

GPU-Accelerated Optical Simulation for Dual Readout Calorimetry with Celeritas

Hayden Hollenbeck
on behalf of the Celeritas Core Team

University of Virginia

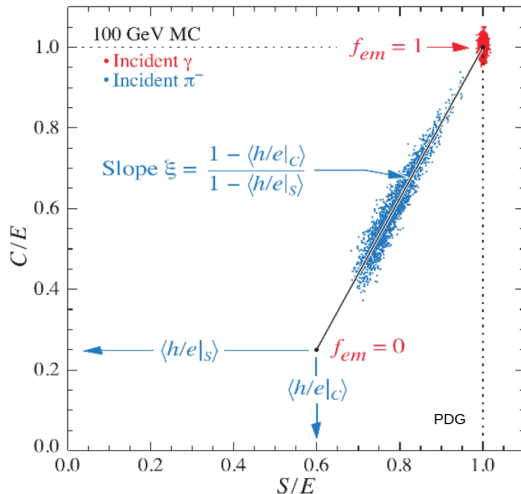
Coordination Panel for Advanced Detectors
University of Washington, October 20-23 2026



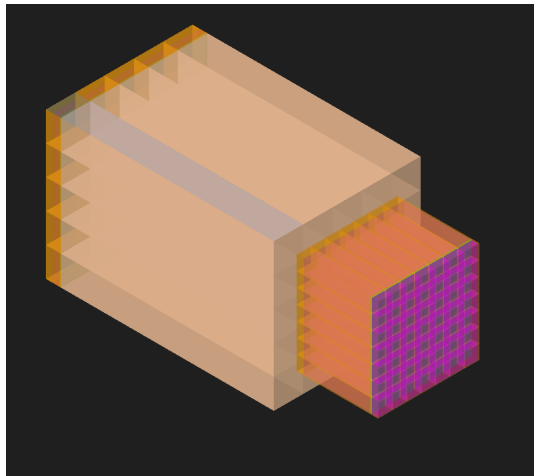
EM and hadronic showers have different Cherenkov and scintillation components

- Measuring both can give better jet energy resolution
- Need to distinguish Cherenkov and scintillation components - require detailed simulation of each

More details in Bob Hirosky's [CPAD talk](#) before this



- 5x5 BGO rear matrix
 - 5.2 photoelectrons / keV
- 7x7 PWO4 front matrix
 - 0.45 photoelectrons / keV
- Each crystal bar is made of:
 - Air gap
 - Crystal
 - Silicone gap
 - SiPM resin window
 - SiPM detector
- Each crystal wrapped in fully reflective material



DRCSim - Full waveform simulation workflow

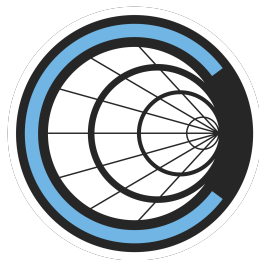
- DD4HEP based Geant4 simulation
- Separate waveform / digitization code
- Suite for testing multiple geometries and incident particles

- Tuning “dials” for approximating optical photon simulations
 - Idea: run reduced set of generated optical photons and reweight the response of the detected ones
 - Good speedup but heavy statistical uncertainties

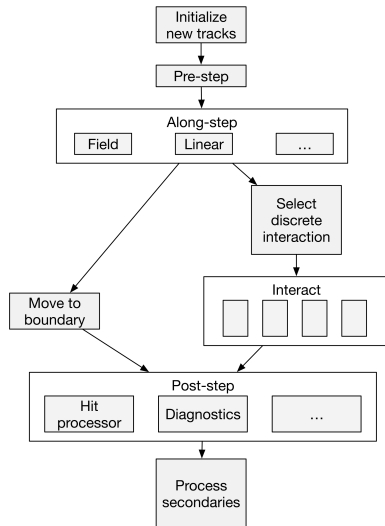
Optical simulation characteristics for a single 2 GeV electron event:

Process	Generated	Detected (Front)	Detected (Rear)
Cherenkov	143,203	5,244	6,788
Scintillation	5,190,977	156,436	514,272
All	5,334,180	161,181	521,061

- **GPU accelerated**, CPU reproducible
- **Full fidelity** Monte Carlo detector simulation
- **Automated Geant4 integration** (geometry, physics, sensitive detectors)
- **Platform portable**: CPU, NVIDIA, AMD
- **Open source** and actively seeking collaborators



- Separate main and optical tracking loop
 - Fine-tune optimizations for high count optical tracks
- Various integration methods
 - Standalone Celeritas full simulation
 - GPU offloading for Geant4 optical photons
 - Fast parameterization generation
 - Replace Geant4 optical physics
- Volumetric processes
 - Absorption
 - Rayleigh Scattering
 - Mie Scattering
 - Wavelength Shifting (up to 2 models per material)
- Surface / boundary physics





Simulation

- Standalone Geant4 simulation
- Using `celeritas::accel::UserSteppingIntegration` interface
- Offloading optical photon generation distributions (when requested)
 - Custom hook to ensure photons *only generated* in optically active regions of interest
- Hits recorded back to same Geant4 data structures

• Optical Physics:

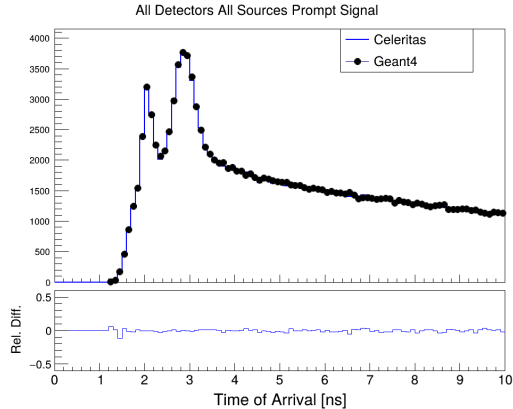
- Generators Cherenkov, scintillation
- Volumetric: Absorption, Rayleigh scattering
- Detector surface fully accepting
- Crystals have fully reflective wrapping
- Otherwise standard dielectric-dielectric surface physics

Hardware - UVa Rivanna / Afton computing clusters

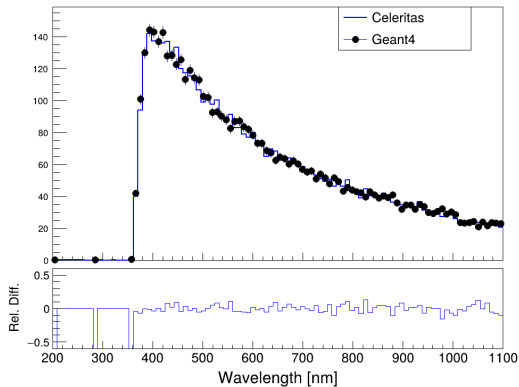
- CPU: AMC EPYC
- GPU: NVIDIA A100

Comparison of Geant4 vs Celeritas optical physics

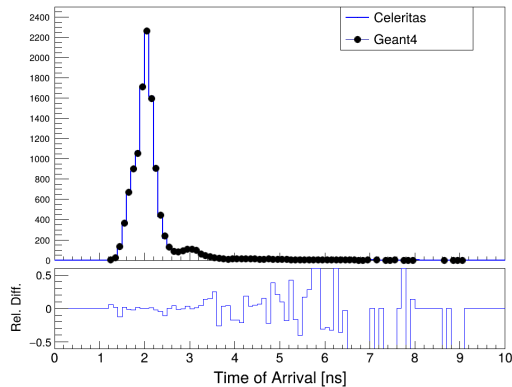
- 2 GeV electron incident at center of the detector
- Same geometry and physics options
- Electron transported in Geant4 only
- Optical photon distributions run in *both* Celeritas and Geant4
- Averaged over 10 events



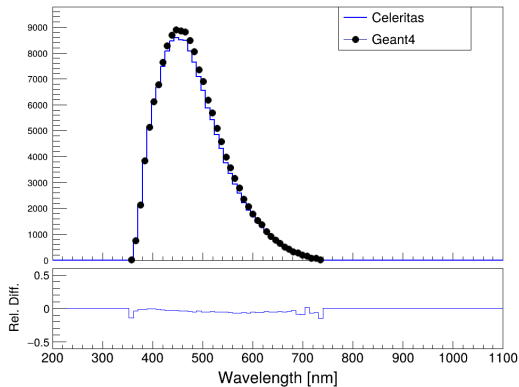
Front Detector Cherenkov Energy



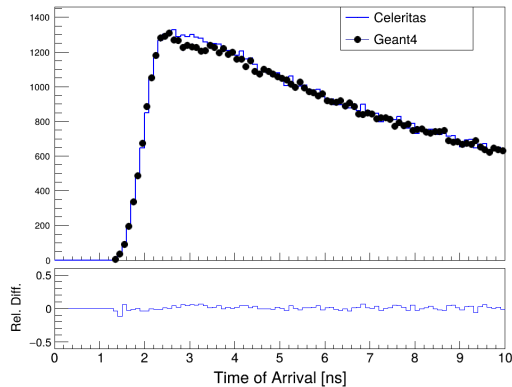
Front Detector Cherenkov Prompt Signal



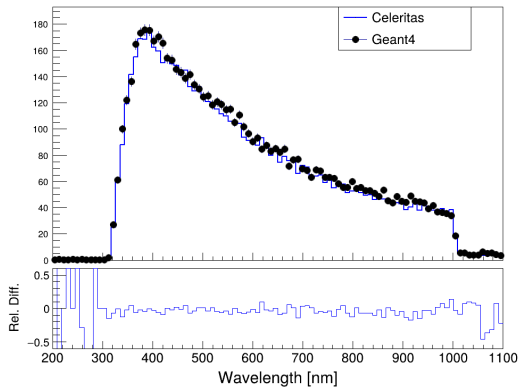
Front Detector Scintillation Energy



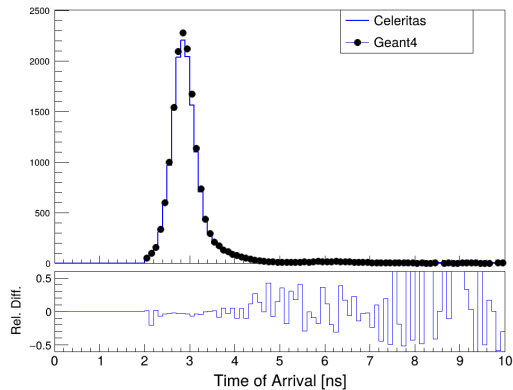
Front Detector Scintillation Prompt Signal

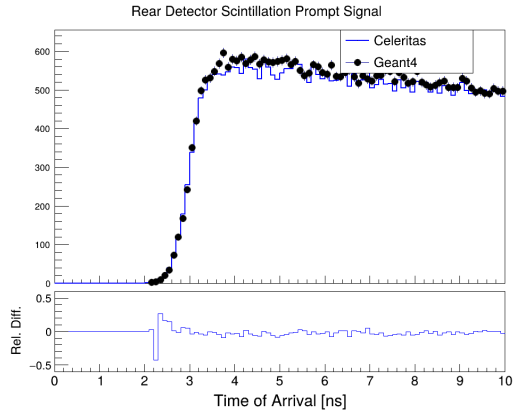
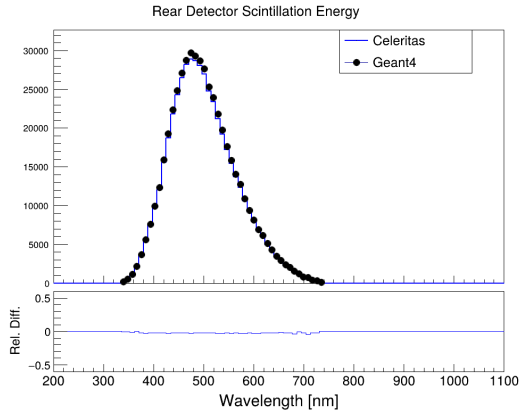


Rear Detector Cherenkov Energy



Rear Detector Cherenkov Prompt Signal

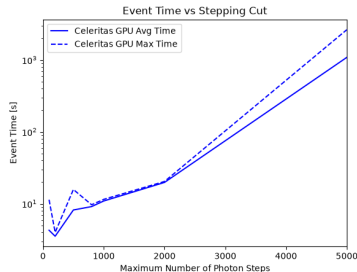
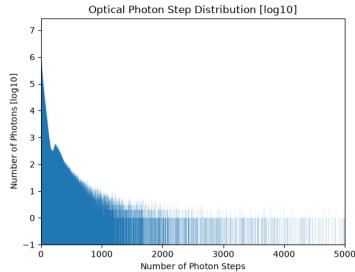




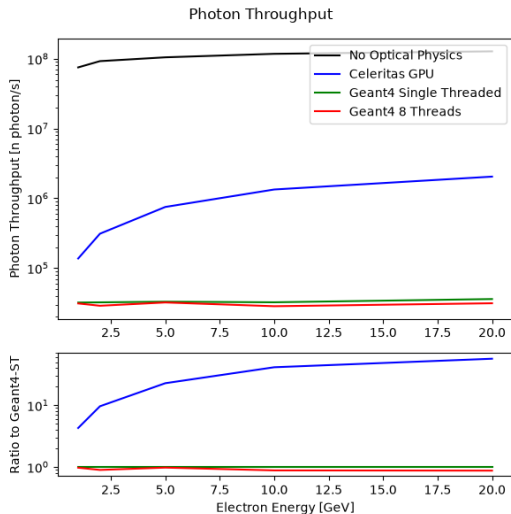
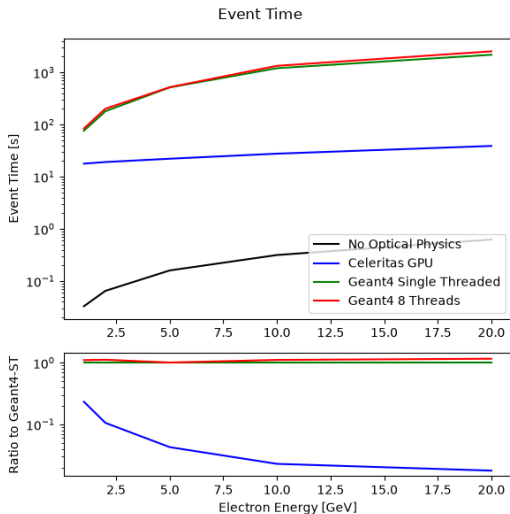
Few photon tracks take large number of steps

- Most take < 2000 but some take up to 5000 steps
- Add a stepping cut to kill tracks past some number of steps
- CPU performance not significantly affected by stepping cut
- Large GPU performance gain with stepping cuts!
- Still examining source of this performance penalty and why some photons take so many steps

Using stepping cut of 2000 for following benchmarking results!



- Single-thread performance per event at different incident e^- energies



On-going Work

- DD4HEP integration
- Further optical physics validation
- Custom detector readout support
- Asynchronous GPU loops
- Celeritas EM offloading (along with optical)

Active collaborations

- DUNE, SBND (LarSoft, Phlex)
- CalVision

Future work and optimizations

- Improved detector hit callback efficiency
- Extended optical surface models
- Multi-threading support
- Helpers for making fast-parameterizations
- Custom MC truth information
- Onloading non-performant tracks back to the CPU

Celeritas v0.6 code contributors:

- Seth R Johnson (@sethrj)
- Amanda Lund (@amandalund)
- Julien Esseiva (@esseivaju)
- Philippe Canal (@pcanal)
- Elliott Biondo (@elliottbiondo)
- Hayden Hollenbeck (@hhollenb)
- Stefano C Tognini (@stognini)
- Lance Bullerwell (@lebuller)
- Soon Yung Jun (@whokion)
- Guilherme Lima (@mrguilima)
- Sakib Rahman (@rahmans1)
- Ben Morgan (@drbenmorgan)
- Damien L-G (@dalg24)

Past contributors:

- Vidor HL Montiel (@VHLM2001)
- Peter Heywood (@ptheywood)
- Tom Evans (@tmdelellis)
- Paul Romano (@paulromano)
- Vincent R Pascuzzi (@vrpascuzzi)
- Doaa Deeb (@DoaaDeeb)

OLCF: This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

SciDAC: This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, Scientific Discovery through Advanced Computing (SciDAC) program.

NERSC: This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award HEP-ERCAP-0023868.

- Code: github.com/celeritas-project/celeritas
- Documentation: celeritas-project.github.io/celeritas/dev/index.html
- Publication: doi.org/10.1051/epjconf/202429511005

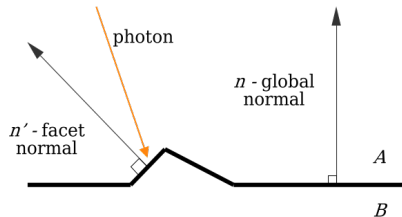
Backup Slides

Geant4 surface physics models:

- GLISUR (Geant3 model, Geant4 default)
- UNIFIED
- Look-up tables (LUT, LUT-DAVIS)
- Dichroic
- Coated dielectric-dielectrics

Goals for GPU Porting:

- Improve code reuse
- Decrease thread divergence
- Increase code readability
- Promote user customization



Three common steps for a surface crossing:

- 1 Sample local facet normal
- 2 Calculate reflectivity
- 3 Sample surface interaction result

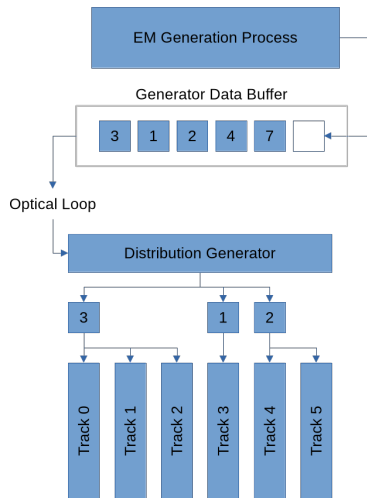
Repeat until photon is leaving the surface

Extensible interface for generation optical photons

- Geant4 optical Cherenkov and scintillation processes in EM physics
- Buffer minimal step information to generate photons
- Generators in optical loop initialize tracks from buffered data

Separates core and optical physics:

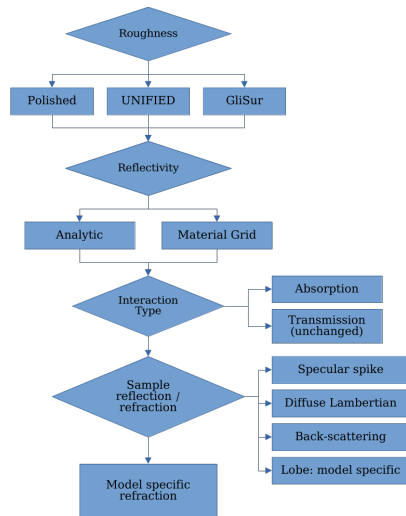
- Generated data can come from Geant4 or Celeritas
- Supports standalone optical simulations
- Supports custom generators
- Can be used by non-EM processes, e.g., optical guns and wavelength-shifting



A single surface's properties are defined by:

- Pre-volume optical material A
- Post-volume optical material B
- Roughness model
 - Polished (No roughness)
 - Uniform smear (GLISUR)
 - Gaussian (UNIFIED)
- Reflectivity model
 - Analytic (Fresnel eqns)
 - User-defined grid
- Interaction model
 - Dielectric $\rightarrow$ dielectric
 - Dielectric $\rightarrow$ metal

Can be mixed, matched and extended. Geant4's physics reproduced by appropriate choice of models!





Geant4 Example: UNIFIED model back painted surfaces

- Crystal $\rightarrow$ air gap $\rightarrow$ wrapping
- Simulate crossing both crystal-gap and gap-wrapping interfaces
- Gap size infinitesimal
 - (use coated model for thick gaps)

Celeritas extends **beyond Geant4** to arbitrary composite surfaces

- *Multiple layers with multiple models*
- Different types of gaps, wrappings, etc.
- Handle multiple reflections and transmissions

