EPJ Web of Conferences **337**, 01267 (2025) https://doi.org/10.1051/epjconf/202533701267
CHEP 2024

A point library for the fast simulation of the LHCb calorimeter

Sergei Kholodenko[1,*], *Michal* Kreps[2], *Michal* Mazurek[3], and *Matteo* Rama[1,**]

[1]INFN Sezione di Pisa, Pisa, Italy
[2]Department of Physics, University of Warwick, Coventry, United Kingdom
[3]National Center for Nuclear Research (NCBJ), Warsaw, Poland

Abstract. The simulation of physics events in the LHCb experiment uses the majority of the experiment's distributed computing resources. Most of the computing time is spent on the Geant4-based detailed detector simulation, with approximately 50% of it dedicated to the calorimeter system. Optimizing the calorimeter simulation is therefore crucial for improving the overall computational efficiency. This article presents a solution implemented in the LHCb simulation software to accelerate the calorimeter simulation. During Geant4 transport, the simulation of particles entering the calorimeter is halted and hits are generated using libraries of pre-simulated energy deposits, which undergo a series of transformations to improve simulation accuracy without increasing the library size. This technique reduces the calorimeter computation time to a negligible level without significant loss of accuracy. A comparison between the outputs of the fast and detailed simulations is presented.

1 Need of a faster simulation of the LHCb detector

The LHCb experiment [1] operating at the LHC at CERN aims to search for physics beyond the Standard Model by studying the decays of b and c quarks, which are abundantly produced in proton-proton collisions at a center-of-mass energy of approximately 14 TeV.

Simulating a sufficient number of physics events is essential to achieving the experiment's scientific goals. For this reason, around 80% of the CPU resources available to the experiment during Run 2 of the LHC were allocated to generating and simulating Monte Carlo events [2]. In the ongoing Run 3, the nominal luminosity is five times higher than in Run 2 and the trigger efficiency has increased by a factor of 2 for most decay channels [3]. As a result, the fraction of total CPU resources expected to be required for Monte Carlo simulations in next few years has increased to approximately 90% [4]. Nonetheless, to avoid limiting the precision of certain measurements, it would be desirable to simulate an even larger number of events. In this context, the need to speed up event simulation while maintaining the same quality is even more pressing.

Two applications, Gauss and Boole [5], are used to simulate physics events in LHCb. Gauss is responsible for event generation, using external generators such as Pythia [6], and for simulating particle transport through the detector volume with the Geant4 toolkit [7, 8].

*e-mail: sergey.kholodenko@cern.ch
**e-mail: matteo.rama@cern.ch

Boole emulates the detector response through the digitization process. Both applications are based on the Gaudi software framework [9]. Since the average CPU time required for event generation and digitization is an order of magnitude smaller than that spent by Geant4 on particle transport, efforts to speed up detector simulation must begin with the latter. In a typical physics event, more than 50% of CPU time is spent in the calorimeter, making it the natural starting point for developing a faster detector simulation.

In LHCb, a parametric simulation of the high-level response of the electromagnetic calorimeter, i.e., at the level of reconstructed quantities, is available to users [10], while a hit-level simulation based on machine learning techniques is under development [11]. This paper presents a hit-level fast simulation of all calorimetric systems in the LHCb detector—not limited to the electromagnetic calorimeter—which is currently at an advanced stage of development. The method combines libraries of pre-simulated energy deposits with a series of transformations to enhance accuracy. This approach and that based on machine-learning produce the same type of output and could, in principle, be used complementarily depending on performance in different kinematic regions or for specific particle types. All three fast simulations are integrated into the Gauss simulation framework [12].

2 The LHCb calorimeter

The LHCb calorimeter subsystem comprises an electromagnetic calorimeter (ECAL) and a hadronic calorimeter (HCAL). During Runs 1 and 2, two additional scintillator pad detectors, SPD and PS, with a 12-mm-thick lead absorber installed between them, were used in front of the ECAL as a pre-shower, as shown in the left plot of Fig. 1.

The electromagnetic calorimeter is based on Shashlik technology and consists of 3312 modules, with three regions in granularity, making a total of 6016 cells. Each module is constructed by 66 alternating layers of 2 mm thick Pb, 120 μm reflective material (Tyvek) and 4 mm thick scintillating tiles. The resulting total depth is 25 X_0. The transverse size of the cells is 40.4×40.4 mm^2, 60.6×60.6 mm^2 and 121.2×121.2 mm^2 in the inner, middle and outer regions, respectively.

The hadronic calorimeter is a sampling calorimeter made of steel absorber and scintillating tiles oriented along the beam axis. It is divided into two regions with different granularities, featuring cell sizes of 131.3×131.3 mm^2 in the inner region and 262.2×262.2 mm^2 in the outer region. The ratio of steel to scintillator is 5.5 to 1, with a total depth of 5.6 interaction lengths.

3 Calorimeter fast simulation

3.1 Overview

When fast simulation is enabled, the generation of hits in the calorimeter by Geant4 is replaced by the simulation of energy deposits using a fast simulation model. This is implemented as a class inheriting from *G4VFastSimulationModel*, an abstract base class in Geant4 that provides an interface for fast simulation. The process is triggered when particles entering the *G4Region*[1] of the calorimeter satisfy the user-defined criteria in terms of particle type, momentum, and angle of incidence on the calorimeter front face (right plot of Fig. 1). Currently, models have been developed for photons and electrons/positrons entering through the front face of the calorimeter. Additionally, a dedicated model has been implemented to cover all particle types entering the calorimeter from the beam pipe region.

[1] A Geant4 class which groups together different geometry volumes, typically a subsystem.

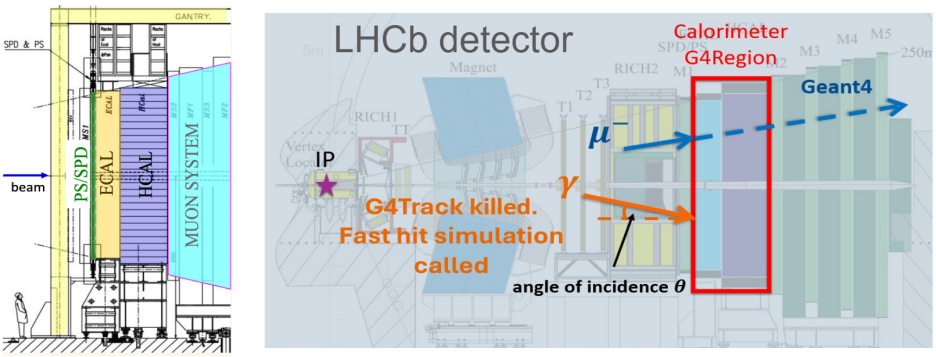

Figure 1. Left: The LHCb calorimeter system in Run 1 and Run 2 [13]. In Run 3, the SPD+PS system has been removed [14]. Right: Sketch illustrating the fast simulation interface, with a muon and a photon entering the calorimeter region. The muon is fully simulated by Geant4, while the photon track is killed and its Geant4 response is replaced by the fast simulation model described in Sec. 3.

When a particle is selected, the corresponding Geant4 track is killed and the energy deposits, which would have been simulated by Geant4, are instead extracted from the library according to the particle type and its p and θ values. The energy deposits then undergo a series of transformations that account for the exact values of the particle's momentum and entry position. Finally, they are converted into calorimetric hits.

3.2 Library of pre-simulated energy deposits: the point library

The information in the library is structured into (p, θ) nodes, representing specific values of momentum and incidence angle for which energy deposits have been pre-simulated and stored. The energy deposits were generated by particles entering the calorimeter at a fixed entry point and with a fixed azimuthal angle[2] ϕ. Currently, the library includes 80 values of p and 13 values of θ, with p varying logarithmically from 20 MeV/c to 200 GeV/c and θ ranging from 0 to 0.6 radians in steps of 0.05 radians. These p, θ ranges account for almost all of the CPU time used to simulate photons and $e^{\pm}$ in the calorimeter in typical LHCb physics events. For each (p, θ) node, 100 collections of pre-simulated energy deposits are stored. Each energy deposit is computed as the sum of *G4Step*[3] energies deposited in a square pseudocell, whose area is 25 times smaller than that of the inner calorimeter cells. This ensures precise energy assignment to the calorimeter cells after transforming the energy deposit positions according to the actual entry point and momentum of the particle. A single collection includes the energy deposits for all calorimeter subsystems: ECAL and HCAL in Run 3, and, additionally, SPD and PS in Run 1 and Run 2. The correlation between hits from different subsystems is preserved, except for the ECAL hit energy, which is regenerated at runtime as described in Sec. 3.3. Tests comparing fast and detailed simulation show no issues related to this, at least for photons and $e^{\pm}$, with excellent agreement also in particle identification variables, as shown in Sec. 4. A point in the library is characterized, in addition to its energy, by its position, deposition time, and an index indicating the calorimeter to which it belongs. The total library size is approximately 90 MB for photons and 90 MB for $e^{\pm}$.

[2]The azimuthal angle is the angle measured in the plane parallel to the front face of the calorimeter.
[3]A Geant4 class that represents a step a particle takes during the simulation.

3.3 Hit energy generation

The main limitation of using a hit library to simulate the calorimeter response lies in the number of collections it contains, which is constrained by the amount of volatile memory available for loading the library on the machine running the simulation. This limitation affects the statistical variability of the energy in calorimeter clusters. While the impact is similar across all calorimeters, its consequences are particularly significant for the electromagnetic calorimeter. To address this issue, the distributions of the logarithm of the cluster energy[4] in the ECAL were obtained, averaged over a given interval of ϕ and a given entry region, for each (p, θ) node. A 10-degree binning was applied to ϕ, while the entry regions are illustrated in the left plot of Fig. 2. Each of these distributions was fitted using the sum of two Crystal Ball functions. The resulting PDF parameters are stored in the library and used at runtime to generate the cluster energy and scale the energy deposits accordingly. The right plot in Fig. 2 presents a typical example of a fit to the cluster energy distribution, highlighting how the function accurately describes the tail of the distribution. This leads to an excellent reproduction of the tails in the reconstructed quantities that depend on the energy measurement in the ECAL, as illustrated, for example, in Fig. 6.

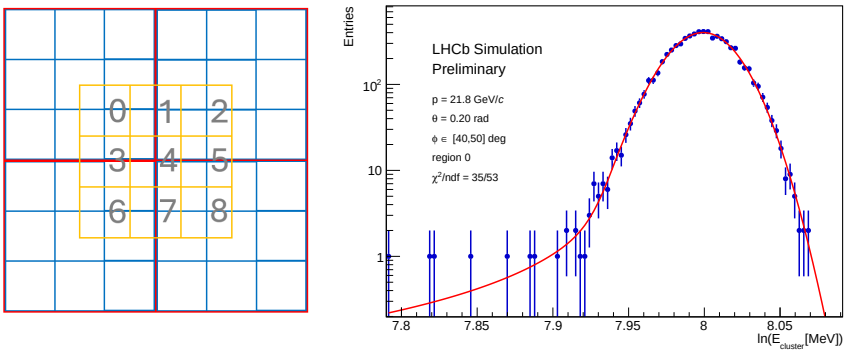

Figure 2. The orange grid in the diagram on the left illustrates the definition of the particle entry regions, used to construct the PDFs stored in the point library. The red square represents an ECAL module, while the light blue squares correspond to the cell sizes in the inner region. The plot on the right shows a typical example of a fit to the ECAL cluster energy distribution. More details can be found in Sec. 3.3.

3.4 Spatial coordinate transformation and inter-node interpolation

For a particle entering the calorimeter from an arbitrary position and with a generic momentum, a series of corrections must be applied. The first consists of an azimuthal rotation of the points to account for the difference between the azimuthal angle of the incident particle and that used in the library generation. The second correction addresses the fact that, in general, the particle's p and theta do not exactly match any of the library nodes. This correction is applied to both the spatial coordinates of the points and their energy, scaling linearly with the distance of (p, θ) from the nearest nodes. Finally, a global translation is applied to the position of the points, corresponding to the difference between the actual entry position of the particle and the reference position used to generate the library.

[4]The cluster energy is obtained by summing the energy of all points.

3.5 Particles entering from the beam pipe region

So far, we have described the implementation of fast simulation for photons and $e^{\pm}$ which enter the calorimeter through its front face, that is, those originating from the interaction region and thus of particular interest for reconstructing the physics events. However, a non-negligible fraction of the CPU time, approximately 20%, is spent by Geant4 simulating the propagation of particles originating from the beam pipe. Most of these particles are absorbed by the calorimeter's shielding material, but those that succeed in entering contribute approximately 6% of the total energy deposited in the calorimeter. To account for these contributions, a dedicated library has been developed, storing the hits produced by all particles coming from the beam pipe in a given event. When the library is enabled, the *G4Track* of all particles from the beam pipe region is killed and a hit collection is picked from the library. It was verified that the hit distribution does not significantly change between events in which a $b\bar{b}$ or $c\bar{c}$ pair is produced, making the same library applicable to most proton-proton collision events of interest at LHCb. A library containing 1000 events has a size of approximately 40 MB.

4 Performance

The performance of the fast simulation based on the point library is evaluated through the simulation, reconstruction, and selection of $B^0 \to K^{*0}\gamma$ decays, with $K^{*0} \to K^+\pi^-$, and $B^+ \to J/\psi K^+$ decays, with $J/\psi \to e^+e^-$. Unless otherwise specified, events are fully simulated, including all particles originating from the proton-proton interaction and entering the detector acceptance, with the detector configuration set to 2024. The comparison is performed between two configurations. In the first, all particles are simulated using detailed simulation based on Geant4, which is the standard for LHCb. In the second, photons and $e^{\pm}$ reaching the front face of the calorimeter are suppressed, and their energy deposits are simulated using the point library.

Figure 3 compares the reconstructed mass of the $B^+ \to J/\psi K^+$ mesons obtained with standard simulation and the point library, while Fig. 4 presents the same comparison for the J/ψ mass. In both cases, an excellent agreement is observed. Figure 5 compares the fast and detailed simulation for two variables used to identify electrons and positrons from J/ψ decays, again showing very good agreement. In this case, the detector configuration corresponds to 2016, and only the decay channel is simulated in each event.

Finally, Fig. 6 shows the comparison of the reconstructed mass for $B^0 \to K^{*0}\gamma$ decays. The reconstruction and selection efficiency for this channel is 3.75 ± 0.03 for the detailed simulation and 3.68 ± 0.03 for the fast simulation, in agreement at a relative precision better than 1%.

The simulation of the calorimeter based on the point library is approximately three orders of magnitude faster than the detailed simulation. This is illustrated in Fig. 7, which shows the average CPU time spent in the calorimeter to simulate a single photon as a function of its energy at the entrance of the calorimeter. The same results apply to $e^{\pm}$.

5 Conclusions and perspectives

In the previous section we showed that the fast simulation of photons and $e^{\pm}$ in the calorimeter based on point libraries accurately reproduces the Geant4 simulation while reducing computation time by a factor of more than 1000. In the standard LHCb simulation, 15% of the time spent by Geant4 in the calorimeter is dedicated to photons and another 15% to $e^{\pm}$, while the propagation of particles from the beam pipe accounts for an additional 20%. Therefore, the fast simulation described here covers 50% of Geant4's total CPU time in the calorimeter.

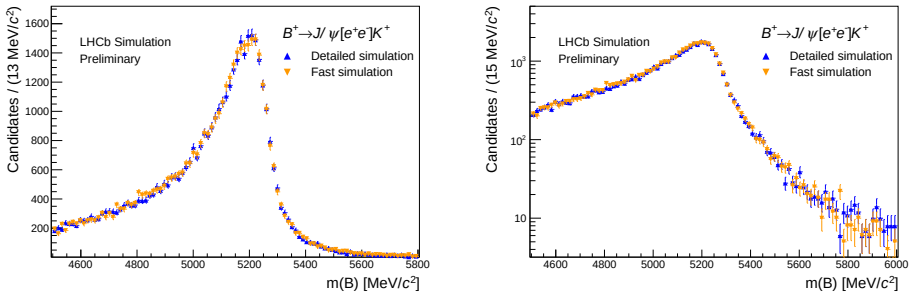

Figure 3. Comparison of the reconstructed $B^+ \to J/\psi[e^+e^-]K^+$ mass spectra using the detailed simulation of the LHCb calorimeter based on Geant4 (blue histogram) and the fast simulation based on the point library (orange histogram). The 2024 detector configuration has been used.

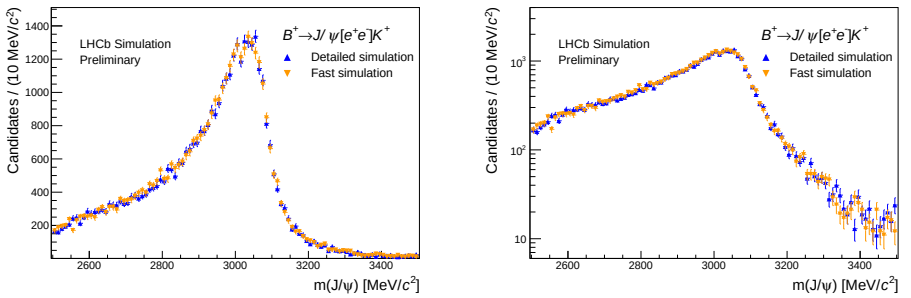

Figure 4. Comparison of the reconstructed $J/\psi \to e^+e^-$ mass spectra using the detailed simulation of the LHCb calorimeter based on Geant4 (blue histogram) and the fast simulation based on the point library (orange histogram). The 2024 detector configuration has been used.

The code is currently under internal review within the LHCb Collaboration with the goal of deploying it in production this year.

The next goal will be to extend the fast simulation to particles contributing to the remaining 50% of processing time, including charged pions and kaons (accounting for 30%), protons and neutrons, thereby approaching full coverage. Fast simulation of hadronic showers will require addressing new challenges, such as accounting for punch-through into the muon system and the increased statistical fluctuations of the showers.

References

[1] LHCb collaboration, A. A. Alves Jr. et al., The LHCb detector at the LHC, JINST **3**, S08005 (2008). https://doi.org/10.1088/1748-0221/3/08/S08005

[2] C. Bozzi, LHCb Computing Resource usage in 2018, LHCb-PUB-2019-004. https://cds.cern.ch/record/2657833

[3] LHCb collaboration, R. Aaij et al., The LHCb Upgrade I, JINST **19**, P05065 (2024). https://doi.org/10.1088/1748-0221/19/05/P05065

[4] LHCb collaboration, Computing Model of the Upgrade LHCb experiment, CERN-LHCC-2018-014, LHCB-TDR-018. https://doi.org/10.17181/CERN.Q0P4.57ON

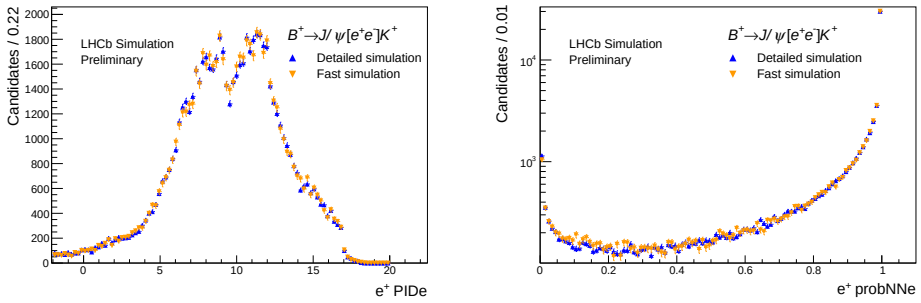

Figure 5. Distribution of two particle identification variables for positrons from $B^+ \to J/\psi[e^+e^-]K^+$ decays. The blue histograms represent the detailed LHCb calorimeter simulation based on Geant4, while the orange histograms correspond to the fast simulation using the point library. The 2016 detector configuration has been used.

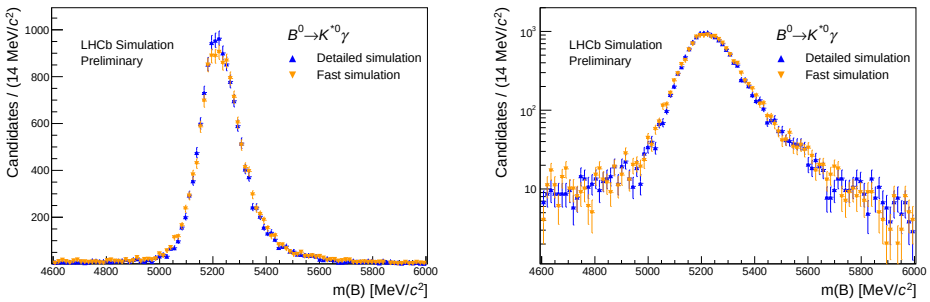

Figure 6. Comparison of the reconstructed $B^0 \to K^{*0}\gamma$ mass spectra using the detailed simulation of the LHCb calorimeter based on Geant4 (blue histogram) and the fast simulation based on the point library (orange histogram). The 2024 detector configuration has been used.

[5] M. Clemencic et al., The LHCb simulation application, Gauss: Design, evolution and experience, J. Phys. Conf. Ser. **331**, 032023 (2011). https://doi.org/10.1088/1742-6596/331/3/032023

[6] T. Sjostrand, S. Mrenna, and P. Skands, A brief introduction to PYTHIA 8.1, Comput. Phys. Commun. **178**, 852 (2008). https://doi.org/10.48550/arXiv.0710.3820

[7] Geant4 collaboration, S. Agostinelli et al., Geant4: A simulation toolkit, Nucl. Instrum. Meth. **A506**, 250 (2003). https://doi.org/10.1016/S0168-9002(03)01368-8

[8] Geant4 collaboration, J. Allison et al., Geant4 developments and applications, IEEE Trans. Nucl. Sci. **53**, 270 (2006). https://doi.org/10.1109/TNS.2006.869826

[9] G. Barrand et al., GAUDI — A software architecture and framework for building HEP data processing applications, Comput. Phys. Commun. **140**, 45 (2001). https://doi.org/10.1016/S0010-4655(01)00254-5

[10] L. Anderlini et al., The LHCb ultra-fast simulation option, Lamarr design and validation, EPJ Web Conf. **295**, 03040 (2024). https://doi.org/10.1051/epjconf/202429503040

[11] M. Mazurek, Generative AI for fast simulations in LHCb, CHEP 2024 conference. https://indico.cern.ch/event/1338689/contributions/6015805/

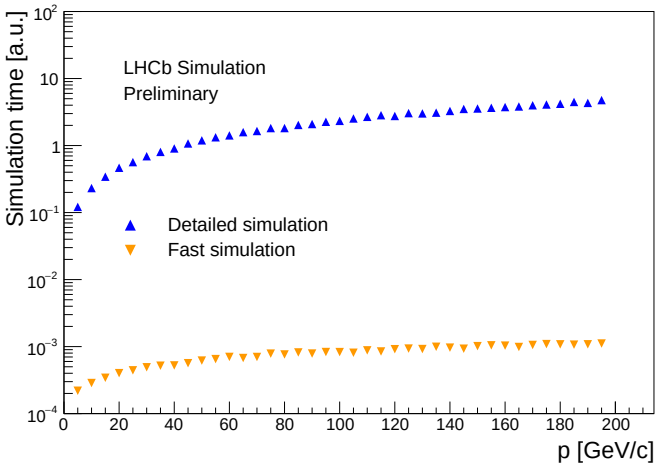

Figure 7. Average CPU time spent to simulate one electron in the LHCb calorimeter, as a function of its momentum, using the Run 2 detector configuration. Blue points: Detailed simulation using Geant4; Orange points: Fast simulation using the point library.

[12] G. Corti et al., New software technologies in the LHCb Simulation, J. Phys.: Conf. Ser. **2438**, 012108 (2023). https://doi.org/10.1088/1742-6596/2438/1/012108

[13] S. Amato et al., LHCb calorimeters: Technical design report, CERN-LHCC-2000-036. https://cds.cern.ch/record/494264

[14] R. Aaij et al., The LHCb Upgrade I, JINST **19**, P05065 (2024). https://doi.org/10.1088/1748-0221/19/05/P05065