BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

Tracking with ACTS

An example from ePIC

Shujie Li
Berkeley Lab

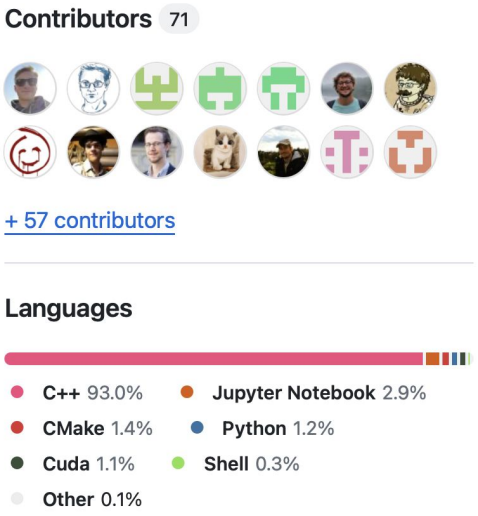
SoLID Opportunities and Challenges of Nuclear Physics at the Luminosity Frontier
Jun 24, 2024 @ ANL

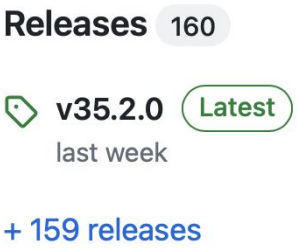
arXiv:1910.03128

A Common Tracking Software

- A C++ library that contains components for assembling a (charged) particle track reconstruction suite for High Energy Physics and Nuclear Physics.
- Initiated in 2016 for ATLAS, now widely used in HEP and NP including sPHENIX, ALICE3, FASER, ePIC …

https://github.com/acts-project/acts



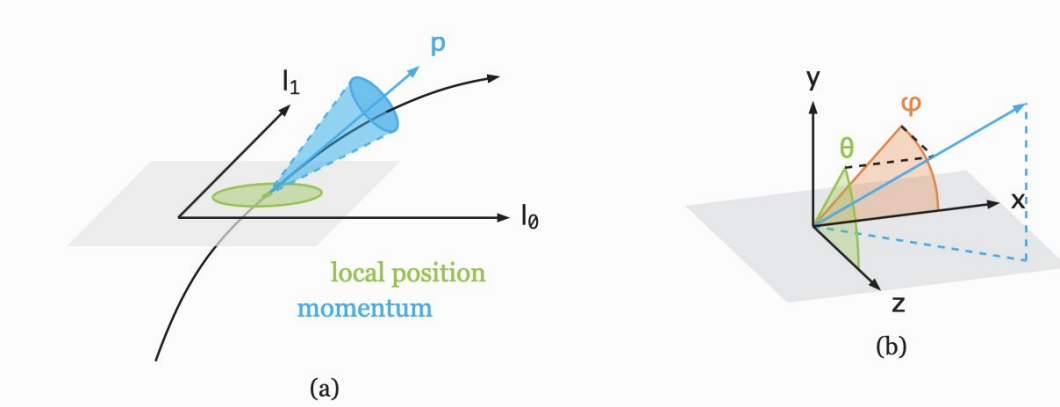


ACTS Track Parameter

https://acts.readthedocs.io/en/latest/tracking.html

Track parameter:

$$\vec{x} = (l_0, l_1, \phi, \theta, q/p, t)^T$$



Covariance Matrix:

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

Track Propagator

Stepper:

- update the track parameter according to the equation of motion through numerical integration
- Default: 4th order Runge-Kutta with adaptive step size. Magnetic field and material effects included
- Pathlength = accumulated step size

Navigator:

Sort out the order of volumes, layers, and surfaces, keeps track of the current position in the geometry and adjusts the step size to reach the target surface

Propagating Through Material

Initial to final step: evolve covariance in time

$$C^f = J \cdot C^i \cdot J^T,$$

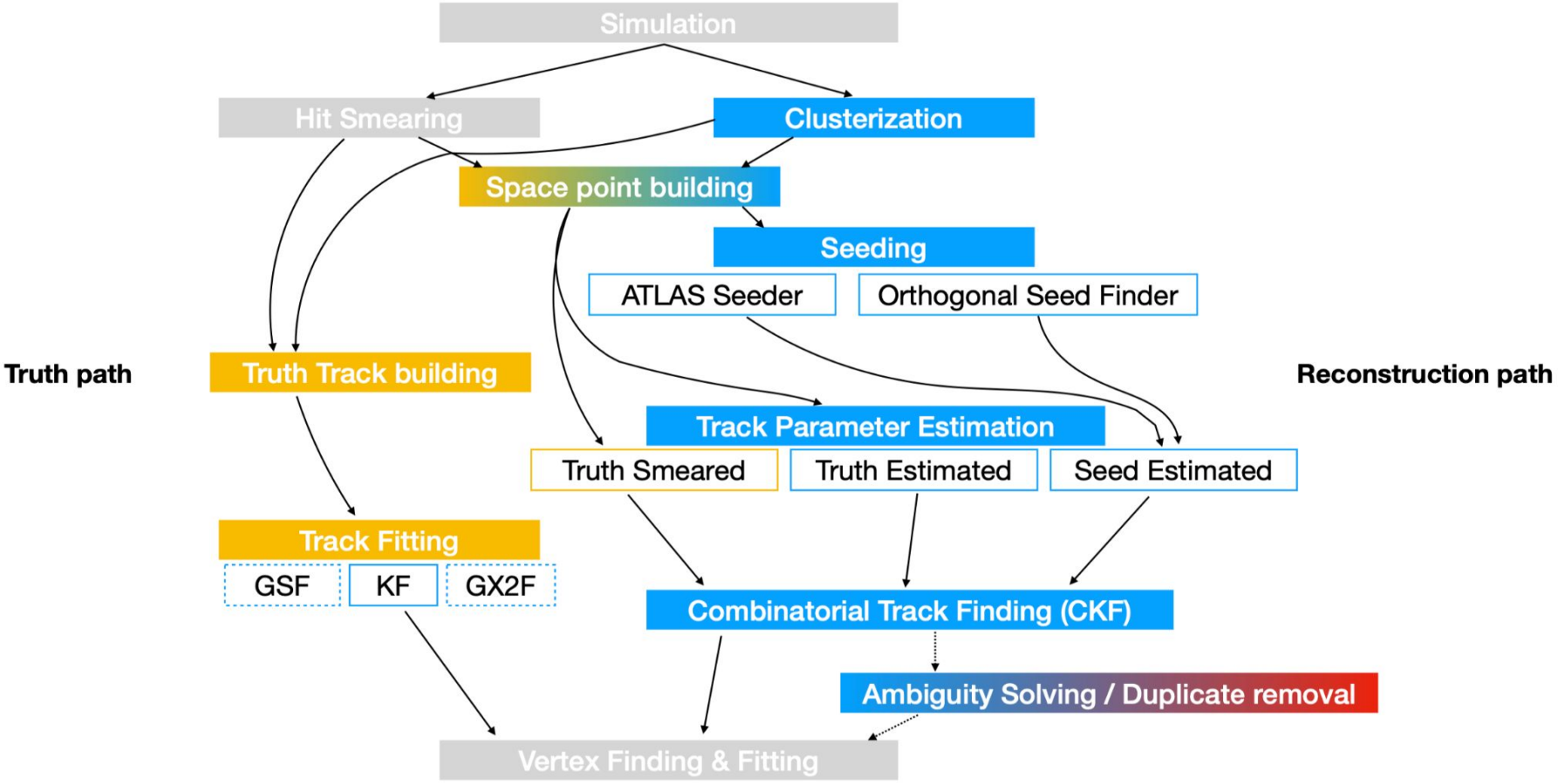
$$J = \begin{bmatrix} \frac{\partial l_0^f}{\partial l_0^i} & \cdots & \frac{\partial l_0^f}{\partial (q/p)^i} \\ \vdots & \ddots & \vdots \\ \frac{\partial (q/p)^f}{\partial l_0^i} & \cdots & \frac{\partial (q/p)^f}{\partial (q/p)^i} \end{bmatrix},$$

Material effects:

- Deflection and offset → averaged to 0, increased uncertainties
- Energy loss → reduced trajectory energy
- Hadronic process → disintegration etc.

ACTS: Core Functionality

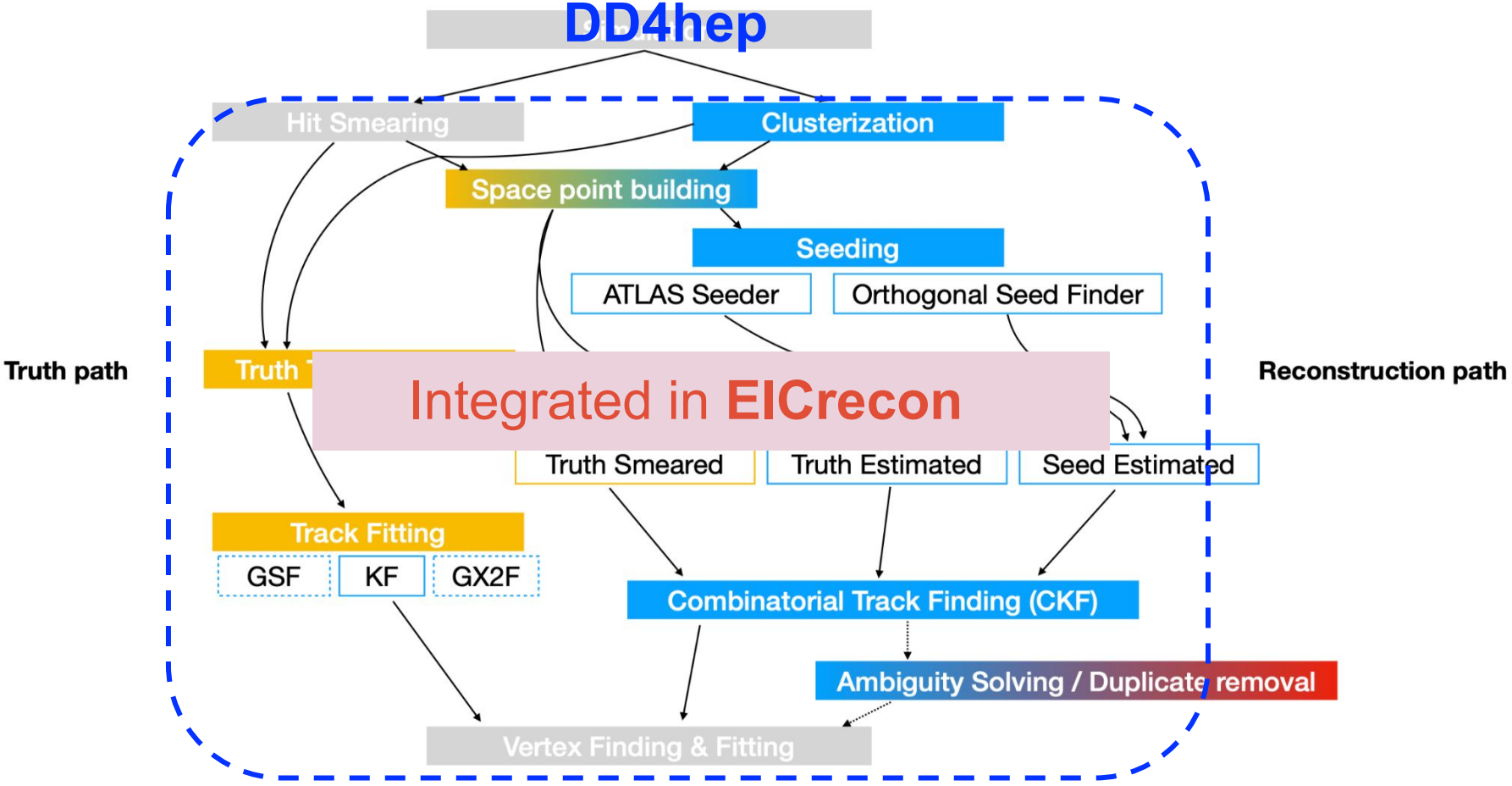
https://acts.readthedocs.io/en/latest/index.html



courtesy of A. Salzburger

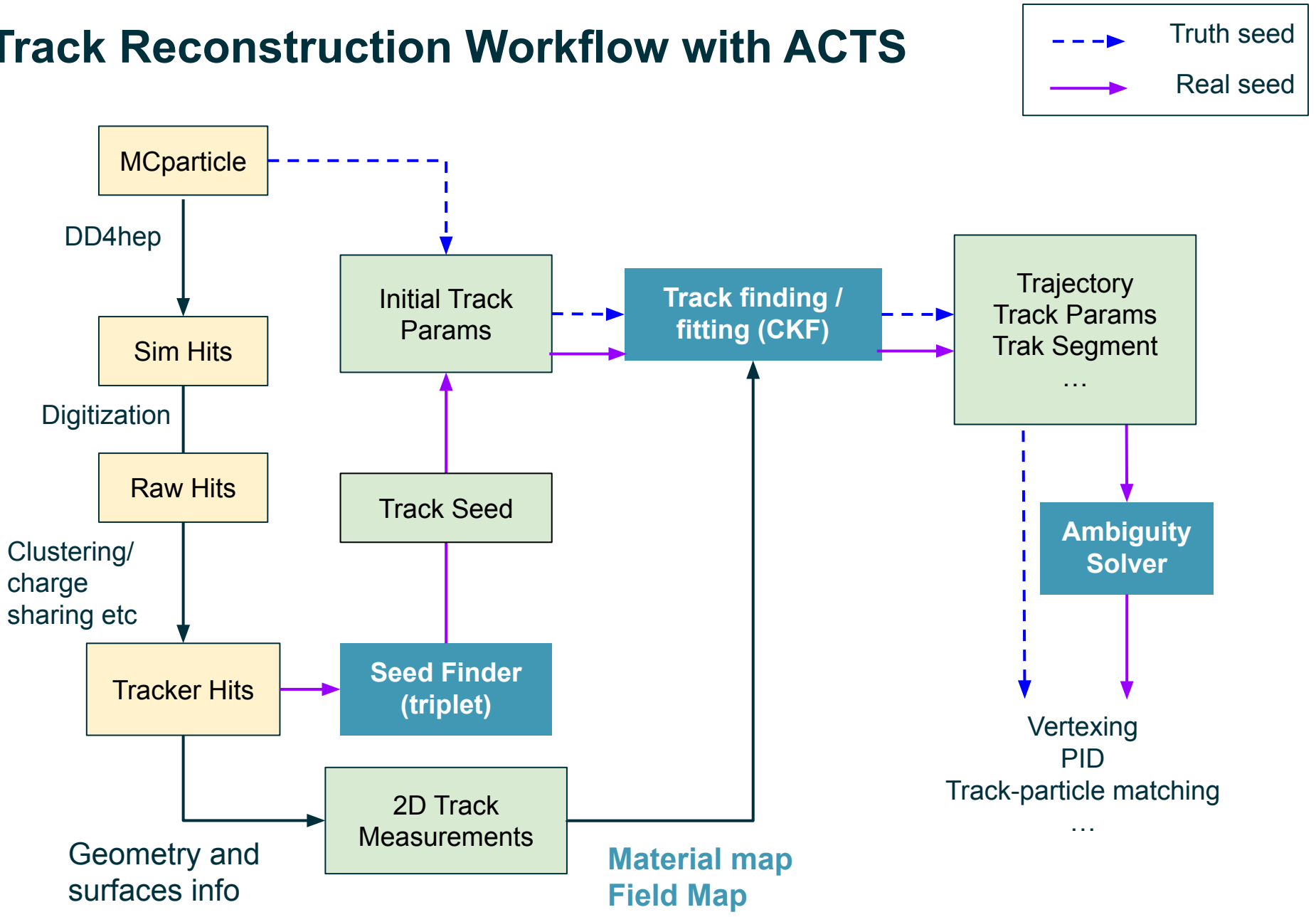
ACTS for ePIC

https://github.com/eic/EICrecon



courtesy of A. Salzburger

Track Reconstruction Workflow with ACTS



Step 1 :Detector Description

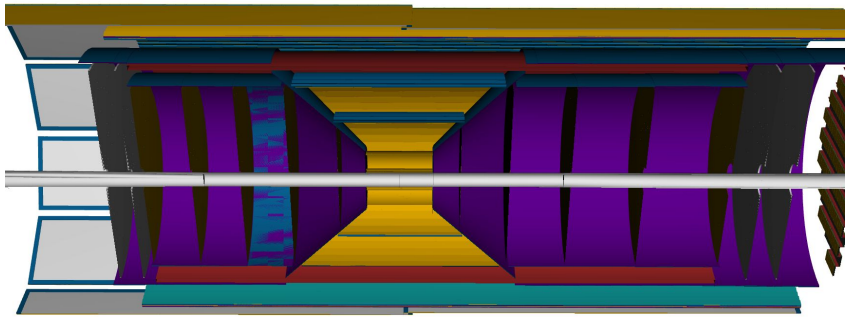
- **Geometry:**
  - Description in DD4hep xml file

main epic / compact / tracking / silicon_disks.xml

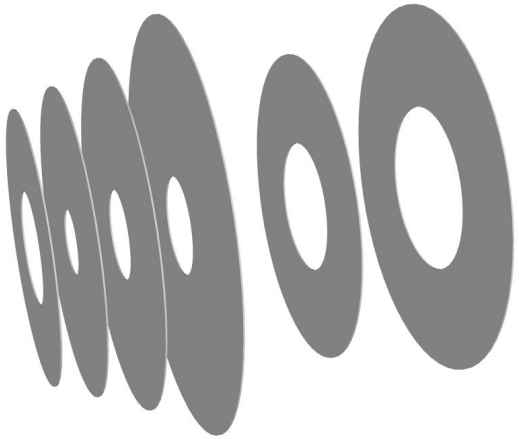
Code Blame 444 lines (429 loc) · 25.7 KB

```
60        <detector
61          id="TrackerEndcapP_0_ID"
62          name="InnerTrackerEndcapP"
63          type="epic_TrapEndcapTracker"
64          readout="TrackerEndcapHits"
65          vis="TrackerVis"
66          reflect="false">
67          <type_flags type="DetType_TRACKER + DetType_ENDCAP"/>
68          <module name="Module1" vis="TrackerModuleVis">
69            <trd x1="InnerTrackerEndcapPMod1_x1/2" x2="InnerTrackerEnd
70            <module_component thickness="SiTrackerEndcapCF_thickness"
71            <module_component thickness="SiTrackerEndcapAl_thickness"
72            <module_component thickness="SiTrackerSensor_thickness" ma
73          </module>
74          <layer id="1">
75            <envelope vis="TrackerLayerVis"
```

ePIC Craterlake Tracker in DD4hep



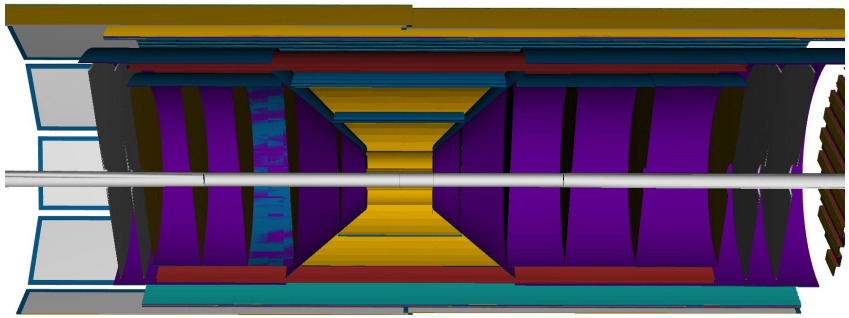
SoLID tracker geometry from Chao Peng



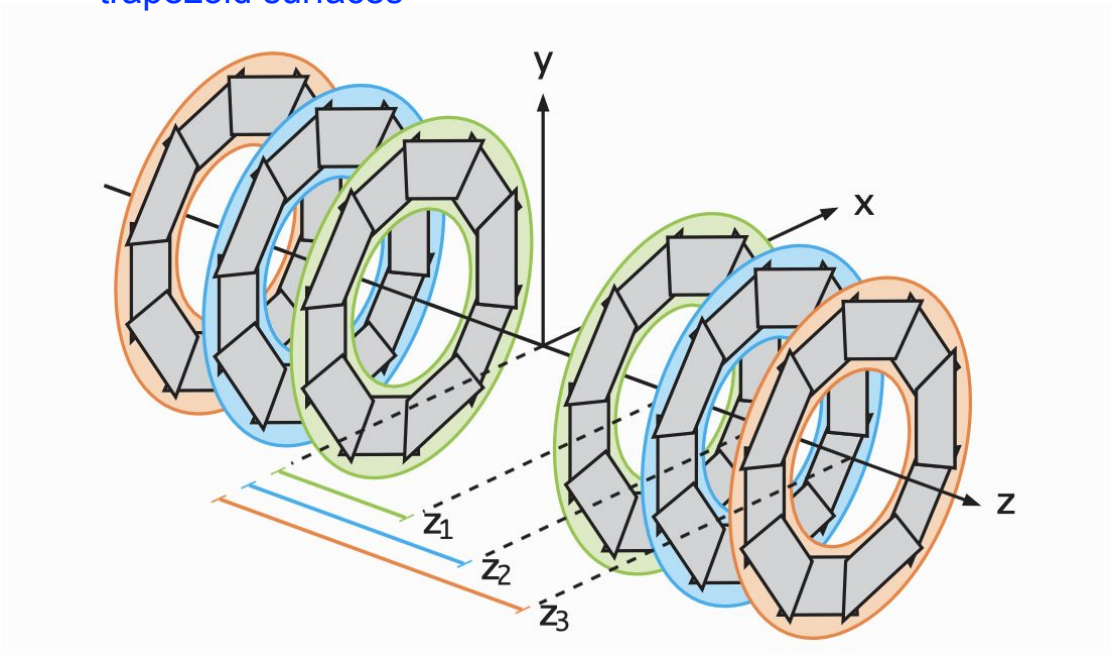
Step 1 :Detector Description

- **Geometry:**
  - Description in DD4hep xml file

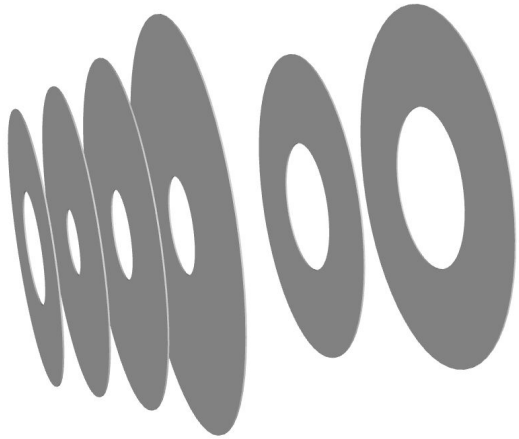
ePIC Craterlake Tracker in DD4hep



ACTS Example: disk description with trapezoid surfaces



SoLID tracker geometry from Chao Peng



Step 1 :Detector Description

- **Geometry:**
  - Description in DD4hep xml file
  - Detailed module layout and ACTS-compatible sensitive surfaces in plugins

main ▾ epic / src / TrapEndcapTracker_geo.cpp

Code | Blame | 314 lines (280 loc) · 13.1 KB

```
33    static Ref_t create_detector(Detector& description, xml_h e, SensitiveDetector sens) {
179           // -------- create a measurement plane for the tracking surface attched to the sensitive volume -----
180           Vector3D u(0., 0., -1.);
181           Vector3D v(-1., 0., 0.);
182           Vector3D n(0., 1., 0.);
183           // Vector3D o( 0. , 0. , 0. ) ;
184
185           // compute the inner and outer thicknesses that need to be assigned to the tracking surface
186           // depending on wether the support is above or below the sensor
187           double inner_thickness = module_thicknesses[m_nam][0];
188           double outer_thickness = module_thicknesses[m_nam][1];
189
190           SurfaceType type(SurfaceType::Sensitive);
191
192           // if( isStripDetector )
193           //  type.setProperty( SurfaceType::Measurement1D , true ) ;
194
195           VolPlane surf(c_vol, type, inner_thickness, outer_thickness, u, v, n); //,o ) ;
196           volplane_surfaces[m_nam].push_back(surf);
197
```

Step 1 :Detector Description

- **Geometry:**
  - Description in DD4hep xml file
  - Detailed module layout and ACTS-compatible sensitive surfaces in plugins
  - Assemble detectors inside-out (onion-like hierarchy with no overlaps)

main ▾ epic / compact / tracking / **definitions_craterlake.xml**

Code | Blame | 204 lines (185 loc) · 11 KB

```
119     <comment> See compact/definitions.xml for reserved detector id
120       ACTS detector volume needs to be built inside out in terms of R. </comment>
121     <detectors>
122       <detector id="VertexSubAssembly_0_ID"
123         name="VertexBarrelSubAssembly"
124         type="DD4hep_SubdetectorAssembly"
125         vis="TrackerSubAssemblyVis">
126         <composite name="VertexBarrel" />
127       </detector>
128       <detector id="TrackerSubAssembly_0_ID"
129         name="InnerSiTrackerSubAssembly"
130         type="DD4hep_SubdetectorAssembly"
131         vis="TrackerSubAssemblyVis">
132         <composite name="InnerTrackerEndcapN"/>
133         <composite name="InnerTrackerEndcapP"/>
134       </detector>
```

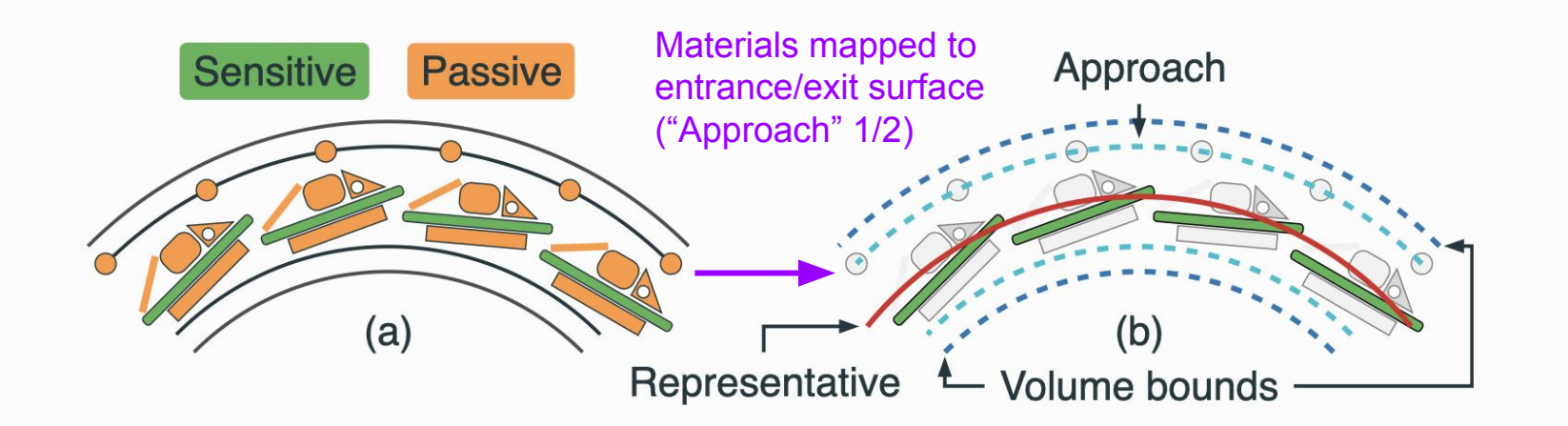
Step 1 :Detector Description

- **Geometry:**
  - Description in DD4hep xml file
  - Detailed module layout and ACTS-compatible sensitive surfaces in plugins
  - Assemble detectors inside-out (onion-like hierarchy with no overlaps)

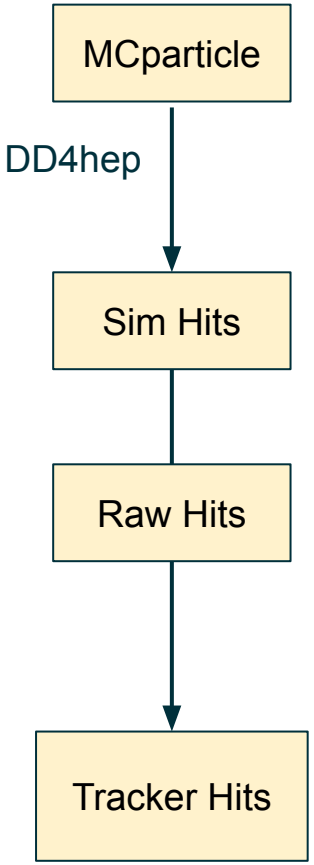
    Dedicated development for telescope detectors

Step 1 :Detector Description

- **Geometry:**
  - Description in DD4hep xml file
  - Detailed module layout and ACTS-compatible sensitive surfaces in plugins
  - Assemble detectors inside-out (onion-like hierarchy with no overlaps)
  - Material mapping



Step 2: Space-time points



Generated physics events

Digitization:
energy threshold, detector resolution, noise, bad channels etc

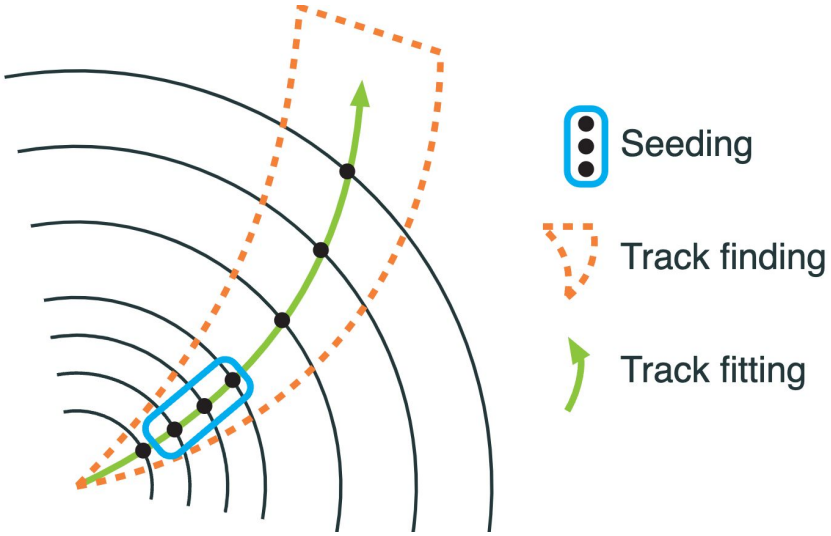
Charge-sharing and clustering, deadtime
?

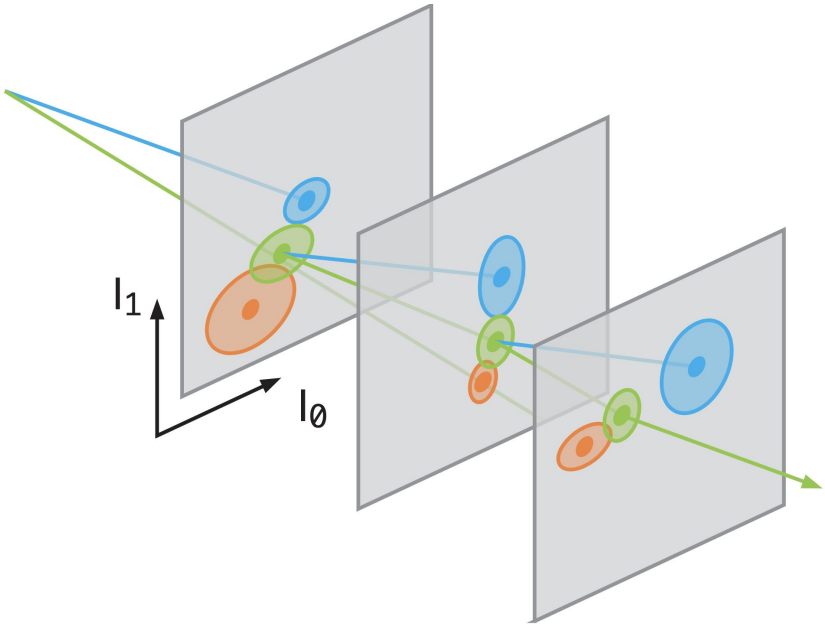
Step 3: Track Finding/Fitting

- **Combinatorial Kalman Filter (CKF)**
  - Combined track finding and fitting
  - Realistic seeder to provide initial guess

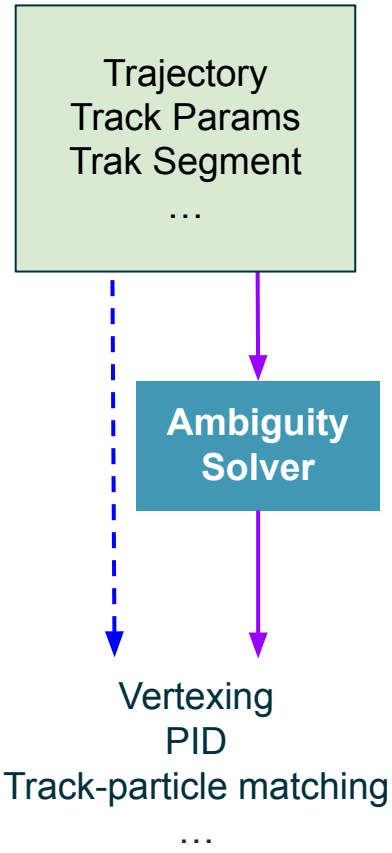
    * Algorithms exist, need tune configuration for the specific detector

Also available, standard KF, GNN etc…





Step 4: Reconstruction with Tracking Info



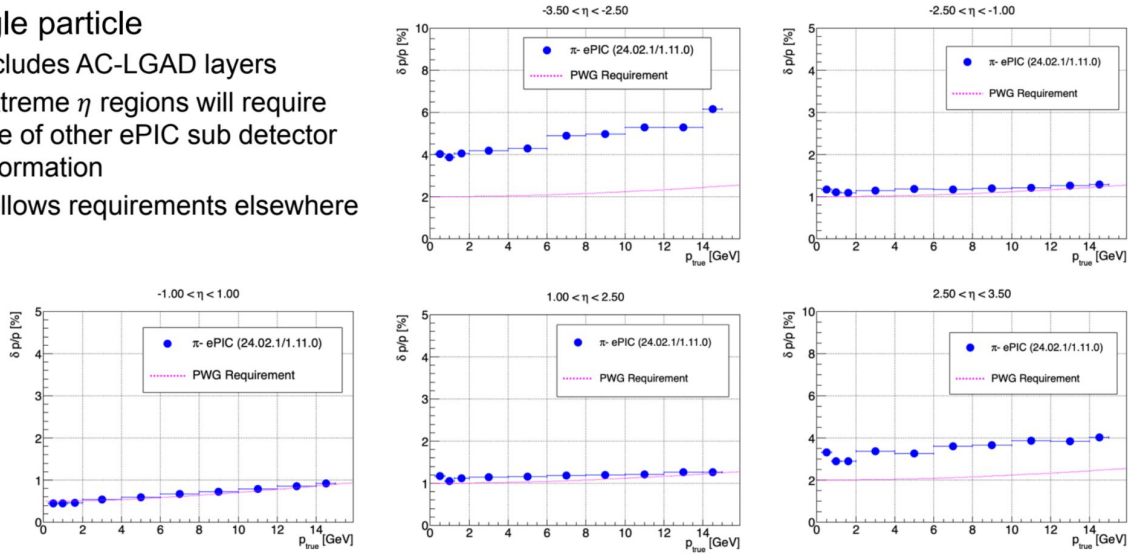
- Hit residual
- Occupancy, efficiency, purity
- PID integration
- Use of timing info

Performance study:

- Single track
- Physics events
- +background
- timeframe/DAQ

Example: ePIC single track performance

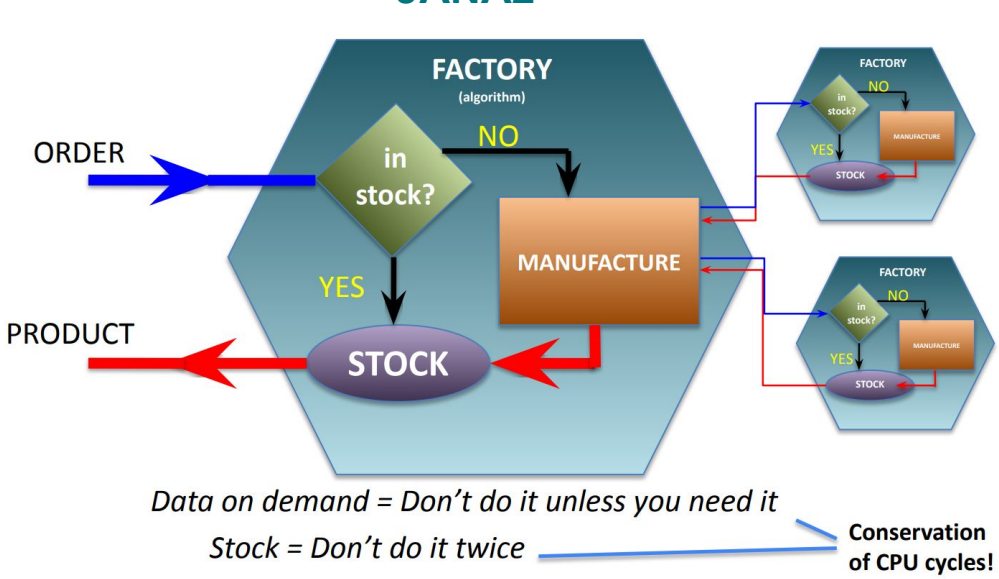
- Single particle
  - Includes AC-LGAD layers
  - Extreme η regions will require use of other ePIC sub detector information
  - Follows requirements elsewhere



Other Considerations

EICrecon:

- JANA2 event processing
- podio-based data structure (edm4eic)
- eic-shell environment (spack packages)



main · EDM4eic / edm4eic.yaml

Code · Blame · 560 lines (516 loc) · 29.2 KB

```
383
384     edm4eic::TrackSeed:
385       Description: "Seed info from the realistic seed finder"
386       Author: "S. Li, B. Schmookler, J. Osborn"
387       Members:
388         - edm4hep::Vector3f         perigee   // Vector for the perigee (line surface)
389       OneToManyRelations:
390         - edm4eic::TrackerHit       hits      // Tracker hits triplet for seeding
391       OneToOneRelations:
392         - edm4eic::TrackParameters  params    // Initial track parameters
393
394     edm4eic::Trajectory:
395       Description: "Raw trajectory from the tracking algorithm. What is called hit here is 2d measurement indeed."
396       Author: "S. Joosten, S. Li"
397       Members:
398         - uint32_t          type              // 0 (does not have good track fit), 1 (has good track fit)
399         - uint32_t          nStates           // Number of tracking steps
400         - uint32_t          nMeasurements     // Number of hits used
401         - uint32_t          nOutliers         // Number of hits not considered
402         - uint32_t          nHoles            // Number of missing hits
403         - uint32_t          nSharedHits       // Number of shared hits with other trajectories
404       VectorMembers:
405         - float             measurementChi2   // Chi2 for each of the measurements
406         - float             outlierChi2       // Chi2 for each of the outliers
```

Thanks!