R&D activities

Plan of work

Witek Pokorski

06.04.2023

Geant4 Technical Forum

Content

- fast simulation techniques
  - Machine-Learning based models
  - Geant4 fast-simulation interface
- exploration of new hardware (GPUs) for full simulation
  - general transport code prototypes
  - domain specific application

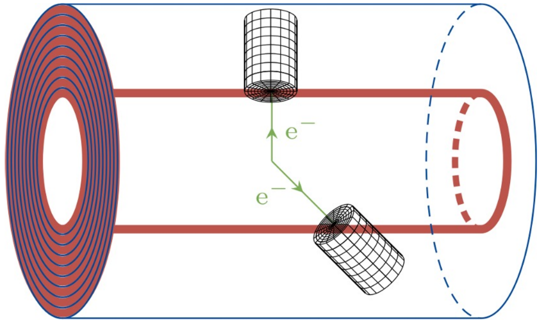
Anna, Dalila, Piyush

Fast Simulation: Overview 2022 (1/2)

g4fastsim.web.cern.ch

Geant4 example with ML fast sim model (Par04)

- Example demonstrating use of ML models (inference within C++ framework)
- Inference libraries extended with LibTorch (additional to ONNX runtime, LWTNN)
- Used to produce EM shower data published on zenodo
- Work on optimisation strategies: ACAT talk
  - at training (parameter selection and visualization)
  - at inference (memory footprint)



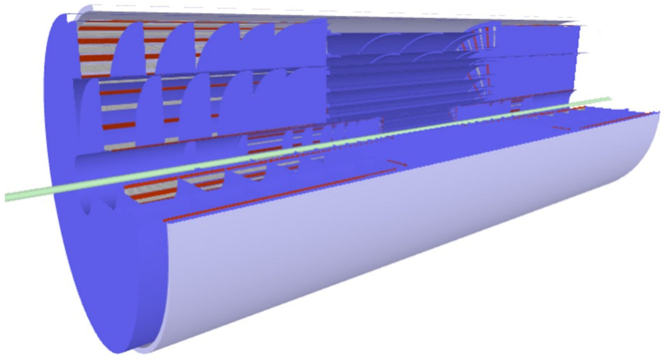
MetaHEP

- Uses the same model architecture as in Par04
- Learning to learn approach (meta learning)
  - tested on simplistic calorimeters (Par04)
  - extended to realistic detectors (FCC-ee)
    ACAT poster
- Finalising paper in the Journal of Physics Letters B

New ML architectures

- Exploration of transformers to build an accurate and general generative foundation model
- Collaboration with Openlab, CMS, and IBM
  - IBM expertise and (soon) resources
- First results discussed at Workshop on Foundation Models for detector simulation

Fast Simulation: Overview 2022 (2/2)

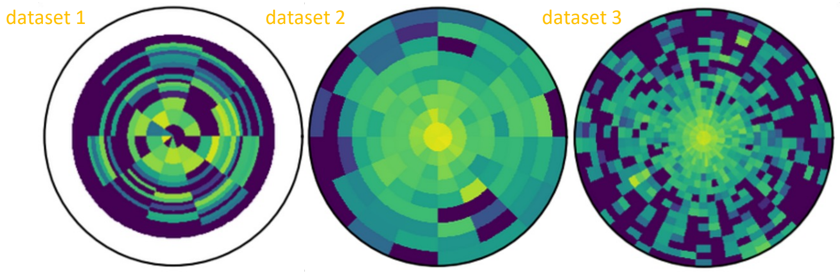


CaloChallenge

- First fast calorimeter simulation ML challenge
- Aim: spur the development and benchmarking of ML fast shower models
- Introduces 3 datasets with increasing difficulty (dimensionality):
  - ATLAS open data 10.5281/zenodo.6234054
  - 2 granularities of EM showers produced with Par04 10.5281/zenodo.6366270 and 10.5281/zenodo.6366323
- First contributions were published and presented during ML4Jets conference

Open Data Detector

- Benchmark detector for algorithmic studies
- Tracker is an evolution of a detector used in a Track-ML challenge
- Extension with EM calorimeter, performance tests in CI jobs completed



Fast Simulation: Plans for 2023

ML-based fast simulation

- Finalisation of MetaHEP paper, results

- Work on the new generative foundation model in collaboration with Openlab, CMS, and IBM
- Focus on accuracy of the model (reference is our previous VAE included in Par04, and other existing models)
- Explore quantitative metrics for evaluating generated showers
- Checkpoint around CHEP (or mid-year) to determine status, accuracy of the model for high granularity calorimetry
  - if successful: (hopefully) continuation with several detectors, incl. LHC experiments
  - otherwise: review strategy, on-going activities/models in the community

Open Data Detector

- Addition of HCal, muon system for completeness
- Validation, documentation
- Production and publication of Geant4 simulation data, serving as input and benchmark to software algorithm development

Geant4 code base (parameterisation)

- Generalisation of GFlash implementation using code developed for ML fast sim (common tools)
- Updates to Par04 example (realistic implementation of sensitive detectors)

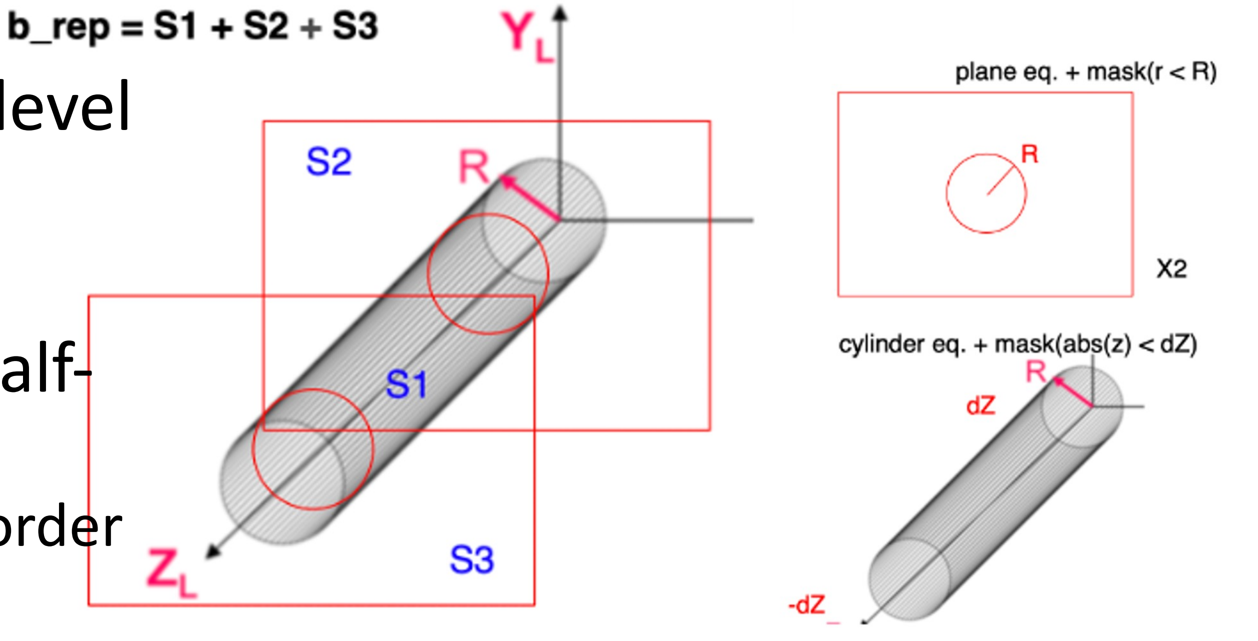
AdePT 2022 status

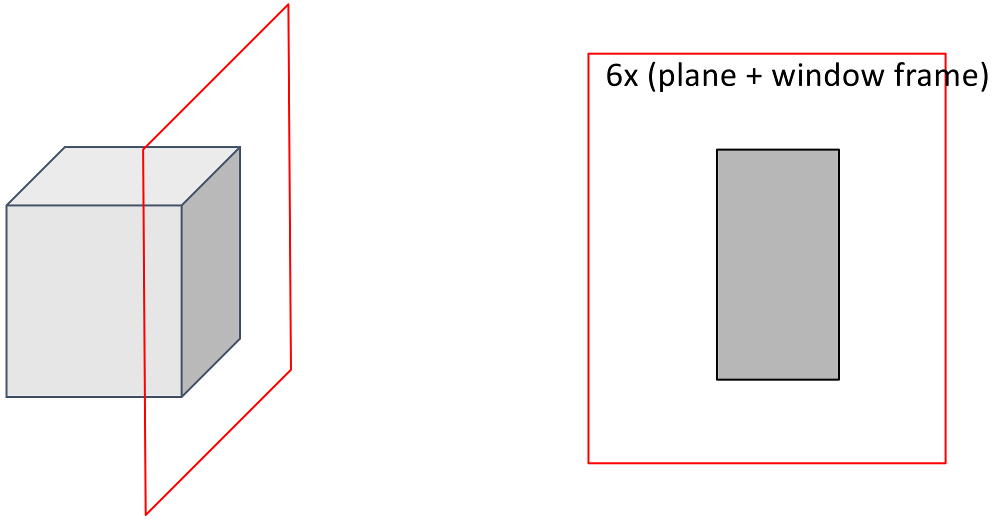
- First prototype for e-, e+ and gammas shower on GPU presented and discussed at HSF Detector Simulation on GPU Community Meeting
  - Full set of interactions of e-, e+, gammas (implemented by G4HepEm)
  - Navigation in complex geometry models using VecGeom (from slabs to CMS, read from GDML)
  - Propagation of charged particles in a magnetic field using helix for constant B-field
    - first version of propagation in non-constant B-field available
  - Simple hit generation code, which is then transferred from GPU to host
  - HepMC interface for input events (Pythia, etc)
  - Implemented both standalone and G4-integrated workflows (using fast simulation interface)

- Performance studies
  - split kernels, refactoring, memory layout, etc, for smaller register footprint and better work balancing
  - main performance bottleneck - current geometry model

Bounded surface modeling

- Decompose navigation at surface level
  - More simple and uniform code
  - Less branching/divergent sections
- Each face of a solid described as half-space + frame = FramedSurface
  - Accurate modeling (first + second order surfaces)
  - Frames defined in the simplest reference frame and carrying a transformation
- Carry hierarchy information for the object, mapping to the Geant4 geometry description





AdePT 2023 plans

- validation and debugging of the current AdePT prototype
- non-constant magnetic field propagator with realistic (CMS) field map
  - validation and optimisation

- experimental AdePT integration into experiments’ frameworks
  - study of the requirements for integration with experiments
  - focusing on user interfaces and handling of scoring
    - may be the main blocker, because transferring individual hits to host is likely performance prohibitive

- looking into improving the workflow and performance of the current integration prototype based on the fast simulation interface
  - currently GPU getting saturated with many CPU threads and acting as bottleneck

Surface geometry model 2023 plans

- Covering all solid types supported by Geant4
  - The surface model is already integrated in AdePT TestEm3 example, but currently only few solids are supported (boxes, tubes, trapezoids)
  - We need to write converters for an extended set, allowing to test more complex setups
- Support for hierarchical logical scenes instead of a single flat scene containing all touchable surfaces
  - The memory footprint in the flattened approach is too high for complex setups
- Optimized BVH navigation adapted for the surface model
  - Including further optimizations of the model, based on profiling on GPU

Celeritas: high-performance HEP detector simulation

- Motivated by HL-LHC computational challenges and by recent success in GPU MC (ECP ExaSMR)
- **GPU**-focused implementation of HEP detector simulation
- Multi-institution collaboration with external contributors (4–5 FTEs)
- Funded through US DOE ASCR/HEP (SciDAC 5, HEP base funding, leveraging ASCR ECP)

Core team:

Philippe Canal *(FNAL)*
Tom Evans *(ORNL)*
Seth R Johnson *(ORNL)*
Soon Yung Jun *(FNAL)*
Guilherme Lima *(FNAL)*
Amanda Lund *(ANL)*
Paul Romano *(ANL)*
Stefano Tognini *(ORNL)*

Active collaborations:

Ben Morgan *(Warwick)*
Geant4 SFT *(CERN)*
Julien Esseiva *(LBL)*
RAPIDS2 *(ORNL)*
US CMS *(FNAL)*

Upcoming collaborations:

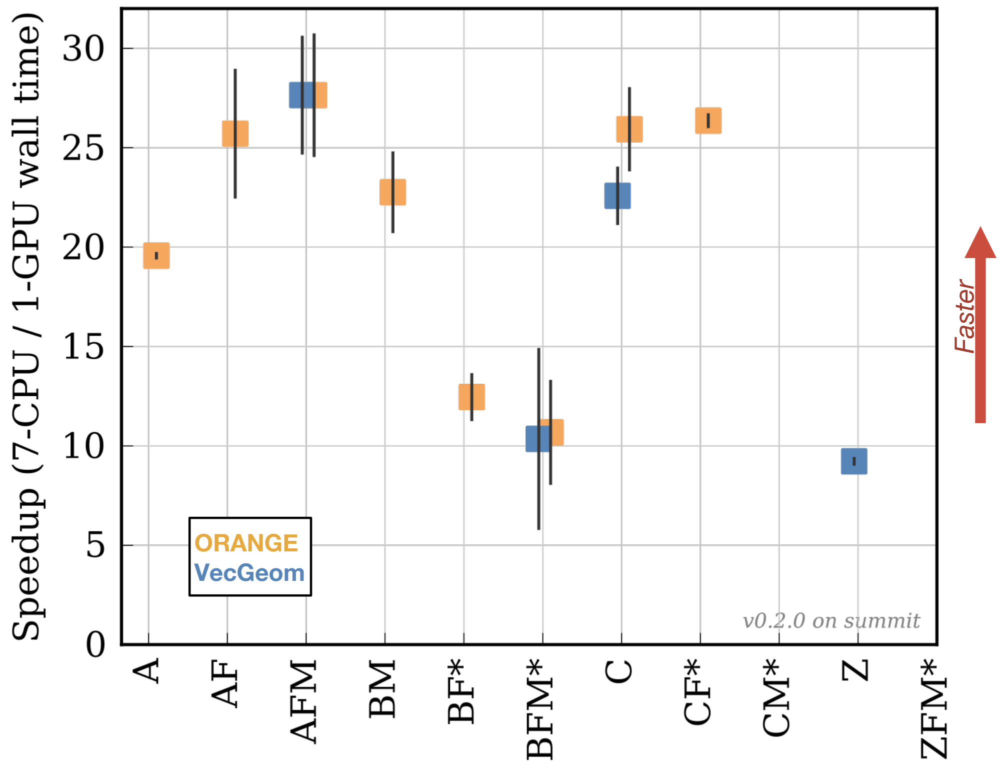
ATLAS, UKAEA, LZ, CalVision

EXASCALE COMPUTING PROJECT

https://github.com/celeritas-project/celeritas

Celeritas: current capabilities

- Roughly 10–25× speedup for tested problems using Celeritas on GPU vs CPU on a full Summit node
- Good agreement with Geant4 for preliminary test problems (energy deposition and step length distributions)
- Support for standard EM physics, GDML geometry, magnetic fields
- “MC truth” output and other diagnostics
- Easy-to-use interface for integrating directly into Geant4 to offload EM tracks
- "Fast simulation"-like capability for transferring GPU hits to user sensitive detectors
- Version 0.2.0 released January 2023

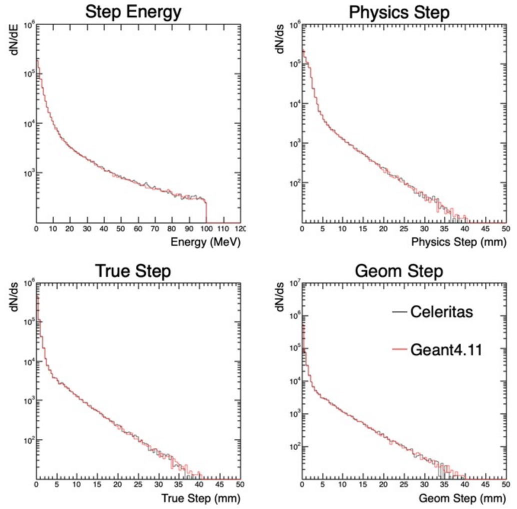


Problem definition

A	testem15
B	simple-cms
C	testem3
Z	cms2018

Modifier

F	1T field
M	MSC enabled
*	failures/unconverged

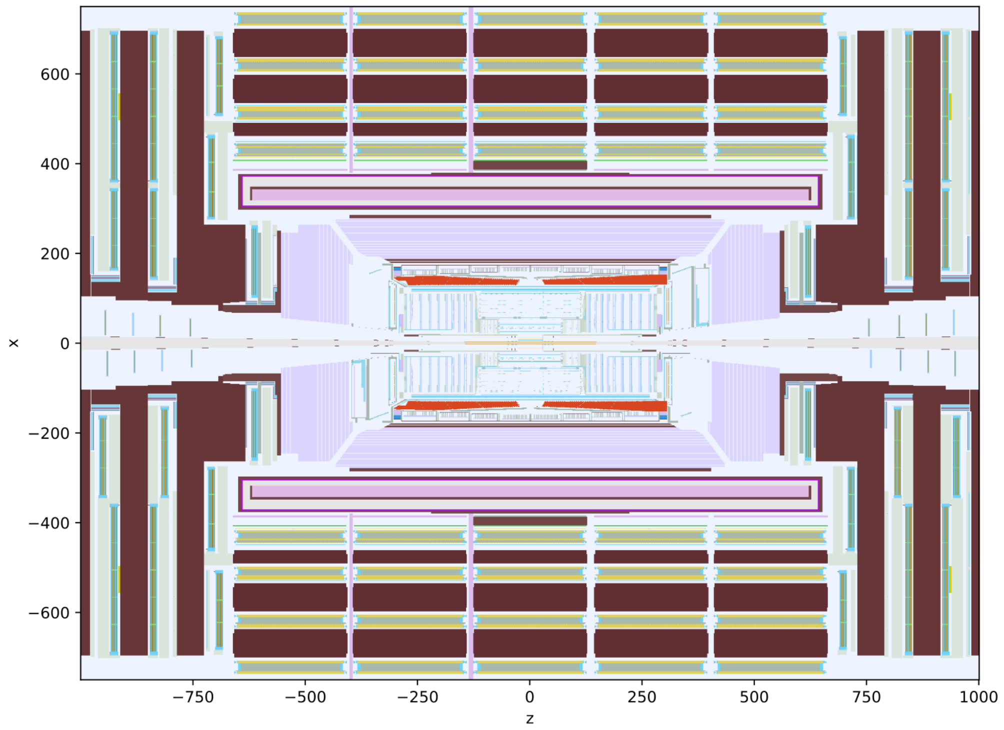


Urban MSC verification

https://github.com/celeritas-project/celeritas

Celeritas: near-term plans

- FY2023:
  - EM physics verification
  - Integration with CMSSW
  - Performance optimization
  - High-bandwidth MC truth (ADIOS)
- FY2024:
  - Photo/electro-nuclear physics
  - CMS readout optimization
  - Integration with ATLAS



CMS ray-traced with Celeritas+VecGeom

Interested in trying Celeritas EM offloading with your Geant4 application? Contact ***johnsonsr@ornl.gov***

https://github.com/celeritas-project/celeritas

Optical photon simulation on GPU using Opticks and CaTS

Opticks is an open-source project that accelerates optical photon simulation by integrating NVIDIA GPU ray tracing, accessed via NVIDIA OptiX.

Developed by Simon Blyth:

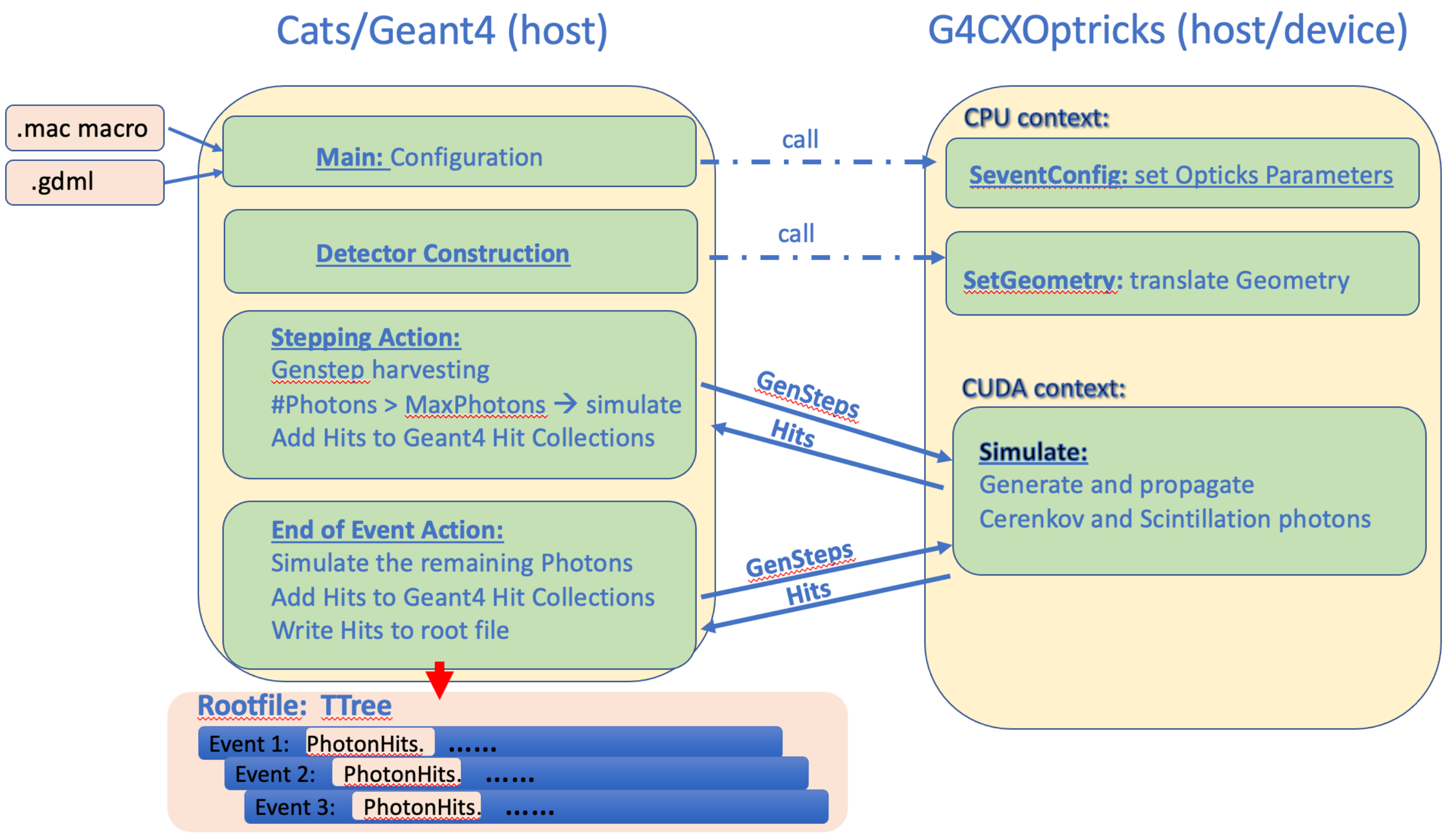
https://bitbucket.org/simoncblyth/Opticks/

CaTS: interfaces Geant4 user code with Opticks using the G4CXOpticks interface provided by Opticks. It defines a hybrid workflow where generation and tracing of optical photons is offloaded to Opticks (GPU), while Geant4 (CPU) handles all other particles. CaTS was included in Geant4 11.0 as an advanced example:

https://geant4.kek.jp/lxr/source/examples/advanced/CaTS/

https://github.com/hanswenzel/CaTS (development)

CaTS workflow using the new version of Opticks based on OptiX (7.x):



Plans

CaTS:

- Make the latest developments and documentation available in the Geant4/CaTS advanced example.
- Achieve true concurrency by using G4Tasking. Allow to configure jobs to fully utilize CPU and GPU resources.
- Change to use in-memory Root file merging (TBufferMerger) when using multi-threading.

G4CXOpticks/Opticks:

- Use the same implementation of the scintillation process on CPU (Geant4) and GPU (Opticks), use the same optical properties/keywords.
- Implement Wavelength shifting process (WLS).

Integration of Opticks with experimental frameworks:

- Starting with framework used by liquid Argon TPC community.

Conclusion

- plan to continue along the already established R&D directions
- ML-based models getting more matured with more advanced techniques involved
  - new generative foundation model under investigation
- GPU-based prototypes providing valuable insights into their application for general HEP transport
  - several bottleneck identified and are in the plan of work for next year
- specializes GPU applications (optical photons) showing already excellent results
  - further improvements planed