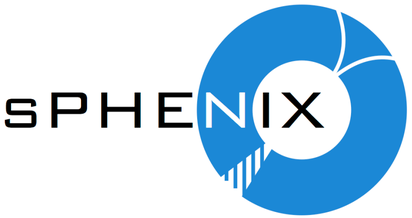
sPHENIX Tracking

Joe Osborn

February 19, 2021

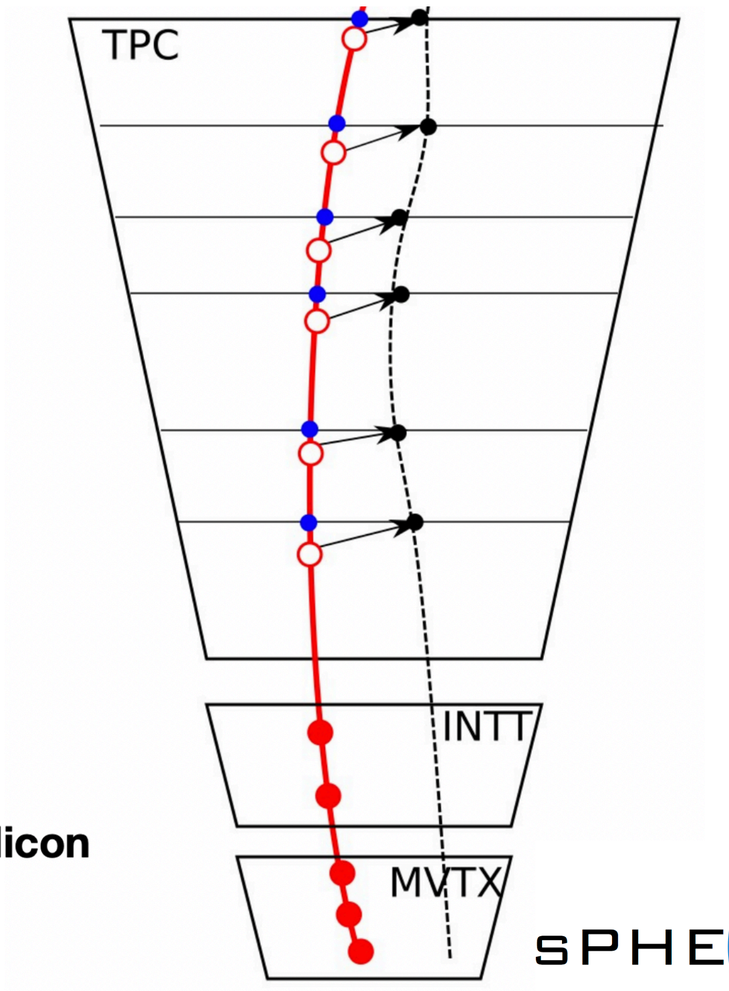


sPHENIX Tracking Overview

- Since last year, sPHENIX has undergone a complete overhaul of the track reconstruction software
- Main driver was replacing GenFit with Acts to improve speed
    - TPC seeding also needed replacing - also driven by computational speed requirements
- As a part of the overhaul, the strategy has shifted to accommodate Acts features

sPHENIX Tracking

- Current sPHENIX track reconstruction uses MVTX, INTT (collectively, silicon) and TPC
- No space charge distortions implemented yet - ongoing development and validation



sPHENIX

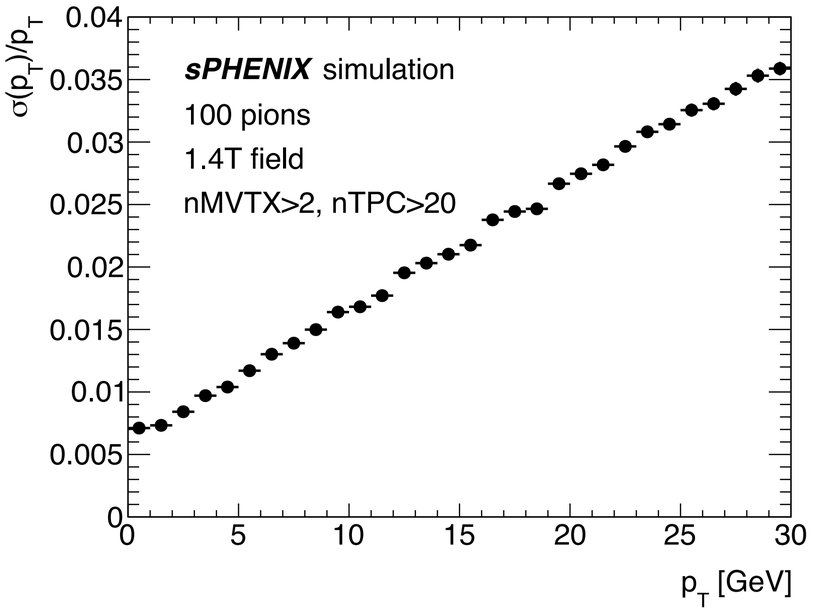
Current Tracking Strategy

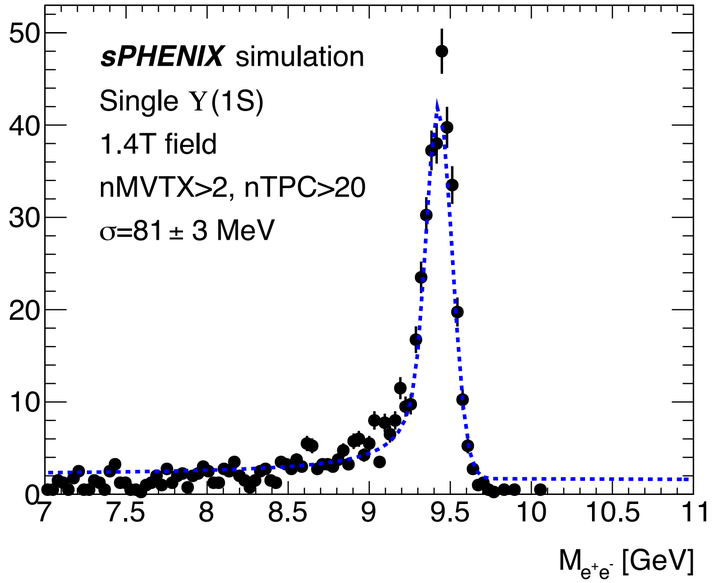
- Cellular automaton based seeder in the TPC (Michael Peters, MIT)
- Acts seeding deployed in the inner silicon layers
- Initial vertexing performed with silicon stubs to get first vertex estimate
- Use a track stub matching module to match TPC track stubs with silicon track stubs based on $\Delta\phi$ and $\Delta\eta$ of stubs
- Fit the entire track with Acts
- Vertexing performed by Acts with final track fits

sPHENIX

Current Tracking Performance

- Performance has largely been tuned only in low occupancy events
  - So far pretty good. A number of improvements could still be made
- Performance in high occupancy is currently being evaluated, in conjunction with the Mock Data Challenge





Ongoing O(~month) Efforts

- Implementing and tuning vertexing (Joe)
    - Final Acts vertexing works; however, initial vertexing with silicon stubs fails ~20% of the time within Acts in high occupancy. Trying to understand why
- Space charge distortions (SCD) (Joe)
    - SCD studies have been performed solely in GenFit. Need to Acts-ify them and make it a part of our default tracking chain
- TPC seeding in high occupancy (Michael Peters, MIT)
    - TPC seeding currently is not very efficient in high occupancy. Improving/tuning
- TPC-Silicon matching (Tony Frawley, FSU)
    - TPC-silicon matching is not great in high occupancy. Working on understanding why and how to solve it
- TPC clustering (Tony Frawley, FSU)
    - Current TPC clustering algorithm is slow and not modular. Tony is modularizing it for concurrent development

sPHENIX

Action Areas

- sPHENIX starts data taking in ~2 years time
- Goal is to have a track reconstruction package that is efficient, fast, and consumes low memory in 50 kHz pileup rates
  - 5 second CPU budget per event, >90% track reconstruction efficiency in high occupancy Au+Au event
- Some projects to help get us there
  - Memory consumption/framework design optimization
    - Currently tracking uses ~12 GB of memory per job. Objects not optimized (i.e. mostly flat maps) and data is copied many places
  - Multi-threading
    - Could improve speed on a number of levels
  - A good high occupancy TPC clustering algorithm
    - Current TPC clustering algorithm is simple. Tony is working on improving the framework and a new student will be working on improving the current algorithm, which is not the best in high occupancy