SW tutorial, Vertexing in FCCAnalyses

This tutorial will teach you how to:

- fit some tracks to a common vertex in FCCAnalyses,
- recontruct the primary vertex and the primary tracks
- retrieve the tracks corresponding to a specific flavour decay in FCCAnalyses
- produce flat ntuples with observables of interest with FCCAnalyses
- build your own algorithm inside FCCAnalyses

It consists in two parts:

- Run basic vertex fits in FCCAnalyses
  - May also use these examples to explain basic things on FCCAnalyses if needed
- Setting up building blocks for a $\tau \rightarrow 3\mu$ analysis
  - More advanced

Based on (existing) Delphes files in the edm4hep format.

Run basic vertex fits in FCCAnalyses

- Reconstruct the primary vertex in Z → had events with a beam-spot constraint
  - Using built-in functionnalities in FCCAnalysis
  - Look at the vertex resolutions and the pulls of the vertex fit

- Compare the number of primary tracks in Z → uds and in Z → bb event

- Plot the total momentum carried by these primary tracks and compare Z → uds and in Z → bb
  - For this, we will need to write a simple piece of user code

It will likely look simple for those who have some prior experience with FCCAnalyses. Still, you should learn something, if you have not played with the vertexing yet.

It may look not so simple for those who discover FCCAnalyses today :-(
But we'll try to guide beginners through it.

Setting up building blocks for a $\tau \rightarrow 3\mu$ analysis

- Start from an example which reconstructs the B decay vertex in a specific (simple) B decay chain. Adapt the example to $\tau \rightarrow 3\mu$.

- Using the tracks that are MC-matched to the muons from the tau decay in a signal file: Reconstruct the tau mass, using the momenta of the muon tracks at the tau decay vertex

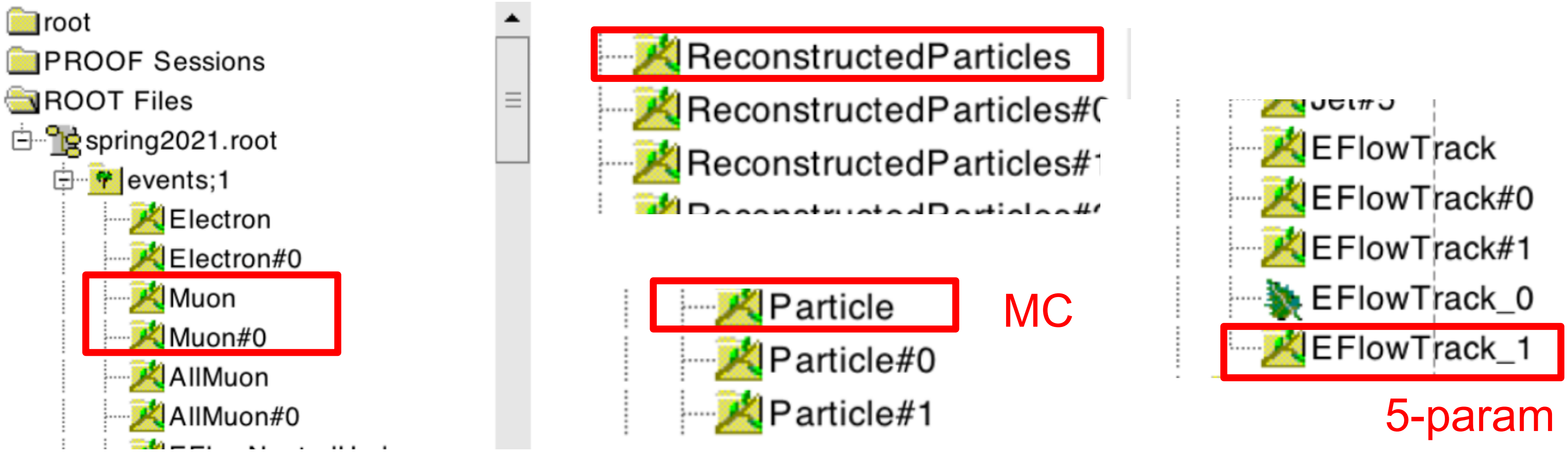
- Build fake muons from $\pi \rightarrow \mu$, such that we can look at the $\tau \rightarrow 3\pi\,\nu$ background

- Run over existing “datasets” using the analysis framework
  - Process the $\tau \rightarrow 3\mu$ signal and the $\tau \rightarrow 3\pi\,\nu$ background (produce ntuples with analysis-level variables)
  - Use the framework to histogram variables with some selections, and produce luminosity-normalised signal + background plots
  - You’ll need to have read access to files under fccsw: please send us your lxplus userid such that we can grant you access.

Backup

Basics: EDM4Hep files

See https://github.com/HEP-FCC/FCCAnalyses/tree/master/examples/basics



root
PROOF Sessions
ROOT Files
spring2021.root
events;1
Electron
Electron#0
Muon
Muon#0
AllMuon
AllMuon#0

ReconstructedParticles
ReconstructedParticles#0
ReconstructedParticles#1

Particle
Particle#0
Particle#1

Jet#5
EFlowTrack
EFlowTrack#0
EFlowTrack#1
EFlowTrack_0
EFlowTrack_1

MC

5-param tracks

- Branches without a pound (#) in their name: Electron (1), Muon (2), AllMuon (3), EFlowNeutralHadron (4), Particle (5), Photon (6), ReconstructedParticles (7), EFlowPhoton (8), MCRecoAssociations (9), MissingET (10), ParticleIDs (11), Jet (12), EFlowTrack (13), EFlowTrack_1 (14). They refer to collections of objects.
  - NB: "Particle" denotes the collection of Monte-Carlo particles. "Muon" contains the isolated muons, while "AllMuon" contains all muons, isolated or not.
- Branches with a pound in their name: Each of the object collections listed above, e.g. "Collection", has up to six associated collections of references, i.e. indices that point to another or to the same object collection. They are labeled Collection#i, with i = 0 ... 5. For example, the Muon collection has one single associated collection of references, Muon#0.

  To figure out which collection is pointed to by Muon#0 (or by any other collection of references), one can look at the value of Muon#0.collectionID (see screenshot below). The collectionID of Muon#0 is the collection number 7 (in the example file used here), which, in the list of "object collections" above, corresponds to the collection of ReconstructedParticles. Indeed, the Muon collection itself contains nothing (see screenshot below): all the information is contained in the ReconstructedParticles. The Muon collection, together with Muon#0, just provides a convenient way to access, among the ReconstructedParticles, those that were identified as muons.

Basics: FCCAnalyses

- See https://github.com/HEP-FCC/FCCAnalyses
  - analysers/dataframe : contains many built-in functions to manipulate the edm4hep collections, and various buikt-in algorithms
  - Example:
    - MCParticle.h contains code to get the phi, theta angles of MCParticles

- Useful analysis information here:
  https://github.com/HEP-FCC/FCCeePhysicsPerformance/tree/master/General

- Oldish talk that describes FCCAnalysis (Clement):
  https://indico.cern.ch/event/982690/contributions/4138504/attachments/2162441/3648904/FCCAnalyses_clement.pdf