CEPC-ACTS integration

Review and Perspective

Outline

Status of CEPC-ACTS framework

Status of Geometry and TPC/FST alternative

A plan & roadmap to be discussed

CEPC-ACTS taste software

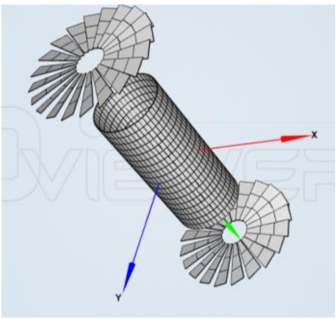
https://gitlab.cern.ch/jinz/acts-framework-cepc, aim to

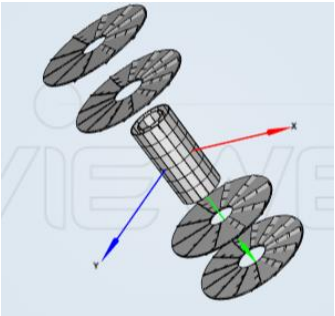
- Keep the progress in this repository for the moment
- trace the updated version of CEPC-ACTS detector(without TPC)
- Include all available examples in the master branch
- facilitate tracking fans to checkout and run easily
  - Possible to profit from the stand-alone examples, to start with your interested part and dive into TRACKING

Examples are tested with CEPC_V0 detector

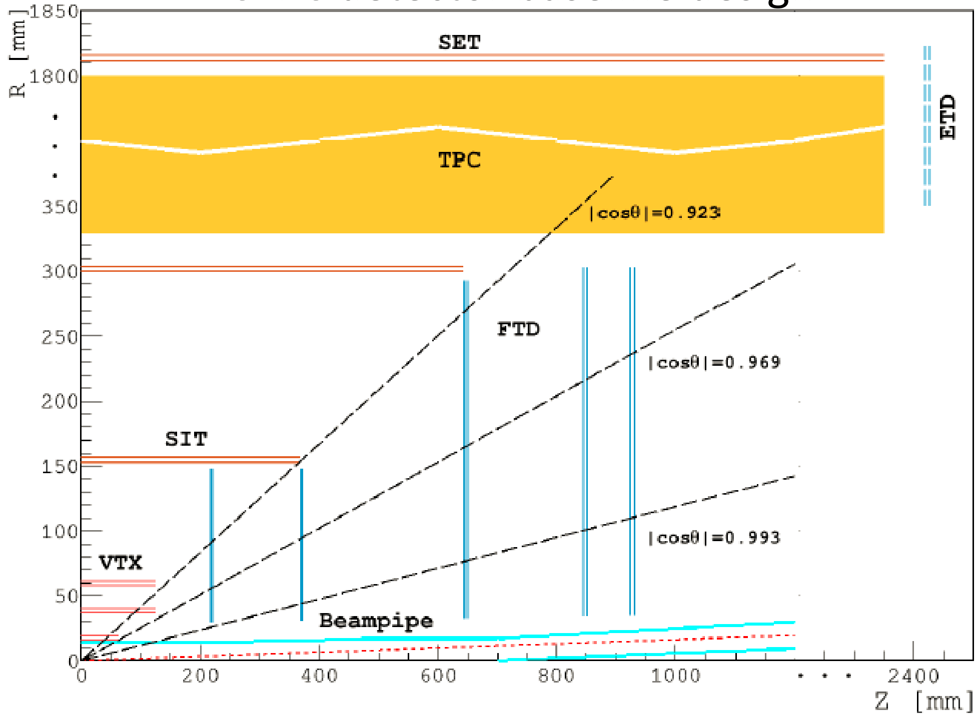
- Geometry
- Propagation (particle motion in detector)
- GeatinoRecording (record material with Geant4 plugins)
- FATRAS (Fast Atlas Tracking Simulation)
- Kalman

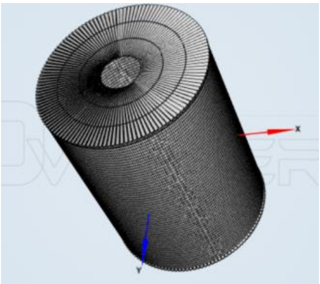
Geometry Example

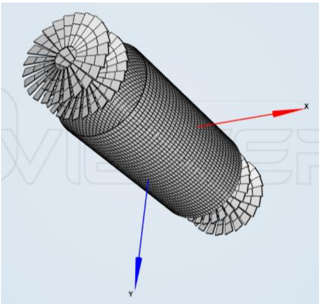




CEPC detector baseline design

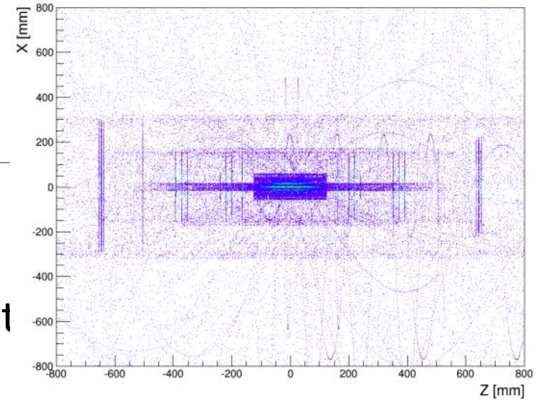






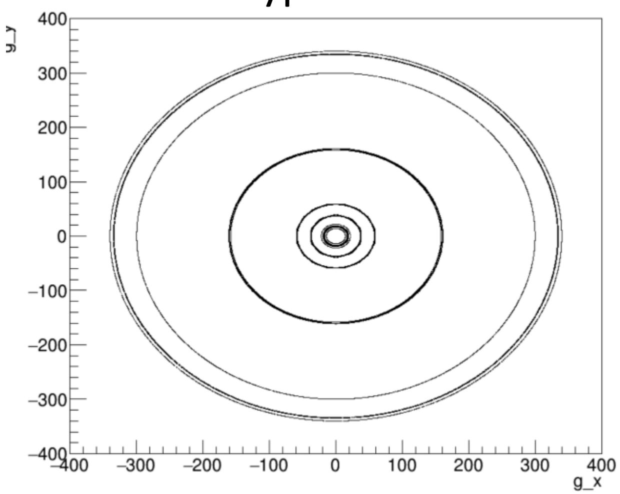
- The Geometry example is to build geometry internally, output some visible objs
- The obj files only show the sensitive detectors, not the real tracking geometry
- Subdetector: barrel + endcap
- ./ACTFWDD4hepGeometryExample --dd4hep-input ../../Detectors/DD4hepDetector/compact/CEPC/cepc_v04_master.xml --geo-subdetectors Tracker_2 Tracker_3 Tracker_1 ETD --output-obj 1

Propagation



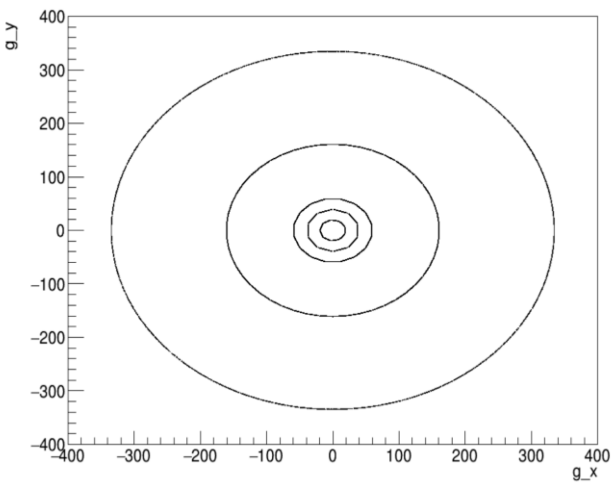
- Generate n events (each contains substantial toy tracks)
- Recording the steps of different type (navigation – Geomet (type==0 …))
- Flexible debug tool for the tracking geometry
- ./ACTFWDD4hepPropagationExample --dd4hep-input ../../Detectors/DD4hepDetector/compact/CEPC/cepc_v04_master.xml -n 10 --bf-values 0 0 3 --output-root 1

type ==0



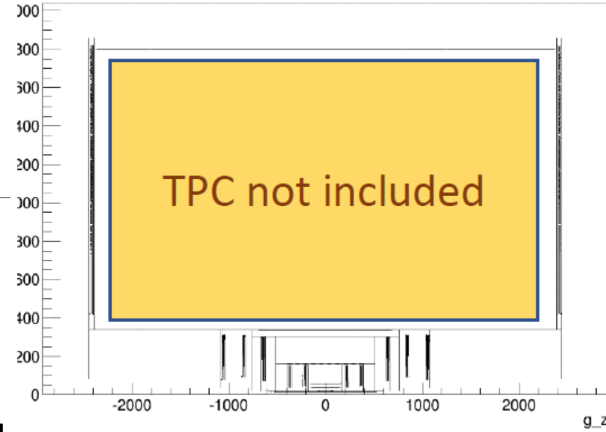
X-Y view : Vertex Navigation detectors

type ==0 && sensitive_id



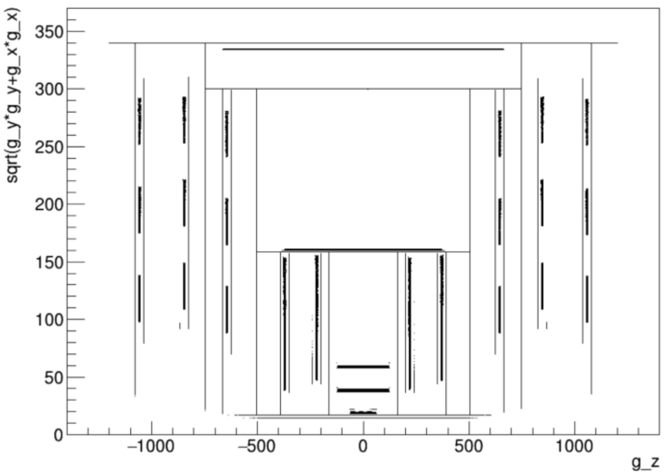
X-Y view: Vertex Sensitive detectors

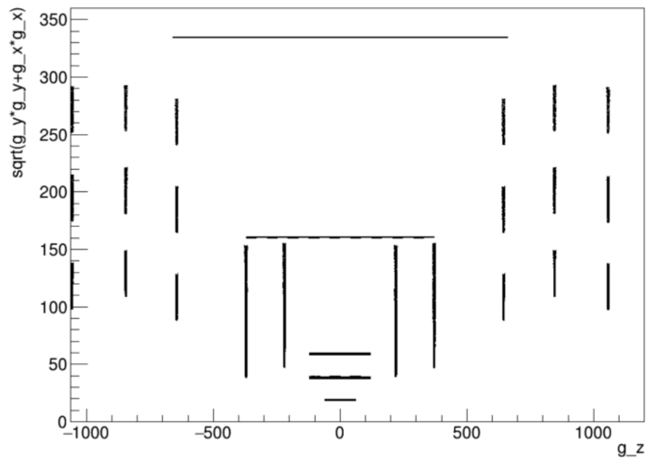
Propagation



propagation_steps->Draw("g_y : g_x","abs(g_z)<100 && abs(g_x)<1000 && abs(g_y)<1000 && type ==0 ")

propagation_steps->Draw("g_y : g_x","abs(g_z)<100 && abs(g_x)<1000 && abs(g_y)<1000 && type ==0 && sensitive_id")





Z-R view : Navigation detectors (w/o SET ETD)

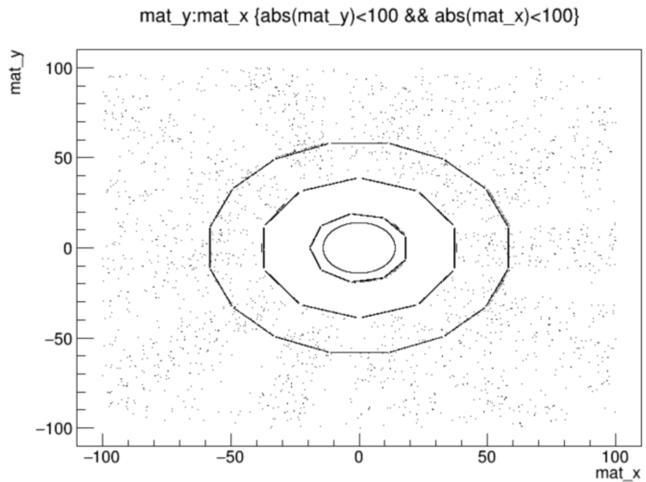
Z-R view : Sensitive detectors (w/o SET ETD)

GeatinoRecording

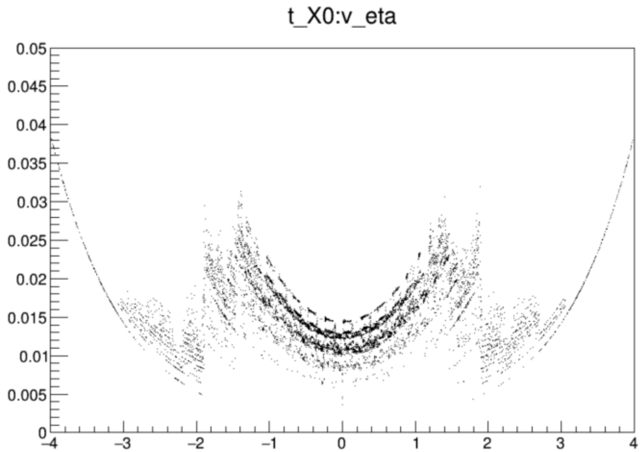
- Use Geant4 Plugin to record the material (with Geant4 extrapolation)
- Geant 4 only allows single thread

./ACTFWGeantinoRecordingExample -n10 --dd4hep-input ../../Detectors/DD4hepDetector/compact/FCChhBaseline/FCChh_DectEmptyMaster.xml --output-root true -j 1

tv__tree->Draw("mat_y:mat_x","abs(mat_y)<100 && abs(mat_x)<100","", 10000, 0);



tv__tree->Draw("t_X0:v_eta>>h(100,-4,4,100,0,0.05)","","", 10000, 0);



Fatras

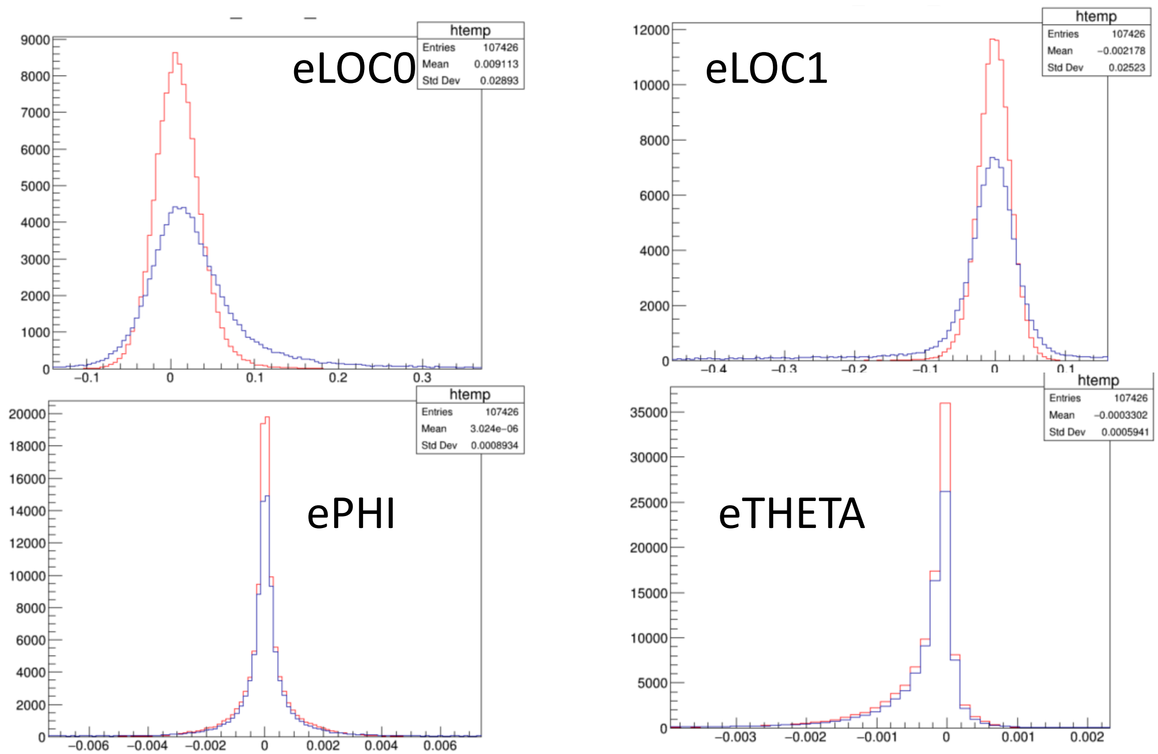
```
./ACTFWDD4hepFatrasExample --dd4hep-input ../../Detectors/DD4hepDetector/compact/CEPC/cepc_v04_master.xml -n 10 --bf-values 0 0 3 --output-root 1
```

Kalman filter

Currently inserted in the FATRAS - split out by Moritez in progress

./ACTFWDD4hepFatrasExample --dd4hep-input ../../Detectors/DD4hepDetector/compact/CEPC/cepc_v04_master.xml --evg-input-type gun --evg-pileup 0 --pg-nparticles 10000 --pg-eta-range 1.3 1.3 --pg-phi-range 0 0 --pg-pt-range 10 10 --output-root 1 --fitted-tracks test_tracks -n 1

Filtered VS Predicted



Status of CEPC-ACTS detector geometry

CEPC CDR detector

- baseline concept - Silicon + TPC
- FST - Full silicon
- IDEA - Silicon + drift chamber

Now : V0 (baseline without TPC)

Next : V1 (impl two alternatives TPC/Full silicon) + more details

CEPC-ACTS detector V0

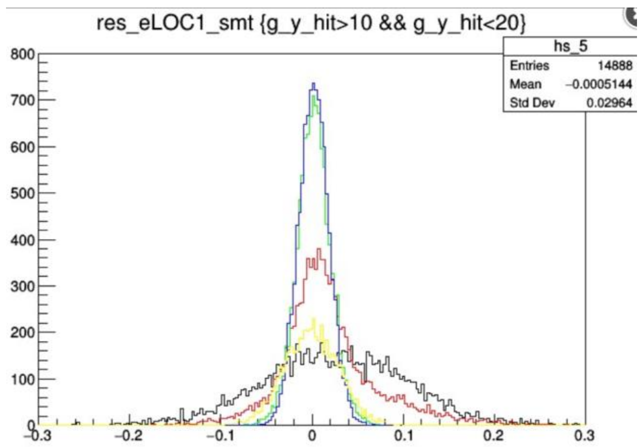
CEPC-ACTS detector (V0) inner tracker baseline concept

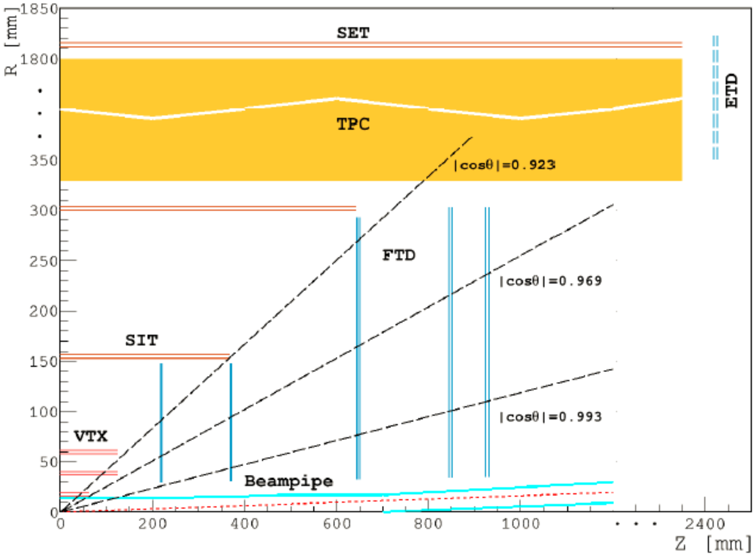
- Silicon trackers : VTX + SIT + FTD + SET + ETD

No TPC/silicon detector

Material budget need to update

Some bugs in the tracking geometry





Resolution of Smoothing from different layers

VTX + SIT : some layers bad smoothing res

CEPC-ACTS detector V1

CEPC-ACTS detector (V1) inner tracker

- Fix bugs in V0
- Add TPC detector
- Add full silicon tracker as an alternative
- Add some material -> apply material mapping

Full silicon version – not difficult

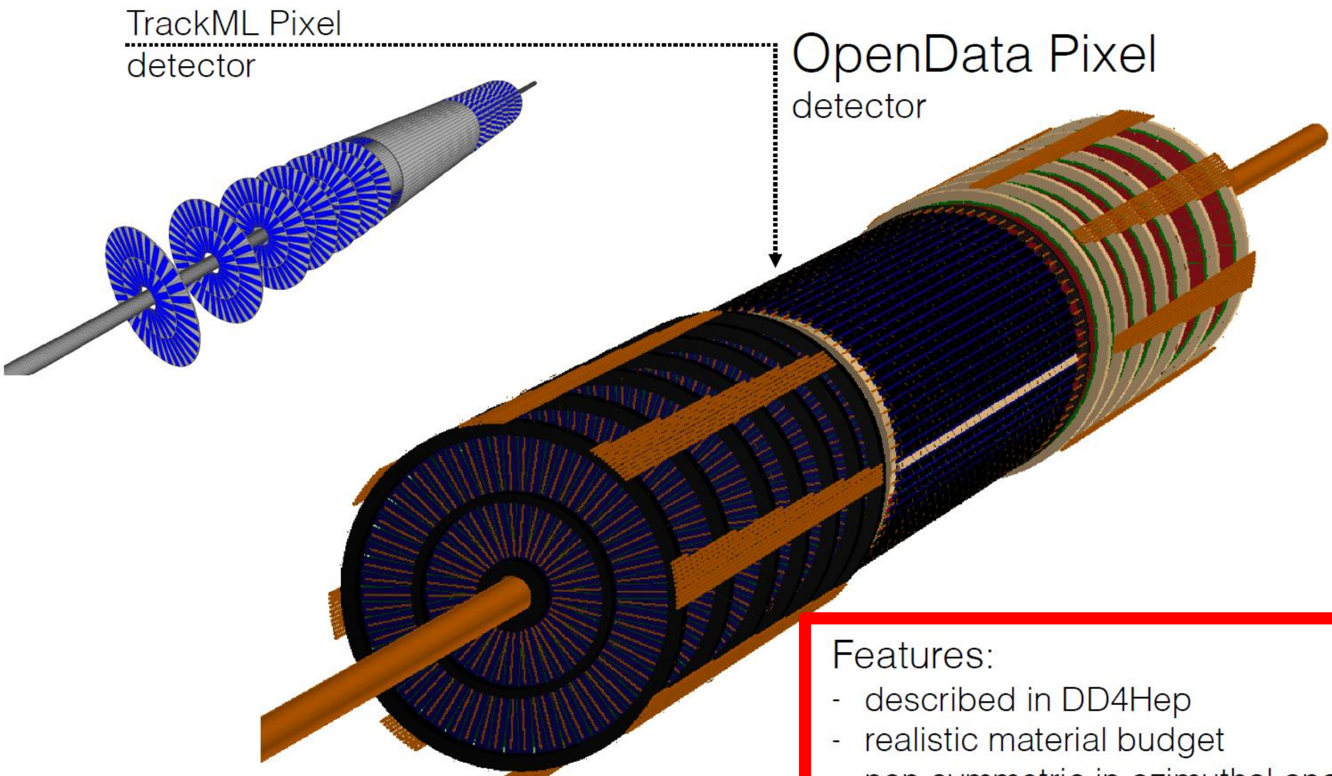
TPC – first try a simple surface-based version

Material mapping : will release soon

For this step, we can gain full experiences from the OpenDataDectector

The OpenDataDetector

TrackML detectors -> OpenData detector in ACTS



Features:
- described in DD4Hep
- realistic material budget
- non-symmetric in azimuthal angle
- full (G4) and fast (ACTS) simulation
- misalignment possibility

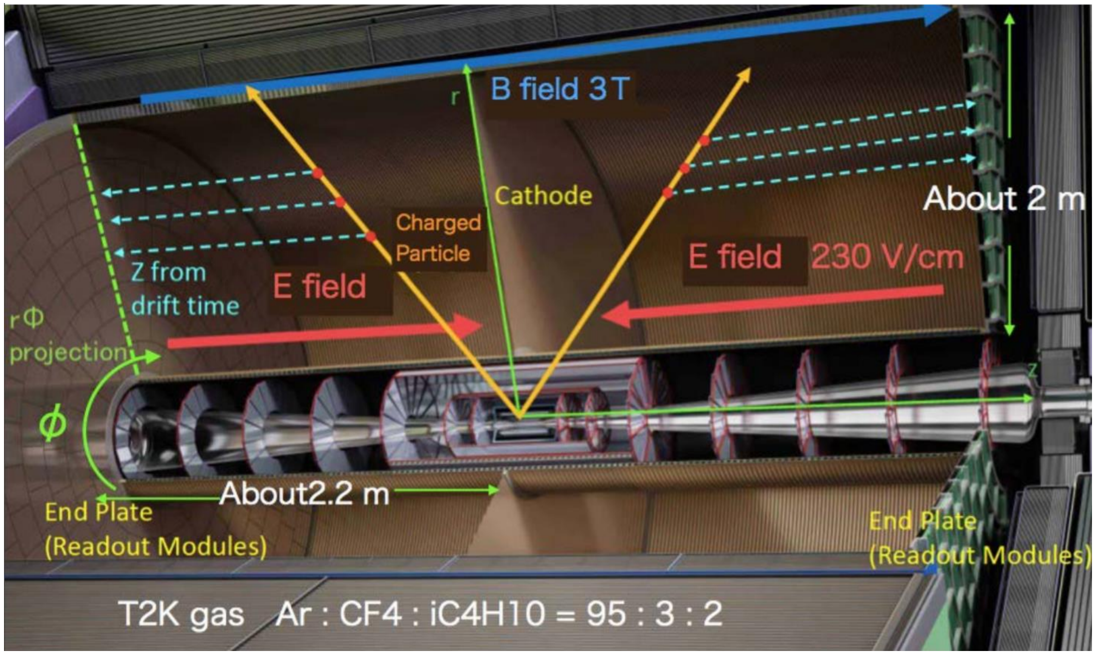
… to be released soon!

TPC

3-D spacepoint

Low material budget

High density spacepoint



TPC Options

Surfaces based concept – personally prefer at first the stage

- 220 Cylinder surfaces – consistent with Pad row
- Define measurement on surface

No surface based concept

- Propagate with denseEnvironment concept
- Create Surface at measurement
- Extremely Different in how to do filter update

No surface based concept – free measurement update – under discussion now

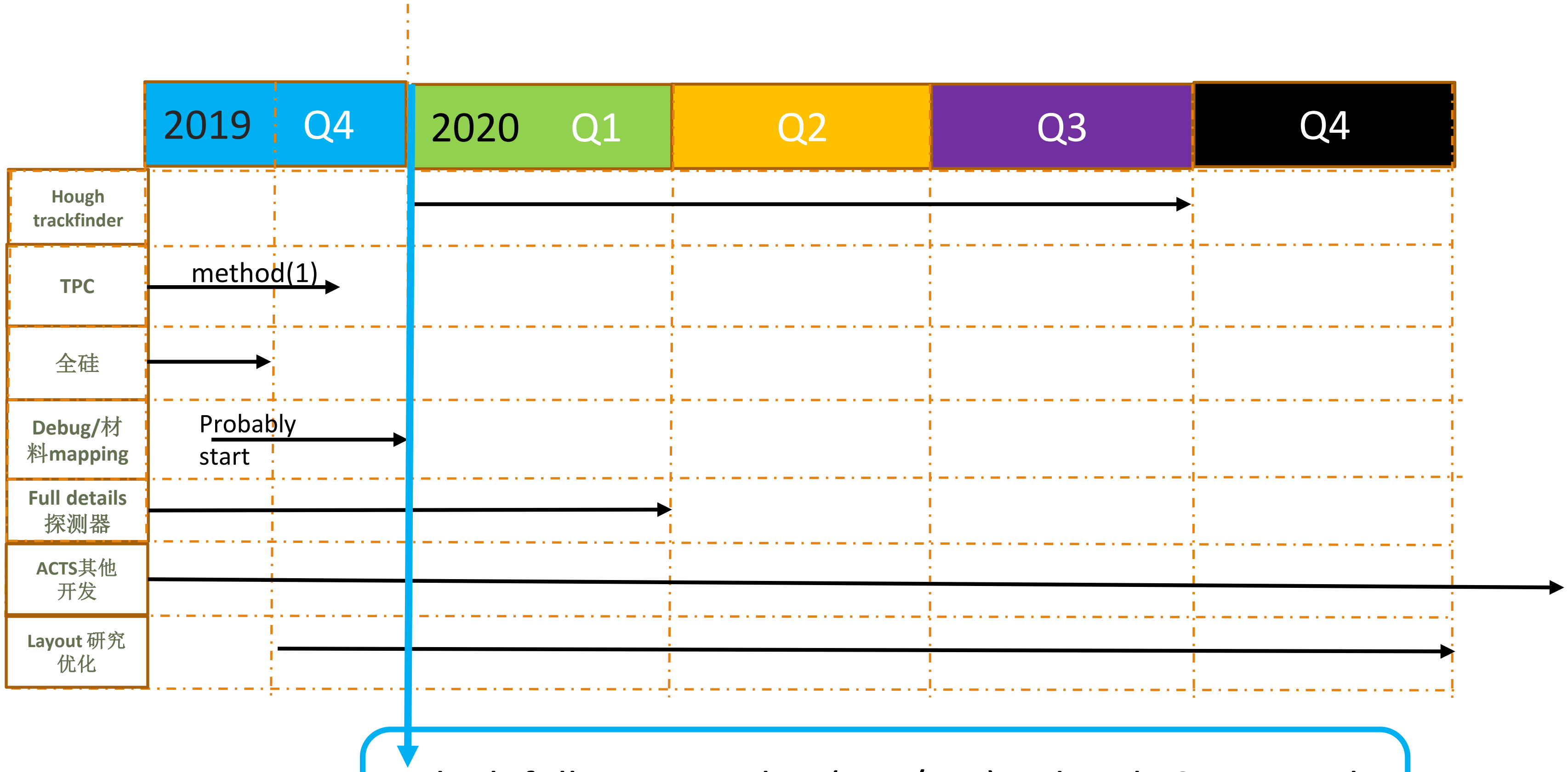
More options??

TPC option (1)

Surface-based concept

- Geometry: 220 cylindrical nested surfaces along 220 pad rows, filling with harmonious material, 220 layers readout
- Propagation: DenseEnvironment propagation (energy loss in each step)
- FATRAS: using 220 surfaces propagation, make it quite simple at first stage
- Digitization: a Cylinder Digitization model is needed (or even more naive version)
- Kalman filter: traditionally on measurement surface

Pre Roadmap of the CEPC integration



Ideal: full inner tracker (TPC/FST) + details & material mapping + some layout optimization

Backup

Propagation

A general issue dealing with particle motion

In the magnetic field

integrated to the tracking geometry

material effect, measurement update

Stepper

Reliable accuracy

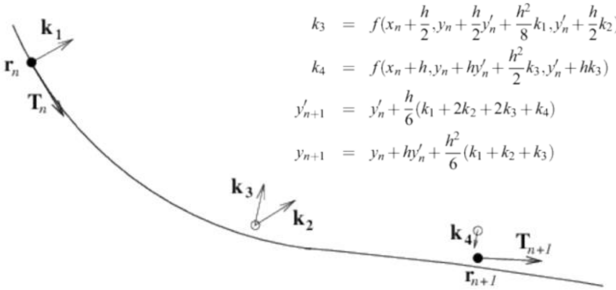
Navigation

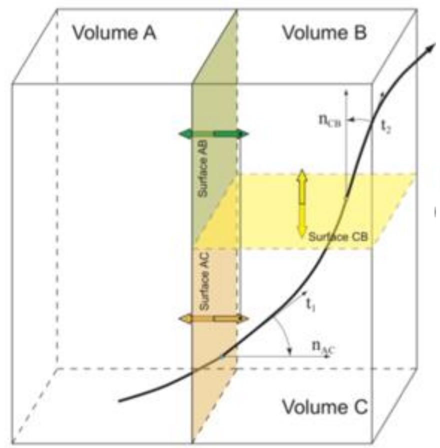
On surface

Actors

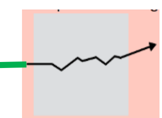
Runge-Kutta-Nystroem method

$$
\begin{aligned}
k_1 &= f(x_n, y_n, y'_n) \\
k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n + \frac{h^2}{8}k_1, y'_n + \frac{h}{2}k_1\right) \\
k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n + \frac{h^2}{8}k_1, y'_n + \frac{h}{2}k_2\right) \\
k_4 &= f\left(x_n + h, y_n + hy'_n + \frac{h^2}{2}k_3, y'_n + hk_3\right) \\
y'_{n+1} &= y'_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
y_{n+1} &= y_n + hy'_n + \frac{h^2}{6}(k_1 + k_2 + k_3)
\end{aligned}
$$

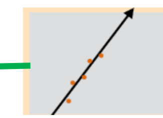




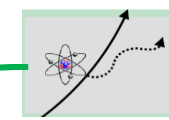
Multiple scattering



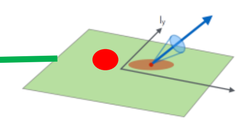
Ionization loss



Bremsstrahlung energy loss



Measurement update



......

Track calculates and returns the error
Adaptive of step size according to error
Vacuum/dense Environment(Calo) is chosen with a voting strategy

our fully connected tracking geometry
Come in convenient!

Aborters

End of the world
Path limit
Target surface
......

TPC strategy (3)

The tracking geometry design is quite determined with the strategy how you implement your propagation -> since tracking is to deal with particle motion issue

Silicon detector carries 'big' materials on surface <->Layer concept

- Propagator must act from surface to surface in terms of discontinuous materials

TPC and other gas chamber issue: if we have continues material / measurement, the layer concept is meaningless

- much time waste
- Meaninglessly material mapping
- Quite confusion of Geometry issue and the Readout issue

Concept we will deal with

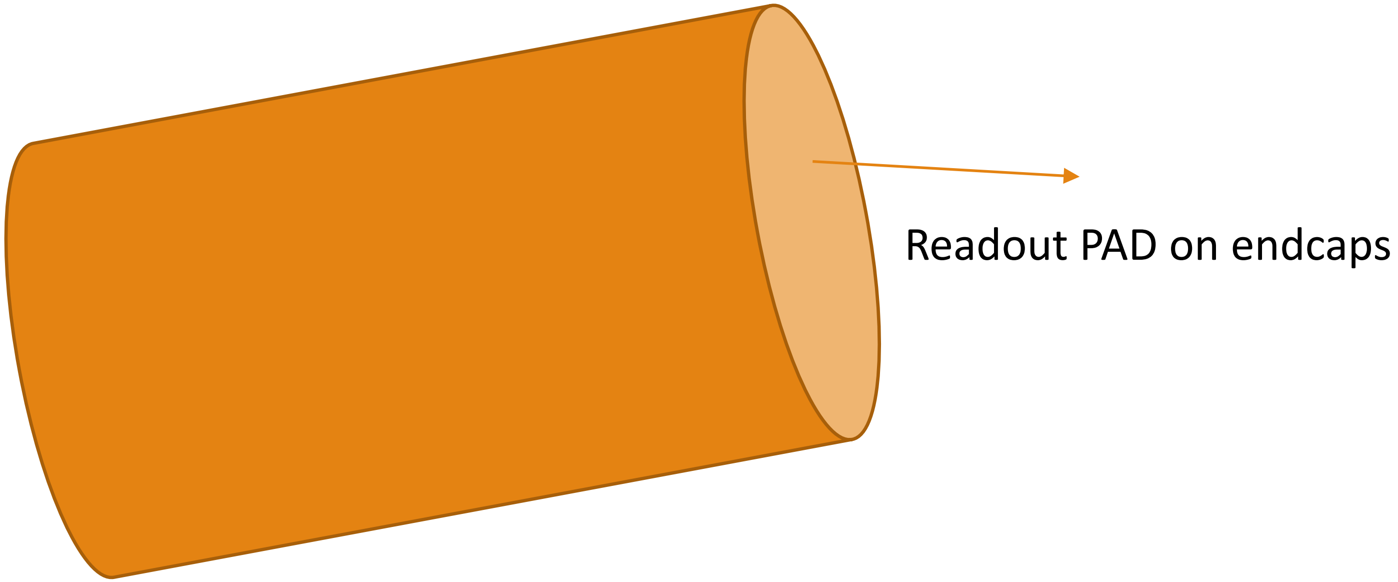
Sense no one tries gas chamber yet(TRT is based on the 40 layers concept), here is the list we should refer to

- Geometry
- Propagation
- FATRAS
- Digitization
- Fitting

TPC strategy - Geometry

Quite simple geometry

- A cylinder with gas filling it
- With readout on the two readout plane



TPC strategy – Propagation

In terms of the TPC gas filled volume, the navigation is just from start to the outer cylinder surface (current layer concept TRT has more than forty layers to navigate)

The propagate extensionList in ACTS

- Default (for the silicon base concept) ->Silicon
  - No energy loss between surfaces
- DenseEnvironment (for the calo) ->TPC probabaly
  - Calculate energy loss along the path

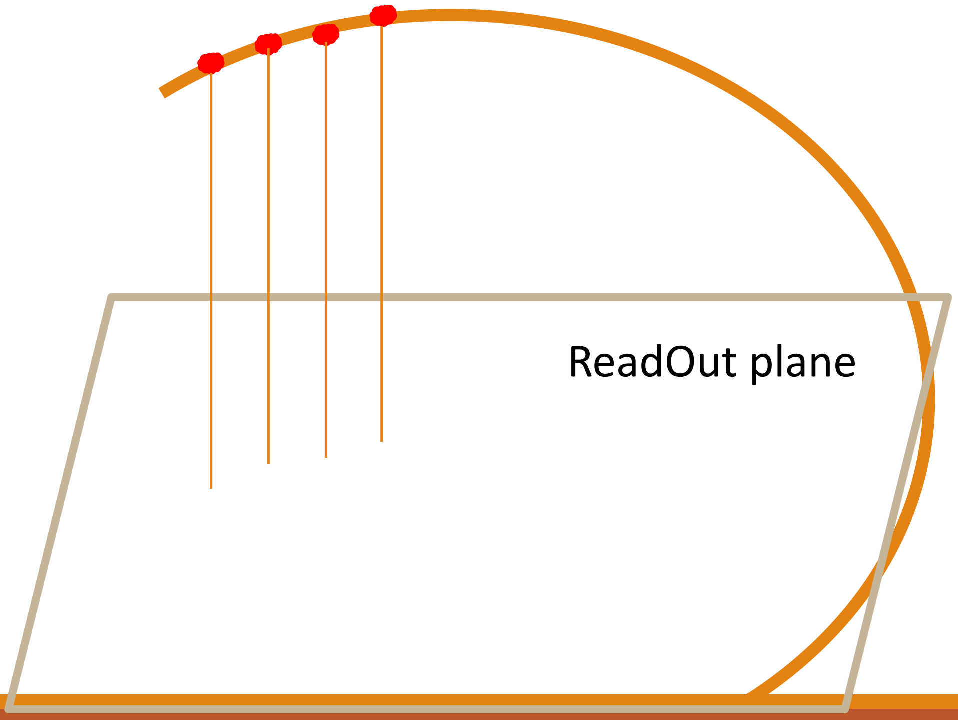
Silicon : add materialEffect only at surfaces + empty between surfaces

Gas : add materialEffect along the road + no surfaces inside the volume

TPC strategy - FATRAS

FATRAS is based on the propagation in ACTS

Add an actor to recording the position along the track with fixed stepSize to create “measurement”



TPC strategy - Digitization

We have to write the digitization ourselves -> that is an issue never be rounded for every experiment

But could we just take the EndCap plane as one readout Plane and consider it as a sensitive surface as silicon and extend it to 3-D with drift distance along Z direction? In that way we can use exited PlaneDigitization models in the ACTS(as same as the CEPC detector)

TPC strategy – Fitting

Measurement list as the Input of Fitting/Propagation

Take the Position of measurement (pos, posCov)

Propagate to the measurement with DenseEnvironment extension

Need to be considered carefully

Kalman update : x_pre + K * (mes – H * x_pre)

- The logic changes : update on Surface -> update globally
- Measurement define (id)
- The measurement dimension
- The H matrix description

At the start, we can simply do something in ACTS-core with

- define a cylinder geometry
- Densenvirment propagate
- Go through the detector …
- …..

End-plate

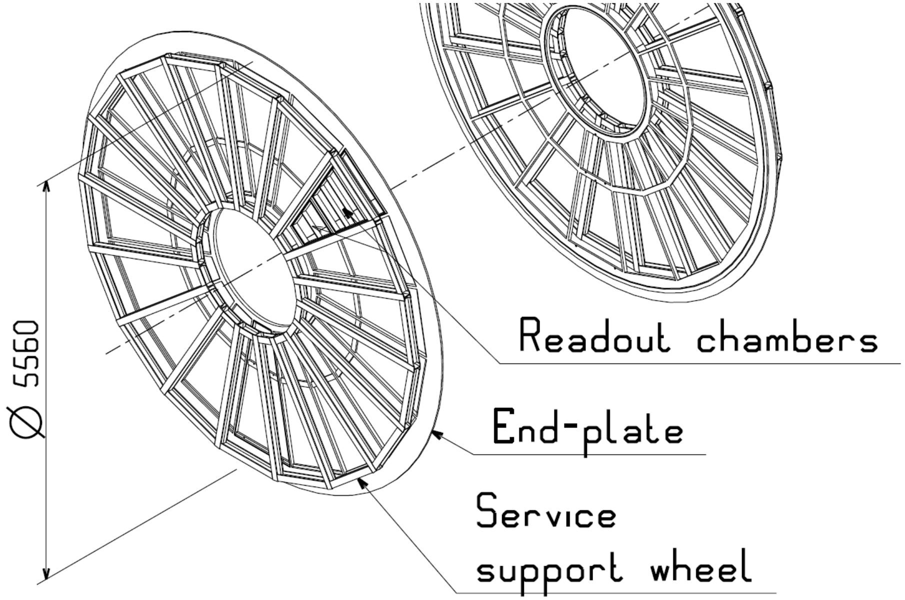


Figure 3.8: Design of the end-plate and the service support wheel.

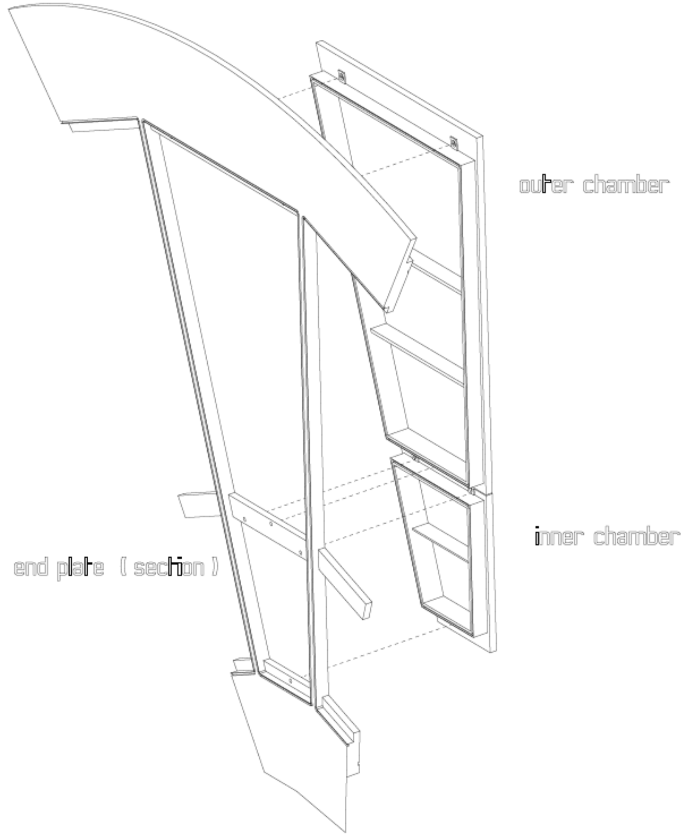
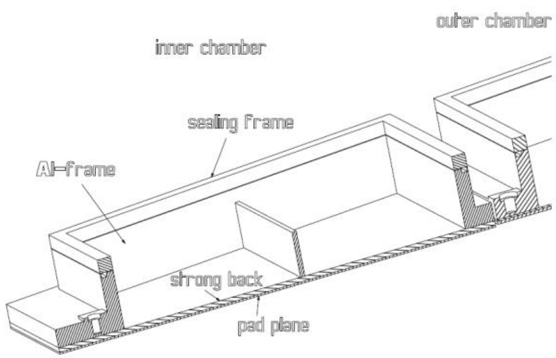


Figure 4.35: Three-point mounting of the readout chambers.

ReadOut Mechanical



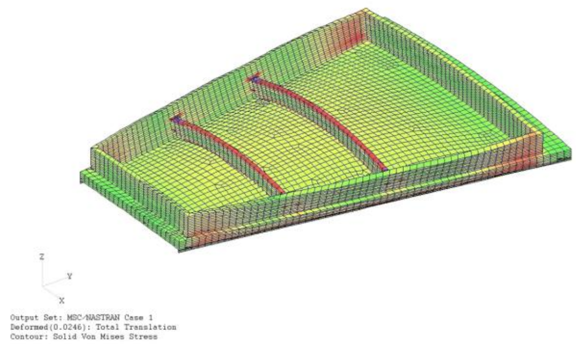


Figure 4.2: Cross-section through an inner readout chamber. Shown are the aluminium frame with the sti ribs, the O-ring groove and threads for the three-point suspension, the pad plane PCB, and the auxiliary back plate.

Figure 4.3: Finite element calculation of the deformation of the outer chamber caused by the wire tension. The maximum deformation is 25 μm.

Pad plane: made of multilayer printed circuit board (PCB)
+
Strong back
+
Trapezoidal aluminum frame