Detector Simulation in SWIFT-HEP

Ben Morgan

WARWICK
THE UNIVERSITY OF WARWICK

Detector Simulation R&D @ March 2023

- **AdePT Project (CERN-SFT)**
  - *https://github.com/apt-sim*
- **Celeritas Project (ECP: ORNL, FNAL, Argonne, LBL)**
  - *https://github.com/celeritas-project*
- **Vecgeom/ORANGE Surface Based Geometry (CERN, Celeritas/ORNL)**
  - *https://gitlab.cern.ch/VecGeom/VecGeom* *(See surface_model branch)*
  - *https://github.com/celeritas-project/celeritas/tree/develop/src/orange*
  - ***ExaTEPP grant from UKRI ExCALIBUR enabling contribution here***
- Regular working and strategy meetings between projects,
- Can only give a shorter overview today, follow links for full details, together with the credits linked in the slides.
  - *See also later presentations from Davide and Andrei/Seth!*

Objectives of AdePT and Celeritas

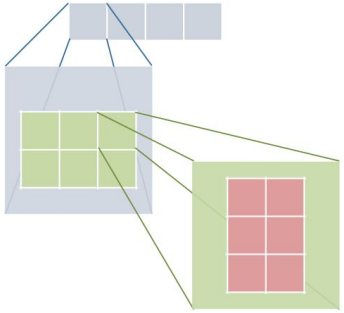
- ***Understand usability of GPUs for general particle transport simulation***
  - *Prototype e+/e−/γ EM shower simulation on GPU, evolve to realistic use-cases*
  - *Focus on EM physics given computational cost in HEP workflows, prior knowledge of applicability of physics models on GPU*
- ***Implement GPU-targeted components for physics, geometry, field, with data models and workflow***
  - *Integrate components in a hybrid CPU-GPU Geant4 workflow (“Fast Sim” approach)*
  - *Offload tracks to GPU/CPU when preconditions like particle type or geometric region met*
  - *Most realistic short-term objective to allow testing/use in existing experiment code*
- **Ensure correctness and reproducibility**
  - *Validate GPU-only, CPU+GPU off/onload against pure CPU Geant4*
- **Understand bottlenecks and blockers limiting performance**
  - *Feasibility and future effort required for efficient simulation workflows on GPU*
- **Celeritas also have a longer term objective to include full hadronic physics**

Challenges for Monte Carlo on GPUs

- **Execution**: divergence and load balancing
  - *GPUs want every thread doing the same thing*
  - *MC: every particle is doing something (somewhat) different*

Structured grid data

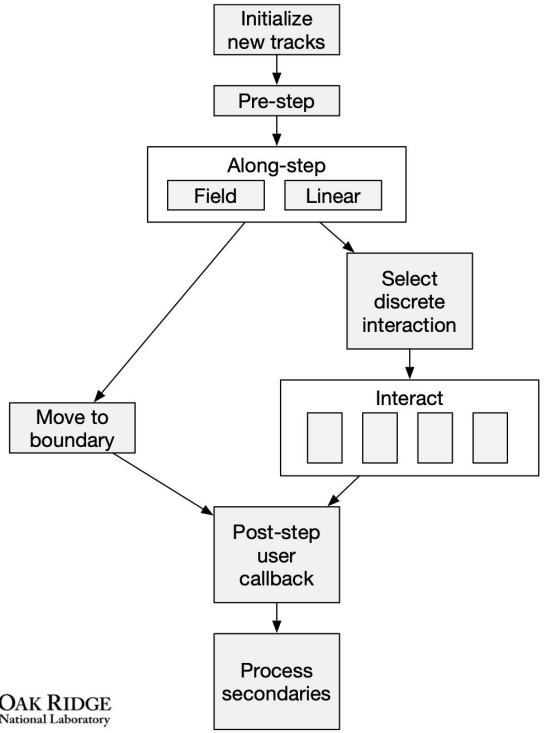
- **Memory**: data structures and access patterns
  - *GPUs want direct, uniform, contiguous access*
  - *MC: hierarchy and indirection; random access*
  - *Memory allocation is a particular problem*



Monte Carlo data

Credit: [Seth Johnson (ORNL)]

Track-parallel Stepping Workflow



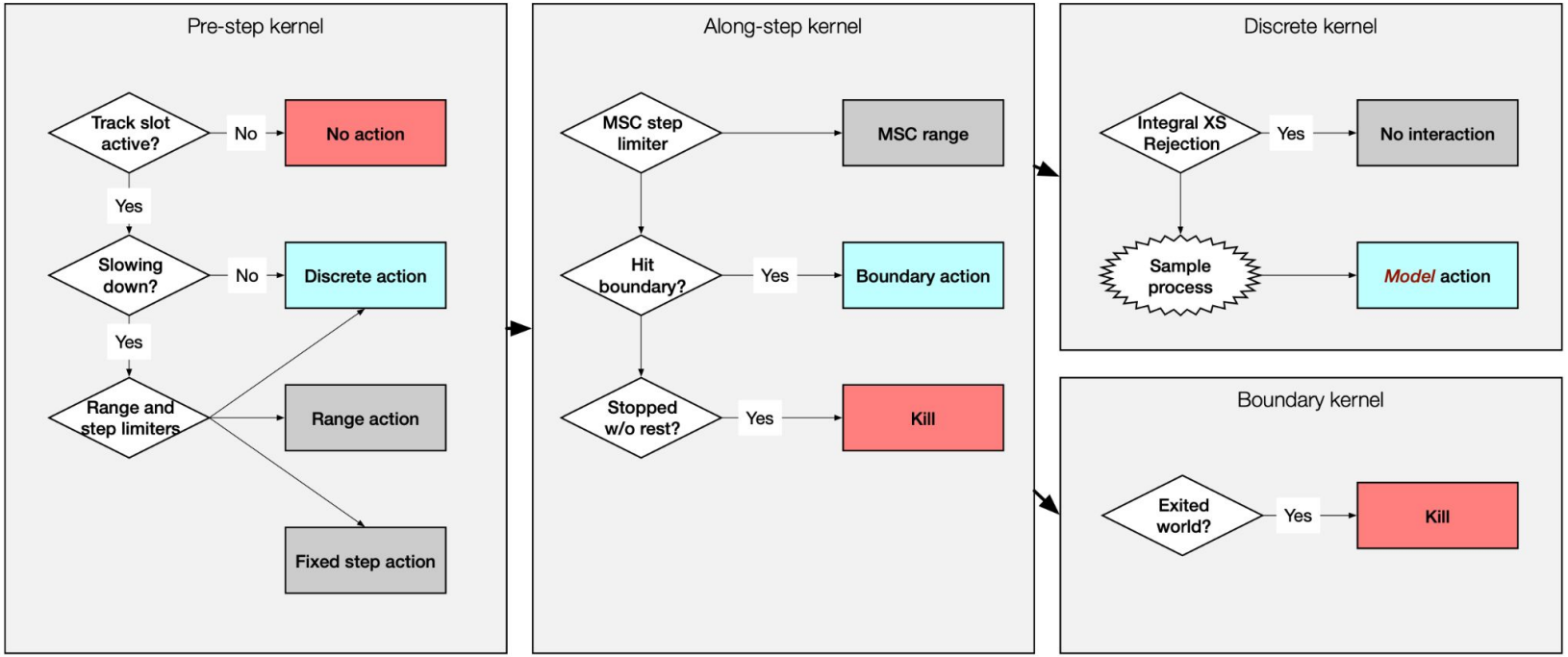

```
extend_from_primaries          ▷ Copy primaries to device, create track initializers
while Tracks are alive do
  initialize_tracks            ▷ Create new tracks in empty slots
  pre_step                     ▷ Sample mean free path, calculate step limits
  along_step                   ▷ Propagation, slowing down
  boundary                     ▷ Cross a geometry boundary
  discrete_select              ▷ Discrete model selection
  launch_models                ▷ Launch interaction kernels for applicable models
  extend_from_secondaries      ▷ Create track initializers from secondaries
end while
```

- Action based control flow
- Kernels determine next **Action**, or perform an **Interaction**
- *Example from Celeritas, AdePT's is similar though with larger kernels*

OAK RIDGE National Laboratory

Credit: Seth Johnson (ORNL)

Celeritas: Inside Kernels

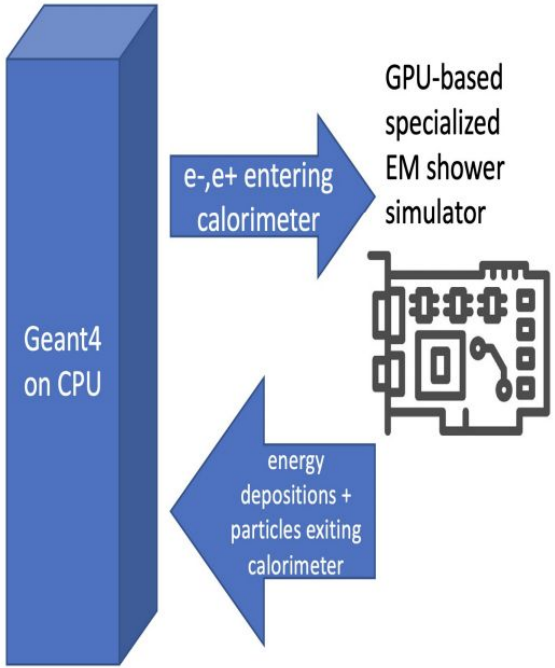


OAK RIDGE National Laboratory

Credit: [Seth Johnson (ORNL)]

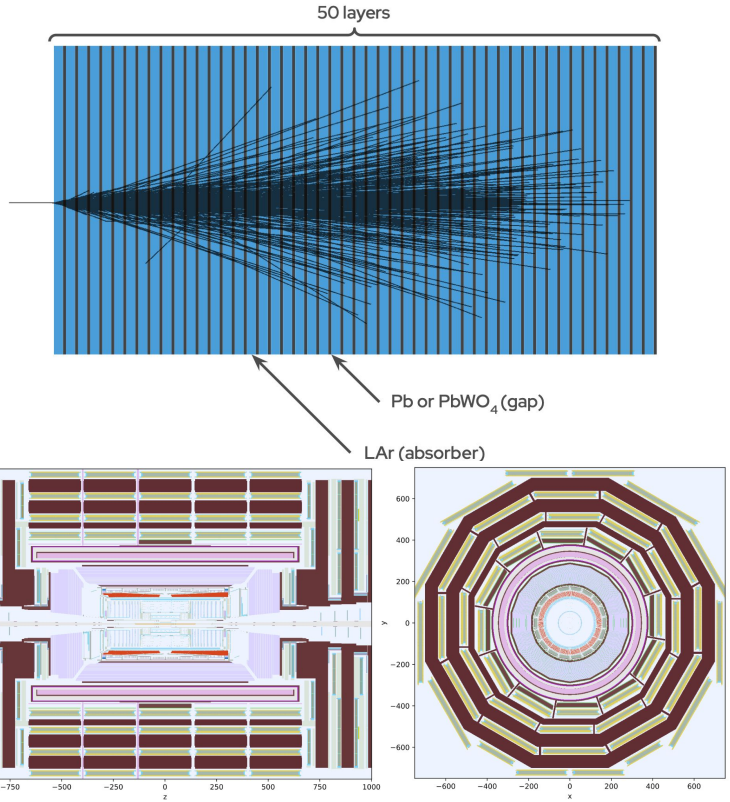
Strategies for integration with Geant4 applications

- AdePT and Celeritas only model e-/e+/g physics at present, so cannot be used standalone for simulating a **full** EM+hadronic experiment
- Instead, use them as a “service” to offload tracks to the GPU according to preconditions such as particle type, or geometric region.
  - *Basically the same as “Fast Simulation” methods*
- Use of Geant4 Fast Simulation and/or Tasking hooks, both with same basic challenges:
  - *Minimizing number/size of on/offload actions*
  - *Allow user-defined actions on GPU, such as scoring/hits*
  - *Handing back particles (e.g. exiting particles, hadrons from photonuclear processes) from GPU to CPU*



Progression Problems for Benchmarking/Validation

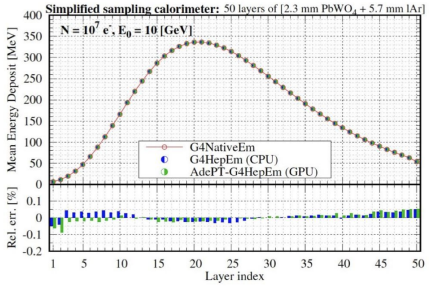
- Both AdePT and Celeritas have adopted two *primary* test cases for benchmarking and validation
  - *Run on CPU, GPU, CPU+GPU hybrid modes*
- "TestEM3" taken from Geant4 examples as a core test case
  - *50 layer Pb (or PbWO4) / LAr sampling calorimeter*
  - *1-10GeV e- primaries in beam*
  - *Validation, basic scoring and performance measurements*
- CMS 2018 GDML geometry
  - *Same primaries, also HepMC3 input*
  - *Use of more complex workflows, scoring*



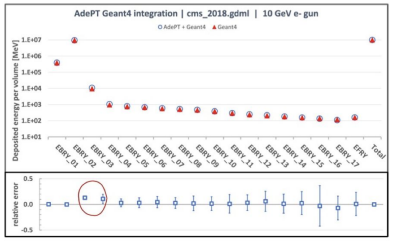
Physics Validation

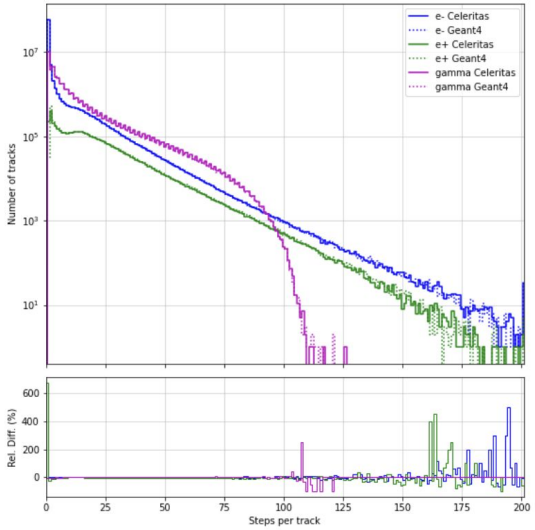
- G4HepEM in AdePT, CPU/GPU implementation/use of Geant4 models/cross-sections in Celeritas.

Sampling calorimeter example



AdePT integration with Geant4





TestEM3 MSC step count verification (Amanda Lund)

Particle	Process	Model
γ	photon conversion	Bethe–Heitler
	Compton scattering	Klein–Nishina
	photoelectric effect	Livermore
	Rayleigh scattering	Livermore
$e^{\pm}$	ionization	Møller/Bhabha
	bremsstrahlung	Seltzer–Berger
		relativistic
	Coulomb scattering	Urban MSC
e^{+}	annihilation	$\rightarrow(\gamma, \gamma)$
μ	bremsstrahlung	μ brems

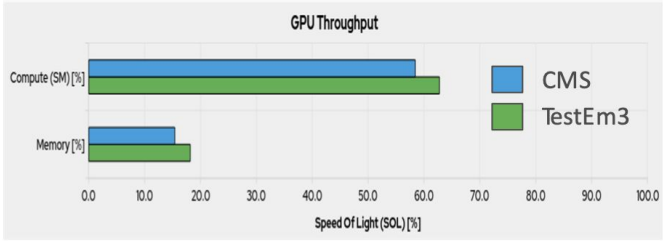
Overall excellent agreement with Geant4, but ongoing validation studies across problem space

Credits: Witek Pokorski (CERN), Seth Johnson (ORNL), Amanda Lund (Argonne)

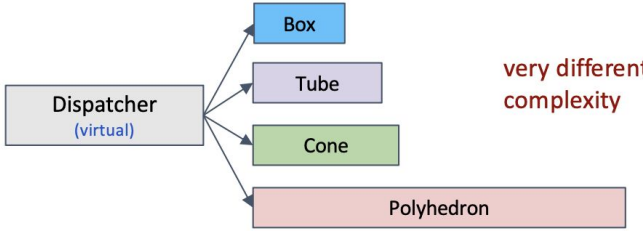
Performance: Geometry on GPUs

- **Several problems in CSG based approach of VecGeom on device:**
  - *Virtual dispatch*
  - *Recursion in relocation algorithms*
  - *Divergence from differences in algorithmic complexity for solids*
- Consequences on GPU:
  - *Large stacks & register-hungry code limits number of concurrent warps*
  - *Divergence limits concurrency per warp*
- Moving from simple to complex geometry => *longer stalls within a warp for same SM compute*

TestEM3 = 100 simple layered boxes
CMS = full CMS_2018 geometry

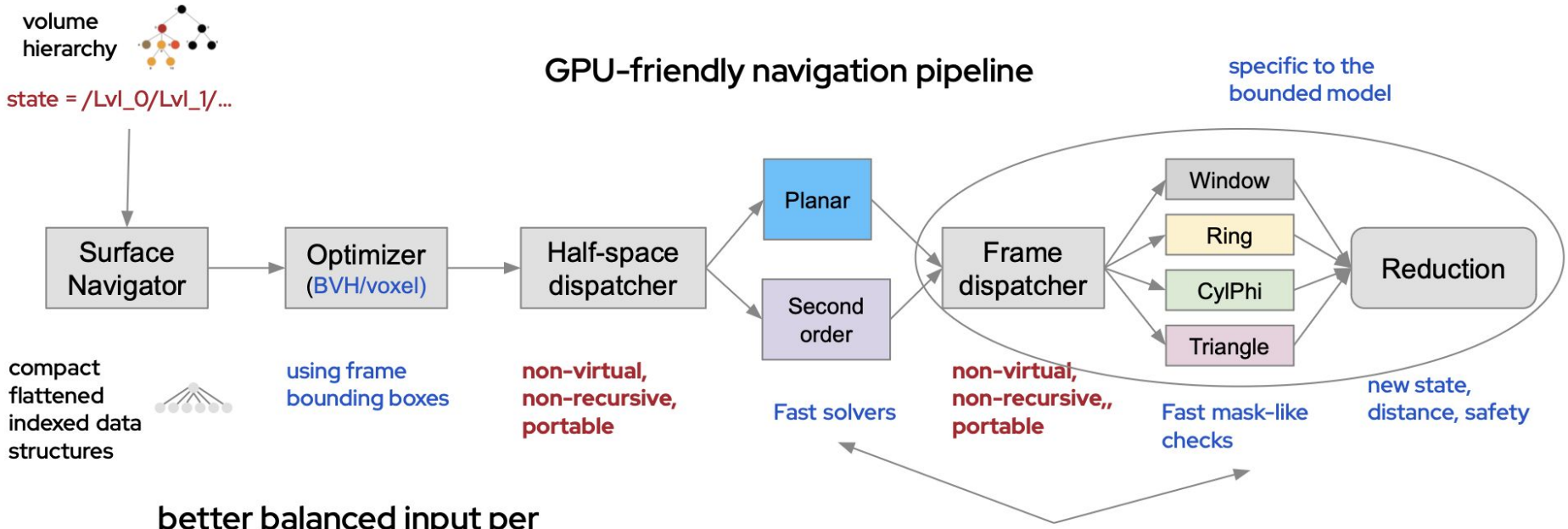


warp stalls for the same SM compute & memory ops



very different code complexity

Credit: Witek Pokorski (CERN)



better balanced input per particle due to flattening and mixing surfaces from different volumes

faster divergent sections with fewer branches

From Solid to Surface Based Geometry Models

See presentations from Andrei and Seth later

Further Progression Problems, Testing with Experiments

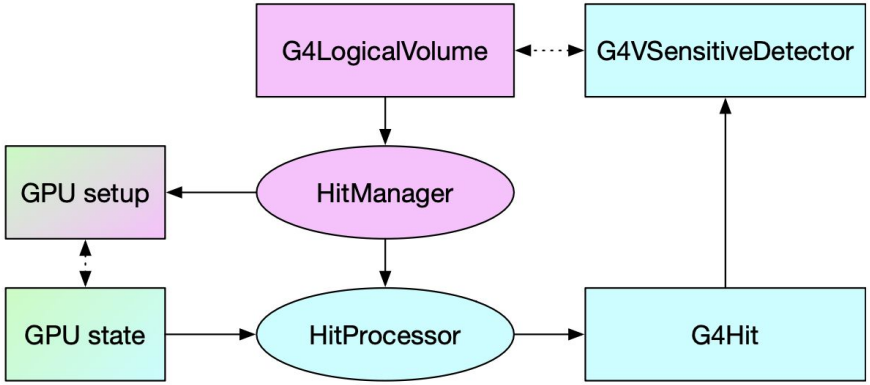
- Ongoing design discussions on additional use cases/setups for validation and performance both on pure GPU and hybrid Geant4+GPU workflows:
  - *Physics models and parameters*
  - *Detector geometries, regions for GPU offload*
  - *Inputs (primaries), Outputs (scoring, hits)*
  - *CPU/GPU hardware, workflow parameters (e.g. CPU threads, GPU tracks in flight, Host/Device memory)* ***- important to measure in realistic setups!***
    - ***Can GridPP provide guidance here, as well as good metrics to measure?***
- Need to consider both simple (~"TestEM3" calo) and complex (e.g. "LHC experiment"), and to isolate different areas (e.g. geometry, offload)
- ***AdePT/Celeritas linked up with ATLAS/CMS simulation teams to investigate integration in their frameworks as a key use case.***
  - ***Collaboration with other experiments and HPC/GPU experts very welcome!***

AdePT/Celeritas in ATLAS and CMS

- Two main lines of work in collaboration with experiment's simulation teams
  - *Test/benchmark "standalone" geometric regions, e.g. CMS HGCal, ATLAS TileCal (see Davide's talk later!)*
  - *Build/Integration of code in software stacks and applications, test full use in Athena/CMSSW*
- Also identifies/motivates improvements to geometries for GPU compatibility (e.g. ATLAS EMEC).
- A key design and development topic is on hits and scoring
  - *Can "SensitiveDetector" concept work on GPU?*
  - *Should user scoring/hit generation code be host or device?*
  - *Primarily an issue of host/device data transfers and complexity of "step-to-hit" implementation*

Approaches to Scoring on Host/Device

- **Celeritas: initial "fast simulation" model**
  - *Searches through GDML/Geant4 geometry for attached sensitive detectors*
  - *Data of interest sent to CPU after each "step iteration", reconstituted as G4Step instances*
  - *Geometric "touchable" updated as required to allow location by SD*
- **AdePT: "fast simulation" model**
  - *User has to implement device side scoring, copy-to-host themselves*
  - *See Davide's talk for an example*
- Both have costs/benefits, ***need additional input from experiments*** to balance specific/generic interface requirements



Credit: Seth Johnson (ORNL)

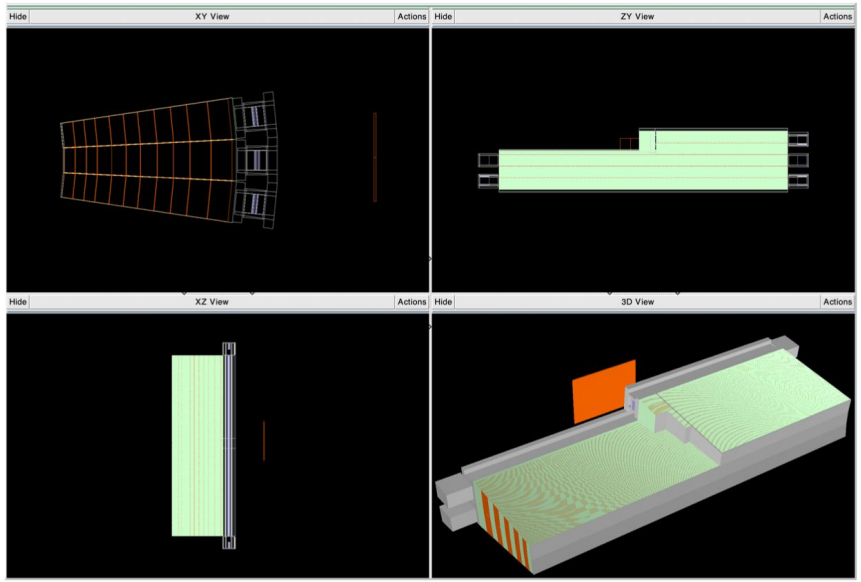
Credit: Seth Johnson (ORNL)

CMSSW integration effort

- Biweekly meeting with Fermilab CMS
  (Special thanks to Kevin Pedro for all the help!)
- Progress in integrating into CMSSW toolchain:
  - Linked Celeritas to CMSSW with Geant4, DD4HEP, and CUDA-enabled VecGeom
  - Added Celeritas offload interface RunManagerMT and TrackingAction in SimG4Core
- Theoretical maximum performance gain offloading EM tracks: **~3.3×**
  (with 1000 $t\bar{t}$ events and CMS Run3 geometry)

ATLAS TileCal in Celeritas/AdePT

- Initial work between ORNL and LBL on use of Celeritas earlier this year.
- Davide Costanzo has also implemented an AdePT example for the same geometry
- Defer to Davide's talk for more detailed info here!
- *ATLAS Full Simulation WG has UK input to assist with testing and integration, but more always welcome!*



Tilecal visualization (Stefano Tognini)

Credit: Seth Johnson, Stefano Tognini (ORNL), Lorenzo Pezzoti, Stephan Lachnit (CERN)

Summary

- AdePT and Celeritas continuing to demonstrate feasibility of detector simulation on GPU
  - *Near full EM physics validated*
  - *Initial workflow of Geant4 CPU offloading EM particles to GPU implemented/profiled*
  - ***Working with ATLAS and CMS on integration, benchmarking, and scoring in their frameworks.***
- **GPU friendly geometry modeling/navigation, scoring, will be key tasks this year**
- **Contributions to projects on GitHub welcome, and especially on experiment integration and validation**
  - ***Is there UK expertise that could contribute here?***