



Simulation and Background Developments

Lindsey Gray / Dimitris Ntounis / Elias Mettner / et al.

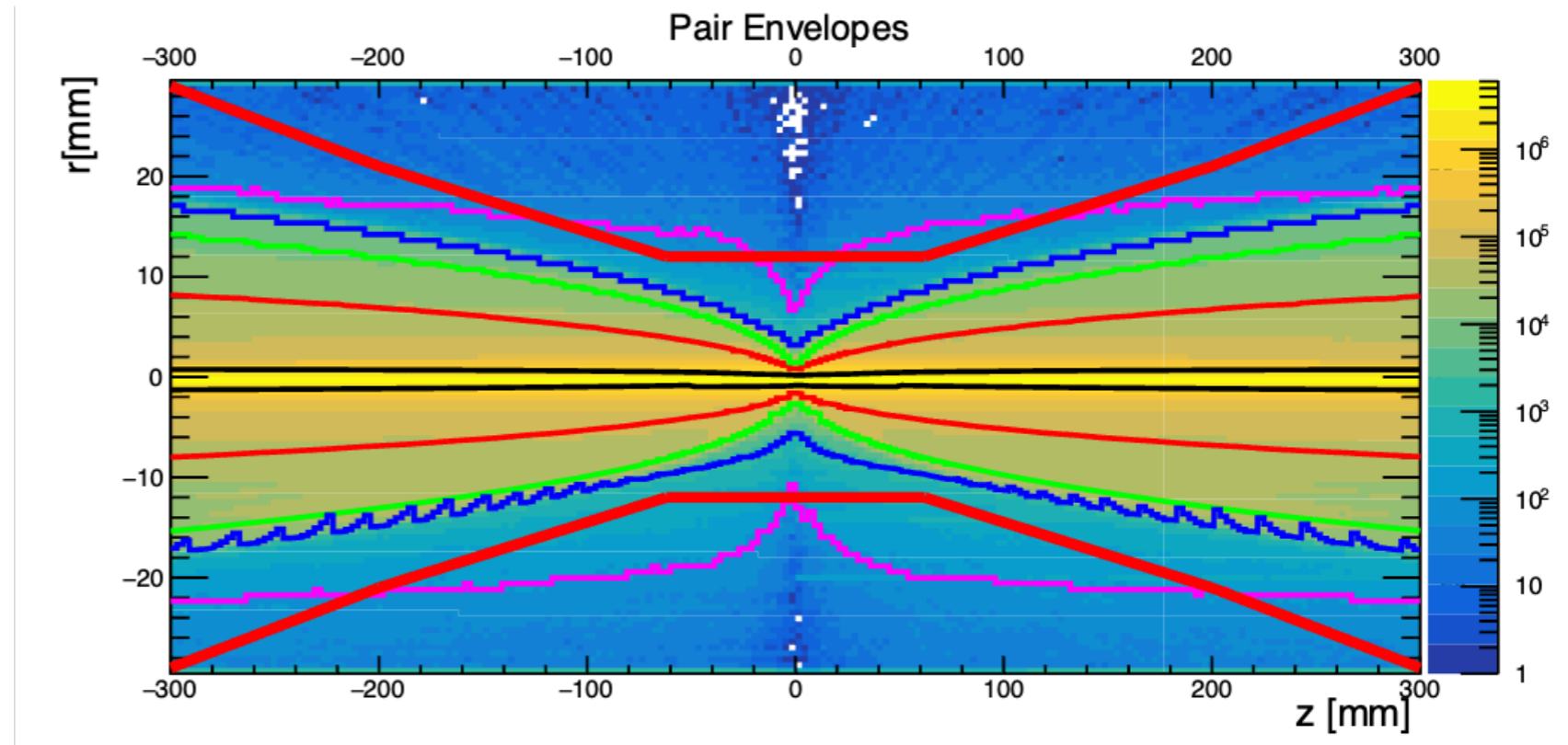
C3 Workshop @ LANL

6 February 2023

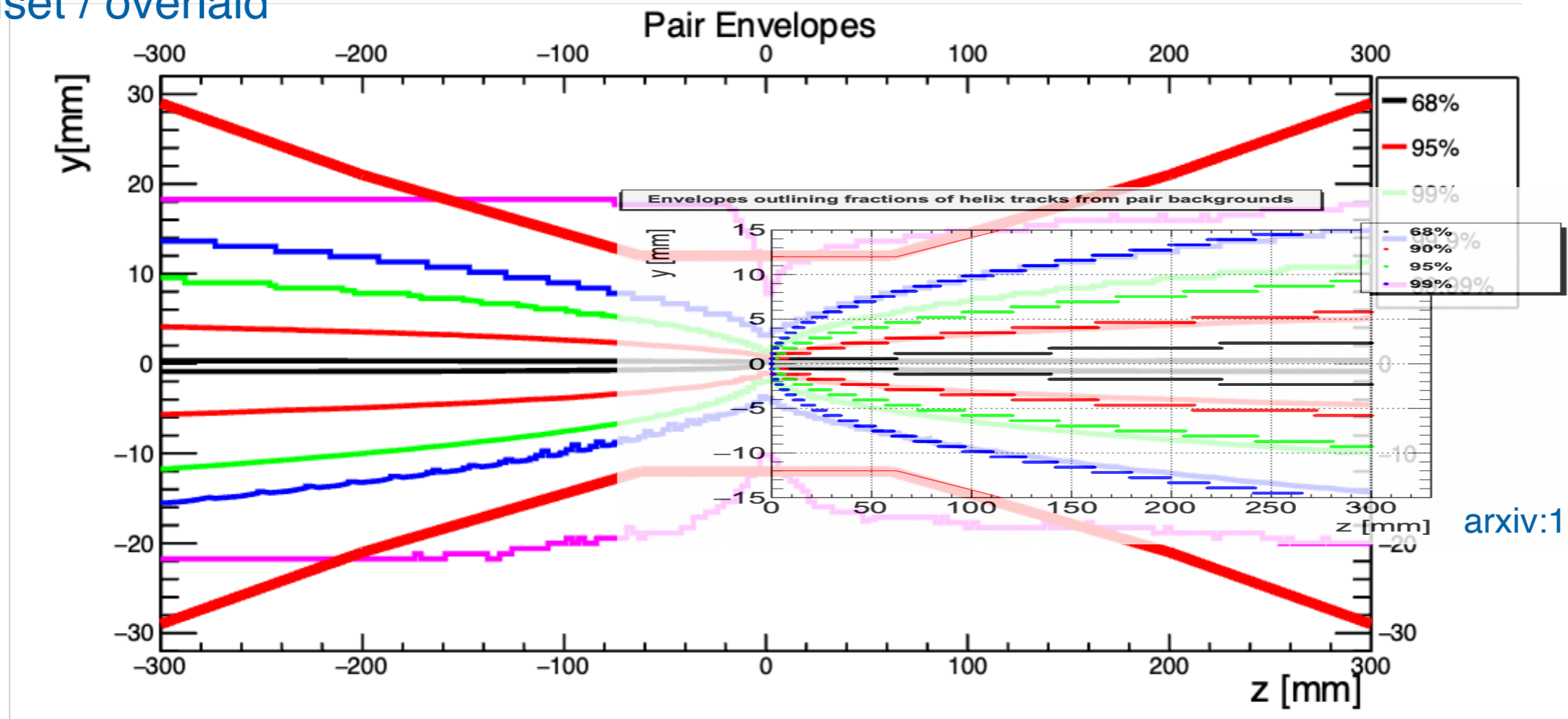
Items to Discuss

- A bit shorter this time - some churn is occurring / pieces moving into place
- Further information about odd features of simulation in pair background
 - Some resolution on strange simulated hits
 - Updating pair background simulation?
- Hadrons / muon backgrounds
- Software strategy / collaboration with muC et al.

Reminder!



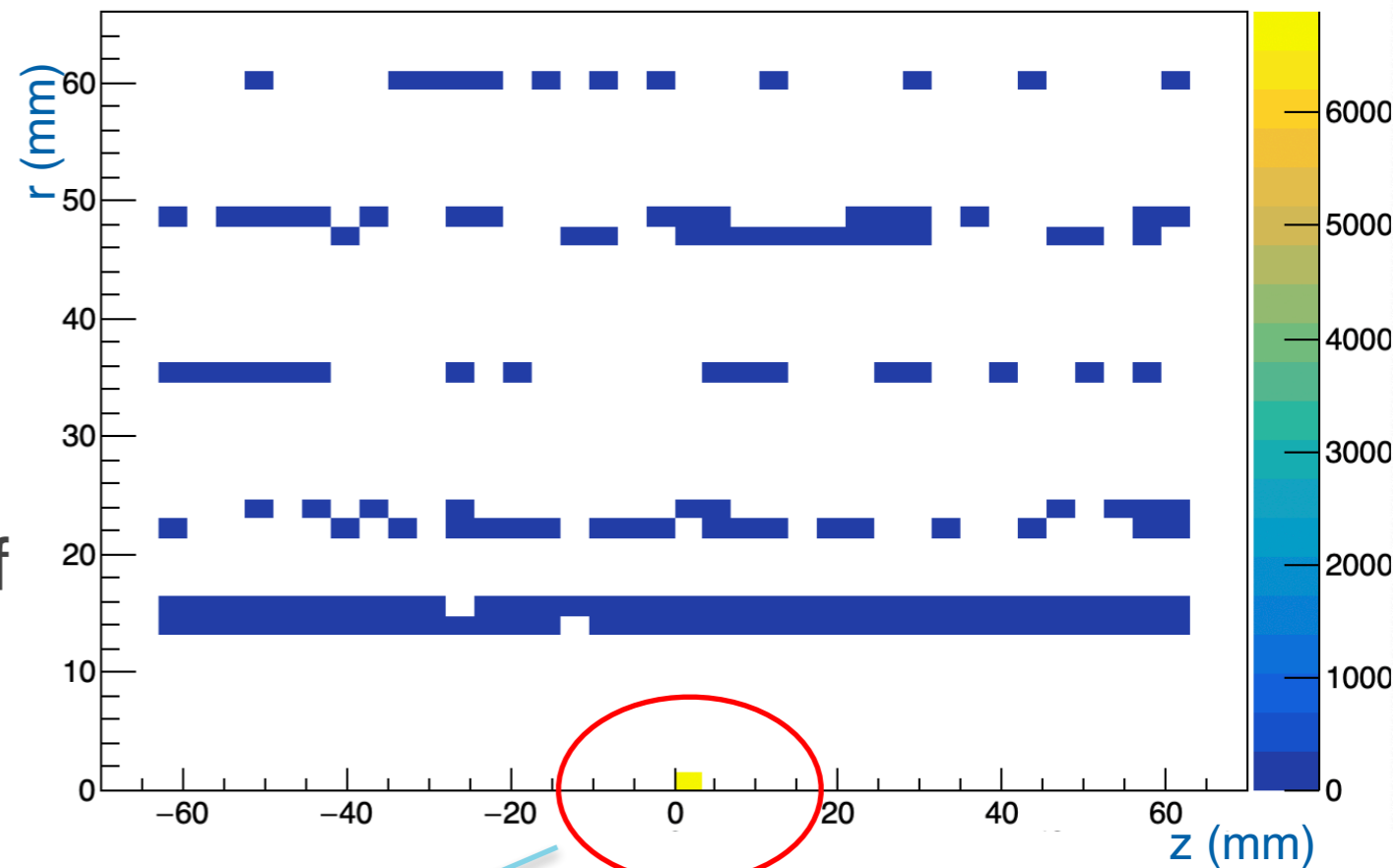
ILC inset / overlaid



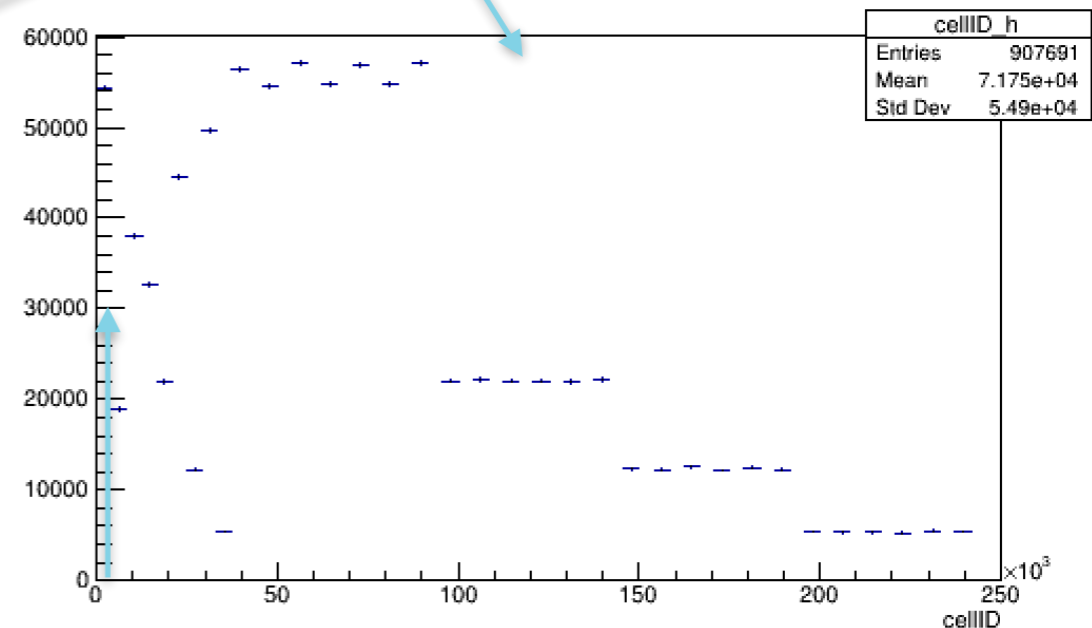
arxiv:1703.05737

Pair Production Background - Simulation Oddities

- Pathological “hits at zero” appear to have two root causes
 - Overlaps in geometry
 - Inability to assign detector ID from physical crossing point
- These are bugs in the description of geometry not caught by overlap checks
 - High “flux” always yields surprises
- To follow up with geometry maintainers

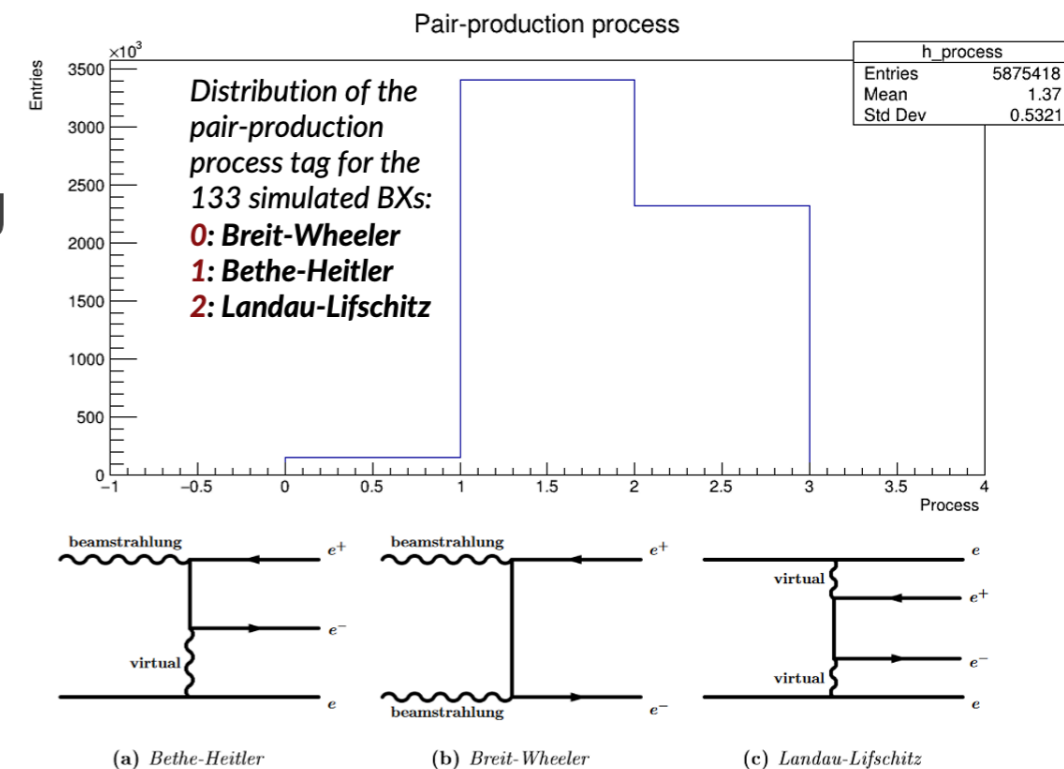


```
EDeposit: 0.0  
EDeposit: 0.0  
EDeposit: 0.0  
EDeposit: 0.0  
EDeposit: 0.0  
EDeposit: 0.0  
EDeposit: 0.0  
EDeposit: 0.0  
EDeposit: 0.0
```



Pair Background Simulation Evolution

- Spurred by a side conversation with Peskin at last meeting, pursued alternative, modern simulations of pair background production
 - In particular, the conversation was about understanding uncertainties
 - C3 bunch charge less than ILC but still best to understand uncertainties in time
 - A gradual update of software related LCs is a good thing
- First thing to do is vary bunch charge in Guinea-Pig and ensure we are not on any “cliffs”
- A clear path to update towards is Ptarmigan (<https://github.com/tgblackburn/ptarmigan>)
 - Pair-background simulation targeting ICS
 - Production cross section and in-bunch transport simulation like guinea-pig
 - Includes updated, detailed calculation of Breit-Wheeler process
 - Provides a framework for integrating more processes more pertinent to LCs



Hadronic and Muon Backgrounds

- Hadronic background to be updated in Pythia 8 (last generation was 5.7!)
 - Pending some clarification with key4HEP and some tooling issues
 - Will proceed with cross-validating low level numbers in GEANT
 - Have events generating, need an LHE reader for raw GEANT
- Muon background archaeology has located the code
 - It's in fortran that won't compile with modern compiler infrastructure
 - I am taking some time to see if it can be brought into shape

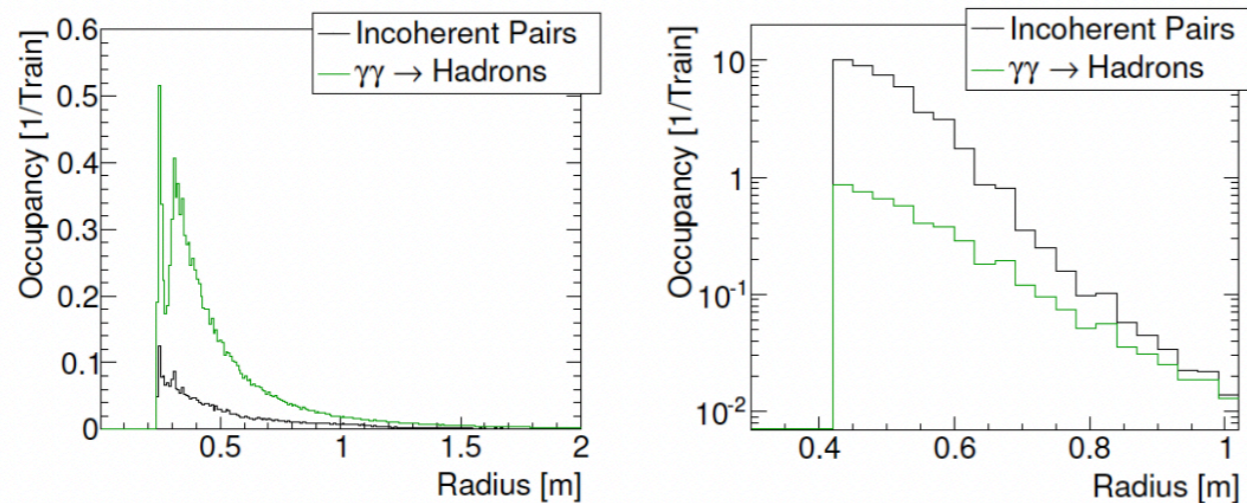


Figure 14. The radial distribution of the train occupancy per pad in ECal (left) and per cell in HCal (right) endcap [10].

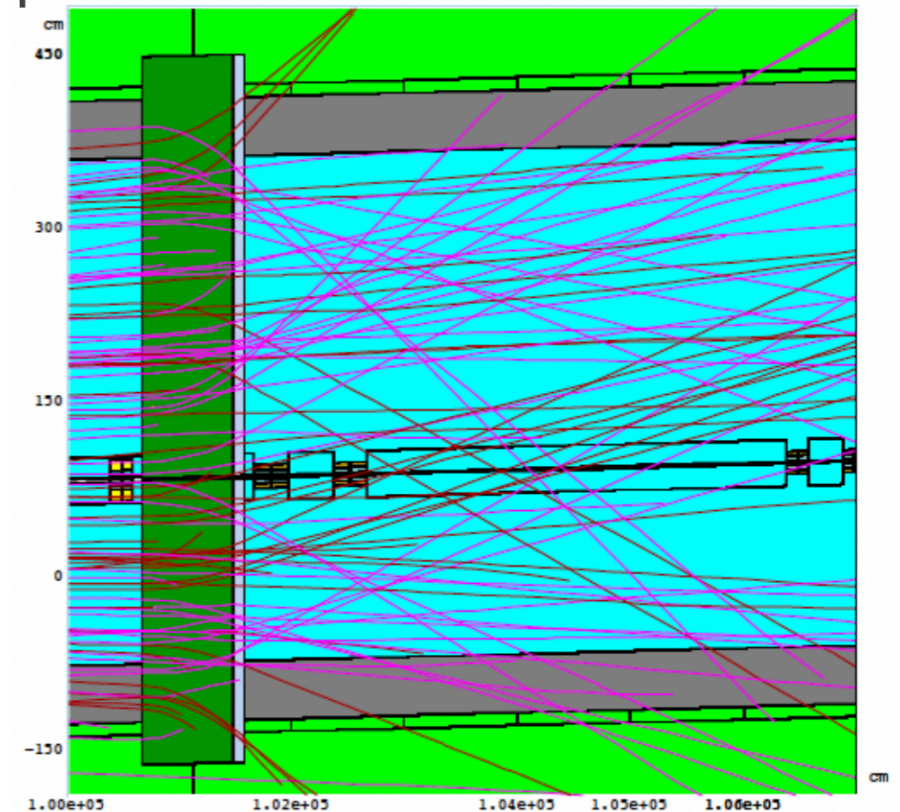


Figure 2. Muon tracks in the spoiler region [3].

Strategy with muC, et al. for software

- muC has already seen progress using key4hep
 - They have built up some analysis infrastructure based around flat ntuples (LCTuple)
 - This includes detector optimization studies
 - It makes little sense to reinvent the wheel, but it is also taking some time to discover what wheels have been built
 - Investment in common software tooling will pay dividends
 - Especially if our students / postdoc can move from one project to another without needing to incur overhead, it could be incredibly beneficial for everyone involved
 - We are not so many, it is better to plan for flexibility
- By summer will have this in place for C3 as well
 - Will work with muC software folks to make sure we converge in practices and general style
- In the mean time:
 - Transferred some physics and background samples to FNAL LPC to start exercising workflows
 - Developing analysis-level infrastructure further with muC folks

Conclusions / Next Step

- Still hammering out minutiae for detector simulation
 - Some errors yet to fix it seems, but extremely rare
- Background simulations proceeding onwards
 - Having more interested folks would be great!
 - There are many processes to hammer down
- Pursuing software framework with muC colleagues (and other LC/ FCC)
 - Aiming for lightweight tuple based analysis, including detector optimization studies
 - Datasets at FNAL to exercise processing workflow