First developments for ldmx-sw track reconstruction

PF

10/22/2020

- not as bad as Tim's talk -

U.S. DEPARTMENT OF ENERGY | Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

Outline

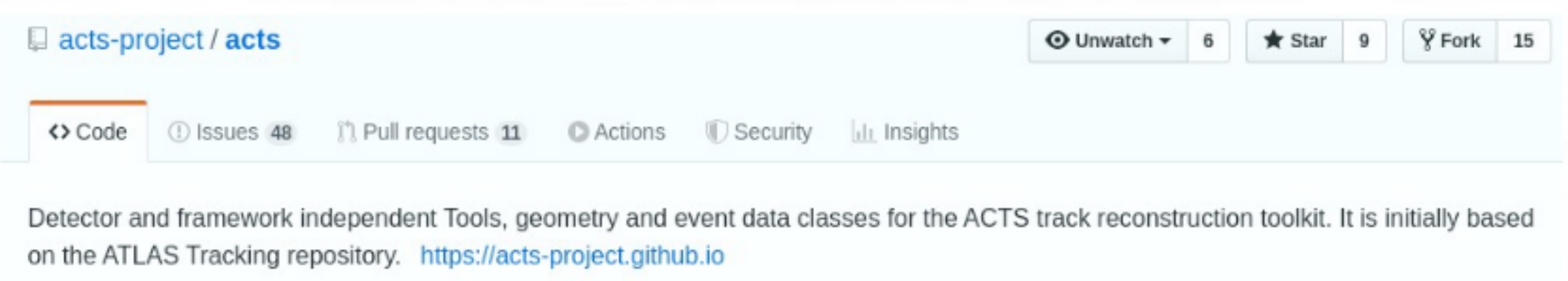
- First look at possible tracking implementation for LDMX
- The idea is to use as much as possible advanced tools provided by the HEP community, without reinventing the wheel

- **Today:**
  - General intro to ACTS tracking framework
  - First applications to LDMX:
    - Seed finding from 3D simulated space points
    - Track propagation and material mapping
  - Next steps and Summary

The ACTS Tracking Software

- The ACTS tracking software is an experiment-independent* toolkit for future detectors
- Preserve and improve ATLAS tracking code
  - Modern C++ 17 concepts
  - Thread safe design
  - Avoid dynamic memory allocation
  - Unit testing
  - Minimal dependencies for integration

- Open source platform for new tracking algorithms and hardware architectures R&D
- Hosted on github, active community. https://github.com/acts-project/acts



Pros/Cons of ACTS For LDMX

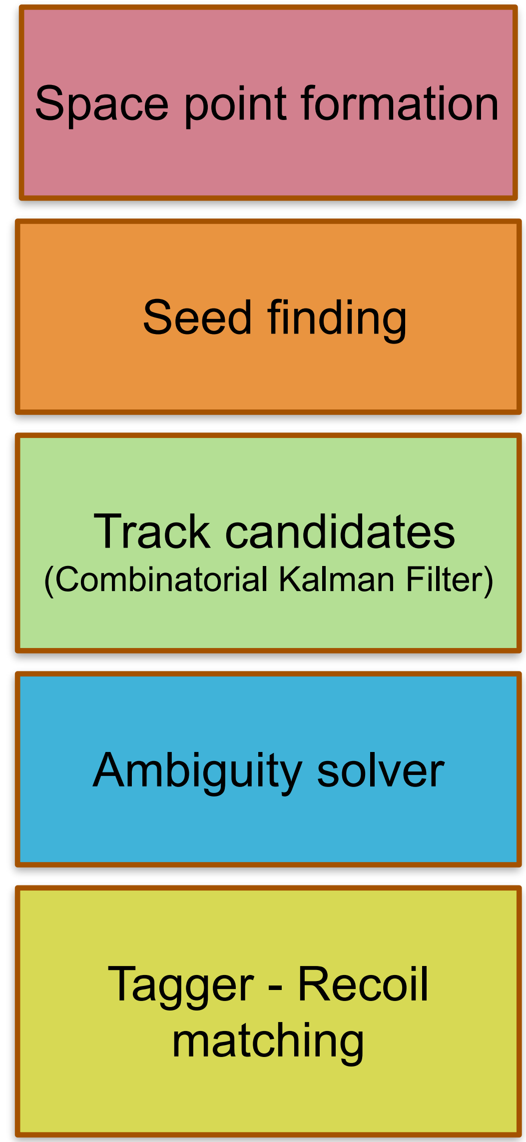
SLAC

- While implementing some track reconstruction basic steps using ACTS, I can say that there are some pros/cons in using an external library, where some effort went into cross-experiment portability
- **Pros**:
  - Active community of experts which can support development and future issues with tracking
  - Other experiments will contribute to test, validate, improve, support the library in the future.
  - Modern C++ package with constants updates
    - Multithreading, ML support, GPU support..
    - Straightforward to integrate in LDMX-sw
  - Detector alignment support will be developed centrally, lot of infrastructure work will be done as a joint effort.
  - Basic Tools are already in place: we can start doing things relatively fast with a reasonable initial push
  - DD4Hep/TGeo Geometry, B-Field non-uniformity, 4D tracking (with time) supported.
- **Cons**
  - Very collider oriented: some issues will remain in terms of coordinate transformations.

Basic tracking procedure

SLAC

- Outlined here are the steps of track reconstruction for LDMX.
- Necessary ingredients are
  - **Write the tracking geometry** as a support for track propagation, track finding and fitting
  - Form **external 3D space points** from raw data clusters.
  - **Provide the 3D space points to the seed finder tools** in order to find track seeds.
  - Provide the hit collection, the initial seed track parameters and the tracking geometry to the **combinatorial kalman filter** for track finding and fitting.
  - Alignment support is initiated (derivatives seem to be there) but not completed (linear system solving not in place).



The LDMX Tracking systems

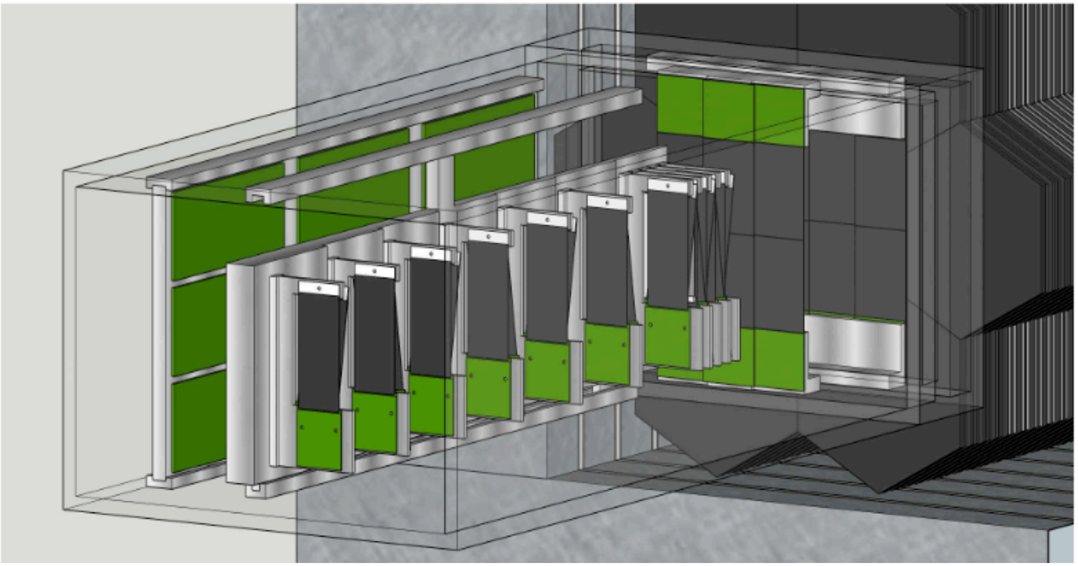


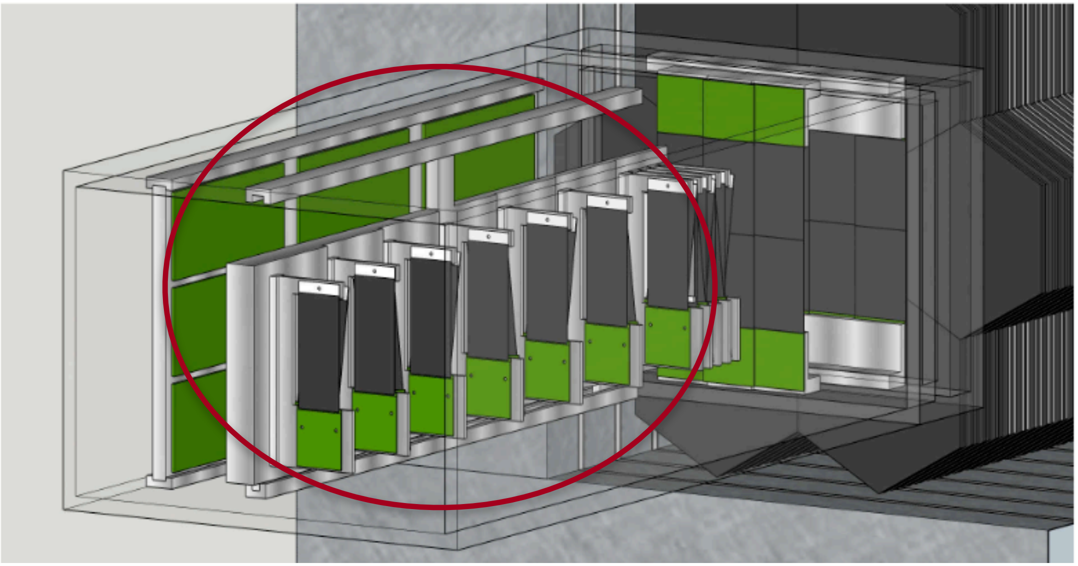
TABLE I: The layout and resolution of the tagging tracker.

Layer	1	2	3	4	5	6	7
z-position, relative to target (mm)	-607.5	-507.5	-407.5	-307.5	-207.5	-107.5	-7.5
Stereo Angle (mrad)	-100	100	-100	100	-100	100	-100
Bend plane (horizontal) resolution (μm)	$\sim$6	$\sim$6	$\sim$6	$\sim$6	$\sim$6	$\sim$6	$\sim$6
Non-bend (vertical) resolution (μm)	$\sim$60	$\sim$60	$\sim$60	$\sim$60	$\sim$60	$\sim$60	$\sim$60

TABLE II: The layout and resolution of the recoil tracker.

Layer	1	2	3	4	5	6
z-position, relative to target (mm)	+7.5	+22.5	+37.5	+52.5	+90	+180
Stereo Angle (mrad)	100	-100	100	-100	-	-
Bend plane (horizontal) resolution (μm)	$\approx$6	$\approx$6	$\approx$6	$\approx$6	$\approx$6	$\approx$6
Non-bend (vertical) resolution (μm)	$\approx$60	$\approx$60	$\approx$60	$\approx$60	-	-

The LDMX Tracking systems



Will focus on the tagger tracker today.

TABLE I: The layout and resolution of the tagging tracker.

Layer	1	2	3	4	5	6	7
z-position, relative to target (mm)	-607.5	-507.5	-407.5	-307.5	-207.5	-107.5	-7.5
Stereo Angle (mrad)	-100	100	-100	100	-100	100	-100
Bend plane (horizontal) resolution (μm)	$\sim$6	$\sim$6	$\sim$6	$\sim$6	$\sim$6	$\sim$6	$\sim$6
Non-bend (vertical) resolution (μm)	$\sim$60	$\sim$60	$\sim$60	$\sim$60	$\sim$60	$\sim$60	$\sim$60

TABLE II: The layout and resolution of the recoil tracker.

Layer	1	2	3	4	5	6
z-position, relative to target (mm)	+7.5	+22.5	+37.5	+52.5	+90	+180
Stereo Angle (mrad)	100	-100	100	-100	-	-
Bend plane (horizontal) resolution (μm)	$\approx$6	$\approx$6	$\approx$6	$\approx$6	$\approx$6	$\approx$6
Non-bend (vertical) resolution (μm)	$\approx$60	$\approx$60	$\approx$60	$\approx$60	-	-

Tagger tracker seeding for ACTS

Space point formation

- No Digitisation available at the moment of making these studies
- 3D points for seeding are taken from Simulated hits

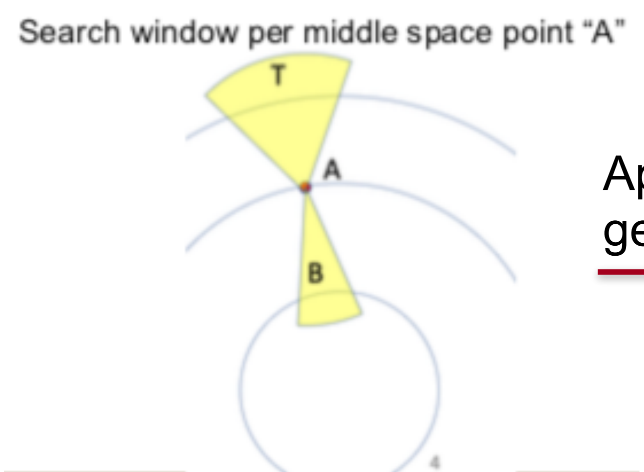
Seed finding

- Bin the available hits in middle-bottom and top points
- Form 3D-space points triplets that are compatible (various algorithms could be used here in principle)
- Fit for track parameters and (to-do) covariance matrix

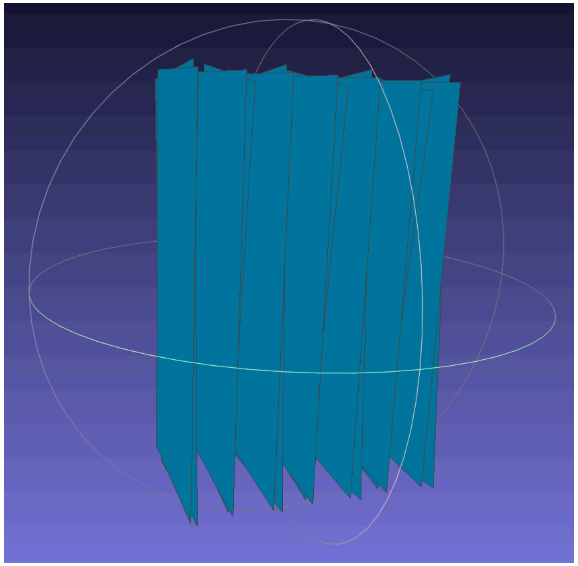
ACTS For LDMX - Seed Finder - Configuration

- I've used the nominal seed finder algorithm provided by ACTS configured for LDMX geometry
- Started by opening up a lot the cuts
  - Notice wide cuts on $\tan(\lambda)$ and $p_T \geq 500$MeV.
  - Assume constant bField of 1.5 Tesla
  - Beamspot in (0,0), wide cuts in z0 (along global Y axis)
- I allowed max 5 seeds for the same middle point, but ended up seeing more… Will need to crosscheck with the developers.

```
//Tagger r max
config.rMax = 1000.;
config.deltaRMin = 3.;
config.deltaRMax = 220.;
config.collisionRegionMin = -50;
config.collisionRegionMax =  50;
config.zMin = -300;
config.zMax = 300.;
config.maxSeedsPerSpM = 5;

//More or less the max angle is somet
//Theta for the seeder is like ATLAS
//Max lamba is of the order of ~0.1 s
config.cotThetaMax = 1.5;

//cotThetaMax and deltaRMax matter to

config.sigmaScattering = 2.25;
config.minPt = 500.;
config.bFieldInZ = 1.5e-3;  // in kT
config.beamPos = {0, 0}; // units?
config.impactMax = 20.;
```



Apply to telescope geometry

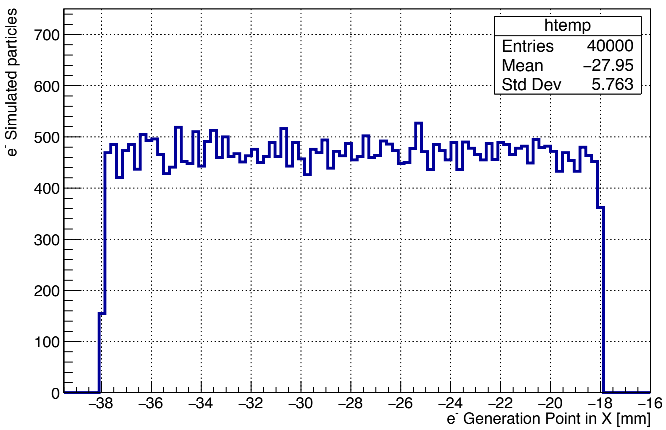


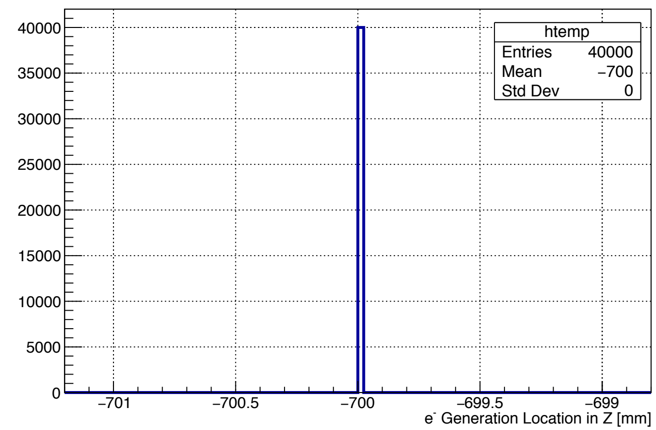
Input Electrons for Seed Finder

- The data used for the seed finder tests are single electrons from particle gun
- Electrons are shot:

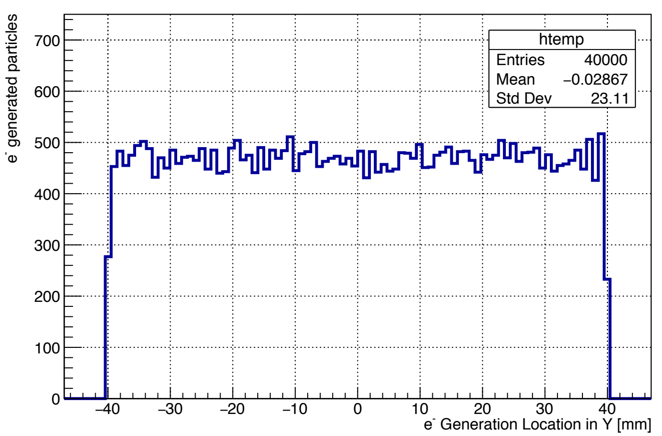
  - Parallel to X-Z plane, so $\tan(\lambda) = 0, \ \ |p| = |p_T| = \sqrt{p_x^2 + p_z^2} = 4$ GeV

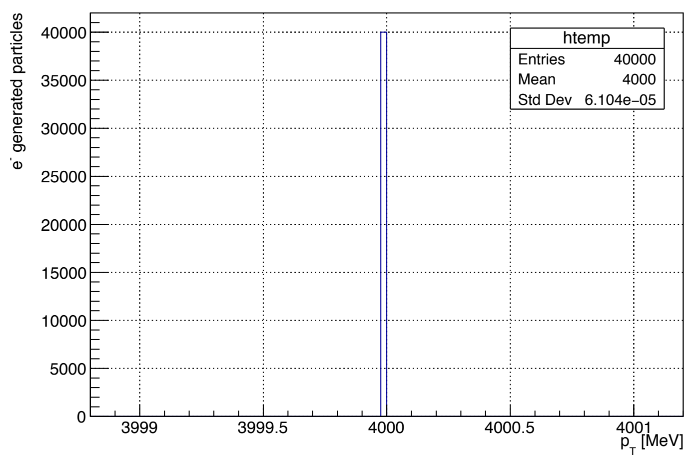
  - Such that they hit the target in a 80x20 region, nominally [-40,40] in Y and [-10,10] in X





Fixed Z=-700 origin
Uniform distribution in X,Y
Fixed transverse 4 GeV pT, with 314 MeV pX





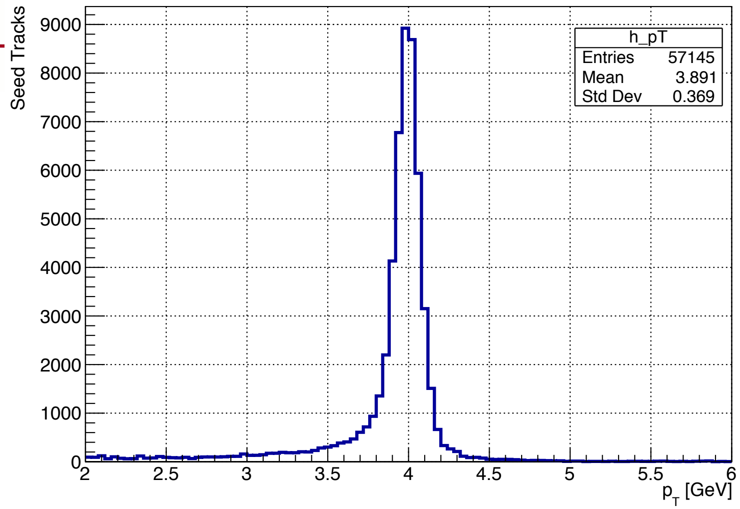
ACTS For LDMX - Seed Finder - Results

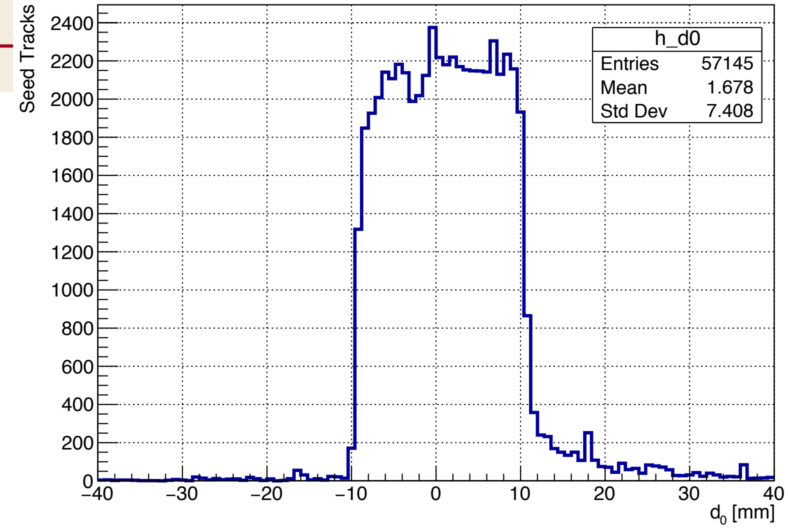
SLAC

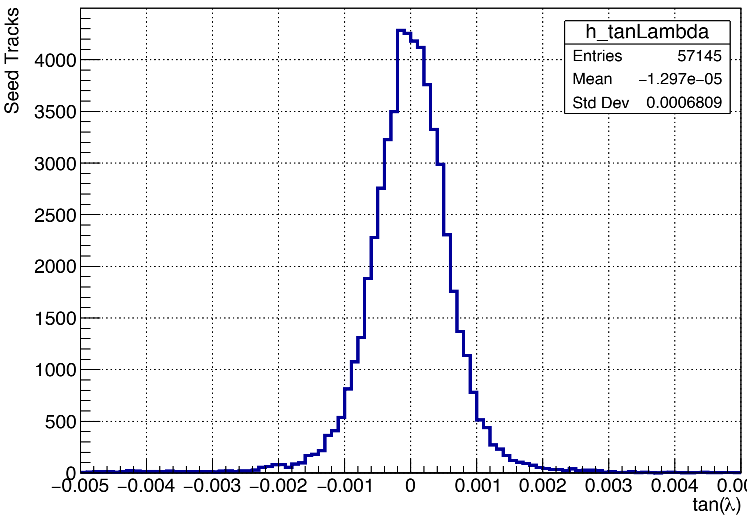
- Example of processing of an LDMX 4 GeV electrons simulated file
- Look for seeds with hits on ly 3,7 and 9 (in 1-14) numbering.
  - Took the simulated hits lefts in the sensors (3D simulated points), read in Ntuple within ACTS framework and interfaced to internal algorithms for seed finding
  - Single iteration, no recursive search, duplicates only present if multiple hits on those layers.
- **All the sim 3D hits boxed in red**
- **Used the 3 3D hits for seeding**
- **58 regions are scanned, 1 seed is found. After seed is found a 3-hit fit is perfomed**
  - I've implemented few fitting procedures:
    - SiTrackMakerTool from ATHENA (ATLAS)
    - Karimaki Circle Fit (NIM 305 1 1991 (187-191))
    - Parabolic fit in conformal space (WIP)

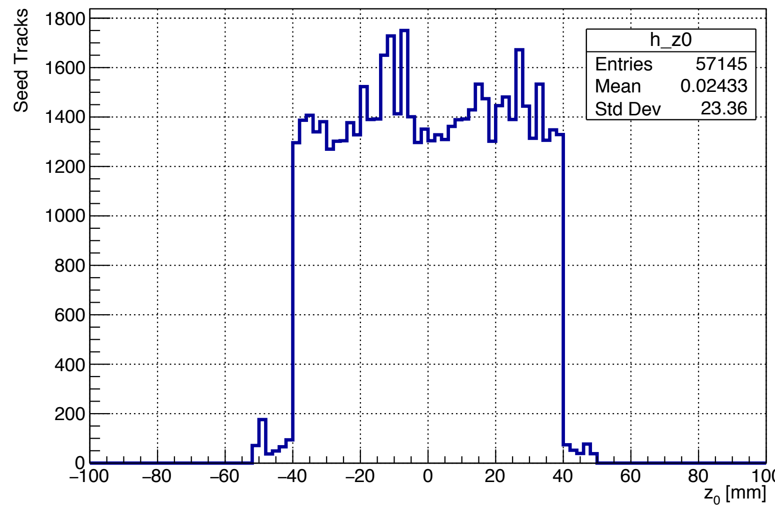
```
13:04:43    Sequencer      INFO      finished event 8
Printing:: LdmxSimHitCollectionwith size 14
Ldmx Sp 67108865 x=-613.1 y=-16.856 z=29.2615
 t=0.291191 ly=1
Ldmx Sp 67108866 x=-607.1 y=-16.4456 z=29.2616
 t=0.311252 ly=2
Ldmx Sp 67108867 x=-513.1 y=-10.5321 z=29.261
 t=0.625423 ly=3
Ldmx Sp 67108868 x=-507.1 y=-10.191 z=29.2627
 t=0.645469 ly=4
Ldmx Sp 67108869 x=-413.1 y=-5.39161 z=29.2897
 t=0.959429 ly=5
Ldmx Sp 67108870 x=-407.1 y=-5.11833 z=29.29
 t=0.979464 ly=6
Ldmx Sp 67108871 x=-313.1 y=-1.35724 z=29.2805
 t=1.29327 ly=7
Ldmx Sp 67108872 x=-307.1 y=-1.15208 z=29.2795
 t=1.31329 ly=8
Ldmx Sp 67108873 x=-213.1 y=1.5334 z=29.2491
 t=1.62697 ly=9
Ldmx Sp 67108874 x=-207.1 y=1.671 z=29.2468
 t=1.64699 ly=10
Ldmx Sp 67108875 x=-113.1 y=3.30894 z=29.2212
 t=1.96059 ly=11
Ldmx Sp 67108876 x=-107.1 y=3.37908 z=29.2204
 t=1.9806 ly=12
Ldmx Sp 67108877 x=-13.1 y=3.95673 z=29.2124
 t=2.29416 ly=13
Ldmx Sp 67108878 x=-7.1 y=3.95988 z=29.2122
 t=2.31418 ly=14
spVec [0] layer 3
(x,y,z) = (-513.1,-10.5321,29.261)
spVec [1] layer 7
(x,y,z) = (-313.1,-1.35724,29.2805)
spVec [2] layer 9
(x,y,z) = (-213.1,1.5334,29.2491)
0 0 -3.11834 1.57076 -0.000250806
3 hits, 58 regions, 1 seeds
```

ACTS For LDMX - Seed Finder - Results





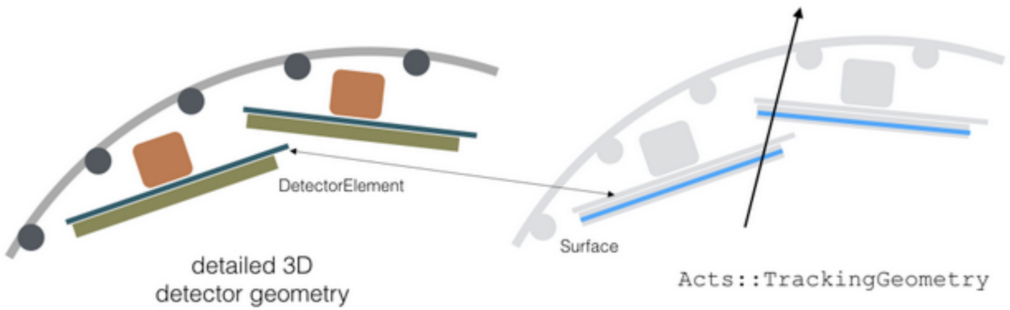


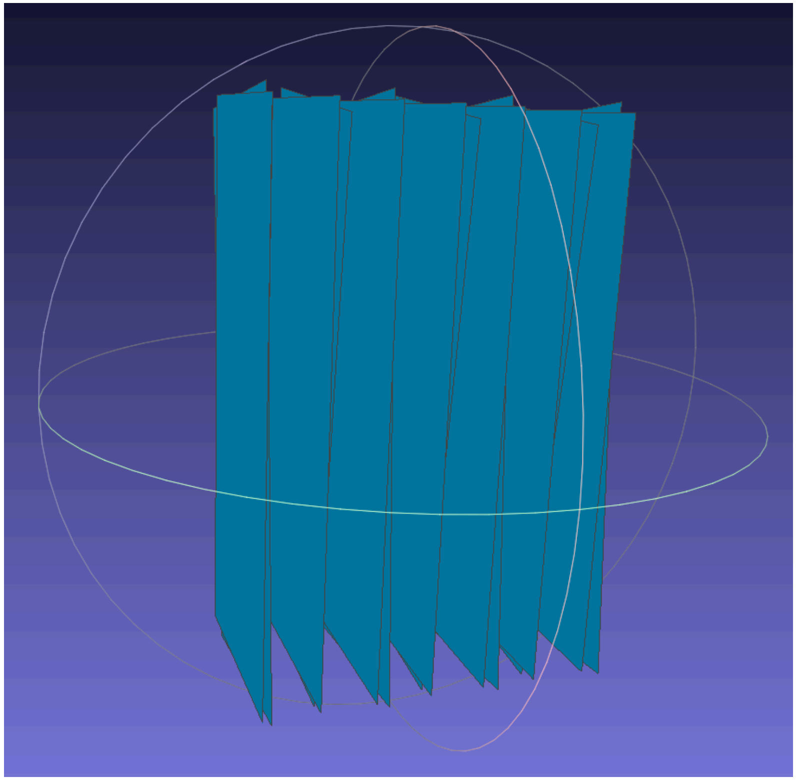


- Seed finder correctly fits tracks to **pT~4 GeV**, **Impact parameters are within the expected limits from simulated production**. Tail in the d0, due to particles fit with low momenta.

ACTS For LDMX - Tracking Geometry

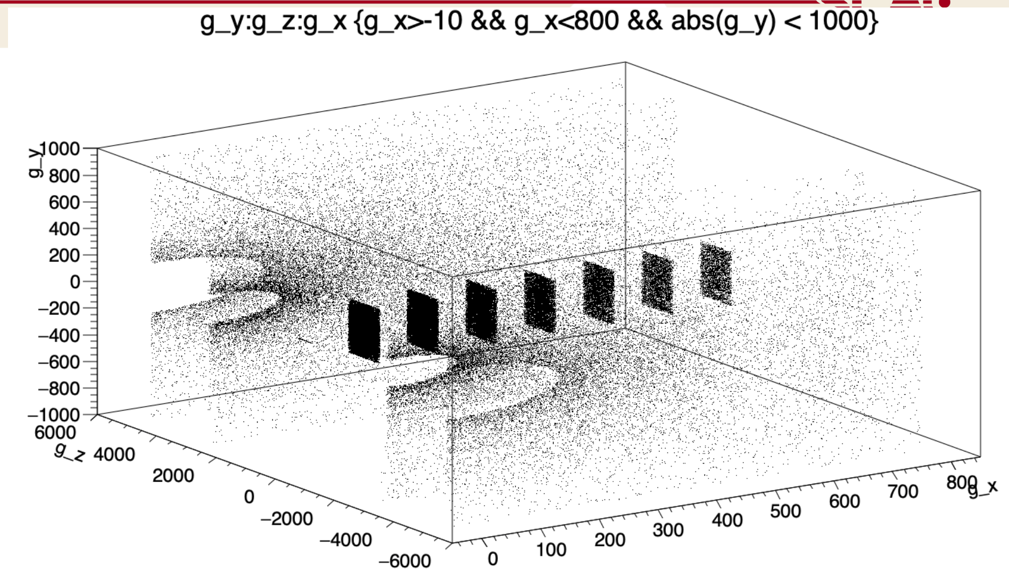
- The ACTS logic follows a lot ATLAS trac' reconstruction logic:
- Construct the tracking geometry:
  - Surface-based description of a detector which is used together with the track propagation module.
  - Time(Event)-dependent description available from context
  - Material description
  - Hierarchical structure of surfaces->layers->volumes is implemented.
- **I built up a mock-up tracking geometry for the tagger tracker, but Omar already progressed to load the tagger into ldmx-sw => new starting point for this exercise**
- I also attached the obj writer to dump the tracking geometry to a format readable by MeshLab, so it can be easily visualised

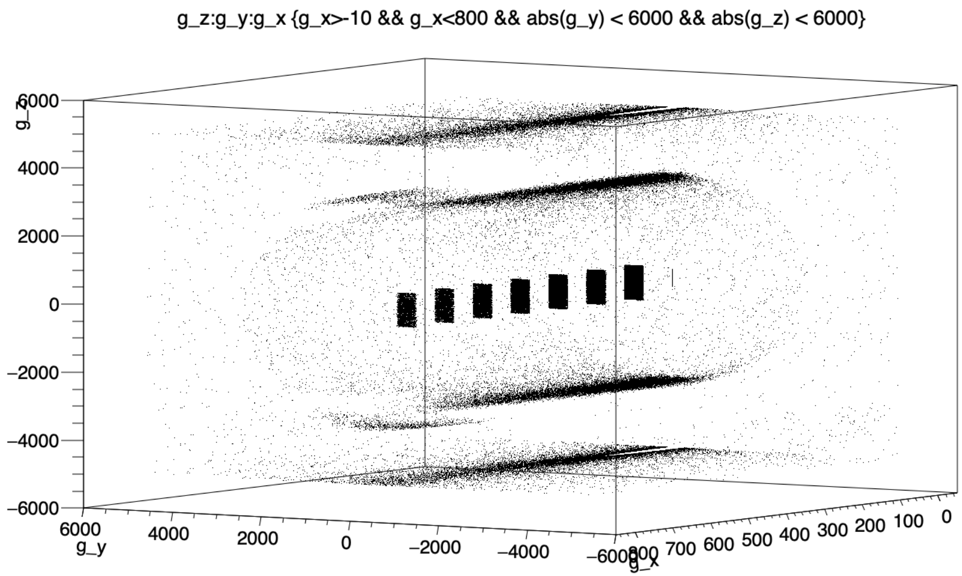




ACTS For LDMX - Tracking Geometry

- As I said tracking geometry is at the basis for track propagation.
- More technically, the instance of the track propagator needs to know where sensitive elements are placed.
- ACTS provides an example to run the track propagator and show the location at each step and intersections with material. More informations available at these tutorials
- I ran the ACTS Track propagator with the LDMX Tagger Tracking geometry, hooking it up to the ROOT Writer to produce validation tuples to show the propagation steps.
- The propagator example follows a cylindrical geometry, starting from a random point along the B-Field axis (z) with a certain ϕ angle.
- The intersections with LDMX tracking geometry can be seen as well as the steps out of acceptance.
- In this example a simpler tracking geometry with no stereo angle was used.
- This should validates the tracking geometry building (at least its internal consistency)

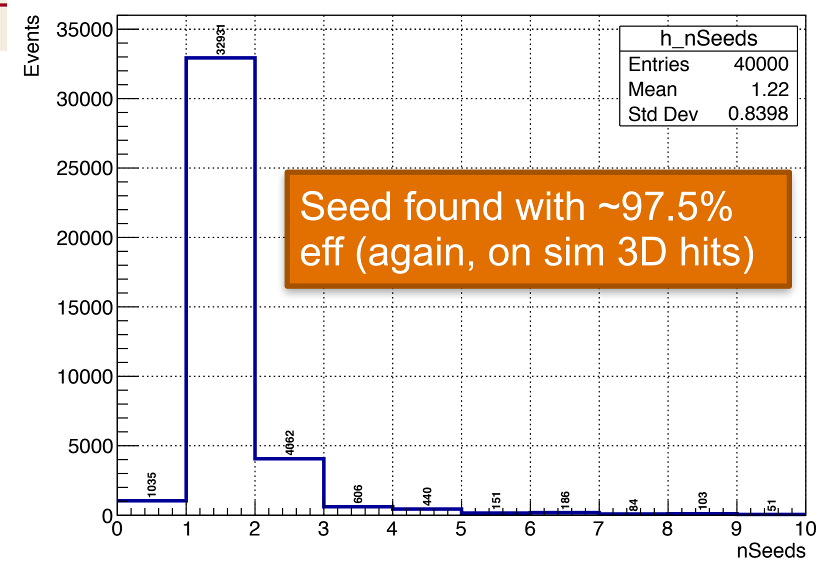


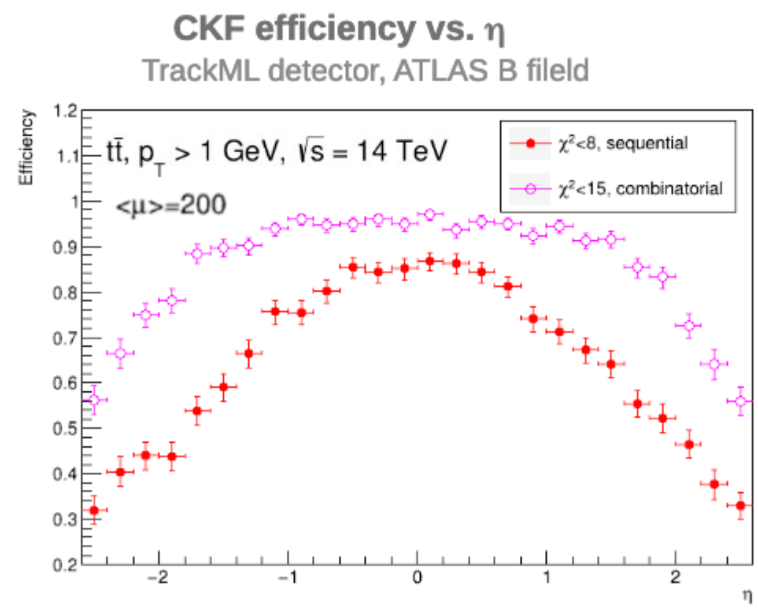


Next Steps

Track candidates
(Combinatorial Kalman Filter)

- After seeds are available next step is to run a proper track finding strategy and track fitting
- ACTS provides a combinatorial Kalman Filter:
  - Support hole search and outlier rejection during fitting
  - Support 2 different approaches for smoothing
    - Smoothing matrix formalism
    - Backward propagation
- Gaussian Sum Filter as non-gaussian extension of KF is available





[More infos on CKF - Xiaocong Ai CTD '20]

Summary

- I’ve started digging more seriously in ACTS for implementing tracking in LDMX.
- I’ve managed to successfully form a basic tracking Geometry and tested the internal consistency running the track propagator and visualising the intersections with the tracker material
- Using the ACTS Framework I managed to:
  - Read in LDMX tuples to obtain 3D space points using the ROOTReader interface provided.
  - Wrote an algorithm to convert LDMX sim hits into ACTS 3D Space points and store them in the transient event store
  - Fed those to the SeedFinder algorithm which, after some configuration jiggles, I set it up to be ~97.5% efficient on those LDMX events (based on Simulated hits)
  - Started contributing to ACTS writing a SeedToTrackParameters tool “inspired” from ATHENA.
- Next steps:
  - Wrap up the seed finder/fitter
  - Interface the hits to the CKF algorithm and see what happens