EPJ Web of Conferences **295**, 11004 (2024) https://doi.org/10.1051/epjconf/202429511004
CHEP 2023

CaTS: Integration of Geant4 and Opticks

Hans Wenzel[1,*], *Soon* Yung Jun[1], *Krzysztof* Genser[1], and *Felipe* De Figueiredo[2]

[1]Fermi National Accelerator Laboratory†, Batavia, IL, 60510, USA
[2]Long Island University, Brookville, NewYork, USA

Abstract. CaTS [6]is an advanced example that is part of Geant4 since version 11.0. It demonstrates the use of Opticks to offload the simulation of optical photons to GPUs. Opticks interfaces with the Geant4 toolkit to collect all the necessary information to generate and trace optical photons, re-implements the optical physics processes to be run on the GPU, and automatically translates the Geant4 geometry into a GPU appropriate format. To trace the photons, Opticks uses NVIDIA OptiX®. In this report, we describe CaTS and the integration of Opticks with Geant4. We demonstrate that the generation and tracing of optical photons represents an ideal application to be offloaded to GPUs, fully utilizing the high degree of available parallelism. In a typical liquid argon TPC simulation, a speedup of several hundred times is observed compared to an equivalent simulation using single threaded Geant4.

1 Introduction

In Geant4 [1], optical physics has an exceptional position among the physics processes, as it adds a new particle, optical photon together with optical properties for materials and optical surfaces. The optical photon can be reflected or refracted at optical surfaces and it can only be created in the optical processes scintillation, Cerenkov radiation, and wavelength-shifting (WLS). This makes the G4OpticalPhoton different from the "usual" high energy physics photon (G4Gamma) in Geant4. Optical properties need to be assigned to the materials whenever optical physics processes are to be considered in the simulation. Every material needs at least a refractive index spectrum (which corresponds to the dispersion relation) in addition an attenuation length spectrum,should be defined though the attenuation length is by default set to infinity if it is not defined. Special optical materials, such as scintillating and WLS materials, additionally require the specification of the emission spectra as well as up to 3 decay times. Optionally, one can also specify the rise time of the scintillation light, but this is not currently supported by the Opticks [2] implementation of the scintillation process. More properties can be assigned to optical surfaces between volumes, for example, the reflectivity of the surface. In Geant4 and Opticks, optical properties (such as the materials refractive index, Rayleigh scattering length or absorption length) are inputs that have to be provided by the user when the detector is constructed. The properties are either bulk properties or boundary properties (G4OpticalSurfaces). Geant4 only traces optical photons for materials for which at least the refractive index is defined, otherwise the optical photons are killed. High-precision modeling

*e-mail: μwenzel@fnal.gov

of light production, transport and detection in general requires the use of the best available values to describe the optical properties.

Liquid argon TPCs are of special interest for current and future neutrino experiments at Fermilab, therefore we decided to use a simplified liquid argon TPC as an example. A good introduction to scintillation light in liquid argon can be found elsewhere [3]. The scintillation process in liquid argon is relatively complicated including various excimer states which can be induced by ionization, but not by the scintillation photons themselves. For that reason, reemmission of scintillation light does not play a role in liquid argon which is highly transparent to its own scintillation light with the absorption length on the order of several meters depending on purity. The wavelength of the scintillation photons produced in liquid argon is in the vacuum ultraviolet (VUV) with a wavelength of 128 *nm*. In the simulation, the emission spectrum is modeled as a Gaussian function centered at 128 *nm* with a width of 10 *nm*. The values of optical properties used in the simulation are given in table 1. Many of the properties were extracted from [4] or were calculated using the formula given therein.

Property	Value
Scintillation yield ($E = 0$)	50000/MeV
Time constant of fast component	6 *ns*
Time constant of slow component	1590 *ns*
Fraction of fast component	0.75
Fraction of slow component	0.25
Rise time	0 *ns*
Emission spectrum peak	128 *nm*
Emission spectrum width	10 *nm*
Absorption length	10 *m* at 128 *nm*
Rayleigh scattering length	90 *cm* at 128 *nm*
Refraction index	1.4 at 128 *nm*

Table 1. Optical properties of the liquid argon used by CaTS.

When an electric field is applied across liquid argon, there are two competing and anti-correlated processes involved: ionization and scintillation. Ionization occurs when ions and electrons are separated due to the electric field, before they can recombine and cause scintillation. This reduces the scintillation yield. The electrons then drift along the electric field lines towards a detection plane where they produce a signal at wire planes in a stereo layout or at pixels. The electron drift is part of the detector response and therefore is not considered in the simulation. For this report, we use a very simple geometrical configuration of a liquid argon TPC which consists of a box of liquid argon with x y z dimensions of 1 x 1 x 2 m shown in figure 1. There are five photo detectors shown in red attached to the side. There is no E-field so we assume all the deposited energy is converted to scintillation. Figure 1 shows the simulation of a 2 *GeV* electron shower. Due to the low atomic number ($Z = 18$) and low density ($\rho = 1.78\ g/cm^3$) of liquid argon, the shower is not fully contained in the detector. In this case, around 7×10^7 VUV scintillation photons are produced. Using Geant4 (11.1.2) to simulate the photon generation and propagation on a single core Intel® Core i9-10900k 3.7 GHz takes more than 5 minutes per event compared to 0.034 seconds per event when no optical photon simulation is performed. Currently, most LArTPC-based experiments use look up tables and parametrizations instead of full simulation for the optical photon response.

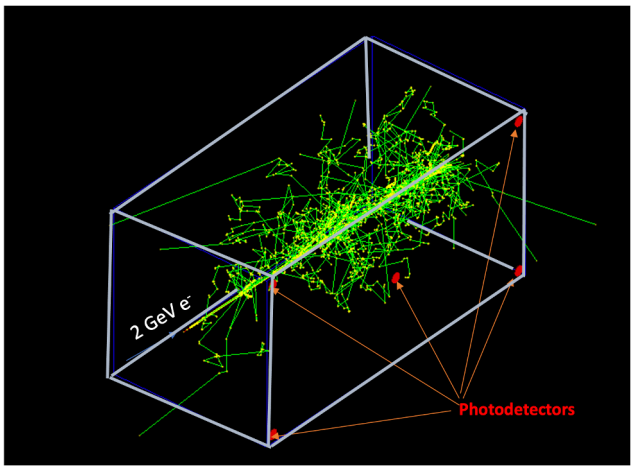

Figure 1. Simulation of a 2 *GeV* electron shower. Shown are only steps and particle tracks, but no optical photons.

2 Simulation of optical photons: an ideal application to be ported to GPUs.

The simulation of optical photons seems an ideal application to be ported to GPUs for the following reasons. Only one particle type, the optical photon is involved, but there are many of them ($\approx 10^7$/event), which allows for massive parallelism. Especially when the number of reflections is limited, one expects low latency and no big fluctuations in computing time for each optical photon. No new particle types besides optical photons are produced. Only a few physics processes need to be implemented on the GPU. The processes involved in generating optical photons are: G4Cerenkov, G4Scintillation (reemission) and G4OpWLS[1]. The processes involved in the transportation of optical photons are G4OpAbsorption, G4OpRayleigh and G4OpBoundaryProcesses with various surface models. These processes do not need a lot of input data. All the information necessary to generate optical photons are collected in so called GenSteps for the Cerenkov and scintillation processes, so there is little data to transfer from host (CPU) to device (GPU). Only a small fraction of photons reaches the photo-detectors and produces a photon-hit, so very little data has to be transferred back from the device to the host. Optical ray tracing is a well-established field so one benefits from the availability of efficient algorithms (in this case OptiX® [5]). Using NVIDIA® hardware and software (CUDA®, OptiX®) has both disadvantages and advantages. On one hand, one is committed to a specific hardware and software vendor compared to a more generic software solution that allows the use of various GPU types. On the other hand, one gains performance benefits from optimal use of the NVIDIA® hardware. For example, hardware acceleration RTX (Ray Tracing texel eXtreme) was introduced in 2018 with the Geforce 20 series and is now in the third generation (Geforce 40) doubling in performance in each generation. RTX is fully supported by OptiX®.

[1]WLS is currently not used in our Geant4 simulation and the process is not implemented in Opticks yet.

3 Opticks

Opticks is an open-source project developed by Simon Blyth [2]. There are two major versions: legacy Opticks based on OptiX6® using the G4Opticks API and the latest Opticks based on OptiX7® using the G4CXOpticks API. Significant re-implementation of Opticks was made necessary because OptiX7 introduced breaking changes to the OptiX API allowing more control of GPU memory allocation and data transfers and allowing to benefit from current and future GPUs, RT cores and RTX. This update also gives the opportunity to improve Opticks, making it more flexible and more modular, splitting libraries by dependencies and splitting up monolithic Cuda files into many small headers. For more details, see reference [2].

Opticks accelerates optical photon simulation by translating the Geant4 geometry to OptiX® without approximation for a subset of the solids available in Geant4, implementing the Geant4 optical processes on the GPU and integrating NVIDIA GPU ray tracing (accessed via NVIDIA OptiX®). G4(CX)Opticks provides an API to interface Geant4 and Opticks. The Geant4 advanced example CaTS (Calorimetry and Tracking Simulation) [6] uses this API to implement a hybrid workflow where the generation and tracing of optical photons is offloaded to a GPU using Opticks, while the rest of the simulation is done on the CPU using Geant4. The workflow is shown in figure 2 with the two big blocks representing the CPU (host) and GPU (device). Geant4 on the CPU/host handles all particle types but the optical photons. In addition, Geant4 is used to collect the GenSteps and uses the G4Cerenkov and G4Scintillation processes to calculate the number of optical photons to be generated at a given step. A Genstep provides all necessary quantities to generate the photons on the GPU. Besides stepping information like the step length, the Genstep provides the deposited energy for the scintillation process and $1/\beta$ for the Cerenkov process. Copying the GenSteps to the GPU and then generating the optical photons on it is more efficient than copying many optical photons to the GPU. The Gensteps are collected in the Geant4 user stepping action and they are passed to the GPU whenever a certain number of optical photons that needs to be generated and traced is reached. At the end of an event, all remaining Gensteps are passed to the GPU to flush each event loop.

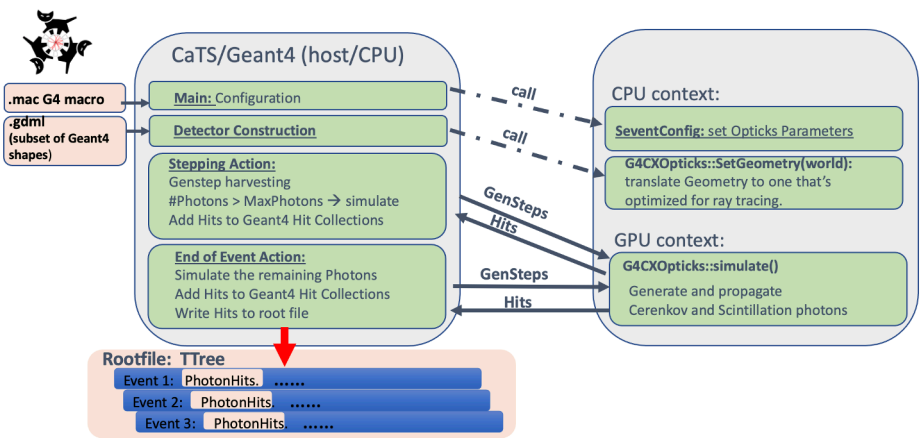

Figure 2. Hybrid workflow where the generation and tracing of optical photons is offloaded to a GPU using Opticks, while the rest of the simulation is done on the CPU using Geant4.

4 CaTS

CaTS (Calorimetry and Tracking Simulation) is a general Geant4 application which was specifically developed with detector R&D in mind. For R&D, it is beneficial if frequent changes in detector geometry and physics configuration can be performed at run-time without having to recompile the program. CaTS is included in Geant4 as an advanced example application since version 11.0. The CaTS application:

- requires no changes to Geant4 to integrate Opticks and only makes use of provided interfaces: UserActions, Sensitive Detectors,
- is modular and extensible, allows to build detector setups from predefined components,
- uses GDML with extensions for flexible detector construction at run-time. GDML extensions are used to:
  - assign sensitive detectors to logical volumes,
  - assign step-limits and energy cuts to logical volumes,
  - assign visualization attributes,
- provides a library of sensitive detector classes,
- creates hit collections and uses automated ROOT-based IO,
- supports both the legacy and new Opticks interfaces,
- uses the G4PhysListFactoryAlt physics list factory to define and configure physics at run-time via command line options (for example, ./CaTS -g simpleLArTPC.gdml -pl FTFP_BERT+OPTICAL+STEPLIMIT -m time.mac)
- provides run-time and build-time options for G4(CX)Opticks/Geant4,
- provides an option to collect either scintillation and Cerenkov Gensteps, or both.

The G4PhysListFactoryAlt physics list factory provided by Geant4 allows the selection of one of the various reference physics lists, specify the electromagnetic option as well as various physics constructors. By default we use the FTFP_BERT reference physics-list, the default electromagnetic option, as well as the optical physics, step-limiter and neutron killer physics constructors.

5 Performance

Using Geant4 11.1.2, it takes around 330 seconds/event to simulate photon generation and propagation with a single core on an Intel® Core i9-10900k@ 3.7 GHz while it takes 0.034 seconds/event without optical photon simulation. When using the legacy version of Opticks, we observe 1.8 sec/event with a NVIDIA GeForce RTX 3090 GPU which is a speed up of ∼190 times for the simple geometry used. With this speed up, it becomes feasible to run full optical simulation event-by-event instead of using look up tables and parametrizations. We observe that one CPU core is sufficient to saturate the GPU.

Modern CPUs are equipped with multiple CPU cores (10 in our case), therefore one should not compare the performance with a single threaded Geant4 simulation but with the performance when all cores (plus hyper-threading) are used using Geant4 event based multi-threading which has been shown to scale well with the number of threads while showing a smaller memory footprint than running the processes in parallel.

6 Making Opticks available to the LArTPC Experiments

artg4tk is a general Geant4 module for the *art* [7] event processing framework used by various Fermilab experiments. It depends only on *art*, Geant4 and dependencies thereof. *artg4tk* is very flexible and can be configured at run-time. The Geant4 User Actions (Stacking, Stepping, Tracking, Event, Run) are implemented as *art* framework services which can be selected and configured at run-time. Like CaTS, *artg4tk* uses gdml files with extensions for the detector configuration and uses the G4PhysListFactoryAlt physics list factory to define and configure physics at run-time. *artg4tk* uses an *art* service to add the Geant4 hit collections to the RootIO based *art* event record.

LArSoft [8] is a general software framework for liquid argon TPCs based on *art* and *LArG4* [9] is a module for *LArSoft*. *LArG4* itself depends on *arg4tk*. Keeping *artg4tk* as a general Geant4 framework, all dependencies (e.g. from lardata objects) introduced by LArSoft are encapsulated in *LArG4*. The dependencies are also shown in figure 3.

We are in the process of integrating Opticks with *artg4tk/LArG4*. This task is made more difficult since build system, package manager as well as Linux distribution are about to change. For now it will require a hybrid approach with customized cmake files to build against Opticks and the libraries it depends on. This is only a temporary solution to make it available to experts who want to try Opticks right now and for integrating *artg4tk* and *LArG4* with Opticks. Once transitioned to the new package manager [10], we expect all products to be made available with it.

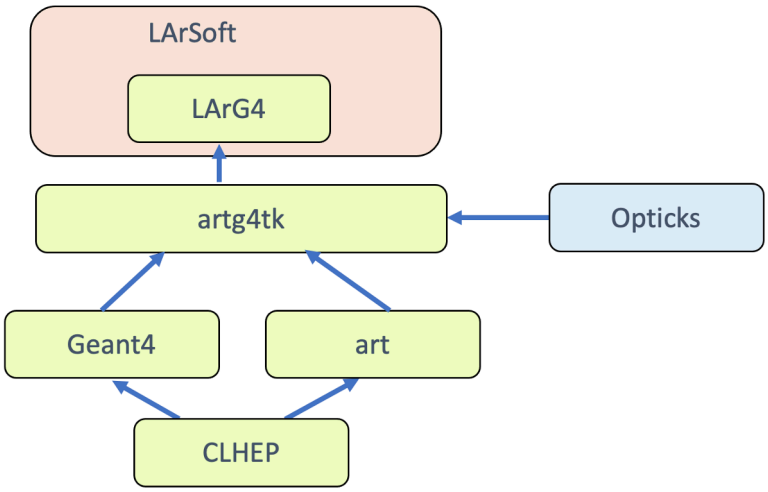

Figure 3. Dependency diagram for *LArSoft, LArG4, artg4tk* and Opticks.

7 Summary

We have created a Geant4 advanced example CaTS which is part of the Geant4 distribution since Geant4 4.11.0. CaTS demonstrates a Geant4 hybrid workflow where the generation and tracing of optical photons is offloaded to a GPU using Opticks, while the rest of the simulation is done on the CPU. The current version is based on the legacy Opticks using NVIDIA OptiX6. CaTS has been modified for the new Opticks workflow using OptiX7 and we plan to make it available with the new release of Geant 4 (11.2) planned for this

fall. Changes to the Geant4 material property APIs in the current version of Geant4 (>11.1) made changes to both CaTS and Opticks necessary while keeping compatibility to previous versions of Geant4 (>10.0). We are in the process of integrating Opticks with artg4tk/LArG4 to make it available to the LArTPC based experiments.

Acknowledgments

† Operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

References

[1] S. Agostinelli et al., Nucl. Instrum. Meth. **A 506**, 250 (2003), J. Allison et al., IEEE Trans. Nucl. Sci. **53** 270 (2006), J. Allison et al., Nucl. Instrum. Meth. **A 835**, 186 (2016), *https://geant4.web.cern.ch*.

[2] *https://bitbucket.org/simoncblyth/Opticks/*, *https://doi.org/10.1051/epjconf/202125103009*, Simon Blyth (presented by Dr. Tao Ling), CHEP2023 Norfolk Virginia May 8^{th} 2023, *https://indico.jlab.org/event/459/contributions/11811/attachments/9215/13376/opticks_20230508_chep_compressed.pdf*.

[3] Ben Jones, *Introduction to Scintillation Light in Liquid argon*, *https://microboone-exp.fnal.gov/public/talks/LArTPCWorkshopScintLight/bjpjone_2014.pdf*.

[4] Emily Grace, Alistair Butcher, Jocelyn Monroe, James A. Nikkel, *Index of refraction, Rayleigh scattering length, and Sellmeier coefficients in solid and liquid argon and xenon*, (arXiv:1502.04213) and references therein.

[5] *https://developer.nvidia.com/rtx/ray-tracing/optix*.

[6] `https://github.com/hanswenzel/CaTS`.

[7] *https://github.com/art-framework-suite/art-g4tk*.

[8] *https://larsoft.org/about-larsoft/*.

[9] *https://github.com/LArSoft/larg4*.

[10] *https://spack.io/*.