

Testing and Performance Evaluation of AdePT in Gaussino

James Peter Gomes

Department of Physics, Independent University, Bangladesh

Center for Computational and Data Sciences, Independent University, Bangladesh

Supervisors: Adam Morris, Witold Pokorski

Date: 16/08/2024

Abstract

To meet the increasing computational demands of LHC experiments, this study explores the potential of GPU acceleration for electromagnetic simulations. The AdePT prototype is integrated into the Gaussino framework and compared to the standard GEANT4. Initial results show that AdePT, while requiring optimization, demonstrates promising performance gains, especially with larger workloads. The successful simulation of electrons using AdePT highlights its potential for accelerating LHCb simulations and contributing to future physics analyses.

1 Introduction

The future LHC runs and upgrades will require the experiments to simulate a larger number of events in order to provide enough statistics to reach the expected accuracy of the physics analysis. One potential way of achieving this is by exploiting GPU-based computing resources. The development of the GPU-based simulation prototype for electromagnetic physics called AdePT (Accelerated demonstrator of electromagnetic Particle Transport) has reached the stage where all the necessary ingredients (physics, geometry, scheduling) for detector simulation are available. The next step is to integrate this prototype into the Gaussino simulation framework by providing all the necessary interfaces. This will demonstrate the use of AdePT in the LHCb simulation setup. The main goal of the project is to explore the interfacing between GEANT4 and AdePT and to see how (and how efficiently) this can be used in concrete applications like Gauss.

2 LHCb Simulation Software

2.1 Gauss

Gauss is an application that performs two of the main steps of simulating a particle physics experiment:

- **event generation**: everything that goes on in the particle collisions as well as the decays of short-lived particles
 - performed by several different external packages developed by theorists/phenomenologists, e.g. Pythia8, EvtGen, BcVegPy, EPOS
- **detector simulation**: the passage of the longer-lived particles through the detector, and their interactions with the detector material
 - predominantly done with GEANT4
 - growing suite of alternative techniques partially or fully replacing GEANT4 (parametric simulation, **GPU-accelerated**, ...)

2.2 Gaussino

Gaussino is both a **library** created from extracting the experiment-independent functionality of Gauss, upon which a full simulation application (such as Gauss) can be built, as well as a **stand-alone application** useful e.g. for rapid prototyping of a new experiment.

3 AdePT

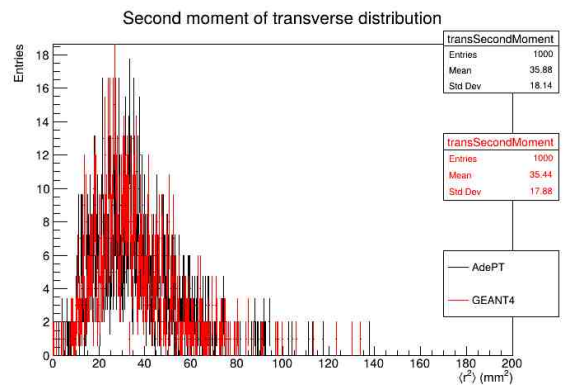
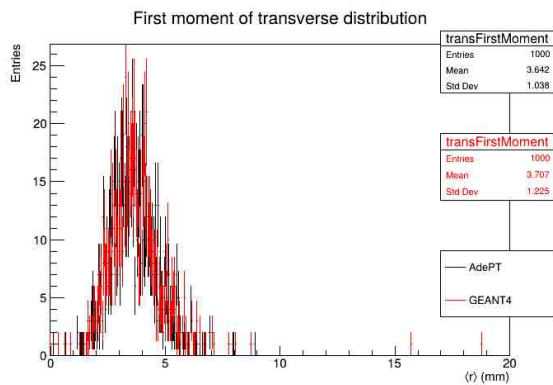
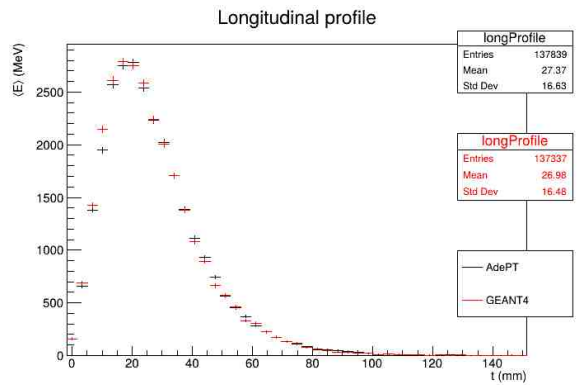
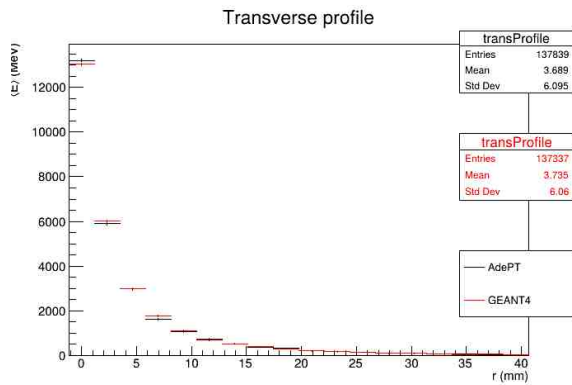
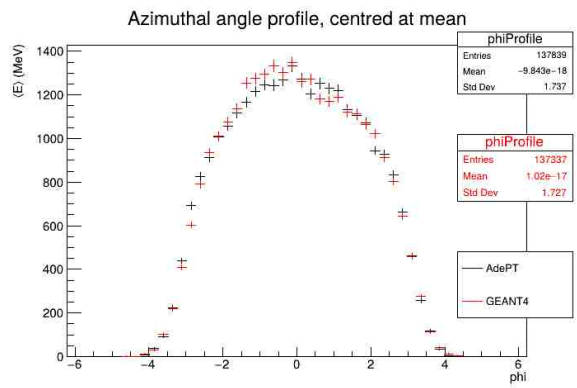
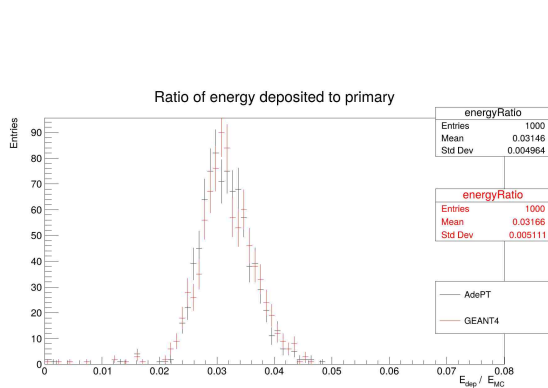
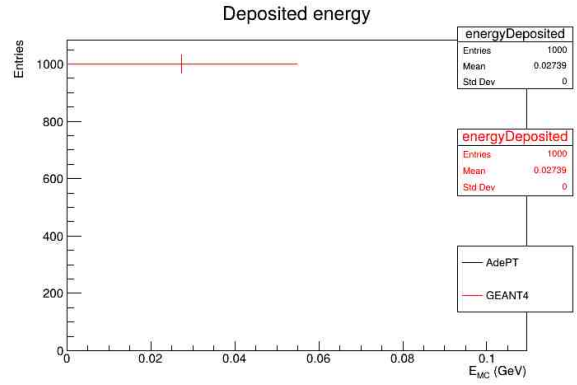
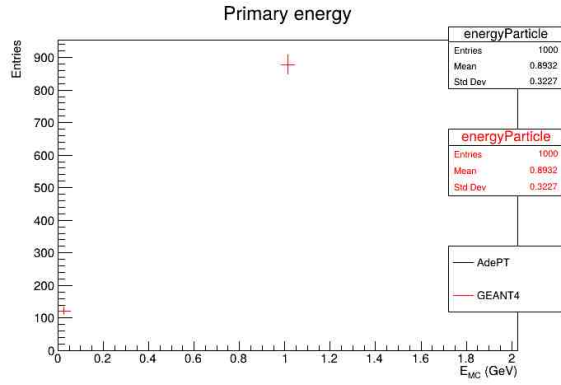
AdePT is an R&D project which aims to explore the possibility for running Monte Carlo simulations for HEP on GPUs. It focuses on the electromagnetic physics simulation. It can be run standalone (for development, testing or EM physics-only simulation), but its ultimate goal is to be called from within GEANT4 in order to offload the electromagnetic part of the particle transport to the GPU and gain some speedup in the overall simulation time.

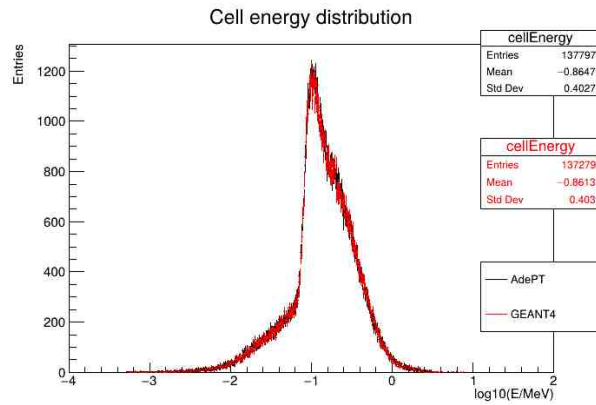
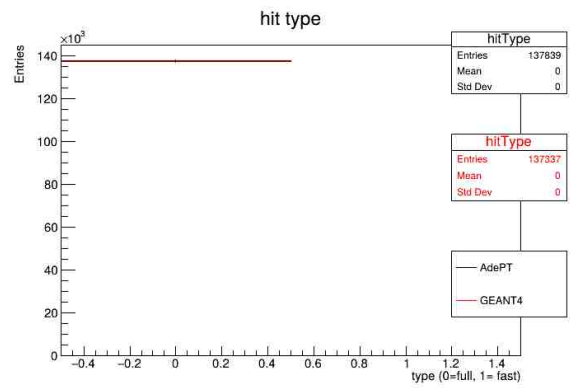
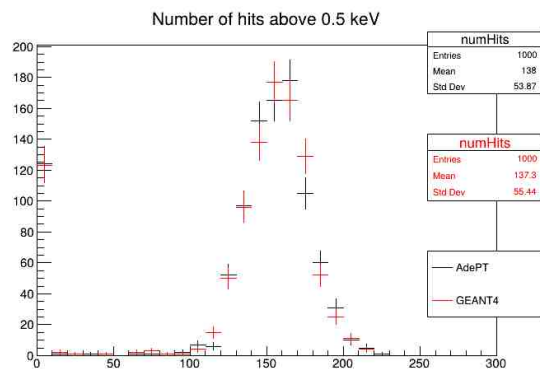
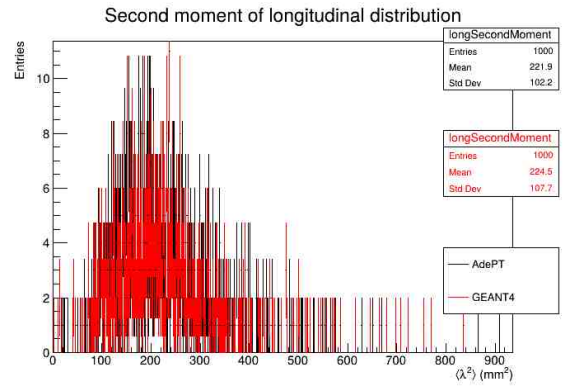
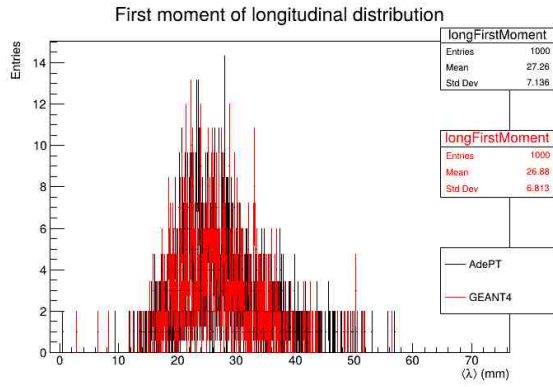
4 Setup and Geometry

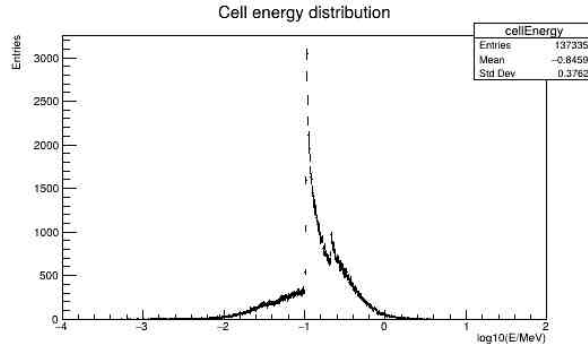
In this work, we have used **ParticleGun** to generate and send electrons (pdg[11]) towards somewhat in the middle of the ECAL sub-detector. We want to see how the detector responds to these hits in case of both AdePT and GEANT4 and compare the results. For our purpose, we have generated 1000 events, using a particular detector geometry, **CaloChallenge** (<https://gitlab.cern.ch/Gaussino/Gaussino/-/tree/mimazure-calo-challenge/Sim/CaloChallenge/python/CaloChallenge>).

5 GEANT4 + AdePT vs GEANT4

For the CaloChallenge detector, the following plots demonstrate the performance of AdePT into Gaussino vs Gaussino stand-alone:







Initially, what we had for cell energy distribution, is shown in the above histogram:

The discrepancy is because, initially, energy loss fluctuations were switched off for AdePT, although they were implemented in **G4HepEm**. AdePT, being GPE-accelerated, is expected to give a faster simulation than GEANT4 or Gaussino stand-alone, but that was not the case in this particular part of the project. But, a different setup:

```
set_particle_gun_momentum_range(
min_particles_no=1000,
max_particles_no=1000,
pdg_codes=[22],
min_momentum=1.0*units.GeV,
max_momentum=1.0*units.GeV,
theta_min=0,
theta_max=constants.pi,
)
```

of the simulation has caused AdePT (Throughput [1/s] : 2.762e+00) to run over 7 times faster than GEANT4 or Gaussino stand-alone (Throughput [1/s] : 3.550e-01). Initially, we were sending 1 gamma per event, but this time, we sent 1000 gammas per event.

6 Conclusion

For the part of the project where we used CaloChallenge detector geometry, AdePT and GEANT4 surely have really good agreement, as can be inferred from the histograms, but the simulation has taken much longer time for AdePT rather than GEANT4. Sending 1000 gammas instead of 1 gamma per event caused AdePT to run faster than GEANT4 and this demonstrates clearly that in order to benefit from GPU simulation, one needs to have a lot of particles per event sent to it. So, maybe depending on the events one may need to decide whether one wants to use the GPU or not. The main current goal of AdePT is to prove that we can run general HEP simulation on GPUs and that we are getting correct physics results, which, in this project, we have successfully shown by producing several good plots. This study shows that GPU-based simulation can be cannot automatically make all the simulations faster. Due to the specificity of the GPUs there is the condition of having many particles to be simulated at once which cannot be always satisfied. AdePT is continuing the R&D to improve the workflows and to allow an efficient use of the GPUs in as many cases as possible. This particular work is going to be one of the Gaussino examples that will be released and available for everyone to try out

Bibliography

- [1] <https://github.com/apt-sim/AdePT>
- [2] <https://gitlab.cern.ch/Gaussino/Gaussino/-/tree/mimazure-calo-challenge/Sim/CaloChallenge/python/CaloChallenge>
- [3] https://lhcb-gauss.docs.cern.ch/master/getting_started/first_simulation_job.html
- [4] https://geant4-userdoc.web.cern.ch/UsersGuides/PhysicsReferenceManual/html/electromagnetic/energy_loss/fluctuations.html
- [5] <https://lhcb-gauss.docs.cern.ch/master/>
- [6] <https://gaussino.docs.cern.ch/master/>