A parameter optimisation toolchain for Monte Carlo detector simulation

Benedikt Volkel[1,*], *Matteo* Concas[1], *Sandro* Wenzel[1], and *Andreas* Morsch[1]

[1]CERN

Abstract. Monte Carlo detector transport codes are one of the backbones in high-energy physics computing. They simulate the transport of a large variety of different particle types through complex detector geometries based on different physics models. Those simulations are usually configurable through a large set of parameters allowing for some tuning on the client side. Often, tuning the physics accuracy on the one hand and optimising the resource needs on the other hand are competing requirements. In this area, we are presenting a toolchain to tune Monte Carlo transport codes which is capable of automatically optimising large sets of parameters based on user-defined metrics. The toolchain consists of two central components. Firstly, the MCReplayEngine which is a quasi-Monte-Carlo transport engine able to fast replay pre-recorded MC steps. This engine for instance allows one to study the impact of parameter variations on quantities such as hits without the need to perform new full simulations. Secondly, it consists of an automatic and generic parameter optimisation framework called O2Tuner. The toolchain's application in concrete use-cases will be presented. Its first application in ALICE led to a reduction of CPU time of Monte Carlo detector transport by 30%. In addition, further possible scenarios will be discussed.

1 Introduction

Monte Carlo (MC) transport simulations are resource intensive applications, in particular due to the navigation in highly complex detector geometries and the large number of physics computations. Increasing experimental data volumes as expected in LHC Run 3 require a compatible increase of the size of simulated samples in order to not be limited by the MC uncertainties for instance in precision measurements. Therefore, there is a strong incentive to speed-up MC transport. This paper presents a new parameter optimisation toolchain developed in ALICE that aims to tune parameter sets to speed up the transport with a negligible impact on physics accuracy.

The performance of a transport simulation depends roughly linearly on the number of simulated MC steps which are the atomic units in typical MC transport simulations. For a given particle, its next step is computed based on its possible interactions with the surrounding detector material. The invoked interaction then determines the path length of the step and the potential creation of secondary particles.

A significant reduction of the number of steps will result in a significant decrease of the overall runtime of the transport. The top row of Fig. 1 sketches possible paths of particles

*e-mail: benedikt.volkel@cern.ch

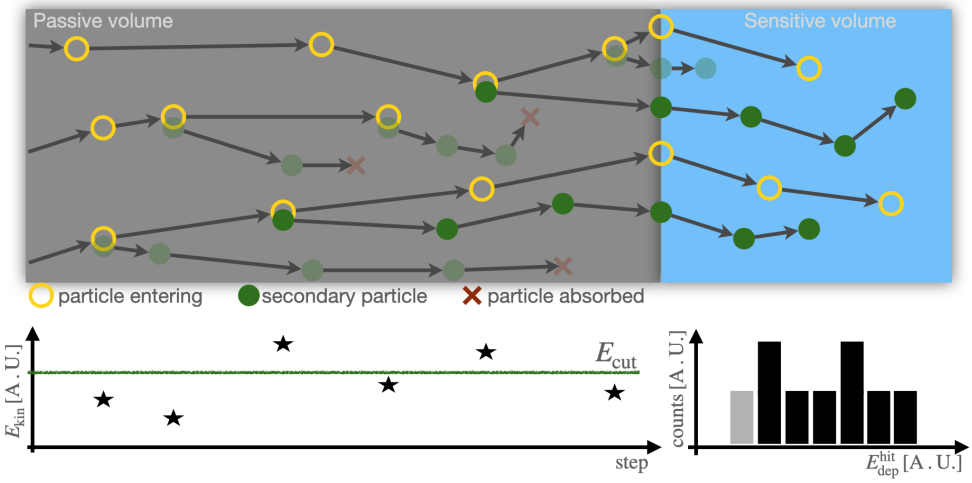

Figure 1. *Top*: Sketch of particles entering some passive detector material (yellow circles) from the left. Some of them produce secondary particles (green blobs) by material interactions. Some of these (low opacity) are already absorbed inside the passive volume (red cross) after a few steps. Others cross the boundary to a sensitive volume where energy deposits induce detector hits. *Bottom left*: Kinetic energy of secondary particles at their production point. *Bottom right*: Histogram indicating the energy of induced hits. Low opacity indicates hits that are not registered if secondary particles with $E_{\mathrm{kin}} < E_{\mathrm{cut}}$ are not transported.

entering a passive detector volume (yellow circles). During their transport, secondary particles might be produced by interactions with the material (green blobs). Some of them are absorbed inside the passive material (low opacity) after a few steps as indicated by the red crosses and they will not reach the sensitive detector region indicated in blue. Introducing an energy threshold cut on the kinetic energy E_{kin} of secondary particles at their production point, as indicated in the bottom-left plot, only secondary particles with $E_{\mathrm{kin}} > E_{\mathrm{cut}}$ will be transported. Hence increasing such cuts results in a lower number of steps while decreasing such cuts increases the number of steps. This might potentially remove some detector hits which would have been induced as indicated in the bottom-right histogram (low opacity). The goal of an optimisation procedure is to determine energy thresholds so that secondary particles below those thresholds are immediately discarded to reduce the number of MC steps to be computed. Thresholds shall be chosen such that the impact on the number of hits and therefore on final physics observables is negligible. A set of energy thresholds is derived per particle type and medium and will be denoted as $E_{\mathrm{cut}}^{\mathrm{particle,med}}$. Similarly to energy thresholds, also other criteria can be used in similar tuning studies such as geometrical ones. The presented study is the first automatic tuning of such a parameter set in ALICE transport simulations.

These proceedings are structured as follows: Section 2 describes the optimisation toolchain, introduces its components and outlines how they work together. Results of such an optimisation run on a realistic scenario are presented and discussed in Sec. 3. A summary, opportunities and an outlook are provided in Sec. 4

2 Optimisation toolchain

To quantify the impact of a chosen set of $E_{\mathrm{cut}}^{\mathrm{particle,med}}$, a simulation with the default parameter set is run first as a reference. In this reference run, all computed MC steps are recorded and written to disk using an instrumentation tool called MCStepLogger [1] as indicated in the top of Fig. 2. This allows us not only to study MC steps after a simulation but also to *replay* a simulation precisely step-by-step without the need to call the simulation engine. The latter is done with a tool that we call MCReplayEngine [1].

The MCReplayEngine was specifically developed for the presented project. It is a quasi-MC transport engine in the sense that it does not do any physics calculations but only injects the steps previously recorded with the MCStepLogger as indicated in the lower part of Fig. 2. This way, a simulation can be *replayed* any arbitrary number of times but under the application of different parameter values. As an implementation of the Virtual-Monte-Carlo (VMC) [2] it can be used as any other VMC-based transport engine which makes it suitable to be used in the O2 software stack of ALICE. It does not need to do any navigation through the detector geometry nor does it need to compute any physics interactions. Hence, the MCReplayEngine can be used as a fast way to evaluate the impact of parameter changes and therefore within parameter optimisation frameworks.

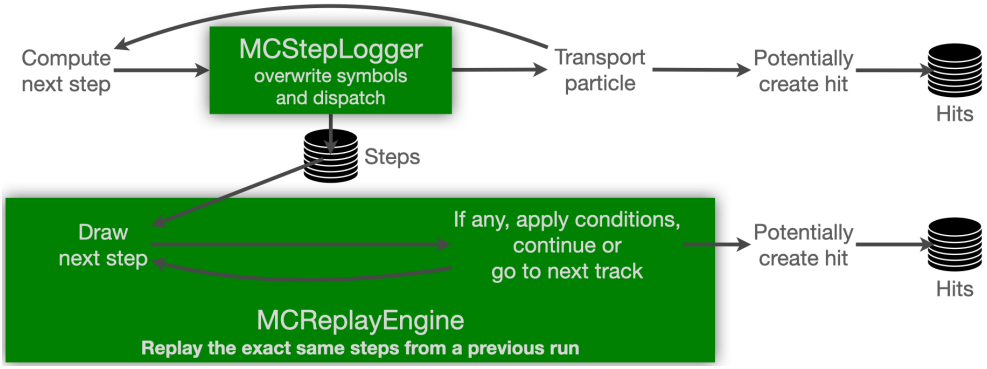

Figure 2. The working principles of the MCStepLogger (*top*) and the MCReplayEngine (*bottom*).

All necessary tasks of data preparation, optimisation, evaluation, and potential closure tests are integrated or part of a steering system called O2Tuner [3]. This is an optimisation toolset on the shoulders of the Optuna hyperparameter optimisation framework [4] and was developed in ALICE to conveniently optimise and tune parameter settings of various algorithms such as simulation, digitisation or reconstruction. O2Tuner makes parameter tuning easily reproducible, the recipes can be shared with other collaborators and it provides a way to make complex optimisation workflows persistent, reusable, and sustainable.

The optimisation workflow followed in the presented study is sketched in Fig. 3 showing how the aforementioned components are linked together. In each trial of the optimisation, a new set of energy thresholds is drawn to be used in the replay simulation. The *Tree-structured Parzen Estimator* (TPE) is chosen as the sampling algorithm for the parameter search.

The choice of the loss function should reflect the intention of the optimisation problem at hand. It is therefore chosen according to

$$L_{\mathrm{t}}\left(s_{\mathrm{ref}}, \{h\}^{\mathrm{d}}_{\mathrm{ref}}, s_t, \{h\}^{\mathrm{d}}_{\mathrm{t}}\right) = \frac{s_t}{s_{\mathrm{ref}}} + \frac{1}{N^{\mathrm{det}}} \sum_{\mathrm{d}\in\mathrm{det}} \alpha^{\mathrm{d}}\left(h^{\mathrm{d}}_t, h^{\mathrm{d}}_{\mathrm{ref}}\right)\left[1 - \frac{h^{\mathrm{d}}_t}{h^{\mathrm{d}}_{\mathrm{ref}}}\right],$$

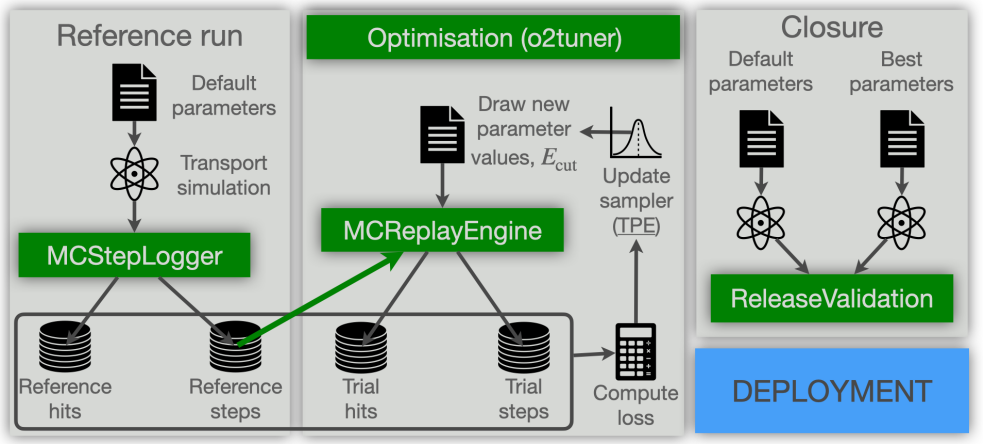

Figure 3. Sketch of the parameter optimisation workflow. A reference run (*left*) is used to assess the impact of a chosen parameter set during the optimisation (*middle*). Closure tests (*right*) are run as the final step to ensure the compatibility of the derived parameter set.

$$\alpha^{\mathrm{d}}\left(h_t^{\mathrm{d}}, h_{\mathrm{ref}}^{\mathrm{d}}\right) = \begin{cases} 10, & \text{if } h_t^{\mathrm{d}}/h_{\mathrm{ref}}^{\mathrm{d}} < 0.95 \\ 1, & \text{otherwise} \end{cases} \forall \mathrm{d}, \tag{1}$$

where s_t and s_{ref} denote the number of MC steps in a certain trial t and the reference run, respectively. The number of hits per detector in a certain trial and the reference run are denoted as h_t^{d} and $h_{\mathrm{ref}}^{\mathrm{d}}$. The ratio of hits can be scaled by a penalty factor α^{d} per detector. In this study it is chosen to be the same for each detector. This penalty factor ensures to increase the loss for those parameter sets that have a large impact on hits in at least one single detector. The sum of detector hits on the right-hand side is scaled by the number of available detectors N^{det}. By doing so, the value of the loss function can be compared independently of the active detectors. The loss function is minimised by minimising the number of steps s_t and keeping the number of hits per detector h_t^{d} as close as possible to the reference. A chosen parameter set is defined to be better the more it decreases the loss.

3 Results

To demonstrate the functionality of the toolchain, the tuning of energy thresholds as described in this section targets all simulation media of the ALICE beam pipe, further denoted as PIPE. It comprises a passive module in the detector geometry and only those particles (and their secondaries) that leave hits in any of the sub-detectors are of importance. Photons, electrons, and positrons are considered first while the transport of all other particles will not be affected by the optimisation. The reason for this is the observation that the majority of steps in the ALICE simulations is made in the PIPE volumes and these steps are caused by electromagnetic showers.

3.1 Optimisation

The left panel of Fig. 4 shows the relative number of steps in all detector modules before (blue) and after the optimisation (orange, hatches). The most significant reduction of steps is

achieved in the PIPE alone, namely by around 20%. These are the steps that were found to leave zero or a negligible number of hits later on. The second most steps are removed in the module denoted as CAVE which acts as the world volume in the simulation.

Importantly, while overall a reduction of the number of steps of about 30% is achieved, only around 1.6% hits have been removed in total as seen in the right panel of Fig. 4. In fact, some detector hits are not influenced at all and most others are impacted by approximately 2%. The largest reduction of hits is around 5% in the FV0 detector which matches the choice of the penalty factor α^{d} in Eq. (1).

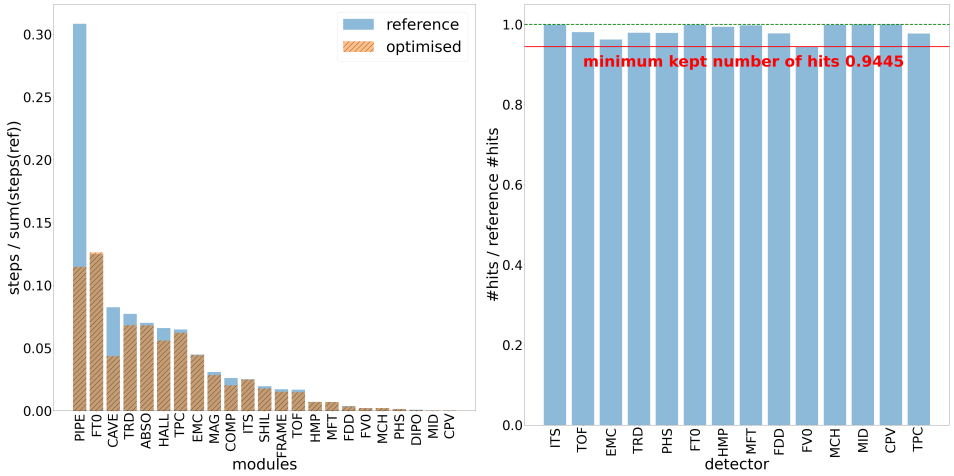

Figure 4. *Left*: Number of steps per detector module in a typical ALICE transport simulation. The blue histograms shows the relative number of steps when using the default parameter sets. The orange-hatched histogram shows the relative number of steps using the tuned parameter set. *Right*: The relative number of hits compared to the reference run per sub-detector when using the tuned parameter set.

The reduction of steps is achieved by increasing the energy thresholds of various media $E_{\mathrm{cut}}^{\mathrm{particle,med}}$. In the left panel of Fig. 5 the ratio of the tuned energy thresholds with respect to the reference values is shown. The color of each cell is proportional to the computed ratio. The two columns refer to photons (CUTGAM) as well as to electrons and positrons (CUTELE) while the rows indicate the various media present in the PIPE geometry. Around 50% of all parameters in this study were increased by the optimisation procedure. For instance, an increase by around three orders of magnitude of the energy threshold on photons in the medium PIPE_AIR_NF was found to be feasible.

3.2 Closure test

In order to asses the impact of the tuned parameters on the output of the full simulation chain, closure tests are done with MC samples. First, two independent simulations were run with the default parameter values and a third simulation was run with the tuned parameter set. The first result of the closure test is that the walltime using the optimised parameter set is observed to be $\approx$ 70% of that when using the default parameter set. That is well in line with the number of steps being reduced to around 70% in the optimisation as presented in Sec. 3.1 and it can be concluded that the newly found parameters have the expected impact on the runtime.

To evaluate potential impacts on the physics accuracy, the compatibility of 1074 *QualityControl* (QC) observables was checked which are are extracted for each of the three simu-

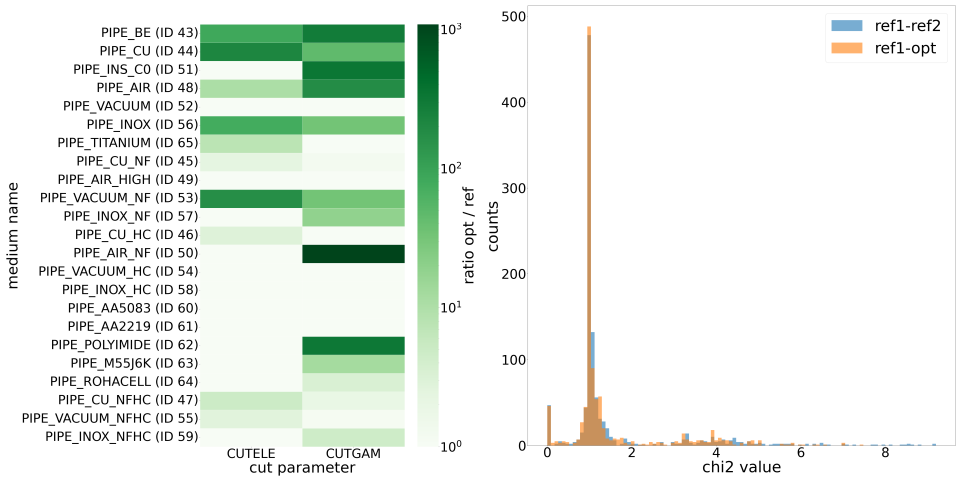

Figure 5. *Left*: The ratio of tuned parameter values with respect to their default values considering all media of the ALICE beam pipe. *Right*: Distribution of χ^2 values comparing two simulations with the default parameter set (blue) and comparing a simulation with the default parameter set to one using the tuned values (orange).

lation scenarios. There are many different observables, per detector, such as distributions of digit and track properties, vertex properties, and various correlations. These observables are then fed into the ALICE ReleaseValidation tool, which allows for a detailed and comprehensive comparison. Its main purpose is to validate software releases but it is well suited to study the impact of different parameter settings. It automatically extracts all observables from the QC files and runs a comparison based on several defined metrics such their χ^2 value. The distribution of the observables were compared between the two reference simulations conducted with the default parameter values and between a reference simulation and the one with the optimised parameter values. The right panel of Fig. 5 shows the histograms of χ^2 values from all comparisons. In blue, the comparison values of the two reference simulations are shown while the orange histogram shows the χ^2 values of the comparison between a reference and the optimised scenario. The χ^2 distributions of both comparisons show good agreement and it can be concluded that the physics accuracy with the tuned parameter set is compatible with the default values. This validates the optimisation procedure.

4 Summary, opportunities and outlook

A toolchain for parameter optimisation of Monte Carlo transport simulations was presented which is the first automatic tuning of such a parameter set in ALICE transport simulations. It consists of four core components, the MCStepLogger, the MCReplayEngine, the ALICE O2Tuner package, and the ALICE ReleaseValidation tool. It was shown how the toolchain can be used to decrease the number of steps in the ALICE simulation by around 30% while the impact on the physics accuracy is shown to be negligible. The MCReplayEngine has been explicitly developed for this effort and also some developments of the ReleaseValidation tool were driven by the presented optimisation effort. However, it should be noted that each single component can be used in other scenarios; O2Tuner can be applied to other parameter tunings, MCStepLogger can be used for stand-alone analyses of MC transport engines, and the

ReleaseValidation tool meanwhile serves as one of the main components for general software and MC productions validations.

Apart from the tuning of energy thresholds, the presented toolchain is well suited for other optimisation scenarios. For instance, one could imagine to tune geometrical parameters of the MC simulation or the number of queries of magnetic field parameters that are necessary to guarantee an accurate transport.

Transport simulations also play a crucial role in astrophysics or medical science such as radiation therapy. The presented toolchain, or at least parts of it, might be helpful to target parameter tuning also in those realms of science.

At present, the implementation of the MCStepLogger and the MCReplayEngine rely on the VMC interface description. Future work could target to remove this requirement to extend the two tools to be used with native GEANT4 or even to make the tuning chain part of GEANT4.

References

[1] MCStepLogger and MCReplayEngine, https://github.com/AliceO2Group/VMCStepLogger

[2] Hřivnáčová I et al: The Virtual MonteCarlo, ECONF C0303241:THJT006 (2003)

[3] O2Tuner, https://github.com/AliceO2Group/o2tuner

[4] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, Masanori Koyama, Optuna: A Next-generation Hyperparameter Optimization Framework, arXiv:1907.10902 (2019)