nu-opticks

Bringing eic-opticks to LAr and WbLS

Brett Viren

March 6, 2026

Topics

- Opticks and EIC-Opticks
- The LDRD-C
  - Promises
  - Plans
  - Progress
- Organization
- Going forward

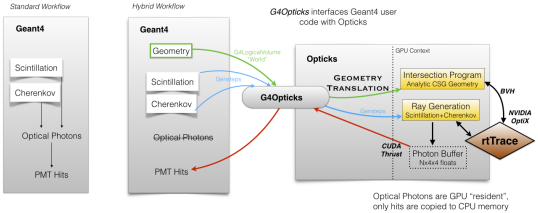
Opticks

Created by Simon Blyth around the time of Daya Bay → JUNO

- Based on NVIDIA OptiX + CUDA for Ray Tracing (RTX) GPUs
- **Accelerates Geant4 optical photon transport by 100x**

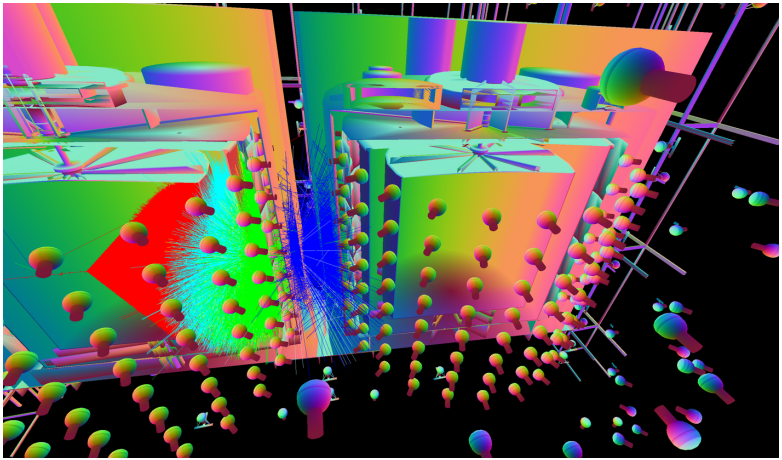
Geant4 + Opticks Hybrid Workflow : External Optical Photon Simulation

`https://bitbucket.org/simoncblyth/opticks`



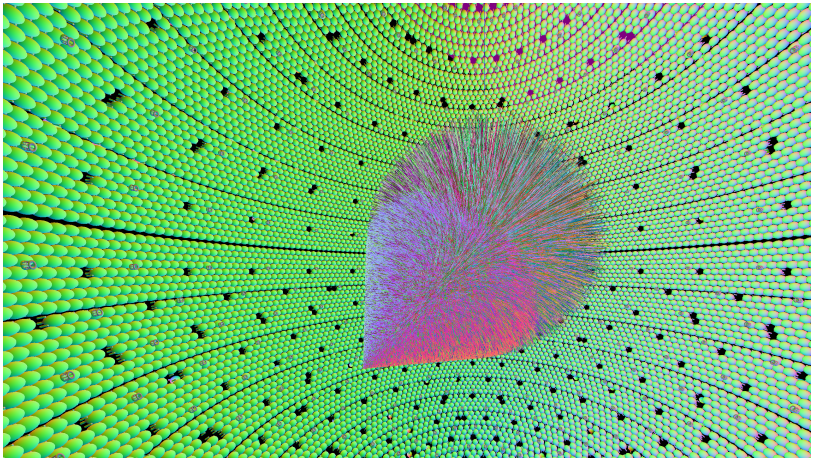
`https://simoncblyth.bitbucket.io/env/Documents/Geant4OpticksWorkflow/Geant4OpticksWorkflow.001.png`

DayaBay/Opticks example output (Simon Blyth)



https://simoncblyth.bitbucket.io/env/presentation/dayabay-principal_half.png

JUNO/Opticks example output (Simon Blyth)



https://simoncblyth.bitbucket.io/env/graphics/ggeoview/jpmt-inside-wide_crop.png

EIC-Opticks = Opticks + more

Started with LDRD for EIC detector R&D

- https://arxiv.org/abs/2512.06061
- **Gabor Galgoczi**, Kolja Kauder, Maxim Potekhin, Sakib Rahman, Dmitri Smirnov, Torre Wenaus

Some of the features added:

- Multi-event photon batching for better GPU utilization.
- Support for spherical mirror and open cones.
- Warning for deeply nested boolean geoms (performance pathology).
- User support scripts for photon leakage etc.

Some future plans:

- Async G4/Opticks interface to bring CPU+GPU parallelism, less idling.
- Grouping of photons by expected number of steps for better GPU utilization.
- Wavelength shifting processes (needed by LAr/WbLS).

The LDRD-C

Extend eic-opticks to support neutrino detectors.

- Circuitous path: initiated by Gabor in NPPS for WbLS, ended up in EDG for LAr/DUNE.
- Short funding period: Feb 15 - Sep 30 2026.
- Xuyang (45%), Gabor (co-PI, 10%), Brett (co-PI, 3%)

Expected results:

- Implement eic-opticks in framework-independent production-ready application.
- Validate its optical physics models as relevant to LAr / DUNE.
- Produce benchmarks and guidance for CPU/GPU hardware specs.
- Gain expertise and planning for integration into future DUNE software framework.

Plans

- Will use Clark McGrew's `edep-sim` as the target Geant4 application.
  - Why: used by DUNE ND, *art* and LArSoft is too "old", phlex is too "new".
- Two phase development plan.
  1. Quickly "hack" opticks into edep-sim, replacing G4 optical, learn both codes, throw away.
  2. Develop proper design with compile-time and run-time choices to use opticks or vanilla G4.
- Along the way: test out further improvements in eic-opticks.
  - New volume shapes (extrusion).
  - Wavelength shifting process.
  - Performance improvements.

Progress so far

- Assembled a group ("nu-opticks" = eic-opticks + LDRD + WbLS + Larsoft)
  - BNL: Xuyang, Gabor, Brett (LDRD-C) and Yousen (WbLS) and Chao (future LDRD-A).
  - SBU: Clark McGrew (edep-sim),
  - Rice: Aaron Higuera and Ilker Parmaksiz (transition LArSoft opticks → eic-opticks)
- Initial toy LAr detector geometry (Xuyang).
  - For initial simplicity, will support official GeGeDe geometry.
- `edep-sim` version 4.0.0 release w/ cherry picks from DUNE fork (Clark)
- `edep-sim + eic-opticks` developer environment (Brett).
  - Simple, Spack-based ecosystem.
- Initial exercising of existing edep-sim optical support (Xuyang).

Future thinking: two main thrusts and some navel gazing

Chao's LDRD-A has an eic-opticks component

- Leverage LDRD-C experience to quickly integrate eic-optics into LDRD-A software.
- Accelerated photons generally support the other LDRD-A R&D goals,
  - In particular, to produce large training sets for LDRD-A's "surrogate detector" ML model.

Future DUNE integration

- `edep-sim` is used for **ND** and will benefit from our work "for free".
- **FD** story is more cloudy in detail but broad strokes are clear:
  - We will work with **Rice** to help them swap opticks → eic-opticks in LArSoft's LArG4.
  - FNAL is porting LArG4 to **phlex** which may carry along opticks additions.

Going further in the phlex era

The patterns we explore in our **opticks**, **SPNG** and **tred** works motivate a **novel, sophisticated integration** that leverages phlex's design in ways that transcend *art* patterns (phlex execution model essentially that of Wire-Cell Toolkit). This could result in software that will better serve **modern needs** such as **ML training, GPU acceleration and multi-threading**.

FIN

Backups