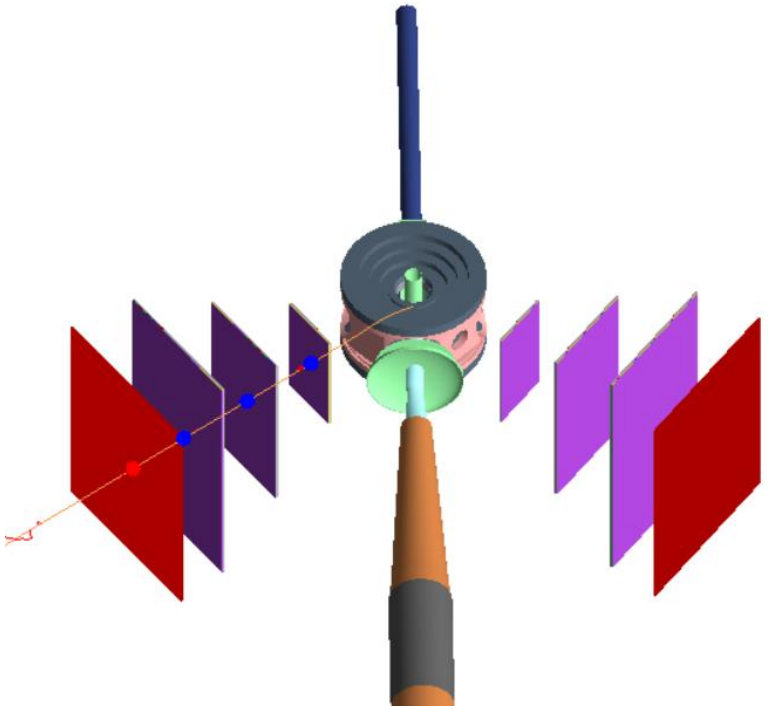
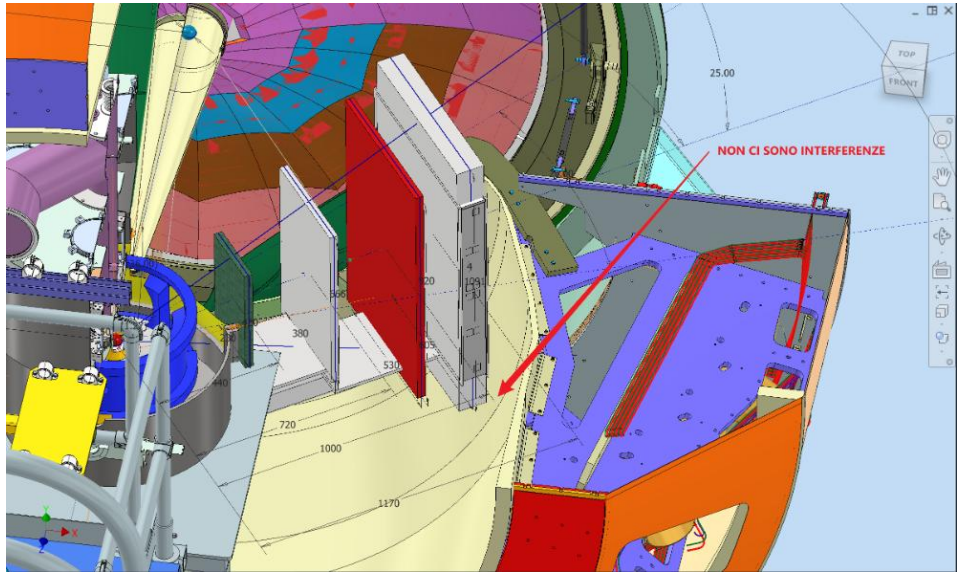
Recoil simulation and reconstruction software, and early performance studies

Silvia Niccolai, IJCLab (France)

RGH Workshop (Jlab)
11/14/2026

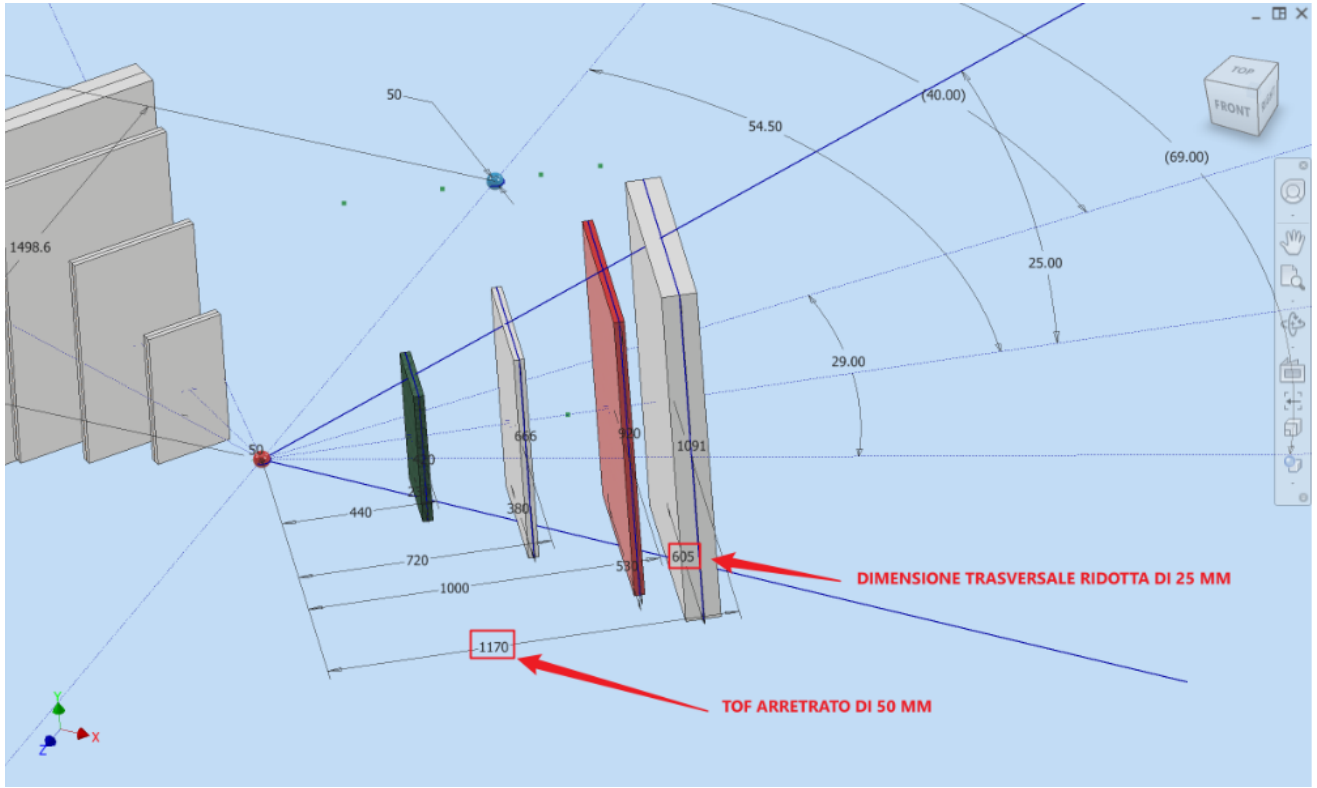


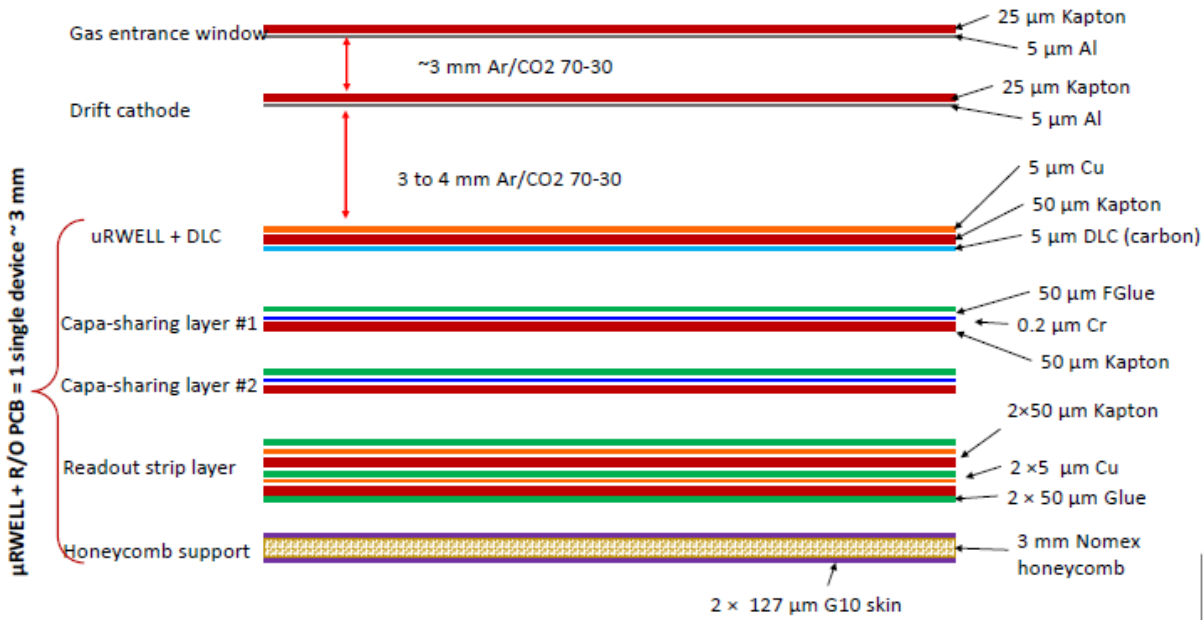


Recoil tracker implementation in GEMC/Coatjava

- Branched Coatjava, GEMC/detectors, and GEMC/source
- Followed the examples of the µRwell for the CLAS12 high lumi, implemented by Mariangela Bondi (lots of thanks to her for the help!)
- Geometry and positioning defined in CJ geometry server
- Number of layers, dimensions, angular opening, number of strips given as parameters, easily modifiable
- Written GEMC/detectors scripts that call CJ to create geometry/material files needed by gemc.
- Current setting:
  - 2 sectors, 3 regions/sector of µRwell, 1 mm strip pitch
  - detectors are placed and dimensioned according to Sandro's drawings
  - each sector and region has x and y (perpendicular) readout

Plane	Name	Distance from IP (mm)	Width (mm)	Height (mm)
1	Tracker Region 1	440	230	420
2	Tracker Region 2	720	380	670
3	Tracker Region 3	1000	530	920
4	Time-of-flight	1180	650	1100





Recoil tracker implementation in Coatjava: banks

- Gemc digitization: RECOIL::adc bank created and filled

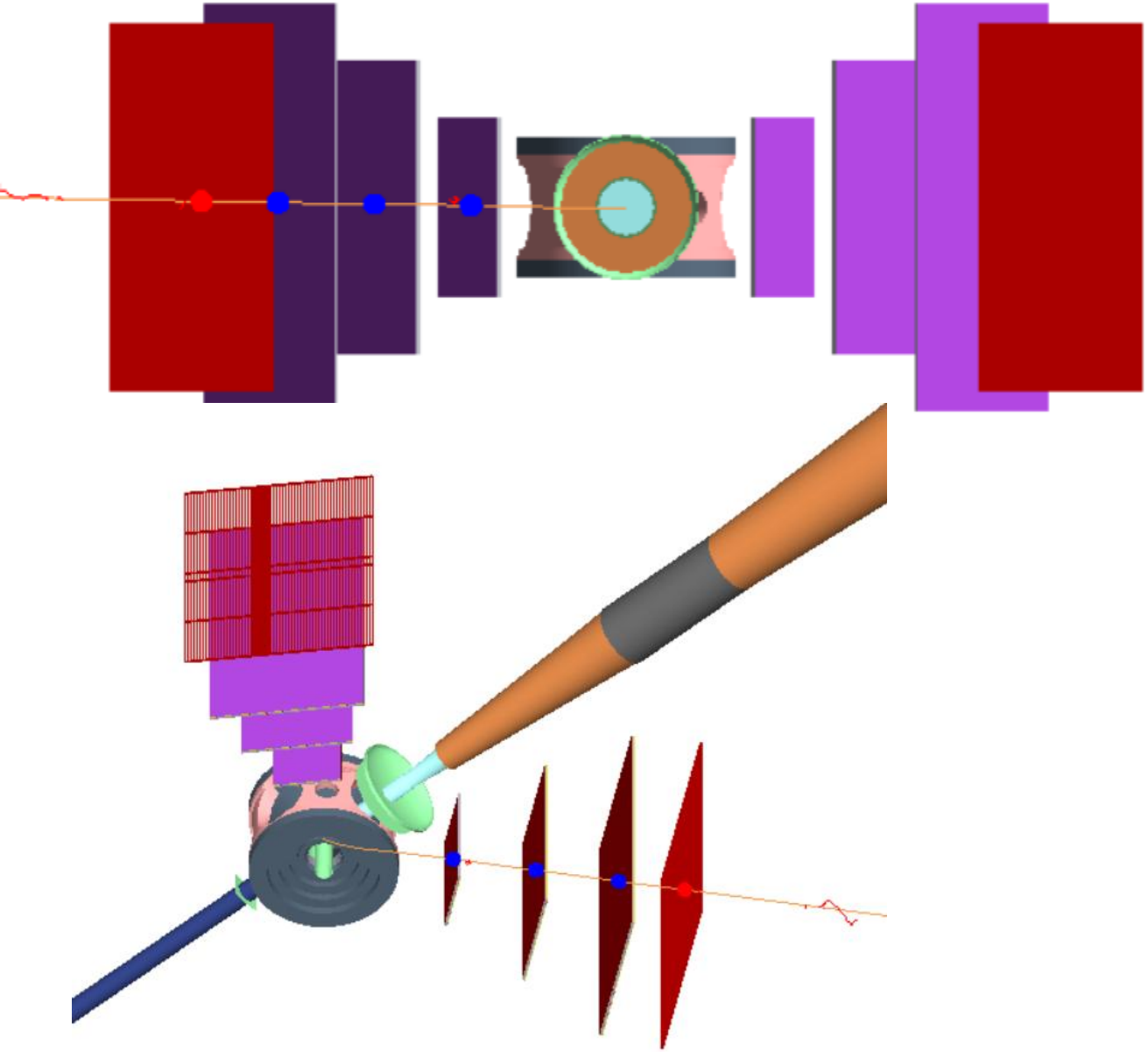
- Reconstruction in CJ: created and filled RECOIL::hits → RECOIL::clusters → RECOIL::crosses banks, again following the example of the high-lumi μRWell.

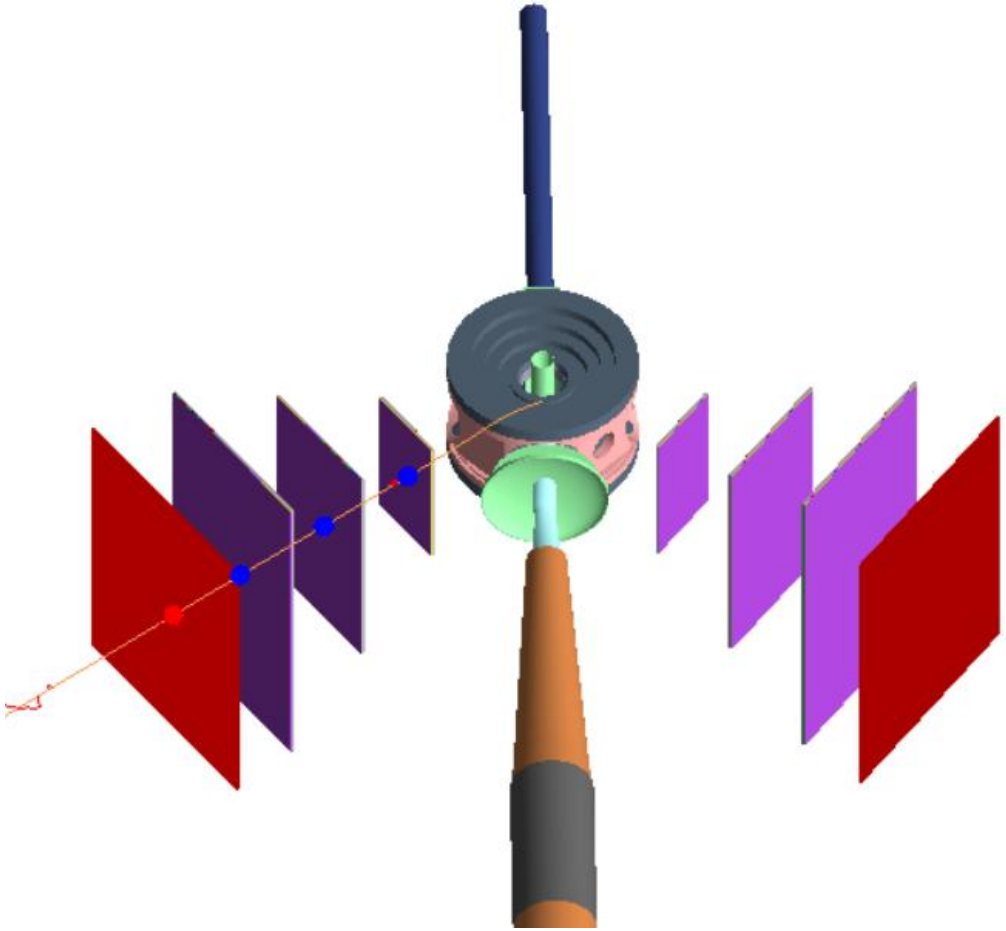
```
DataBank bankS = de.createBank("RTRK::hits", strips.size());
for(int h = 0; h < strips.size(); h++){
    bankS.setShort("id",        h, (short) strips.get(h).getId());
    bankS.setByte("sector",     h,  (byte) strips.get(h).getDescriptor().getSector());
    bankS.setByte("layer",      h,  (byte) strips.get(h).getDescriptor().getLayer());
    bankS.setShort("strip",     h, (short) strips.get(h).getDescriptor().getComponent());
    bankS.setFloat("energy",    h, (float) strips.get(h).getEnergy());
    bankS.setFloat("time",      h, (float) strips.get(h).getTime());
    bankS.setShort("status",    h, (short) strips.get(h).getStatus());
    bankS.setShort("clusterId", h, (short) strips.get(h).getClusterId());
}
```

```
DataBank bankX = de.createBank("RTRK::crosses", crosses.size());
for(int c = 0; c < crosses.size(); c++){
    bankX.setShort("id",        c, (short) crosses.get(c).getId());
    bankX.setByte("sector",     c,  (byte) crosses.get(c).getSector());
    bankX.setByte("region",     c,  (byte) crosses.get(c).getRegion());
    bankX.setFloat("energy",    c, (float) crosses.get(c).getEnergy());
    bankX.setFloat("time",      c, (float) crosses.get(c).getTime());
    bankX.setFloat("x",         c, (float) crosses.get(c).point().x());
    bankX.setFloat("y",         c, (float) crosses.get(c).point().y());
    bankX.setFloat("z",         c, (float) crosses.get(c).point().z());
    bankX.setShort("cluster1",  c, (short) crosses.get(c).getCluster1());
    bankX.setShort("cluster2",  c, (short) crosses.get(c).getCluster2());
    bankX.setShort("status",    c, (short) crosses.get(c).getStatus());
}
```

```
DataBank bankC = de.createBank("RTRK::clusters", clusters.size());
for(int c = 0; c < clusters.size(); c++){
    bankC.setShort("id",        c, (short) clusters.get(c).getId());
    bankC.setByte("sector",     c,  (byte) clusters.get(c).get(0).getDescriptor().getSector());
    bankC.setByte("layer",      c,  (byte) clusters.get(c).get(0).getDescriptor().getLayer());
    bankC.setShort("strip",     c, (short) clusters.get(c).getMaxStrip());
    bankC.setFloat("energy",    c, (float) clusters.get(c).getEnergy());
    bankC.setFloat("time",      c, (float) clusters.get(c).getTime());
    bankC.setFloat("xo",        c, (float) clusters.get(c).getLine().origin().x());
    bankC.setFloat("yo",        c, (float) clusters.get(c).getLine().origin().y());
    bankC.setFloat("zo",        c, (float) clusters.get(c).getLine().origin().z());
    bankC.setFloat("xe",        c, (float) clusters.get(c).getLine().end().x());
    bankC.setFloat("ye",        c, (float) clusters.get(c).getLine().end().y());
    bankC.setFloat("ze",        c, (float) clusters.get(c).getLine().end().z());
    bankC.setShort("size",      c, (short) clusters.get(c).size());
    bankC.setShort("status",    c, (short) clusters.get(c).getStatus());
}
```

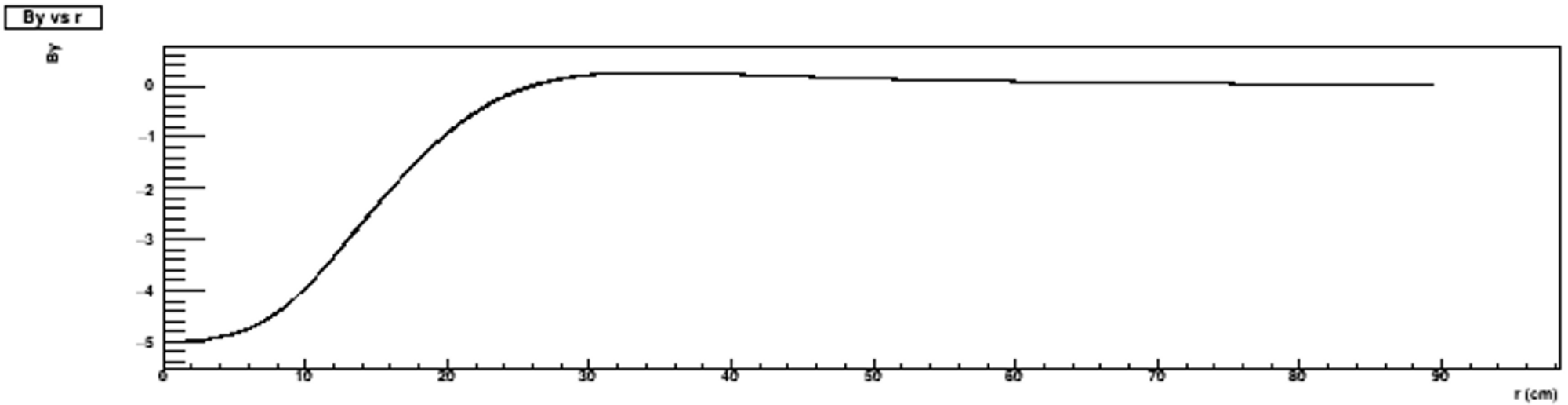
Inclusion of TOF in the recoil simulation and reconstruction





TOF gemc and CJ implementation done by Nilanga
Images from early implementations

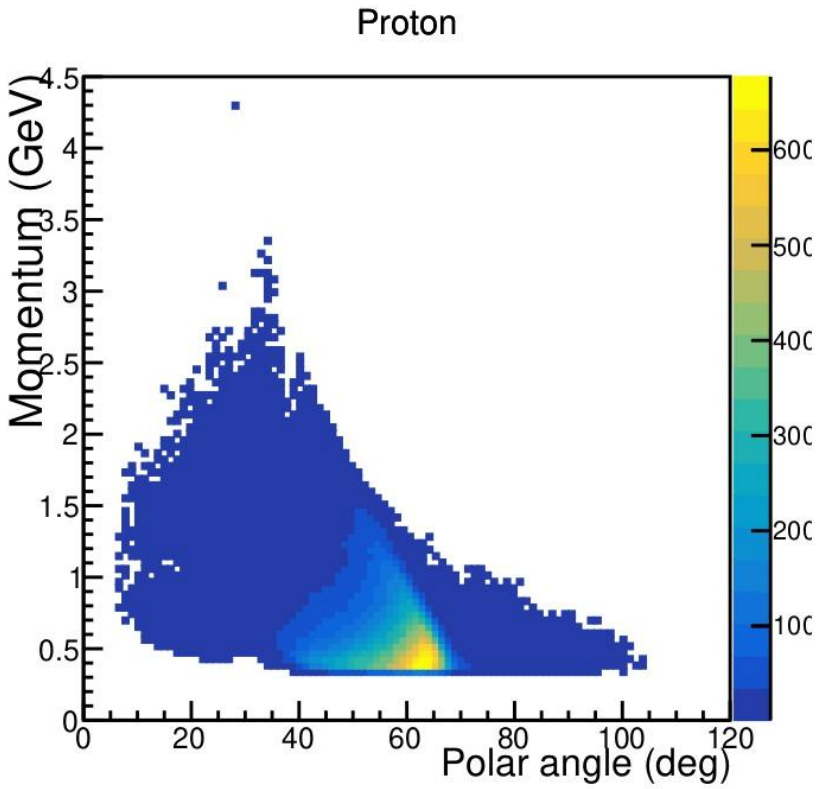
Field map (gemc version)

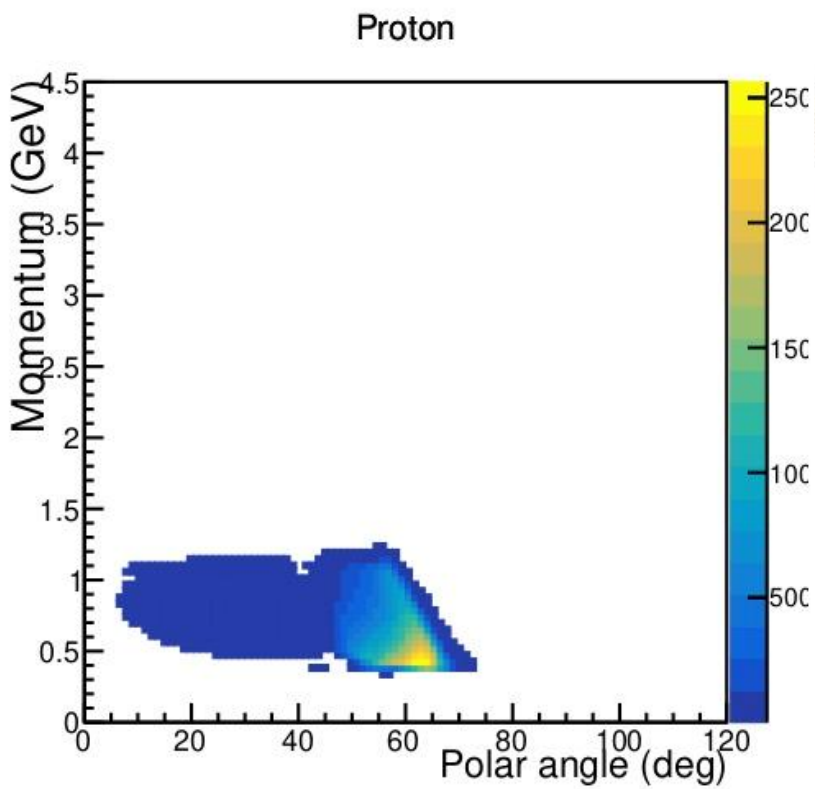


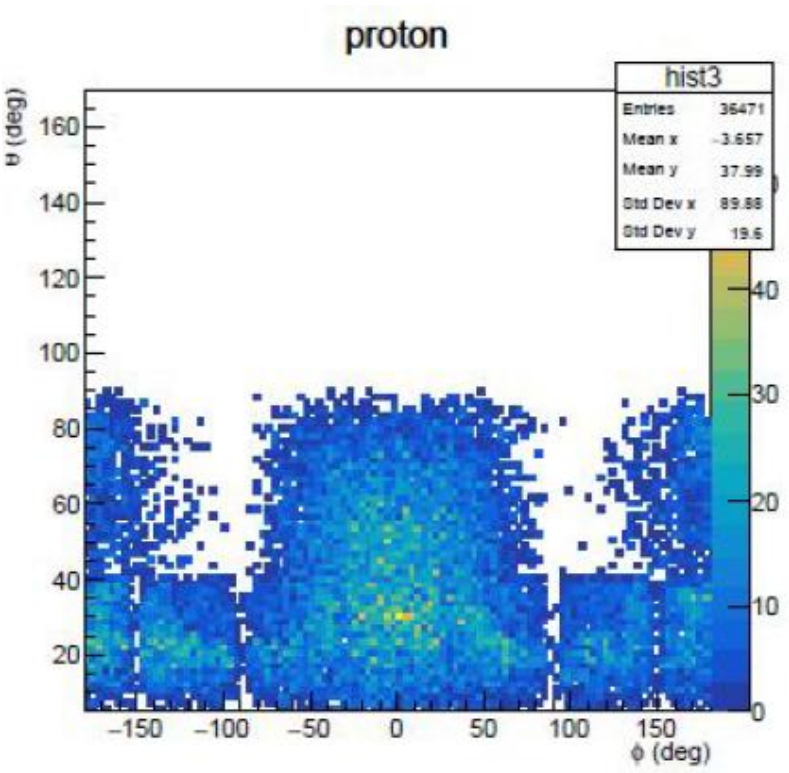
The y component of the field decreases quickly as a function of the radius, drops to 0 at r ~ 30 cm
The x and z components of the field are negligible

DVCS proton kinematics in RGH

J.S. Alvarado's analysis





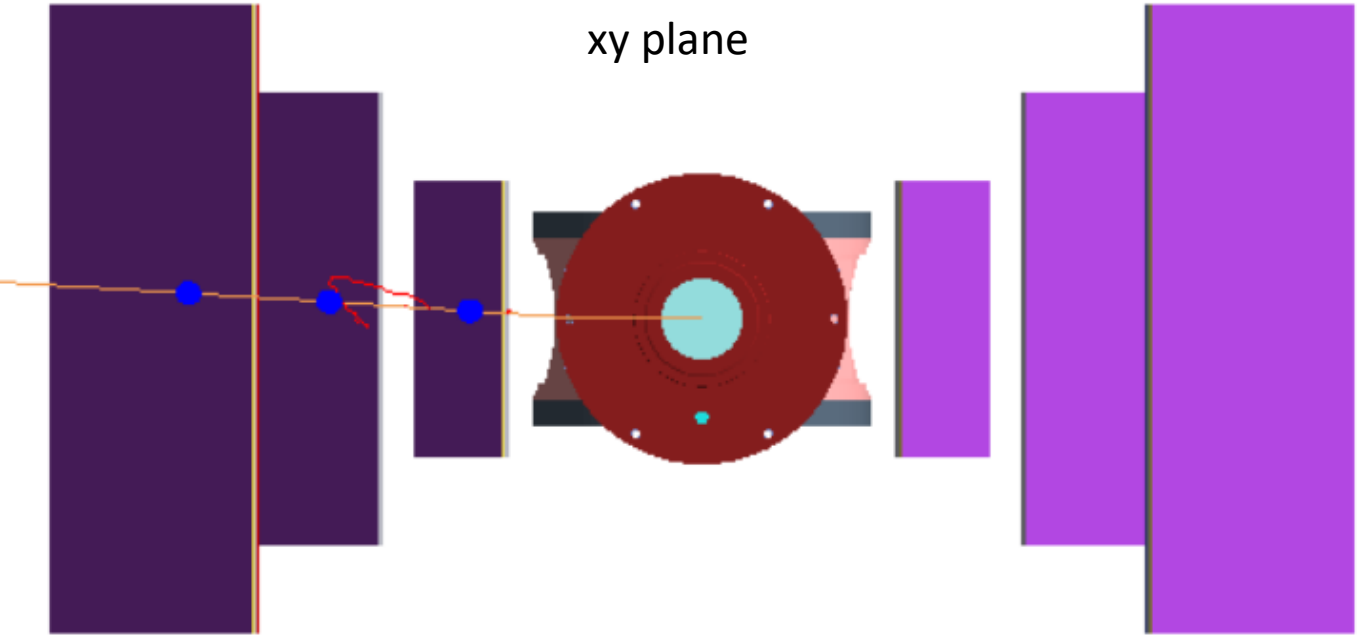


RGA data

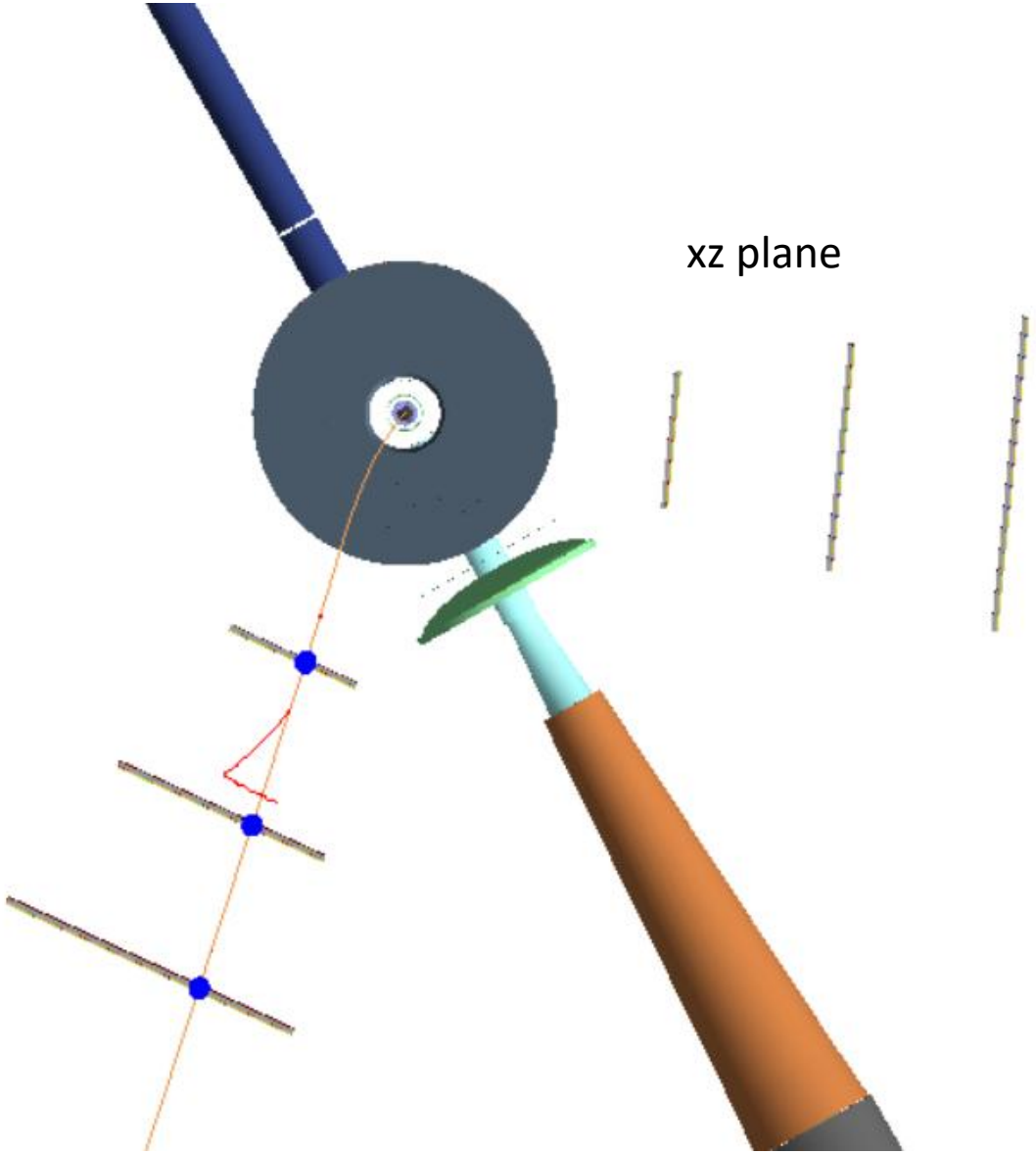
MC rec (dvcsgen) in RGH acceptance (FD+recoil)

P = 0.6 GeV, θ=70°, ϕ=180°

xy plane



xz plane

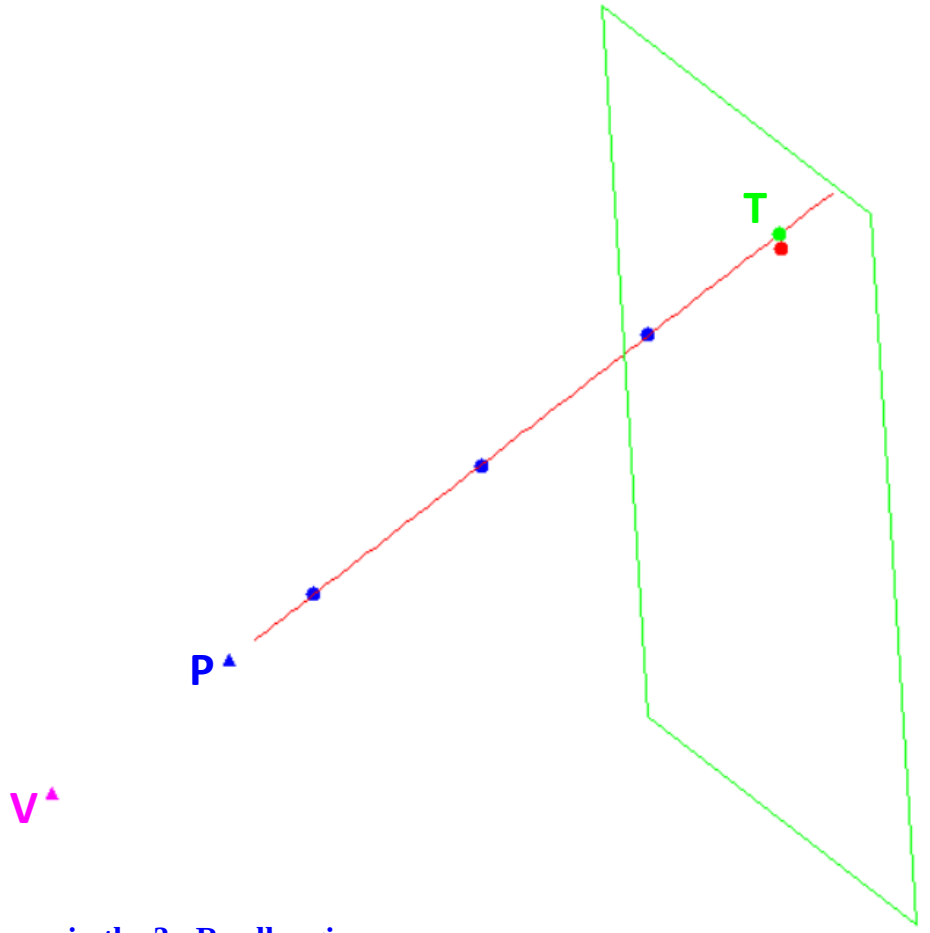


The bending is only in the xz plane
The magnetic field in the recoil region is so weak that linear tracking can be adopted

« Tracking »/reconstruction strategy (used for the DVCS proposal)

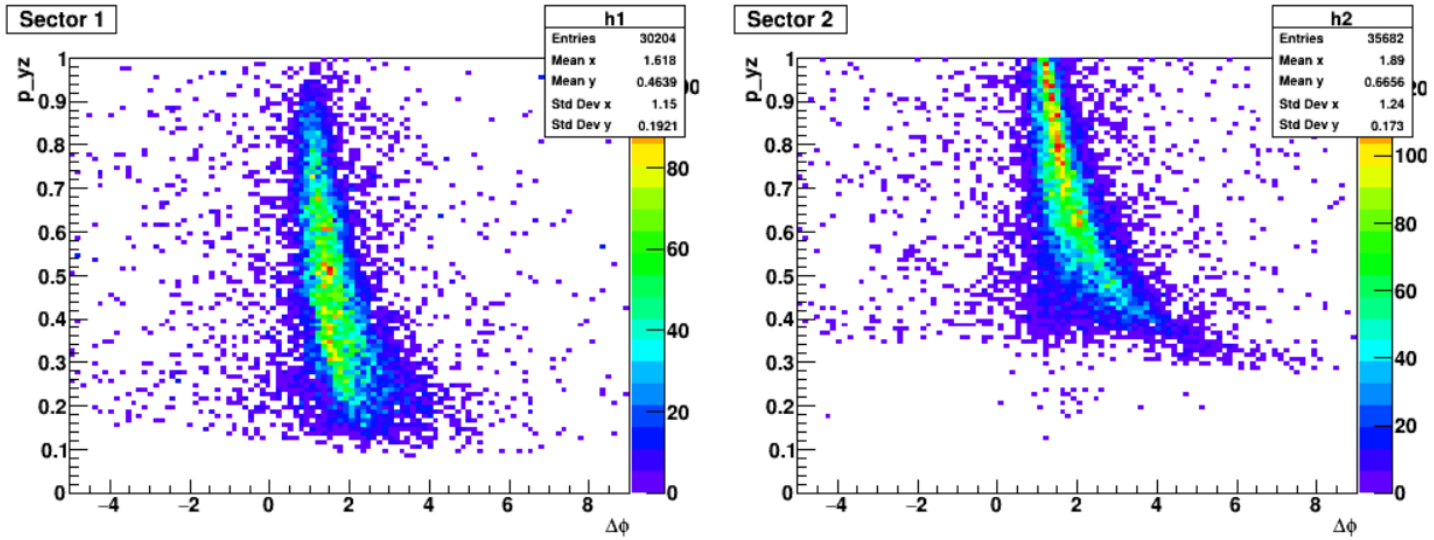
- Loop over all the crosses of each μRwell layer
- Fit a straight line through each combination of 3 crosses
- Choose the fitted line that gives the best chi2 and is closest to the interaction vertex (obtained from the forward tracking)
- The intersection ”T” of the fitted line with the TOF panel is determined
- The closest TOF cluster to T is found, and its timing is taken to compute β
- For the path-length calculation, the intersection P of the fitted line with a cylinder, coaxial with the target magnet, of radius 30 cm is determined (this radius corresponds to where the target magnetic field starts to become non-negligible)
- The last piece of trajectory is approximated with the segment VP, connecting the interaction vertex V and P
- The path length is defined as the sum of the two segments TP and VP
- β is computed combining the path length and the measured time-of-flight (knowing the electron start time)
- The **absolute value of the momentum** was deduced from β assuming that the track belonged to a proton

To handle background, added time matching between the 3 crosses

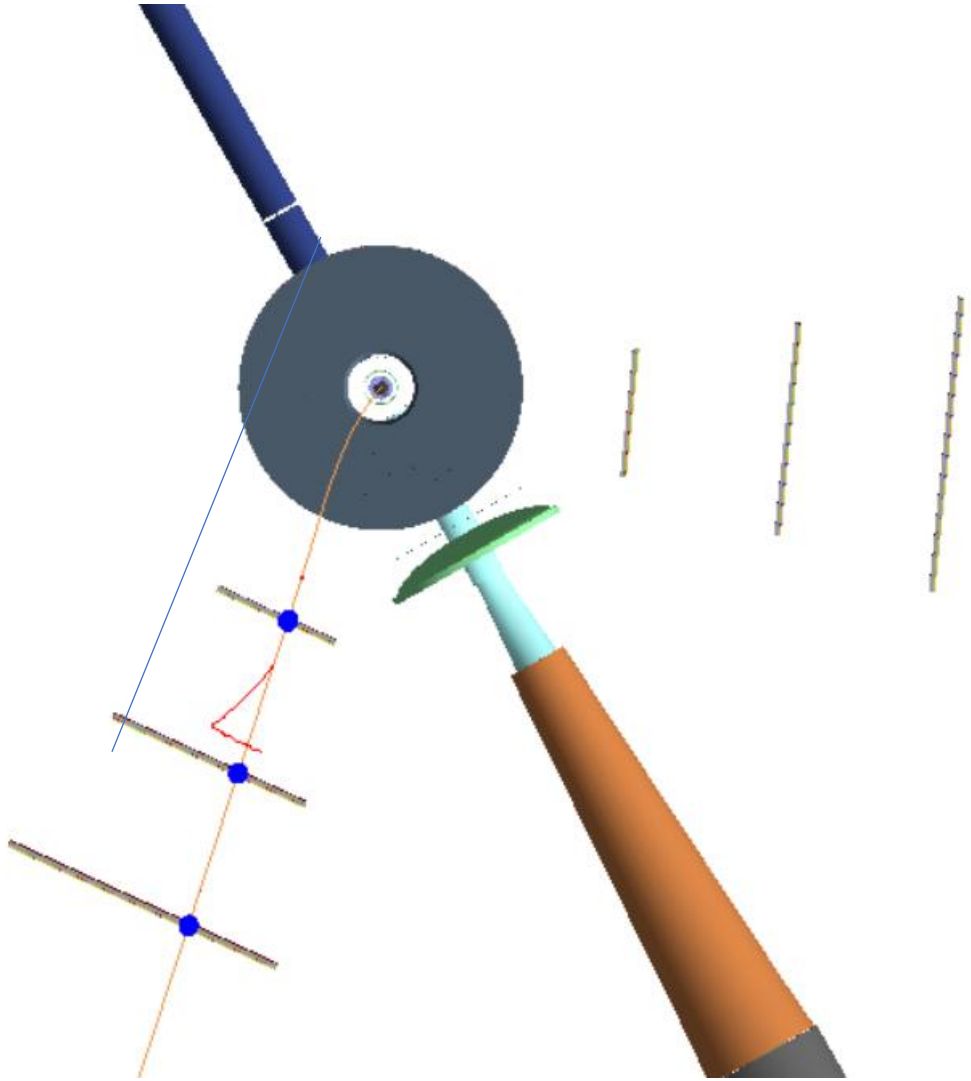


Crosses in the 3 μRwell regions
Line: linear fit to the three crosses
T: Intersection point of the trajectory with TOF
Point in TOF closest to the fitted track
P: intersection point of the trajectory with a 30-cm radius cylinder
V: Electron vertex from FD

Calculation of momentum components from trajectory deviation



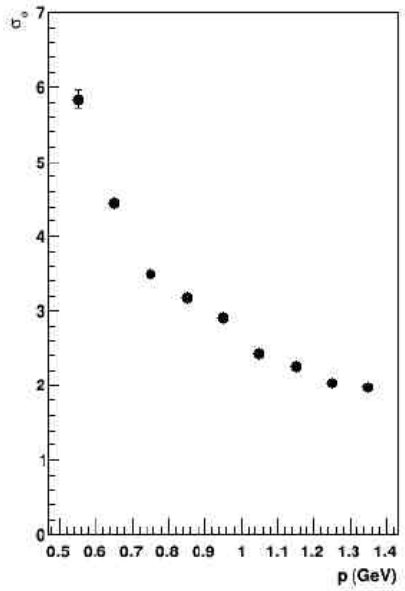
$\Delta\Phi = \Phi_0 - \Phi_1$
Φ_0 = atan2(vertexToCross.Z(), vertexToCross.X());
Φ_1 = atan2(bestDirectionForSector.Z(), bestDirectionForSector.X())

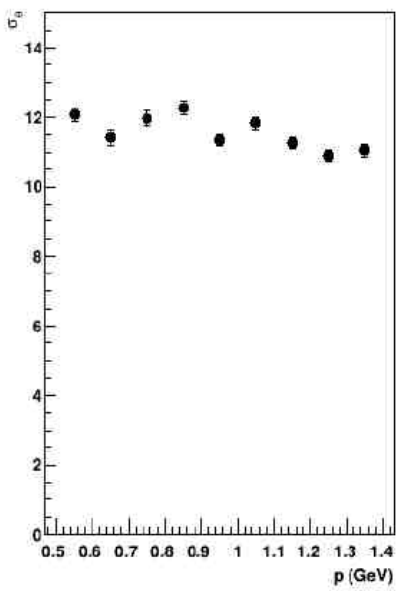


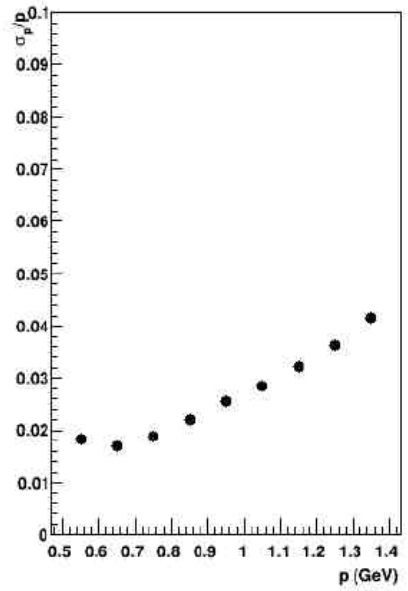
- We observed that the track does not bend in the xy plane, and therefore **the angle ϕ** can be obtained from the xy position of the Region-1 cross
- We found that there is a one-to-one correspondence between the yz components of the momentum and the angle between the fitted track and the line joining the interaction vertex to the cross in Region 1
- This dependence is parametrized with a polynomial fit and used to deduce the momentum yz component
- Knowing ϕ, p_{yz}, and p, **the other momentum components and the angle θ** are computed

Obtained angular and momentum resolutions

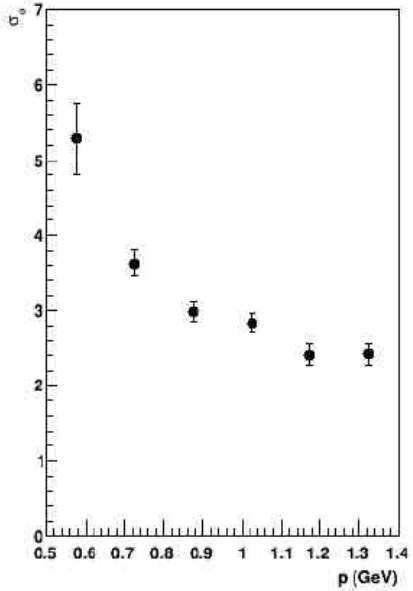
Without background

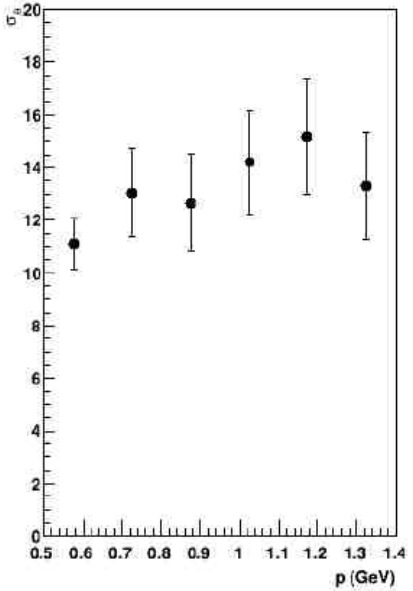


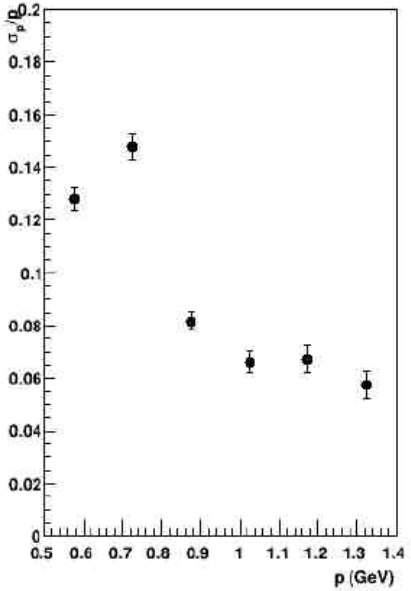




With merged background







The many approximations in the reconstruction coupled to the non-ideal magnetic field conditions yield poor resolution performances, especially in θ

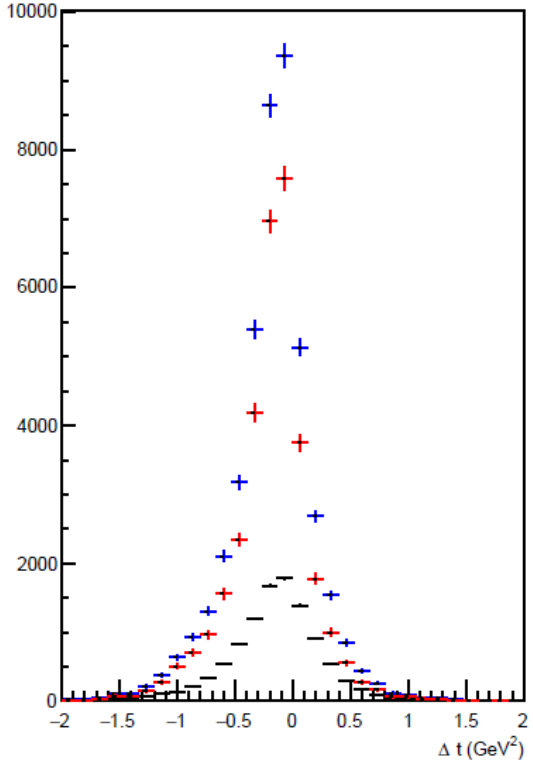
→ **See Maggie's talk** for the improved performances adopting neural-nets based reconstruction

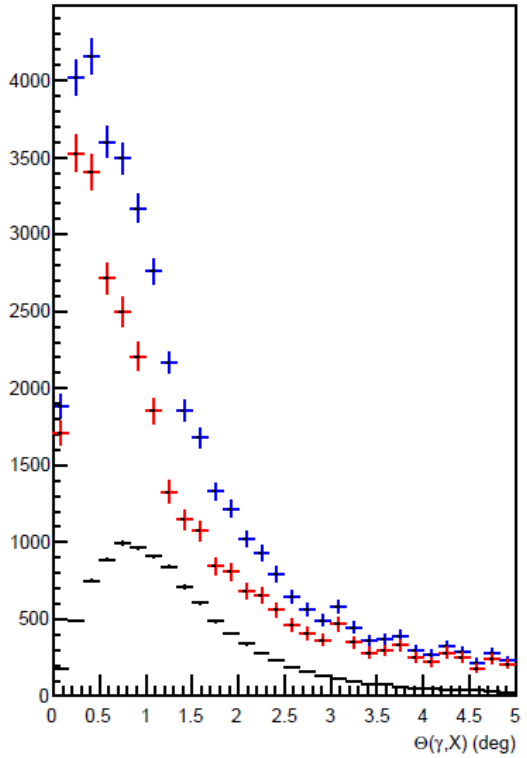
π^0 background evaluation for DVCS

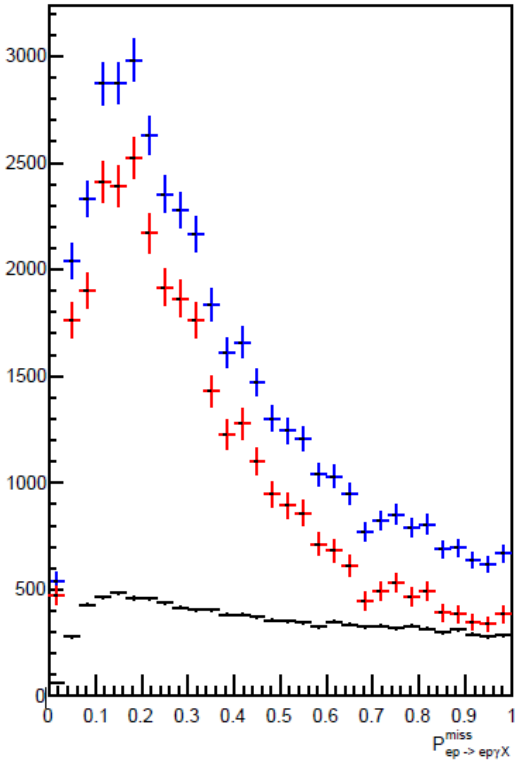
Evaluation of the π^0 background for DVCS events in the RGH setup :

- Monte-Carlo exclusive $ep\pi^0$ events produced on an NH3 target are generated and passed through GEMC and reconstruction chain, for both the RGH and the RGA configurations.
- The π^0 yield is obtained scaling the RGA $ep\pi^0$ experimental yields by the ratio of acceptances of RGH and RGA times the luminosity ratio.

Projected yields in RGH conditions for DVCS, $ep\pi^0$, DVCS+π^0





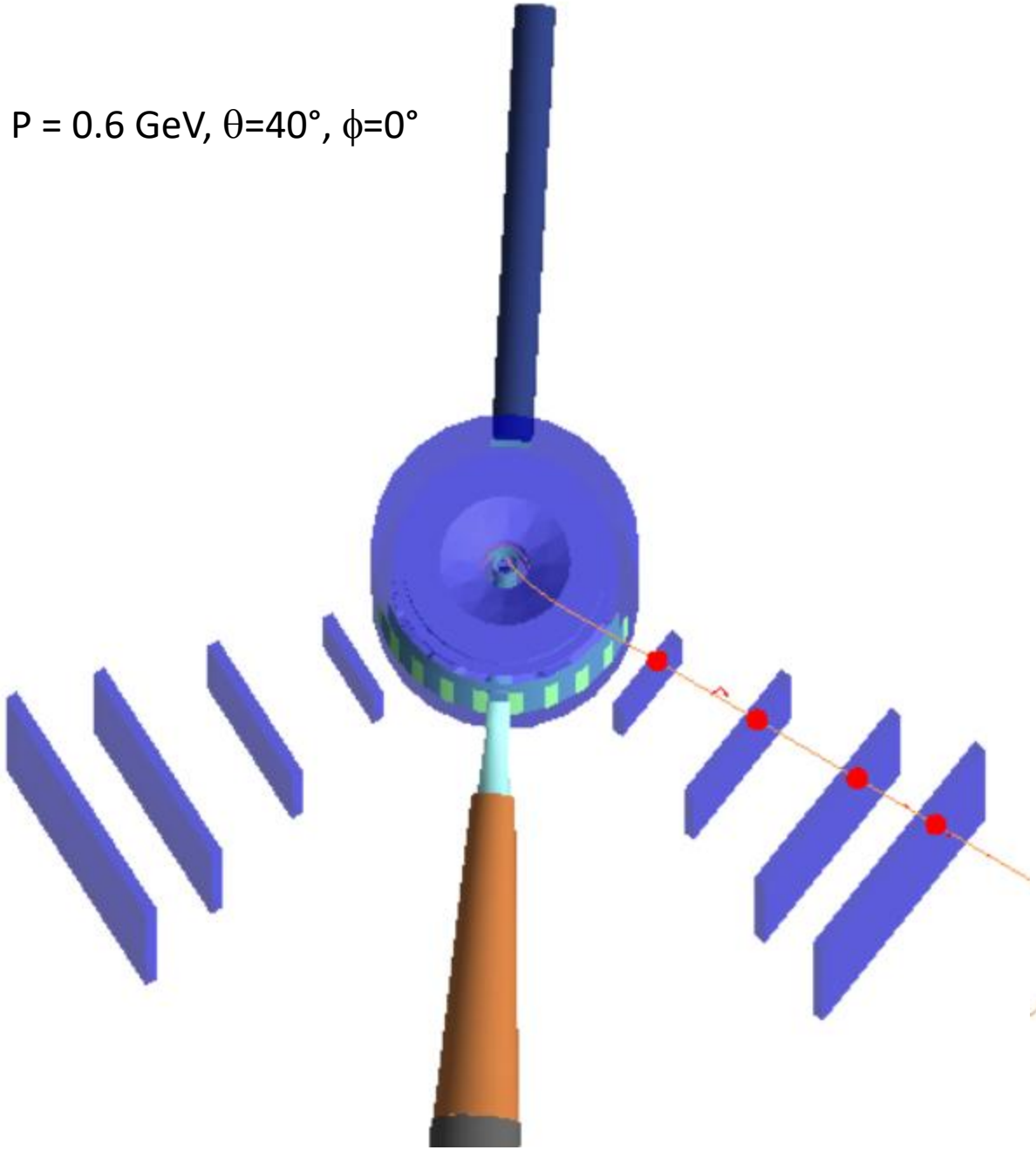


Exclusivity cuts:

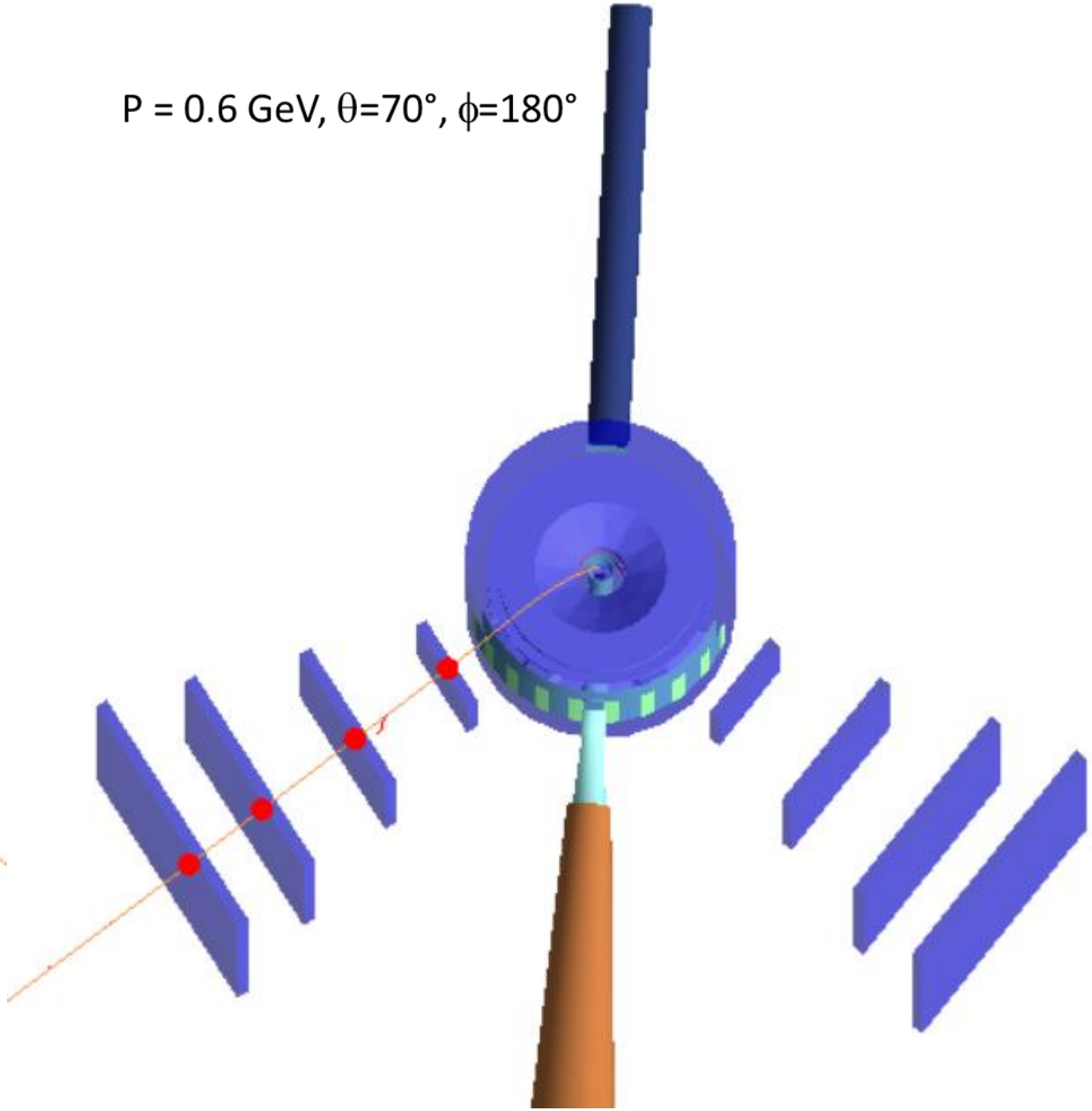
$E_\gamma > 2$, $Q^2 > 1.0$ GeV^2, $W > 2$ GeV, $p_p > 0.3$ GeV, $p_e > 1.0$ GeV, $\Delta\phi < 1.5°$, $\Delta t < 2$ GeV^2, p_X ($ep \rightarrow ep\gamma X$) < 1 GeV, $\theta_{\gamma X} < 5°$

By comparing the black and blue histograms within the regions of the DVCS selection cuts, the **π^0 contamination is roughly ~30%,** consistent with other DVCS experiments at CLAS12

P = 0.6 GeV, θ=40°, ϕ=0°

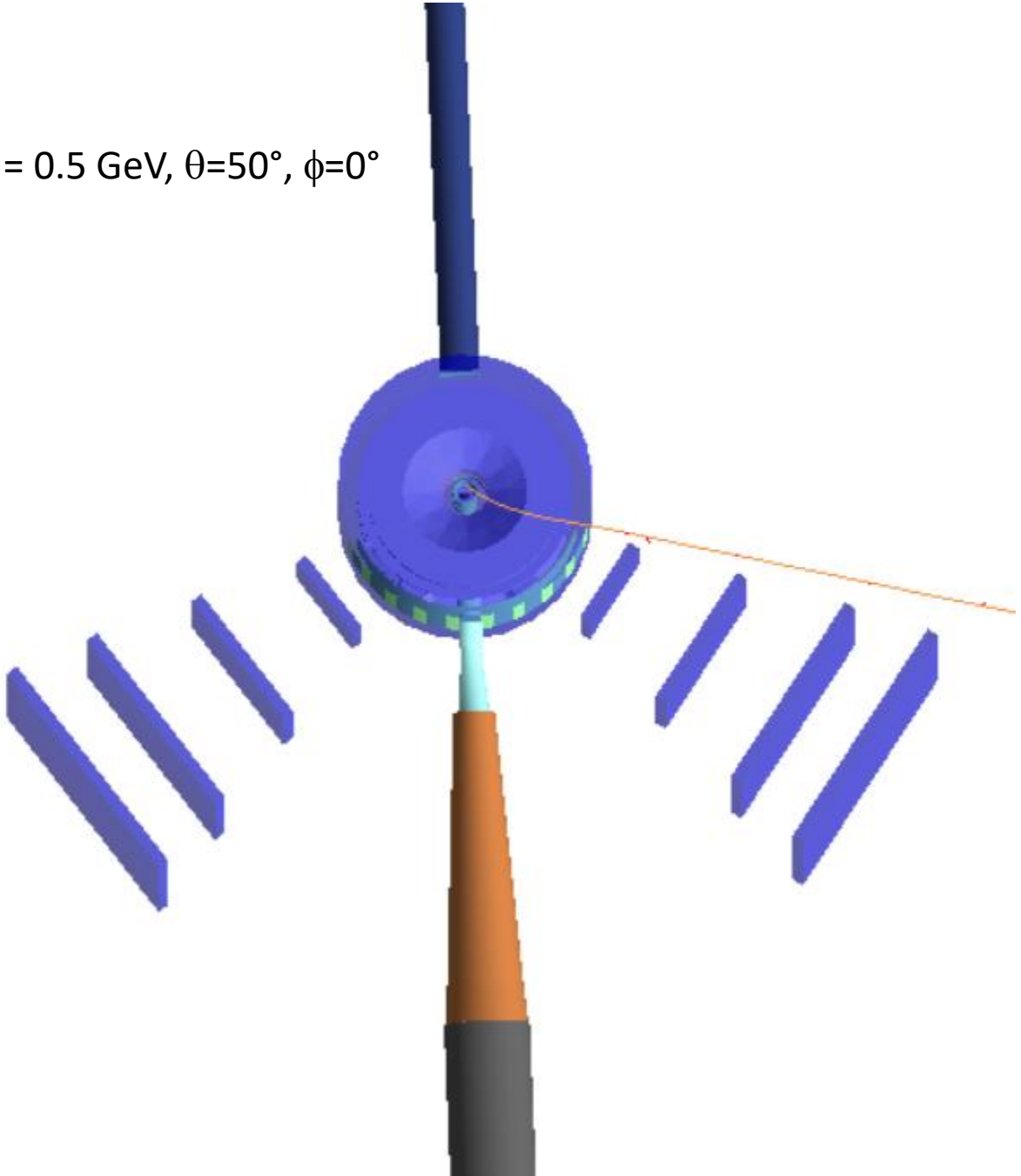


P = 0.6 GeV, θ=70°, ϕ=180°

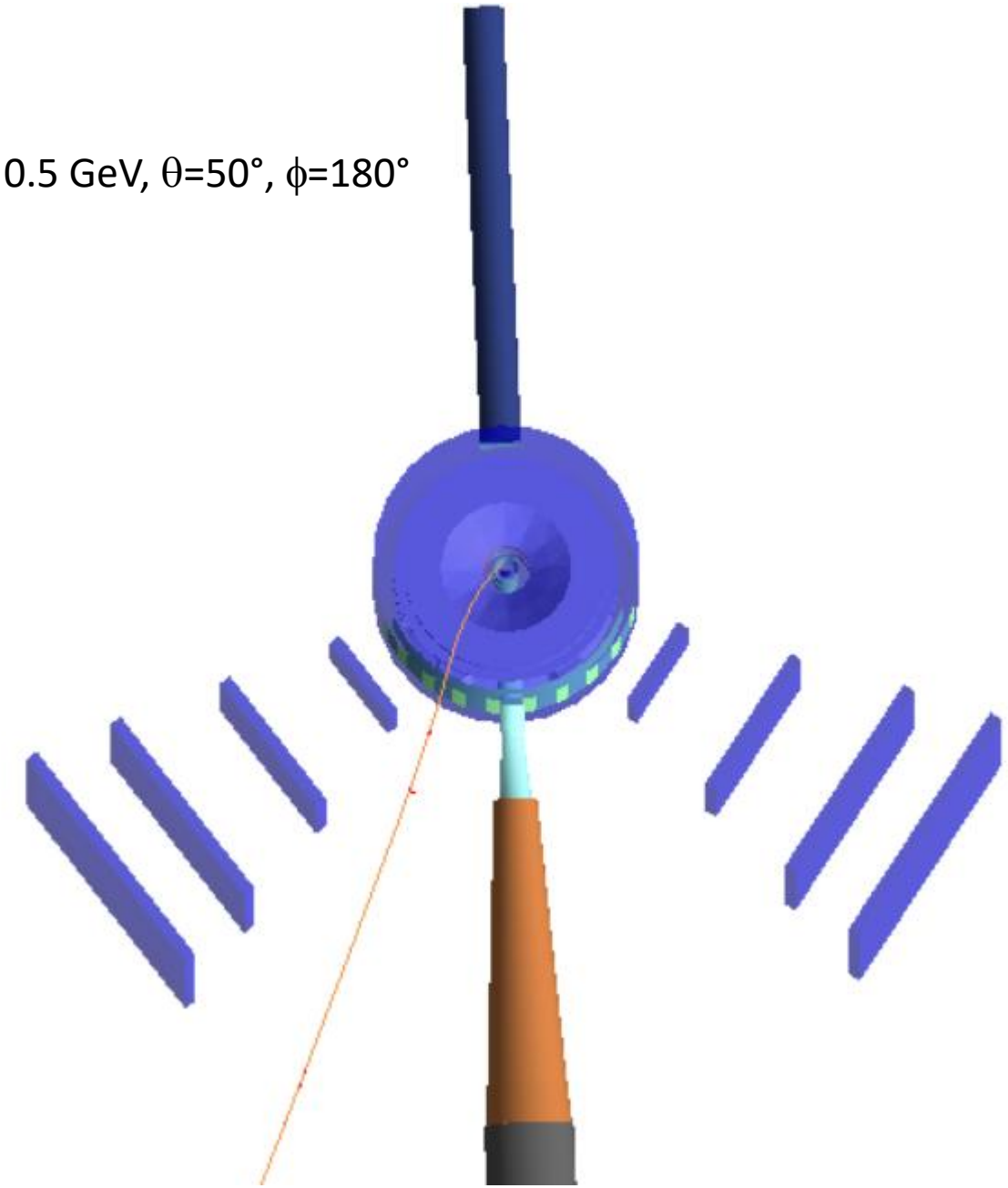


Can a different angular positioning of the two sectors improve the acceptance for DVCS protons? See Maggie's talk

P = 0.5 GeV, θ=50°, ϕ=0°



P = 0.5 GeV, θ=50°, ϕ=180°



Recent modifications in CJ/Gemc (work in progress)

Following Raffaella's request, the Tracker and TOF components of the recoil software were put in the same directory (before it was recoil/ and recoil_tof/)
Changes implemented by my and Nilanga in CJ geometry server, CJ reconstruction and getmc digitization
To do: implement this new structure in the gemc/detector scripts to create the geometry/materials files for gemc

coatjava / reconstruction / recoil / src / main / java / org / jlab / service / **recoil** /

c-dilks fix: javadoc warnings (#1332) ✕ c628fa3 · 4 months ago History

Name	Last commit message	Last commit date
..		
tof	fix: javadoc warnings (#1332)	4 months ago
trk	RGH recoil (trk and tof) (#1312)	5 months ago

coatjava / reconstruction / recoil / src / main / java / org / jlab / service / recoil / **trk** /

3 people RGH recoil (trk and tof) (#1312) ✓ 243ba98 · 5 months ago History

Name	Last commit message	Last commit date
..		
RTRKCluster.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTRKCross.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTRKEngine.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTRKParameters.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTRKStrip.java	RGH recoil (trk and tof) (#1312)	5 months ago

coatjava / reconstruction / recoil / src / main / java / org / jlab / service / recoil / **tof** /

c-dilks fix: javadoc warnings (#1332) ✕ c628fa3 · 4 months ago History

Name	Last commit message	Last commit date
..		
ClusterFinder.java	fix: javadoc warnings (#1332)	4 months ago
HitFinder.java	RGH recoil (trk and tof) (#1312)	5 months ago
Parameters.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTOFCluster.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTOFEngine.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTOFHit.java	RGH recoil (trk and tof) (#1312)	5 months ago
RTOFRawHit.java	RGH recoil (trk and tof) (#1312)	5 months ago
RecoBankWriter.java	RGH recoil (trk and tof) (#1312)	5 months ago

Conclusions/outlook

The recoil detector (tracker+TOF) is fully implemented in Gemc and Coatjava (banks)
The geometry is flexible and easy to modify
A new implementation (only cosmetic fixes) is nearly complete

An offline, approximate proton reconstruction was developed for the DVCS RGH proposal

The simulation and this reconstruction allowed to demonstrate the feasibility of the DVCS TTSA measurement, produce projections, and obtain PAC approval

Next steps:

- Establish the optimal geometry/position, keeping into account simulation results by Maggie and hardware constraints
- Implement neural-nets-based reconstruction and PID (see Tong Tong and Maggie's talks), including background → EB?
- Elastic simulation to verify feasibility of PbPt measurement