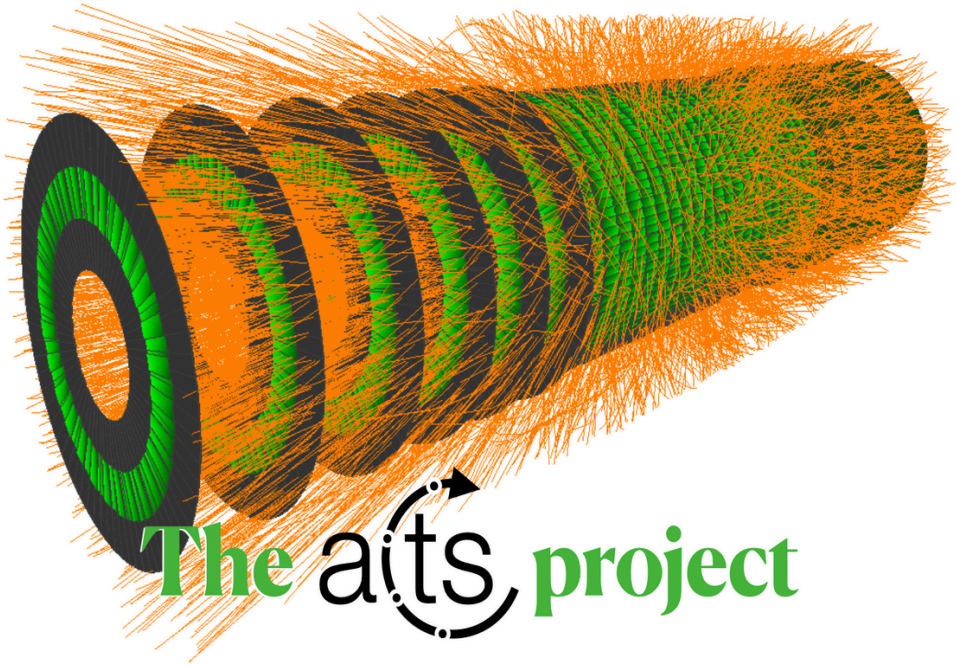
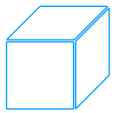
ACTS : A Common tracking software



Corentin Allaire

Acts: A Common Tracking Software

- Open source and **experiment-independent** toolkit for track reconstruction: https://github.com/acts-project/acts
- Developed with **modern C++** with unit testing and continuous integration
- Minimal dependency for ease of building
- **Community** project: used by many **experiments** and **R&D** projects



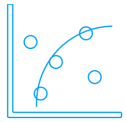
Geometry/Detector*
(Surface based geometry)

Plugins to DD4hep, TGeo, etc.

Event Data Model
target track reconstruction

backend separation
with different I/O models

Propagation
parameter + covariance
transport through
magnetic field



Seeding
Seed finding with
Triplet seeder,
OrthogonalSeedFinder

Track Fitting
parameter estimation
with Kalman Filter,
GSF, GX2F**

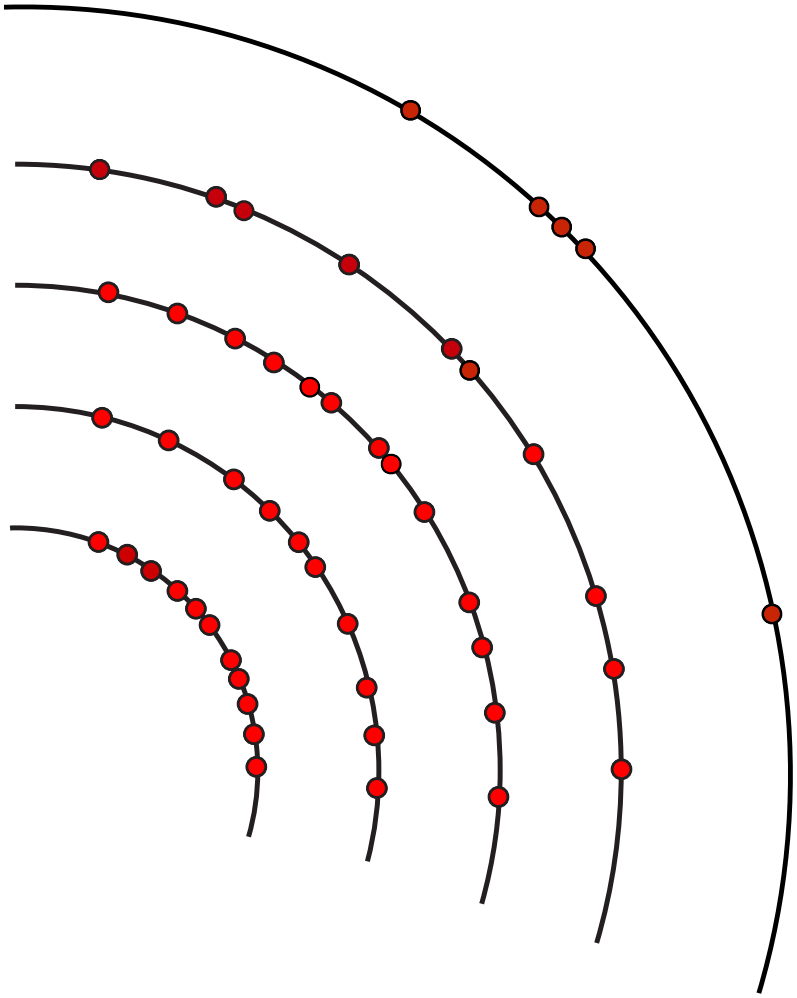
Combinatorial track finding
Combinatorial Kalman Filter
for track finding

Vertex finding + fitting
Iterative, multi variant
primary vertex finders
and fitters

Detector alignment
KF based alignment
functionality

Track Reconstruction

Space point formation
Clustering



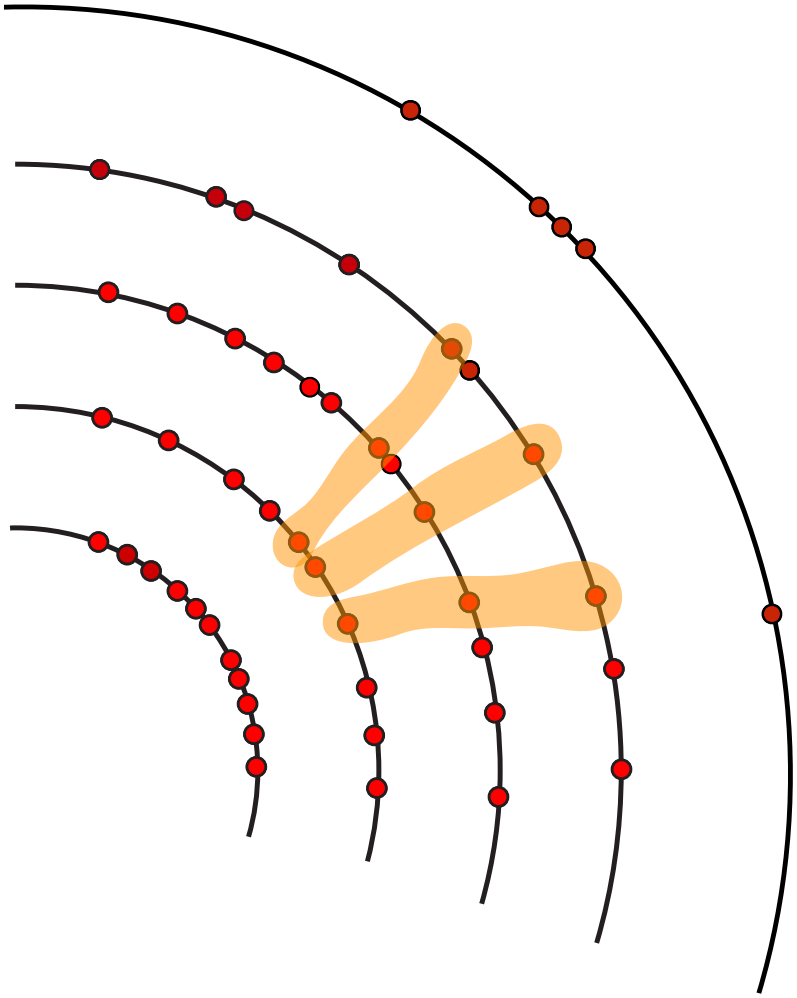
- Hits in the detector are collected to create measurement points

Track Reconstruction

Space point formation
Clustering

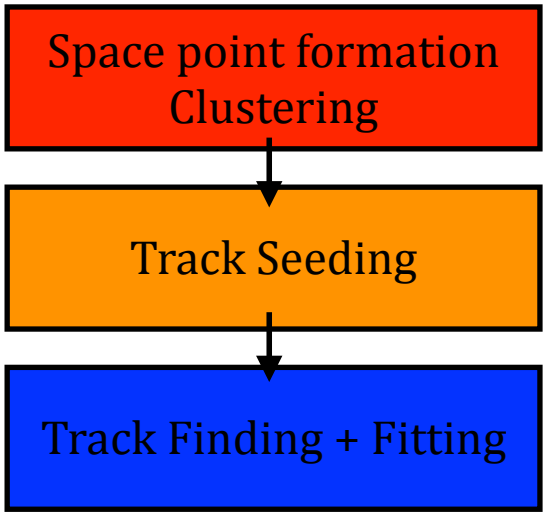
↓

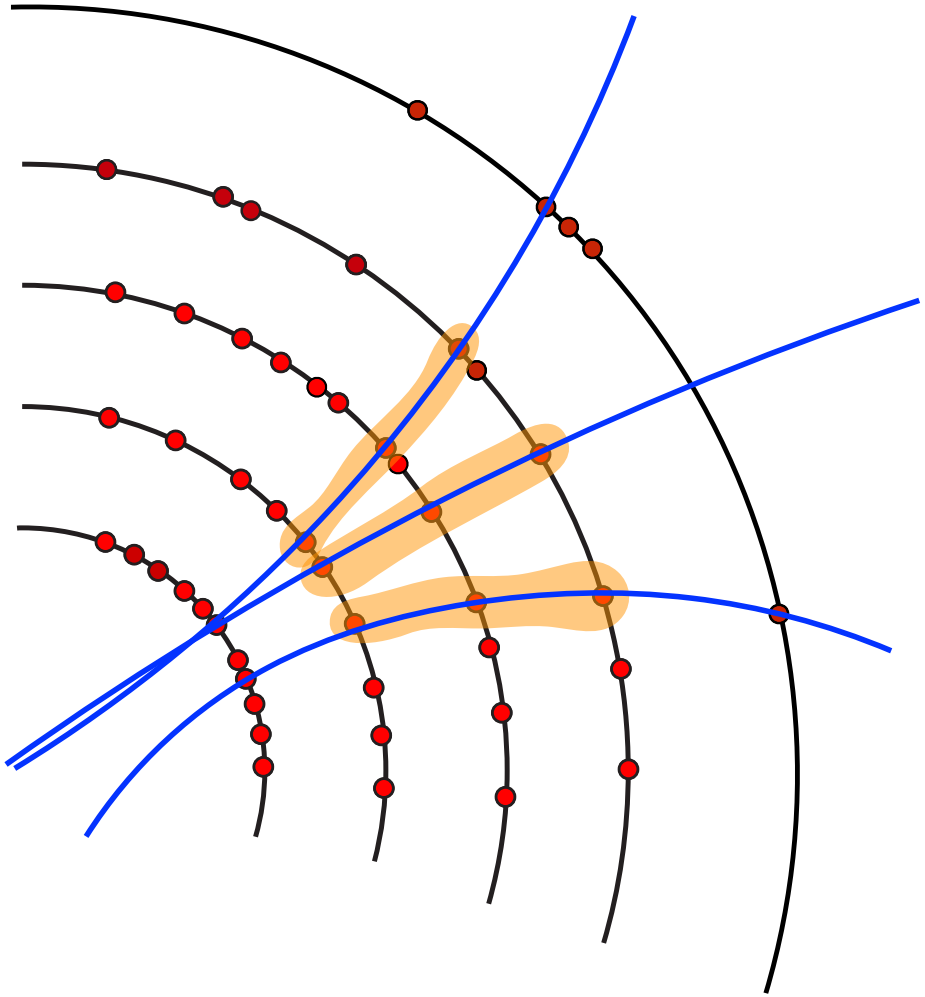
Track Seeding



- **Seeds** are groups of **three measurements** compatible with the basic track hypothesis
- Will be extended to create the track candidate

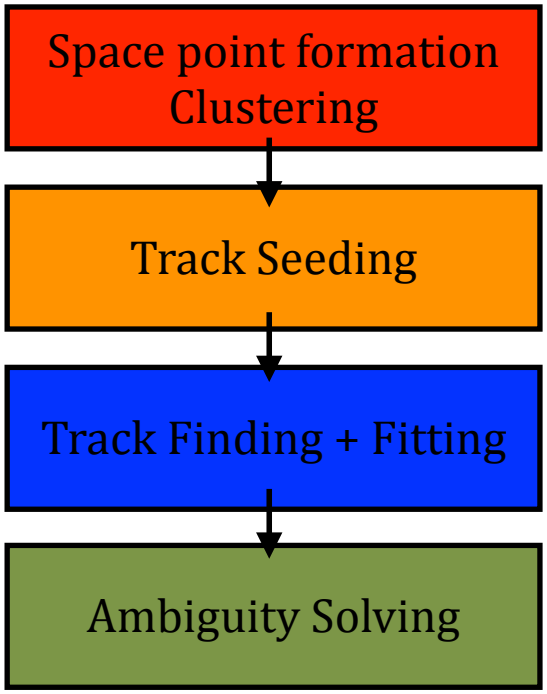
Track Reconstruction

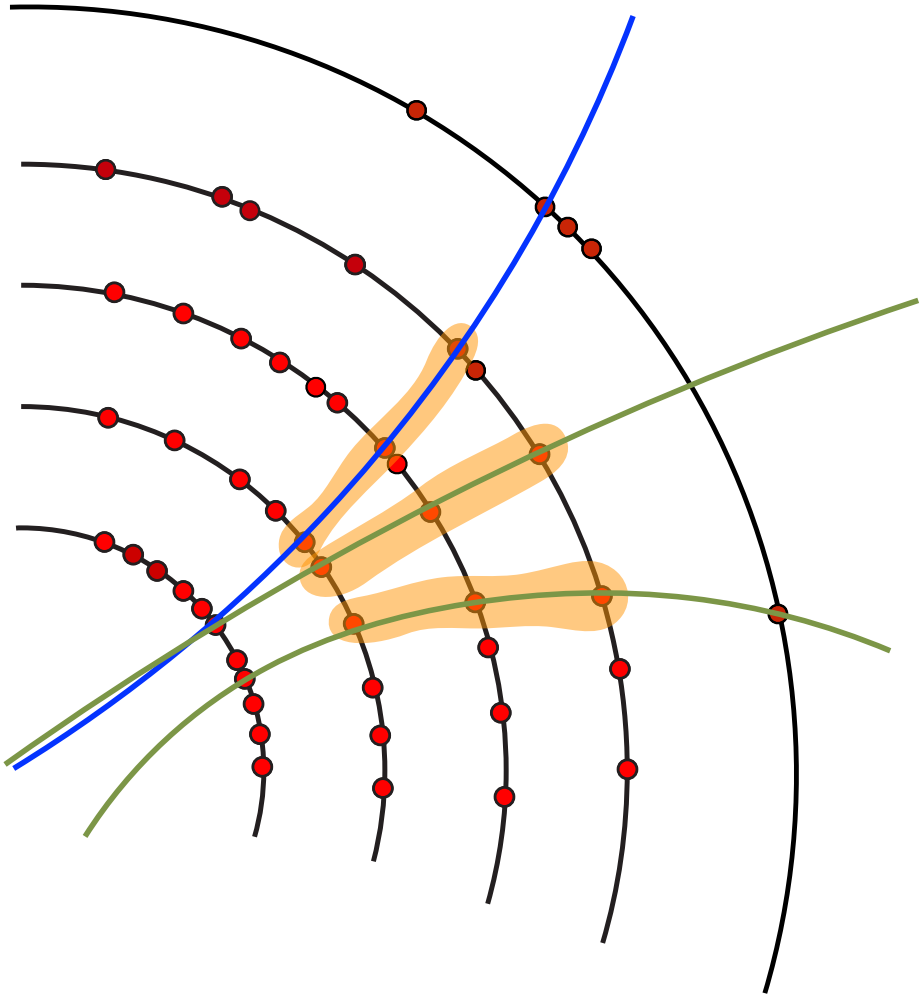




- A combinatorial Kalman filter is then used to build track candidates
- **Track Candidate** = Seed + compatible measurements
- More than one **track candidate** can be built from a seed if multiple paths are possible
- **Tracks** are fitted on the fly

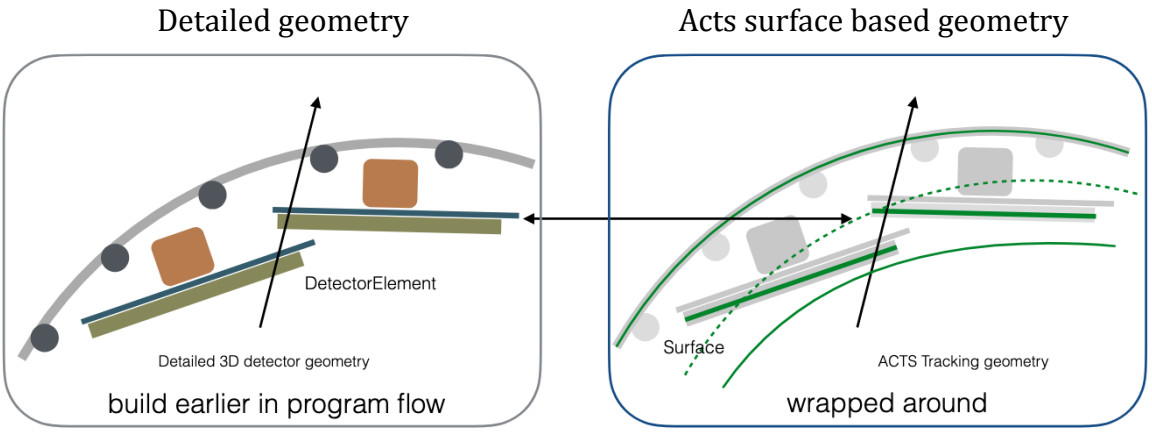
Track Reconstruction

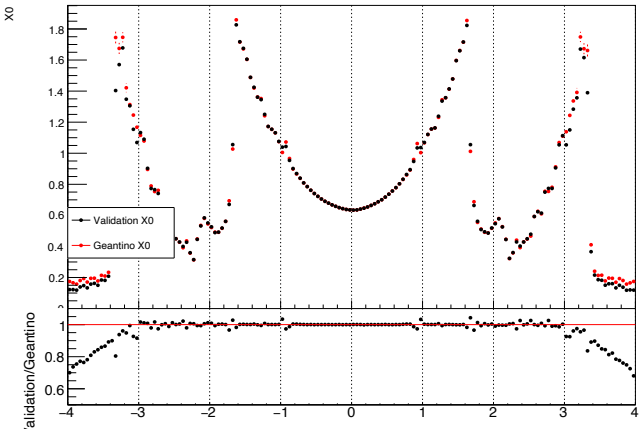




- A score is then associated with each track candidate
- Resolve ambiguity: reconstruct tracks in descending order of a track score

Geometry and detector





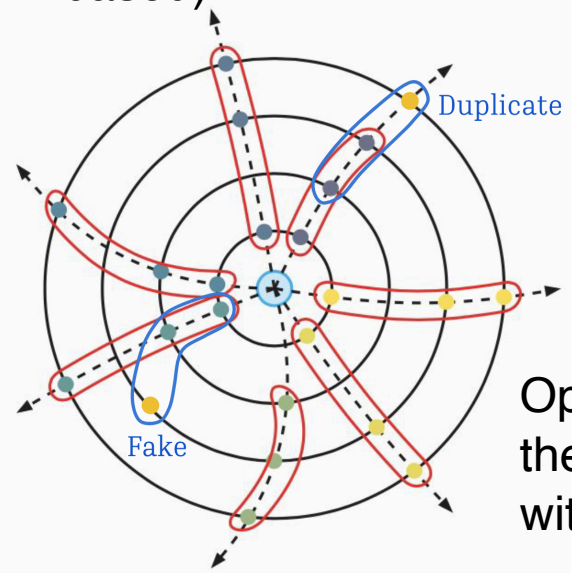
- Detector implementation is done through **DD4Hep**, **TGeo** or **GeoModel**
- Automatic **translation algorithm** to go from those to a simplified tracking optimised geometry
- Implement a **material interaction** module to account for particle/matter interaction during particle propagation
- And a **material mapping** module to extract a simplified material representation of the detector from the full geometry

Trajectory reconstruction

- Developed as a **modular toolkit**: different implementations of the different tracking steps available
- Easy to develop and add new ones adapted for specific detector/physics case

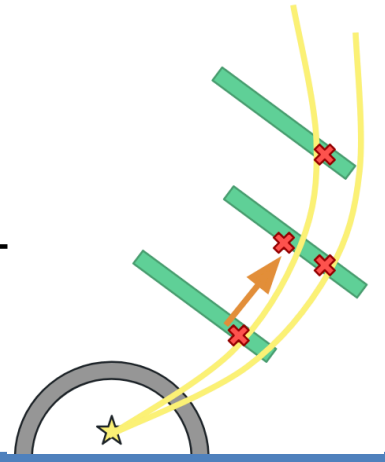
Two **seeding** algorithm:

- ATLAS inspired one
- Orthogonal seeding (KD-Tree based)



Multiple **track fitting/finding** algorithm:

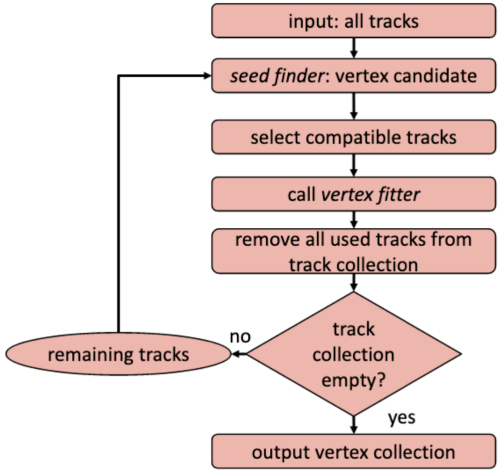
- Combinatorial Kalman filter (fitting+filtering)
- Chi2-based track finder
- Gaussian Sum Fitter (fitting for electrons)
- Graph Neural Network-based track finding
- …



Opportunity to develop new state-of-the-art algorithms and easily share with the wider HEP community

Vertexing

- Find back the interaction point from reconstructed tracks
- Two different implementations currently available: **Iterative Vertex Finder** (IVF) and **Adaptive Multi Vertex Finder** (AMVF)
- It is being used right now in ATLAS for vertexing

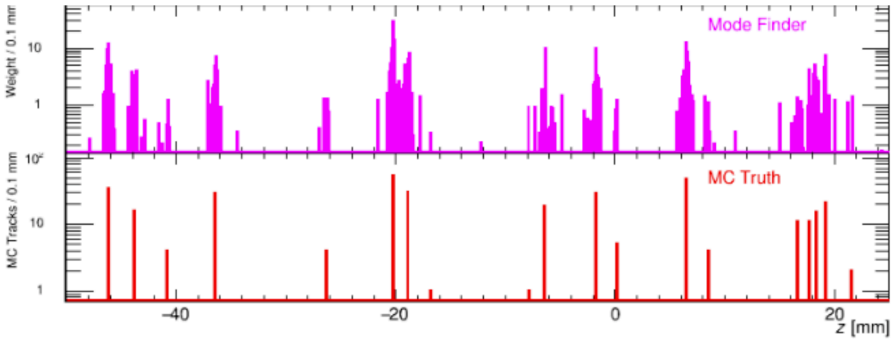


IVF :

- Reconstruct vertex seed based on an analysis of Z0 density
- Vertex are fitted based on the nearby track (with outliers being progressively removed)

AMVF:

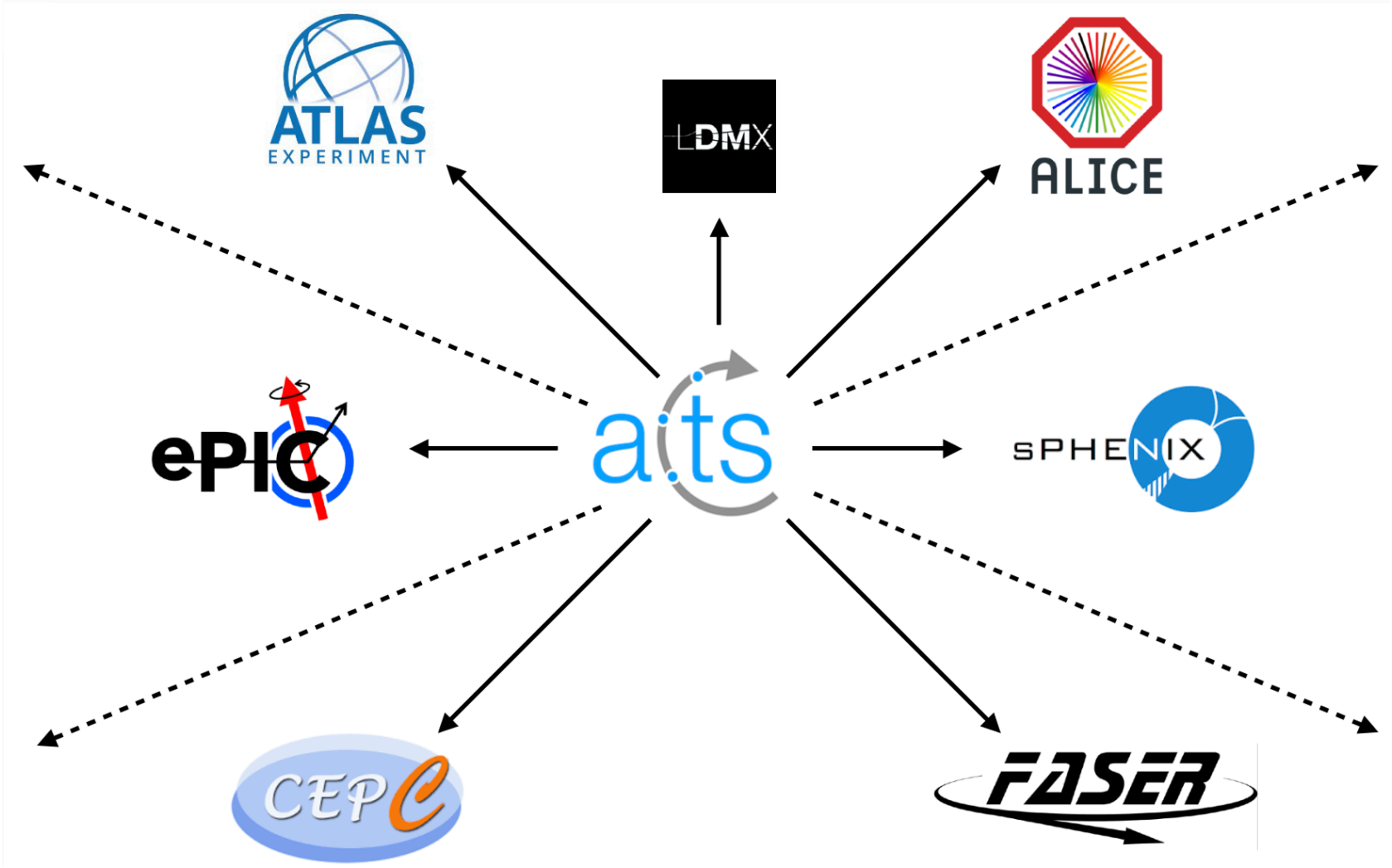
- At the start of the fit, tracks can belong to multiple vertex (but at the end, only 1 vertex per track)
- More performant in high PU environments



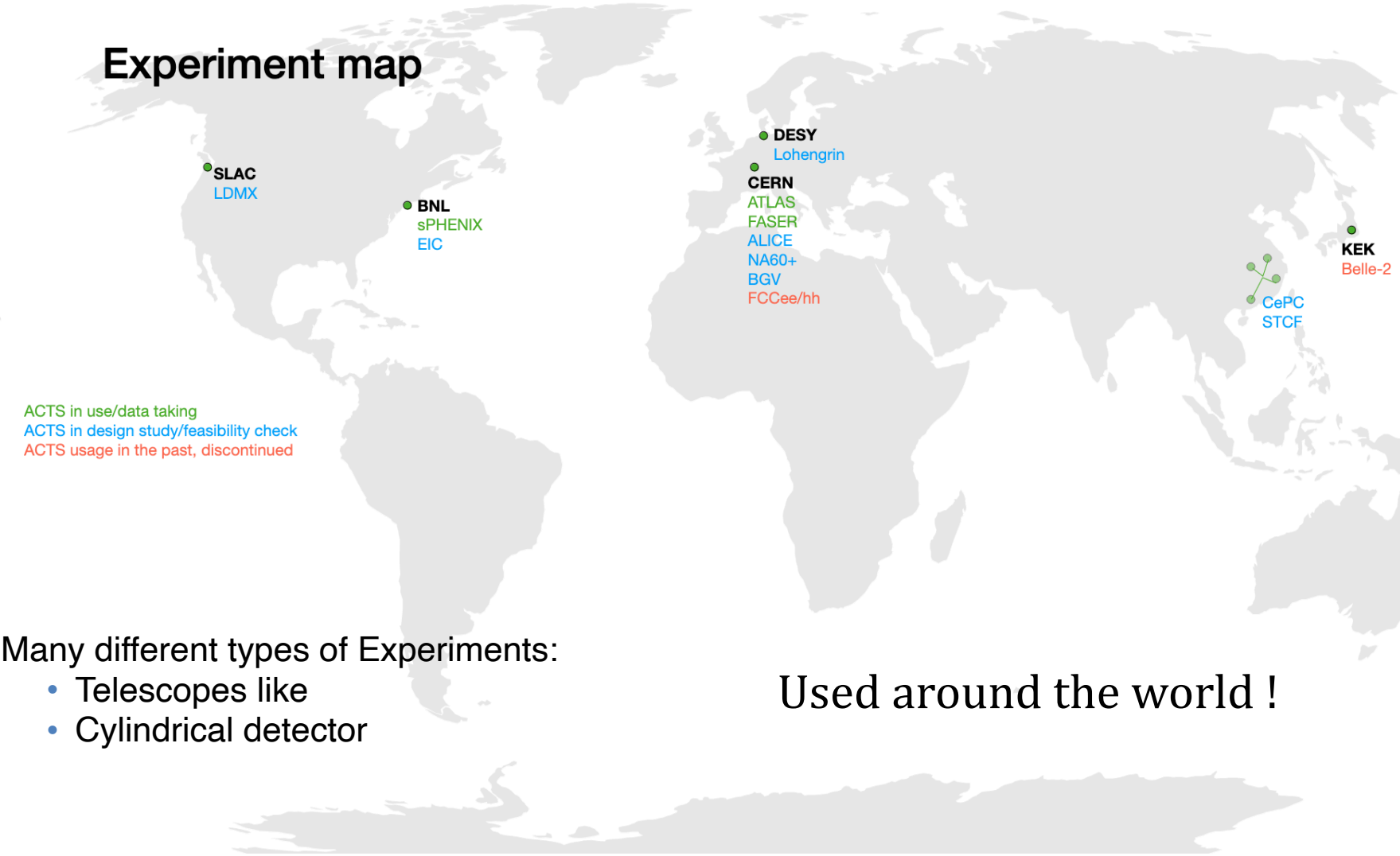
How to use ACTS ?

- ACTS: **not** a full event reconstruction software
- Provide a set of **building blocks** to be used to perform tracking tasks, needs to be integrated/interfaced with the experiment software suit (ie: JANA2 at the EIC)
- But it does provide a series of Examples that let you test those different functionalities, and that can be used as a basis when trying to interface the ACTS methods
- Those examples run by default on the OpenDataDetector, a virtual detector similar to the ATLAS ITk, but they can also be used with most detectors described with a DD4Hep description
- For information on how to build and use ACTS, you can check the git page and the documentation

Many different users



Many different users



Many different types of Experiments:

- Telescopes like
- Cylindrical detector

Used around the world !

Development and R&D

Core

acts-developers@cern.ch

CPU multi-threaded library of tracking reconstruction components

Presented at AI4EIC 2022:
Machine Learning in ACTS

R&D1

acts-parallelization@cern.ch

CPU/GPU "single source" demonstrator re-implementing the main Core chain

R&D2

acts-machinelearning@cern.ch

Machine learning and ML assisted modules for track reconstruction

- Different R&D Branch are part of the project
- Development of a GPU based tracking chain
- Test of Machine learning based tracking algorithm with easily available test bench

Large involvement of IJCLab

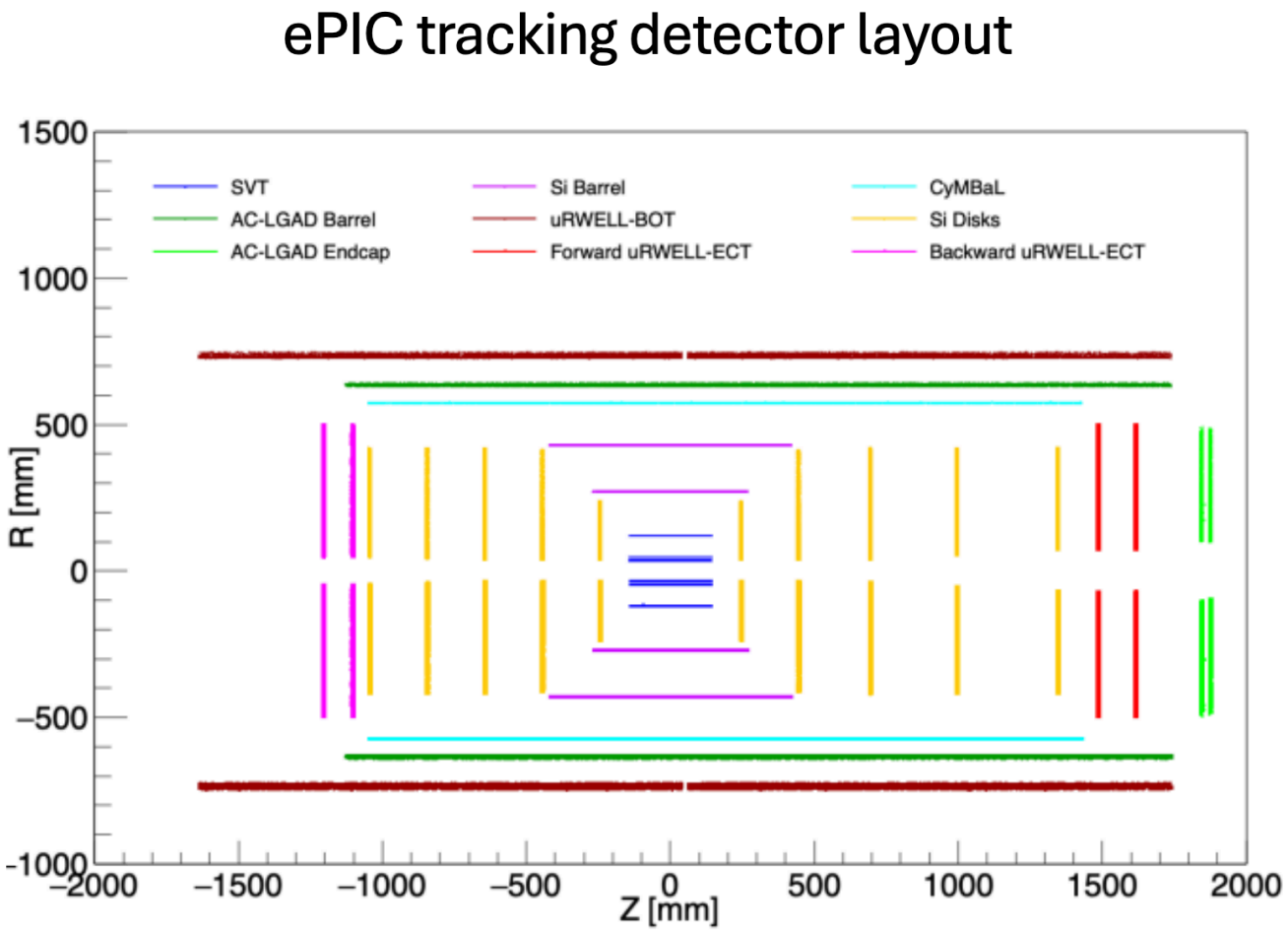
The ACTS Team

- The Core ACTS developer team are CERN based ATLAS tracking expert
- Local development at IJCLab:
  - Material Interaction module
  - ML for tracking
  - Code optimisation
- Open Sourced project:
  - Contribution from the different ACTS users
  - Large involvement of ATLAS, Faser and sPhenix

ACTS for the ePIC detector

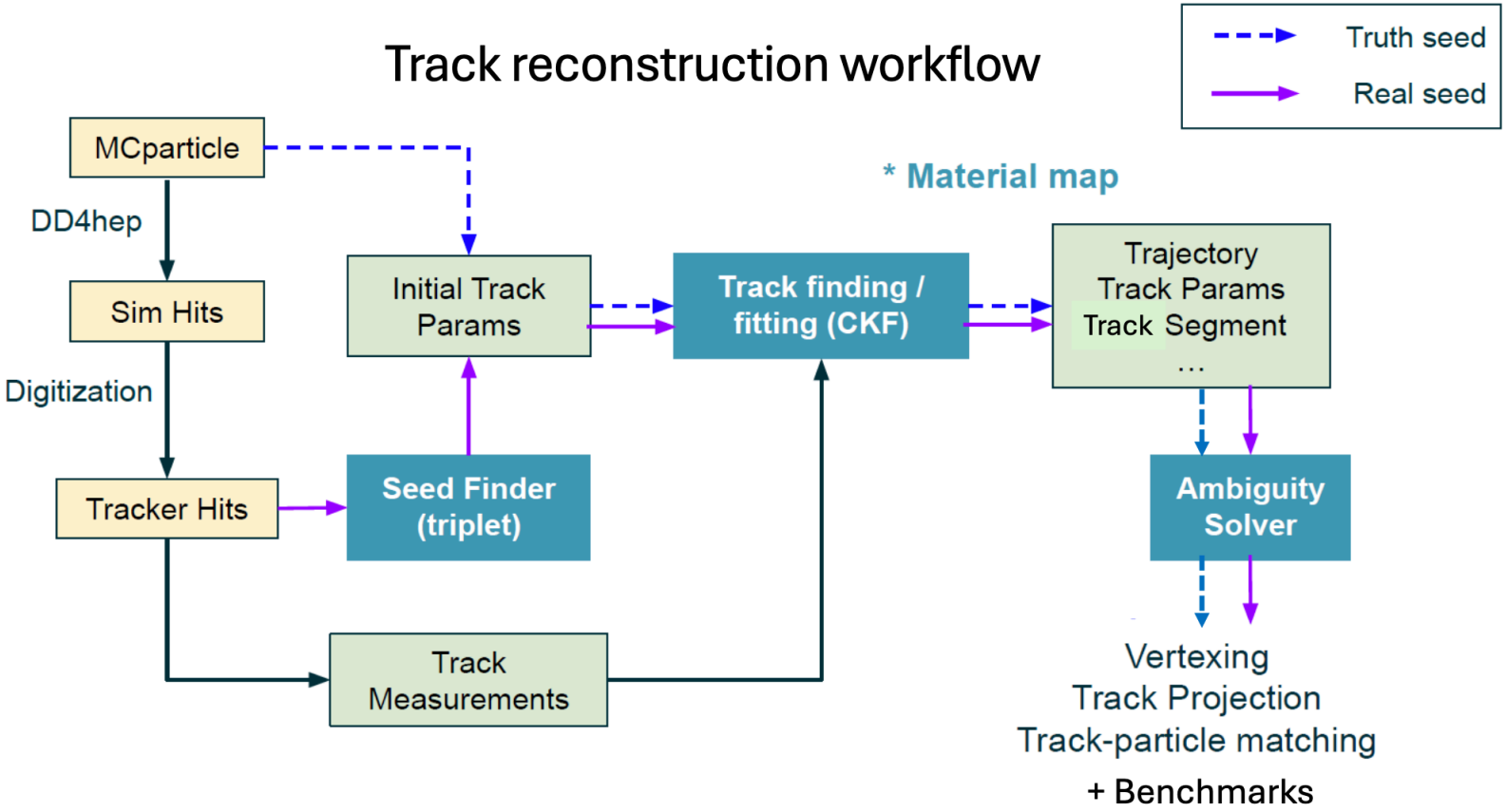
A preliminary **ePIC geometry** has been implemented by colleagues from Berkley

Implemented via **DD4Hep**: « Easy » to implement new detectors and test their impact on the performance



Slides borrowed from **Barak Schmookler**

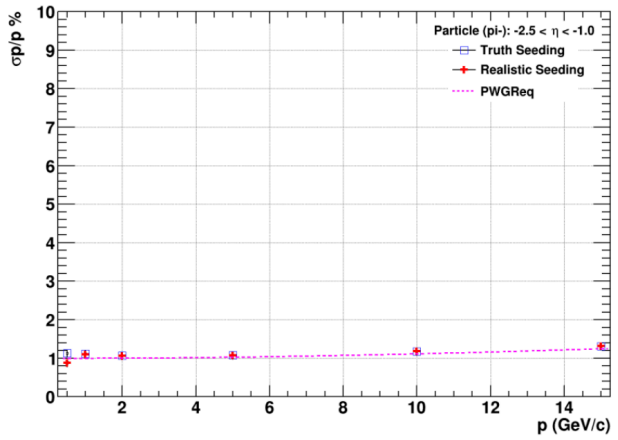
ACTS for the ePIC detector

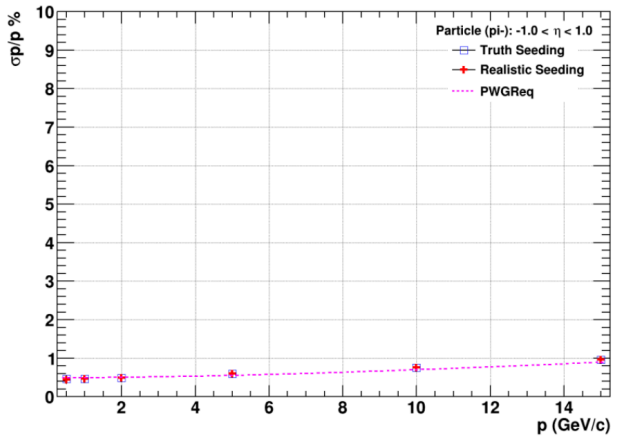


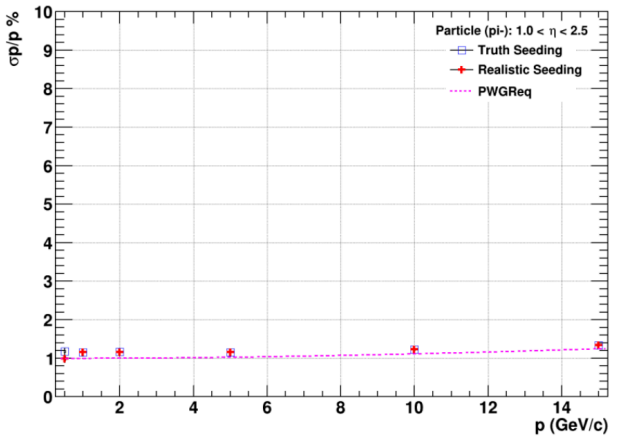
ACTS **tracking chain** implemented for the ePIC detector, used for the evaluation of tracking performances of the detector

First Evaluation of the performance

- First evaluation of the detector performances with a realistic tracking chain is being performed
- Use a realistic tracking chain
- Run in a CI: constantly updated with detector changes





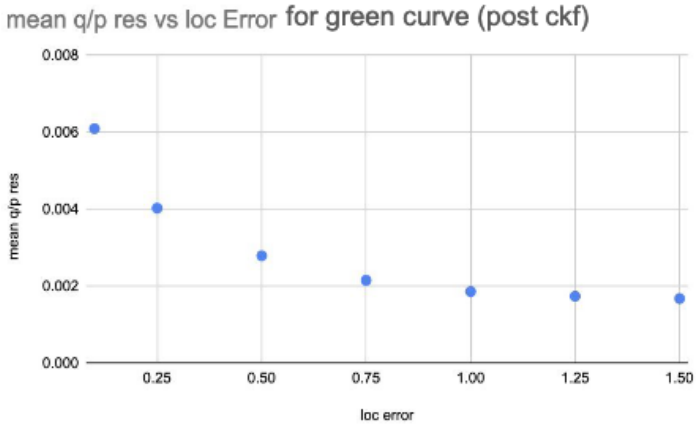


Work by Shyam Kumar

Some ongoing work

$$C = \begin{bmatrix} \sigma^2(l_0) & \mathrm{cov}(l_0, l_1) & \mathrm{cov}(l_0, \phi) & \mathrm{cov}(l_0, \theta) & \mathrm{cov}(l_0, q/p) \\ . & \sigma^2(l_1) & \mathrm{cov}(l_1, \phi) & \mathrm{cov}(l_1, \theta) & \mathrm{cov}(l_1, q/p) \\ . & . & \sigma^2(\phi) & \mathrm{cov}(\phi, \theta) & \mathrm{cov}(\phi, q/p) \\ . & . & . & \sigma^2(\theta) & \mathrm{cov}(\theta, q/p) \\ . & . & . & . & \sigma^2(q/p) \end{bmatrix}$$

- Performance evaluation is so far only using single particles; full physics events are expected soon
- Some issues with the CKF: might need to be **tuned for low momentum** particles, studies needed on **chi2 cuts** and the effect of the **input covariance matrix**
- Issue with the outmost tracker (MPGD): hits not properly accounted for in the tracking



What next ?

- Right now, ACTS has been fully implemented with the ePIC detector
- Some tuning of the algorithms to use them on full simulation events with noise and background
- Implementation of timing information in tracking (information available but not currently exploited)
- New algorithms might need to be developed to exploit the Physics/ Detector specificity fully
- New ML techniques are being developed for tracking; they might prove interesting for EIC physics

Conclusion

- ACTS has been fully implemented with the ePIC detector
- We have been awarded a **France-Berkeley Funding** for next year, which will probably be used to organise a workshop on tracking for the EIC and to facilitate collaborations
- If you are interested in joining this effort, **don't hesitate to get in touch with me**
- A meeting on tracking and vertexing is held semi-weekly: https://indico.bnl.gov/category/542/

BACKUP