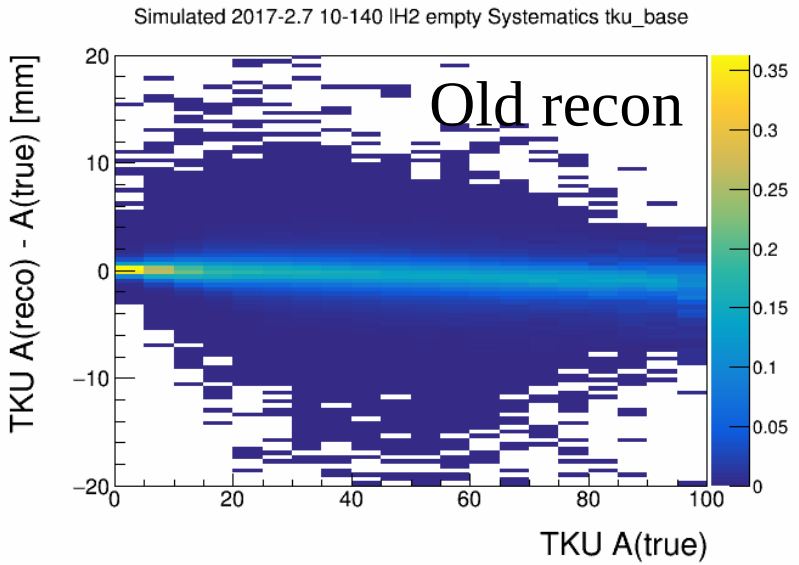


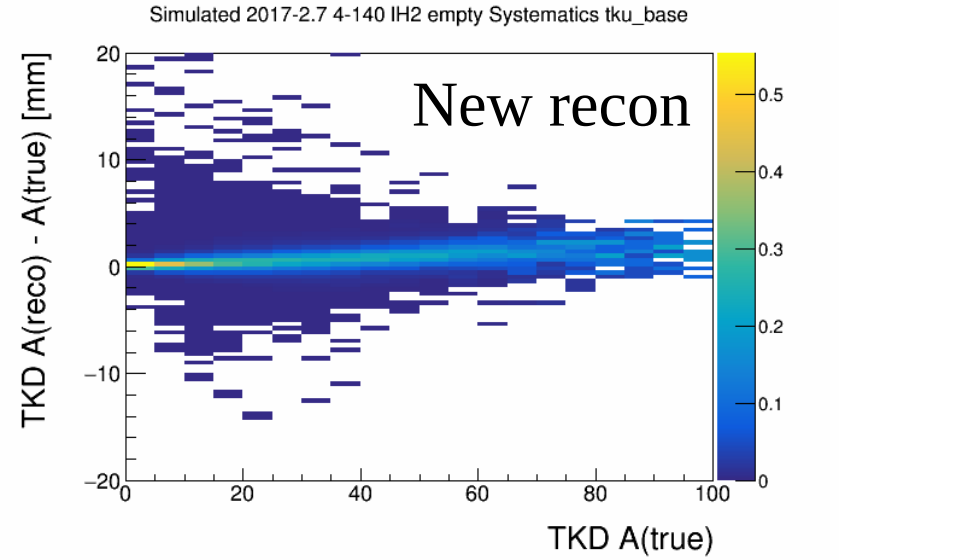
All

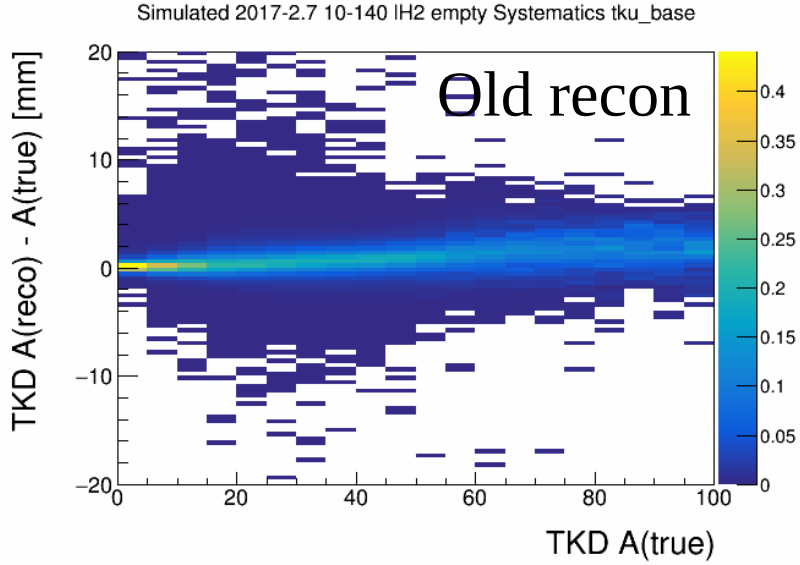


Amplitude residual

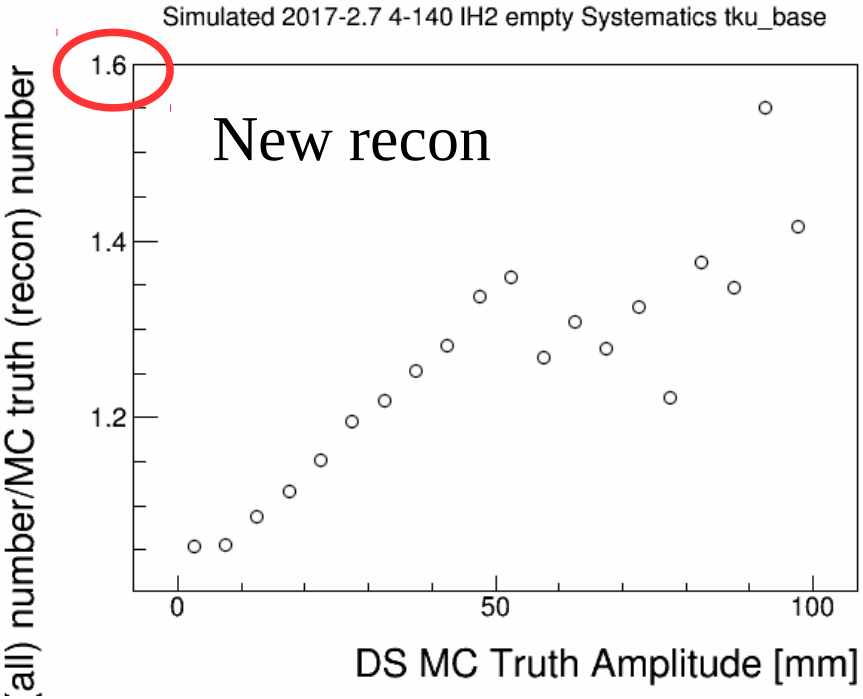


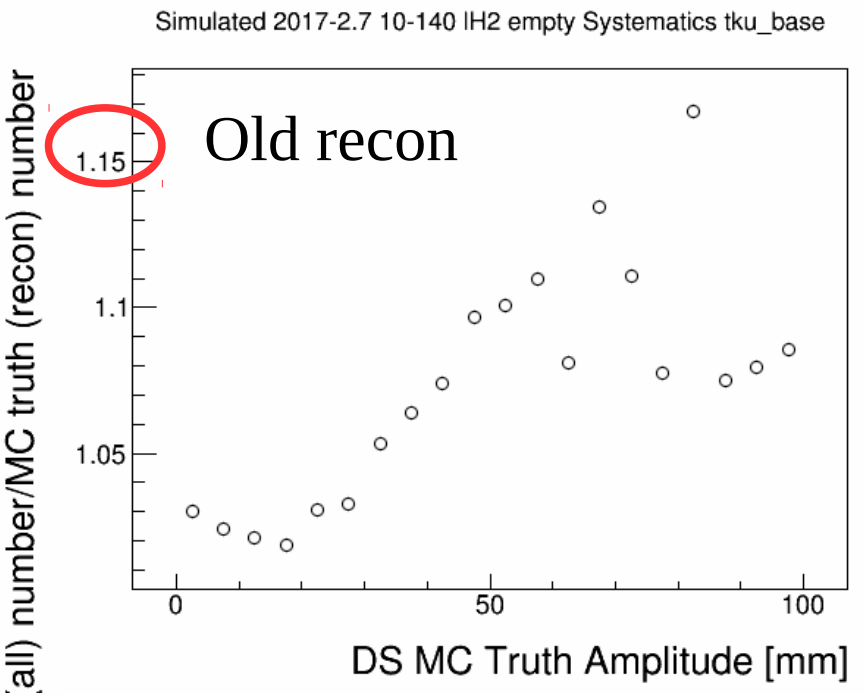






Amplitude efficiency correction





Conclusions

- Efficiency looks worse
- Guess tracks are being reconstructed with too high radius (or pt)
  - Leads to poor chi2
- Probably it means I have made a mistake when running the reconstruction job!