Progress on particle flow for ALLEGRO

Giovanni Marchiori
(APC-Paris CNRS/IN2P3)

29/1/2025

NUCLÉAIRE
& PARTICULES

Intro

- *Technical* demonstration by Archil that PandoraPFA can run on ALLEGRO reconstructed events
  - Managed to make reconstruction work for electrons/photons/pions/KL in barrel, at given polar angle (60 degrees) in detector, using CLD pandora settings (turning off muon reconstruction), tracks and ecal/hcal barrel hits
  - PFO objects constructed and with decent calibrations, but typically misidentified
- Plan:
  - Import/consolidate Archil's changes into key4hep
  - Understand what is missing or needs to be changed in order to make Pandora
    - run for the whole detector and particle types
    - have a good physics output
    - This also means going through Pandora documentation (and, more broadly, papers on p-flow) and through its software
  - Ideally, try to split actions needed in granular subtasks that can then be worked on independently / in parallel

Starting point

- As shown by Archil, making PandoraPFA technically run for ALLEGRO requires changes to a certain number of packages:
  - **k4RecTracker**: extrapolate tracks to calorimeter
  - **k4geo**: add information for reconstruction to detector extension
  - **DDMarlinPandora**: retrieve information from reco and use it to fill the objects passed as input to Pandora
  - **FCC-config**: the reconstruction steering script, to schedule the EDM4hep<->LCIO conversions and running Pandora
  - Additionally: a hack of **LCContent** to temporarily disable a check that crashes on ALLEGRO (reason to be understood and better solution to be found)
- Re-enabling muon reconstruction also requires muon cells so changes to **k4RecCalorimeter** in order to have some muon digitisation (we only have muon hits ATM)

Recent progress on the code

- **Track extrapolator**:
  - Determine automatically B field from Geant rather than passing it as an option to the algorithm
  - Obtain ECal barrel inner radius from detector data rather than passing it as an option
  - Make the extrapolation work also for tracks in the endcap region
- **Muon reconstruction**:
  - Pandora's inputs are tracks and ecal/hcal/muon hits, so basically the main missing ingredient for us are muon hits
  - Bug fixes to simple cylindrical volume used to model muon tagger in endcap (w/ Archil and Alvaro) and implementation of muon digitiser (w/ Archil)
- **Reconstruction steering**
  - Re-enable muon algorithms, pass muon cell collections to Marlin wrapper
  - Little fix in EDM4hep<->LCIO conversion to get rid of a warning at run time

Current status

- **k4RecTracker**: algorithm for track extrapolation
  - https://github.com/key4hep/k4RecTracker/pull/43, ready for review
- **k4geo**: simple cylinder detector (SimpleCylinder_geo_o1_v02) for muon tagger, with both endcaps in same mother volume and calo data for Pandora; calorimeter data
  - https://github.com/key4hep/k4geo/pull/419, almost ready (need to check Archil's calculation of interaction/radiation lengths in various sub detectors, should probably be at normal incidence rather than 60 degrees)
  - more work needed in the future e.g. to add calo data to ECAL endcap and also to compute properly information about cell size (now passing a fixed size in cm, should rather switch to a "pointing" theta-phi grid for the ECAL but needs modifications on Pandora's side)
- **k4RecCalorimeter**: cell positioning tool for SimpleCylinder_geo_01_v02 with FCCSWGridPhiTheta_k4geo readout, allows using existing Calo digitiser (CreatePositionedCaloCells) to digitise muon hits (simple sum of energies of hits in same cell)
  - https://github.com/HEP-FCC/k4RecCalorimeter/pull/136, ready for review
- **DDMarlinPandora**: wrap info from reconstruction into track and hit objects for Pandora
  - PR prepared by Archil: https://github.com/iLCSoft/DDMarlinPandora/pull/31, review already ongoing
- **LCContent**: no PR yet (code is in branch)
- **FCC-config**: added option —pandora to reco steering file (run_digi_reco.py) that schedules the PandoraPFA algorithms (and uses legacy IO rather than IOSvc since EDM4hep<->LCIO does now work yet with functionals)
  - https://github.com/HEP-FCC/FCC-config/pull/230, draft (want to do couple more checks about input cell collections)

Doc

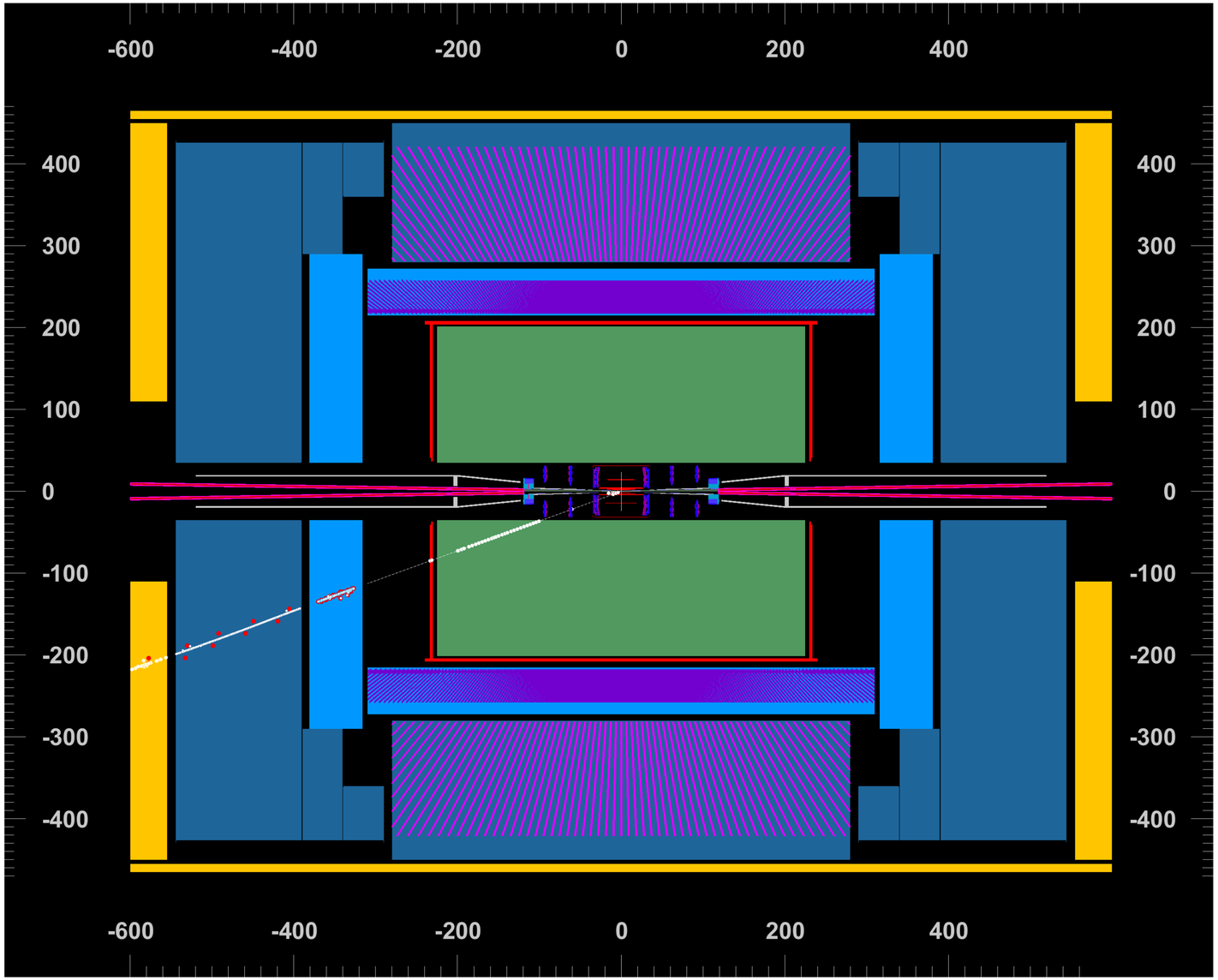
- I am keeping notes of the process/more details in
  - https://docs.google.com/document/d/1OCx4XoUMeOlNsqBgI8WvUauzzB15CB7RxQqn8-fiNJo/edit?tab=t.0#heading=h.560gy9whw9a4
  - https://docs.google.com/presentation/d/1wcsVH0zXNwo9Da8IOmBw-2nvegkBs0aIczUQpE62aIk/edit?usp=sharing
  - For instance, (1) list of algorithms run by default sequence copied from CLD and links to their definition in the code, and (2) how/where Calo hit information is used in PandoraPFA (to understand if we're filling the info correctly in the wrappers and when calculating the detector data)

- positionVector: of course
- x0: only for LArTPC usage
- expectedDirection: in src/LCHelpers/FragmentRemovalHelper.cc, src/LCClustering/ConeClusteringAlgorithm.cc, src/LCTopologicalAssociation/IsolatedHitMergingAlgorithm.cc, src/LCTopologicalAssociation/ProximityBasedMergingAlgorithm.cc, src/LCParticleId/MuonReconstructionAlgorithm.cc, in src/Objects/Cluster.cc to calculate initial direction ..
- cellNormalVector: in src/Helpers/ClusterFitHelper.cc, src/LCPlugins/LCShowerProfilePlugin.cc, src/LCPlugins/LCParticleIdPlugins.cc ..
- CellGeometry: used in CaloHitManager::MergeCaloHitFragments or for CaloHit::CalculateCellLengthScale()
- CellSize0, CellSize1: used in LCSoftwareCompensation but also to calculate CellLengthScale
- CellThickness: to calculate energy density (E/volume) in LCSoftwareCompensation (with CellSize0, CellSize1)
- nCellRadiationLenghts: EM shower ID (link)
- nCellInteractionLengths: not used
- Time: src/LCHelpers/ClusterHelper.cc
- inputEnergy: src/Helpers/ClusterFitHelper.cc, src/LCPlugins/LCEnergyCorrectionPlugins.cc
- mipEquivalentEnergy: calculated as energy * eCalToMip * absorberCorrection, user in src/CaloHitCreator.cc, src/LCPlugins/LCEnergyCorrectionPlugins.cc, src/LCUtility/CaloHitPreparationAlgorithm.cc, ..
- electromagneticEnergy: calculated as energy * eCalToEMGeV or hCalToEMGeV depending on detector in src/CaloHitCreator.cc, used in src/Objects/Cluster.cc to calculate cluster EM energy, used for instance in src/LCParticleId/PhotonRecoveryAlgorithm.cc and src/LCPfoConstruction/PfoCreationAlgorithm.cc and src/LCPlugins/LCParticleIdPlugins.cc, ..
- hadronicEnergy: calculated as energy * eCalToHadGeVBarrel/EndCap or hCalToHadGeV in src/CaloHitCreator.cc, used in src/Objects/Cluster.cc, src/LCPlugins/LCSoftwareCompensation.cc, src/LCHelpers/ClusterHelper.cc, src/LCTopologicalAssociation/ConeBasedMergingAlgorithm.cc, include/LCReclustering/TrackDrivenMergingAlg.h, …
- isDigital: in src/LCUtility/CaloHitPreparationAlgorithm.cc
- hitType: ECAL, HCAL, MUON … yes
- hitRegion: BARREL, ENDCAP, … include/Pandora/PandoraEnumeratedTypes.h .. yes
- CellLengthScale: used in fit to cluster centroids, in src/LCUtility/CaloHitPreparationAlgorithm.cc, src/LCClustering/ConeClusteringAlgorithm.cc, src/LCHelpers/FragmentRemovalHelper.cc, src/LCTopologicalAssociation/MipPhotonSeparationAlgorithm.cc, src/LCParticleId/MuonReconstructionAlgorithm.cc
- Layer: src/LCPlugins/LCParticleIdPlugins.cc,
- pseudoLayer: src/Objects/OrderedCaloHitList.cc, src/Objects/Cluster.cc, src/LCHelpers/FragmentRemovalHelper.cc, src/Managers/CaloHitManager.cc, src/LCUtility/CaloHitPreparationAlgorithm.cc, src/Helpers/ClusterFitHelper.cc, src/LCClustering/ConeClusteringAlgorithm.cc, …

- CaloHit preparation: https://github.com/PandoraPFA/LCContent/blob/master/include/LCUtility/CaloHitPreparationAlgorithm.h
  - Initialize a binary tree with k-dimensional data representing the hits. Calculate number of nearby hits
- Event preparation: https://github.com/PandoraPFA/LCContent/blob/master/src/LCUtility/EventPreparationAlgorithm.cc
  - Filter track list to select tracks to be used during clustering (keep tracks without daughters), save the filtered list and set it to be the current list for subsequent algorithms
  - Split input calo hit list into ecal/hcal and muon calo hits, save the lists, setting the ecal/hcal list to be the current list for subsequent algorithms
- Standalone muon reconstruction: https://github.com/PandoraPFA/LCContent/blob/master/src/LCParticleId/MuonReconstructionAlgorithm.cc
  - Run ConeClustering algorithm on muon calo hits and create clusters (https://github.com/PandoraPFA/LCContent/blob/master/src/LCClustering/ConeClusteringAlgorithm.cc, https://github.com/PandoraPFA/LCContent/blob/master/include/LCClustering/ConeClusteringAlgorithm.h)
  - Simple cuts on cluster properties
  - Loop over all non-associated tracks in the current track list to find bestTrack associated to given cluster
  - Apply simple cuts on track properties
  - Create helix, extrapolate to muon detector, Calculate separation of helix and cluster inner centroid, use it to keep association or not
  - Add calohits to muon cluster
  - Create muon PFO
  - Save lists of tracks and hits without muon tracks/hits and set them as current lists for subsequent algorithms
- Standalone photon reconstruction: https://github.com/PandoraPFA/LCContent/blob/master/include/LCParticleId/PhotonReconstructionAlgorithm.h, https://github.com/PandoraPFA/LCContent/blob/master/src/LCParticleId/PhotonReconstructionAlgorithm.cc

For fun

- Updated event display code to also show muon hits & cells (barrel and endcap) and ECal, HCal endcap hits/cells



To-do

- Implement comments from PR review
- Identify/determine list of tasks for next steps
  - Consolidate calculation of inputs to Pandora (with better understanding of how/where they’re used)
  - Determine proper calibrations for various particle types (particularly for MIP and hadronic scale)
  - Particle reco/ID: retune/modify/develop new algorithms for proper identification of particles
  - …
- Share tasks with anybody interested in the effort