Geometric Hyperparameter Optimisation of ALICE Monte Carlo Transport Simulations - Summer Student Report

Anthony Swain

Supervisors: Benedikt Voelkel, Sandro Wenzel

Abstract

Monte Carlo (MC) transport simulations often require a significant amount of computing resources due to the detailed physics simulations. There are various physics models and thousands of particles are transported through, in the particular case of this report, a complex geometry resembling the ALICE detector. This report outlines the development and use of automatic optimisation procedures using a geometric approach to reduce the number of relatively computationally heavy MC steps whilst removing a negligible number of hits in GEANT4. The approach was able to reduce the number of steps by roughly 8-9% whilst removing <1% of hits in any detector in the main barrel region of ALICE. Further study should be conducted, using the developed framework, to look at the absorber materials inside the detector. Alternative methods should be explored to optimise the region including the ZDC detectors.

Keywords: Hyper-parameter optimisation, Monte-Carlo transport, ALICE, Simulations, Steps, Hits, GEANT4

1 Motivation

Monte Carlo (MC) transport simulations often require a significant amount of computing resources due to the detailed physics simulations. There are various physics models and thousands of particles are transported through, in the particular case of this report, a complex geometry resembling the ALICE detector. Figure 1 shows the wall times of all tasks in a simulation chain, starting from the MC transport (sgnsim) and resulting in the objects that will be used for physics analysis (aod). It demonstrates the sheer amount of computing resources required by MC transport simulations, taking 59.7% of wall time in a run of 10,000 minimum bias proton-proton (pp) collisions using GEANT4 [1]. Any increase in efficiency in the MC transport process of simulations will have a considerable impact on the efficiency of the entire framework. This will have multiple benefits, particularly: reducing power consumed and time taken to run, allowing more events to be ran with the same resources.

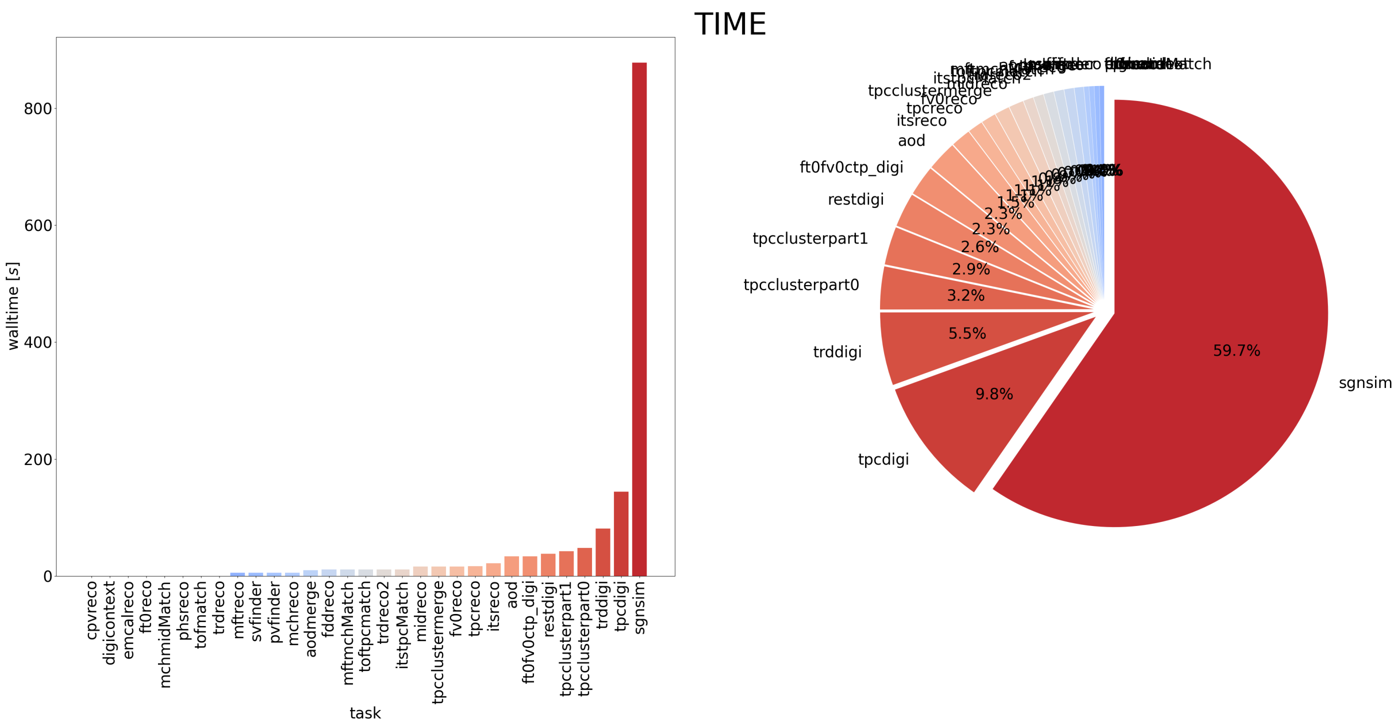


Figure 1: Wall times for all tasks in a simulation chain of 10,000 pp collisions. The process 'sgnsim' describes the Monte Carlo transport part of the simulation and takes, by far, the most wall time compared to any other process with 59.7% of wall time.

Section 2 outlines the approach taken to perform the optimisation, Sec. 3 describes the development and testing of the optimisation framework, Sec 4. presents the results and Sec 5. is the conclusion.

2 Approach

In order to describe the approach taken, it is useful to introduce a few terms commonly used in MC transport simulations:

Term	Description
Step	A particle's trajectory is broken into discrete steps, at each step the interactions between particle and detector material (e.g. scattering, energy loss, decays) are calculated.
Hit	Recorded energy deposit induced by steps. Hits are then digitised to resemble the detector readout, and eventually, physics objects such as tracks are reconstructed from this information.
Passive Material	Interact with particles, no information stored about interactions.
Sensitive Material	Interact with particles, additionally capable of recording hits.
World Volume	The volume which contains the entire simulation, i.e. nothing is simulated, or exists, outside of the world volume.

The computationally heavy part of MC transport are the MC steps; if the number of steps can be reduced, the resources required will drop by an approximately proportional amount. In order for no physics to be lost from the simulations, the number of MC hits removed should be negligible. Therefore, the goal is to reduce the number of steps whilst maintaining an almost identical number of hits by means of automatic optimisation procedures.

The general approach explored in this report involves cutting volumes out of the detector geometry where steps taken do not contribute to many, if any, hits in the detectors. Figure 2 illustrates this principle using a voxel map where the detector geometry is broken into cubic voxels each assigned a value of true or false, if a particle enters a voxel set to true, the particle's transport is stopped.

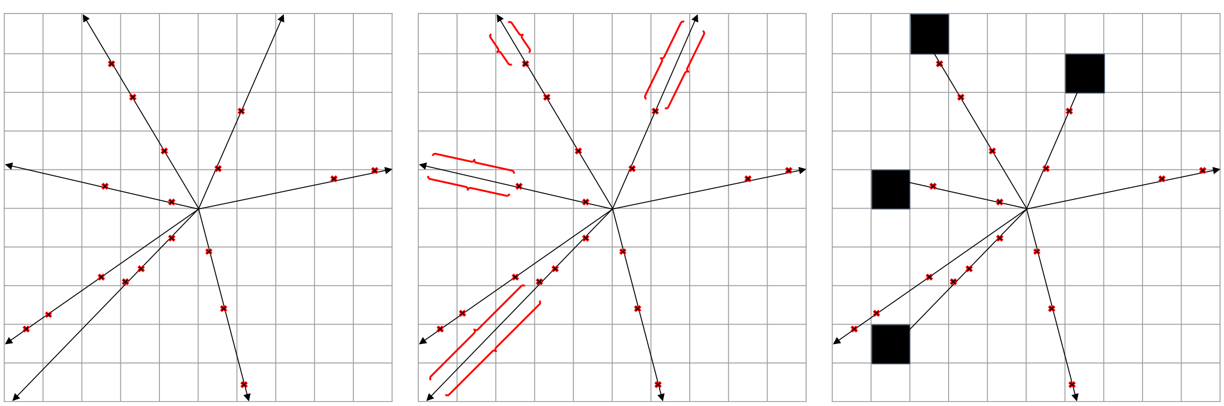


Figure 2: 2D demonstrations of the working principle of a voxel map superimposed on the detector geometry. The particle transport is stopped when particles will no longer leave any more hits. Left: Some MCTracks (arrows) and hits (red crosses) that are left in the detectors. Center: Areas where the particles are being transported but no additional hits are left in the detectors. Right: The said areas being cut out, resulting in a lower amount of steps being calculated but leaving the detector readout, i.e. number of hits, the same.

In principle, all methods to find the best voxel map follow the basic outline described in Fig. 3; they vary in how the voxel maps are sampled. A reference simulation is created and used as seeding, therefore benchmark, for trial simulations using the MCReplay tool [2]. Applying the voxel map in a replayed simulation allows to study the impact on the number of steps and hits when comparing it with the benchmark simulation. This is evaluated by a loss function, $\mathcal{L}$, defined in Eq. 1. The loss is used to assess the quality of a voxel map, and is used for iterative improvement as described in Fig. 3. A lower loss indicates a better voxel map.

The loss function is chosen according to:

$$\mathcal{L} = (\frac{s_t}{s_r})^2 + \sum_{\text{detectors}} (P \times (1 - \frac{h_t}{h_r}))^2 \ , \quad P = \begin{cases} 1, & \text{if } \frac{h_t}{h_r} > 0.99 \\ 50, & \text{otherwise,} \end{cases} \tag{1}$$

where s_t, s_r refer to the number of steps taken in the trial and reference simulations respectively and h_t, h_r refer to the number of hits recorded in the trial and reference simulations respectively. A penalty, P, is applied if the number of hits in any detector falls below a chosen percentage of their original value, in this case 1%. This is to maintain almost an identical number of hits.

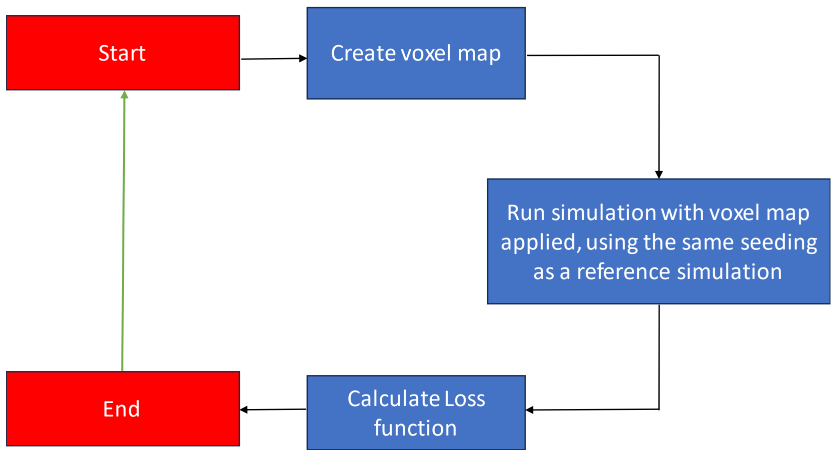


Figure 3: Basic outline of the methods used to perform the optimisation. A voxel map is created and applied to a simulation using identical seeding as a reference simulation. The loss for the voxel map is then calculated and used to choose parameters for future voxel maps.

2.1 Geometrical Optimisation

The main optimisation approach explored in this report involves creating an envelop as tight as possible around the detector geometry; in essence shrinking the world volume around the detector. Particles that have left the region of the detector are very unlikely to leave anymore hits; therefore, their transport can be stopped once they have left the detector region with a minimal loss of hits. The methods involved utilise the cylindrical symmetry of the detector as indicated in Fig. 4. The methods include:

1. Individual cylinders:
   (a) Define a single cylinder whose axis is aligned with the beam line (z-axis) with an initially large radius and defined z extent. Any particle beyond the cylinder's radius and within the cylinder's defined z extent has its transport stopped.
   (b) Shrink the radius of the cylinder and calculate the loss function.
   (c) Repeat (b) until the loss function starts to grow. The cylinder defined for the minimum loss is the best envelope.
   (d) Repeat (a)-(c) for different cylinders defined in other regions of the z-axis until the z-axis can be fully encompassed in cylinders.
2. Fine tuning:
   (a) Create a mapping containing the cylinders found individually, each having its z extent and radius defined. The z extents of the cylinders should be defined such that there is not a z co-ordinate in the detector region which isn't encompassed by a cylinder. This can be done by working inwards from the furtherst cylinders away from the interaction point:

i. Define the cylinders furthest away from the nominal interaction point on either side.

ii. Define the next furthest away cylinders from the interaction point on either side, letting the z extents of the cylinders go up to the previously defined cylinder on their side but no closer to the interaction point than previously found.

iii. Repeat until cylinders fill the z-axis.

(b) Using the algorithm defined in Fig. 3, optimise the z extents for the cylinders, whilst maintaining the condition that all the cylinders should be joined, and their radii to minimise the loss.

3. Hyper-fine tuning:

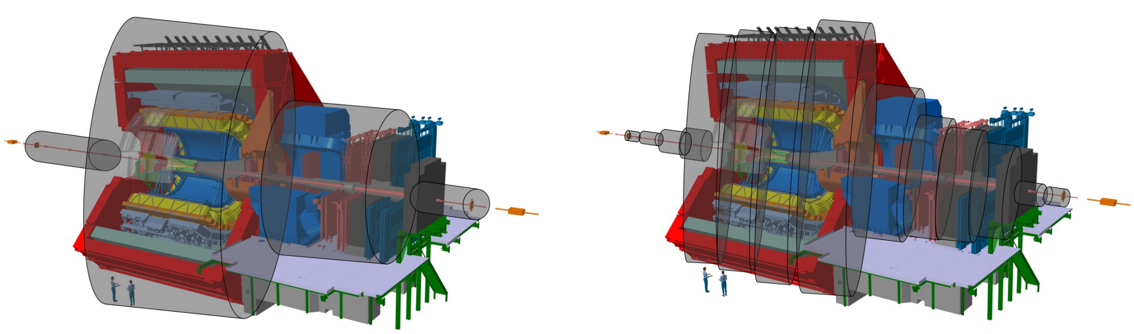
(a) To optimise further, cut all the cylinders found via Fine Tuning into equal sized slices, cutting in the $x-y$ plane, using the following the method:

i. Split the z-axis into a given number of equally sized slices; greater than the number of cylinders used in fine tuning.

ii. Assign each slice the radius of the cylinder at the corresponding z location found in fine tuning. If the slice encompasses 2 cylinders, assign the greater radius.

Each slice can have its radius varied but fixed z extent.

(b) Optimise further using the algorithm described in Fig. 3.



(a) Fine Tuning **(b)** Hyper-Fine Tuning

Figure 4: Graphic demonstration of fine tuning (a) and hyper-fine tuning (b). This are example images, fine tuning involved around 5-10 hollow cylinders and hyper-fine tuning optimisation runs included 100+ hollow cylinders. Any particle beyond a cylinder's radius and within the cylinder's defined z extent has its transport stopped.

2.1.1 Note on voxel maps

Whilst the end goal is to be able to provide a single voxel map which is a perfect envelope around the detector geometry, the creation of voxel maps, particularly ones with many cylinders involved, is a time-consuming process. This significantly increases the time required to run optimisations which involve many trials. Therefore, the hollow cylinder dimensions were often passed to the simulation which emulated the role of the voxel map without having to create the voxel map. Over many events this would consume more time than creating the voxel map. However, for many trials, where a different voxel map is required for each trial, and a handful of events (fewer than 100), this provides a significant speed-up. Once an optimal set of cylinders were found, they were converted to a single voxel map.

2.1.2 ZDC detectors

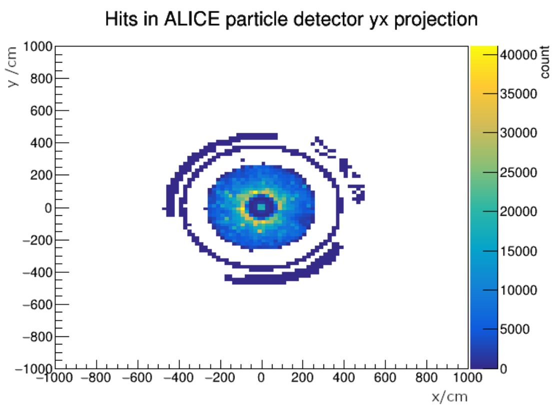
The ZDC detectors are situated on both sides of the nominal interaction point, roughly 112.5m in the z-directions [3]. To compare, the main barrel part of the ALICE detector is roughly 26m long [4]. The vast distance that particles are required to be transported along the beam-line leads to a substantial increase of steps, therefore memory usage and CPU time in the simulations. Optimisations were initially run excluding the ZDC detectors to find an optimal voxel map for the main barrel of the detector. It was then studied whether it was possible to include the simulation regions including ZDC detectors. However, it was seen not to be feasible with the presented approach as discussed again in Sec. 4. Hence, this region was excluded from the optimisation.

3 Testing and Development

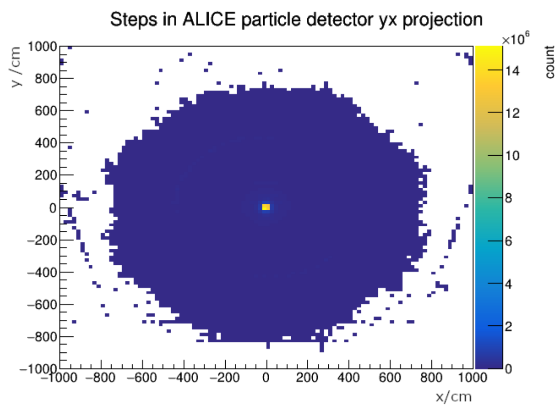
The optimisation framework involved many parts, this section describes the framework development and testing to validate the functionality of the voxel map implementation.

3.1 Step and Hit Graphical Representation

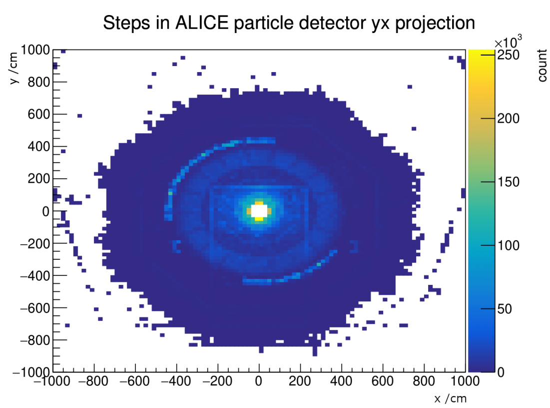
A graphical representation of where steps and hits take place in the detector is useful to provide an informed initial guess for the voxel maps as well as confirm that the maps work as expected. A light-weight implementation was added to the stepping function, found in [5], to save the location and particle type (PDG code) of steps. A set of macros, found in [6], was developed to analyse the step and hit data. Figure 5 demonstrates the $x-y$ projection of steps and hits of 100 simulated pp events.



(a) Hits



(b) Steps



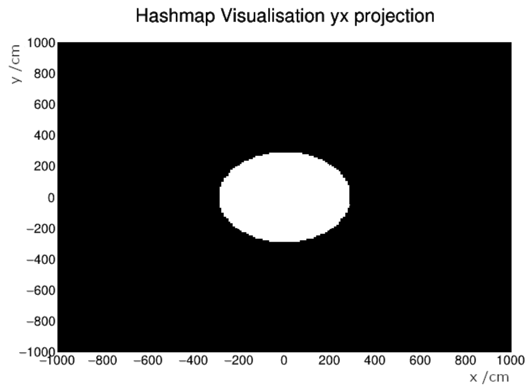
(c) Steps, Beam-pipe data-points removed

Figure 5: Projections in the $x-y$ plane of hits (a) and steps (b) of 100 pp events using Pythia8 generator with no voxel map applied. The TPC and calorimeters are easily identifiable in the hits projection. The intricate details in the steps projection have been washed out due to the beam pipe containing orders of magnitude more steps than the rest of the detector. Figure 5c shows the $x-y$ projection of steps with the data from the beam pipe removed allowing the calorimeters to be identified.

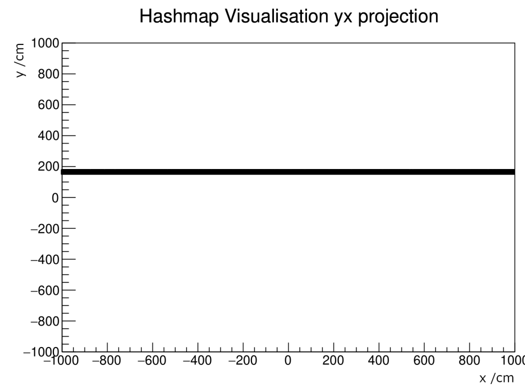
3.2 Voxel Maps

The VecGeom library [7] was used to create and query voxel maps. Macros developed and used to quickly create, manipulate and image the voxel maps from VecGeom can be found in [6].

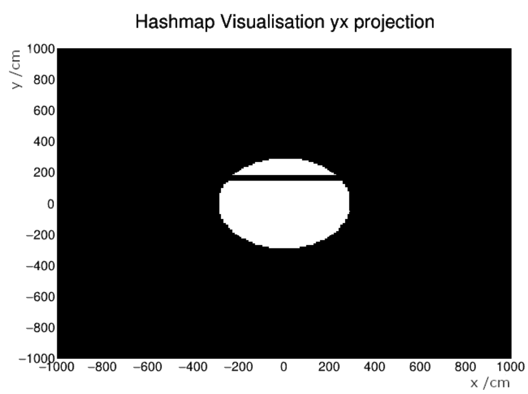
Figure 6 shows two different voxel maps, a cylinder whose cylindrical axis is the z axis and a wall splitting the $x-y$ plane in two. Figure 7 shows the $x-y$ projection of steps and hits with the voxel maps from Fig. 6 applied. It is clear that the cylindrical voxel map is working as expected. However, for the half-plane wall voxel map, steps are seen above the wall. Further inspection, shown in Fig. 8, shows that thickening the wall significantly reduces the number of steps and hits. The reasoning on why steps are taken beyond the wall is likely due to the distance between some steps being larger than the thickness of the wall; therefore, some particles are never stopped as they stepped over the entire wall.



(a) Cylindrical voxel map

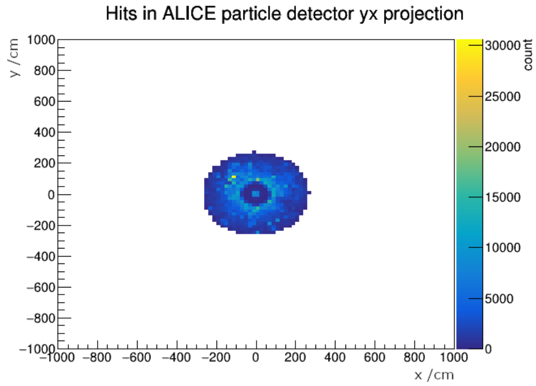


(b) Half-plane wall voxel map

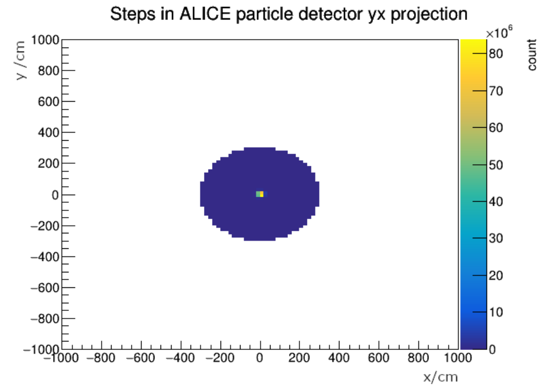


(c) Addition of cylindrical and half-plane voxel maps

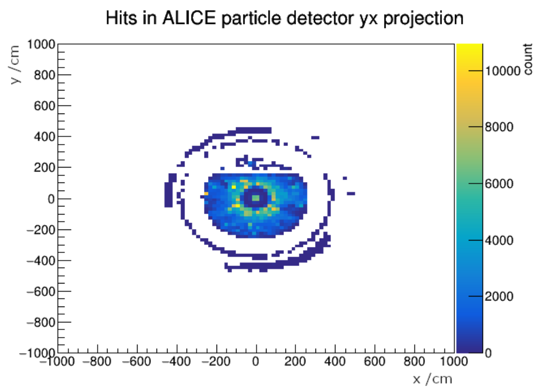
Figure 6: Three voxel maps that were used in development and testing. The map shown in Figure 6c is the combination of the maps in Figures 6a and 6b.



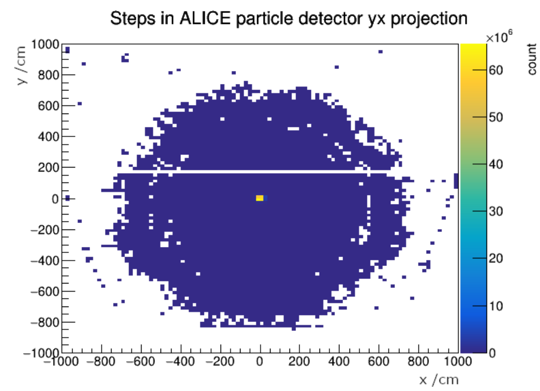
(a) Cylindrical voxel map - Hits



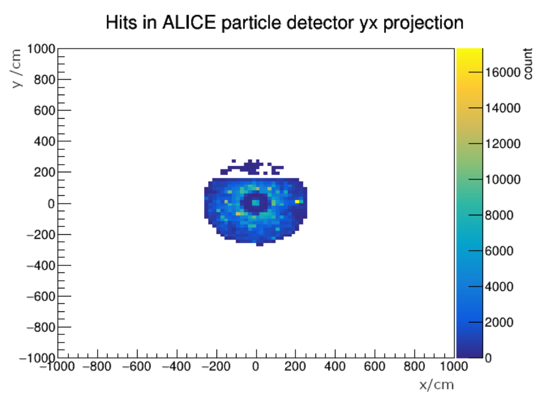
(b) Cylindrical voxel map - Steps



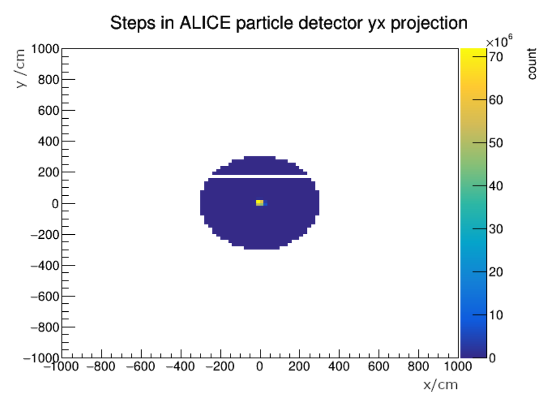
(c) Half-plane wall voxel map - Hits



(d) Half-plane wall voxel map - Steps



(e) Cylindrical and half-plane voxel maps - Hits



(f) Cylindrical and half-plane voxel maps - Steps

Figure 7: Projections in the $x - y$ plane of steps and hits with different voxel maps applied. The cylindrical voxel map works as expected. However, for the half-plane wall voxel map, steps are seen above the wall; this is also seen when the cylindrical and half-plane wall maps are combined. The likely reason for this is that the distance between some steps are larger than the thickness of the wall; therefore, some particles are never stopped as they stepped over the entire wall.

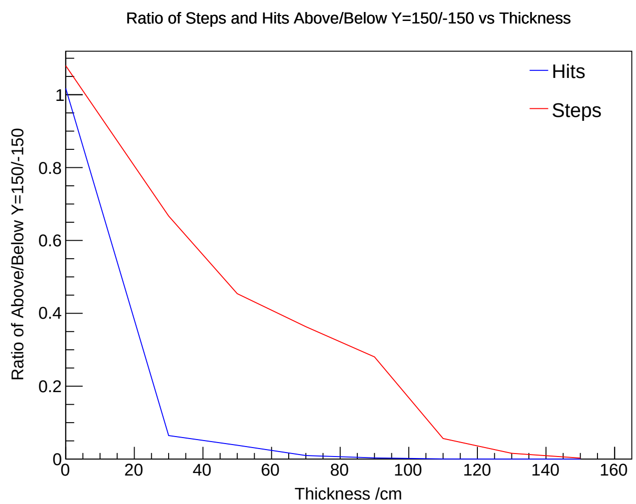


Figure 8: Graph showing the ratio of steps and hits above $y = 150$cm and below $y = -150$cm vs thickness of the halfplane wall. Above $y = 150$cm (above the half-plane wall) and below $y = -150$cm, assuming cylindrical symmetry, should have a similar number of hits/steps if there were no half-plane wall voxel map. Relative confidence can be put in this assumption given that when the thickness of the plane is 0 (i.e. it does not exist) the ratio is approximately 1. It is seen that thickening the wall significantly reduces the number of steps and hits on the other side.

3.2.1 Particle-Specific Voxel Maps

Implementation that allows voxel maps to stop, or only allow through, specific particle's based on their PDG code was also added; this can be found in [5].

3.3 Optimisation

The O2Tuner library [8] was the core to the optimisation framework. The optimisation algorithms developed from this library can be found in [9]; they will not be discussed individually here but follow the approaches described in section 2.1. This repository additionally includes implementation for optimisations inside the main barrel.

4 Results

An 8-9% reduction in steps in the main barrel region of the ALICE detector with less than 1% of hits removed in any detector was achieved. This is relative to a reference simulation using the same seeding with no voxel map applied (500 minimum bias pp events generated with Pythia8 [10], GEANT4). The voxel map was created using 100 hollow cylinders.

The ZDC detectors were not included in the presented optimisation. The number of steps cut away by placing cylinders around the beam pipe outside the barrel detector was found to be negligible. Most steps in these regions arise from electromagnetic showers inside the beam pipe material. Reducing the number of steps in these regions requires additional studies.

5 Conclusion

Optimising ALICE's monte carlo transport simulations by means of using voxel maps to remove steps whilst removing a negligible number of hits was successful. Implementation was developed and integrated into ALICEO2; allowing for utilisation of custom voxel maps within the stepping function. An automatic optimisation framework to find an optimal voxel map which enveloped the ALICE detector was developed and used. The procedure for creating voxel maps, their use, and optimisation framework was validated.

Geometric based optimisation aimed at creating an envelope around the main barrel of the ALICE detector demonstrated a potential 8-9% reduction in the number of steps whilst maintaining over 99% of hits, as compared to a reference simulation using identical seeding with no optimisation applied.

The optimisation framework is also ready to attempt optimisations for absorber material inside the main barrel. It would be interesting to see if a voxel map, or set of particle-specific voxel maps to account for different particles properties, using the developed framework, could be found which overlay the absorber material and stop the transport of particles once a certain point inside the material has been reached; instead of computing many steps to reach the same final result.

An alternative approach is required for the ZDC detectors, the geometric approach introducing an envelope around ALICE is not suitable; further study should be conducted looking into alternative approaches to optimise these regions.

Acknowledgements

I want to extend my deepest thanks to my supervisors, Benedikt Voelkel and Sandro Wenzel, for their continued support and guidance this summer. I would also like to thank the CERN summer school organisers for organising and granting such an amazing opportunity filled to the brim with experiences to last a lifetime. Lastly, but by no means least - a big thank you to all the other summer students; this has been a summer of many adventures and memories that I will cherish - I hope to see you back here one day.

References

[1] S. Agostinelli, J. Allison, K. a. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand *et al.*, "Geant4—a simulation toolkit," *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250–303, 2003.

[2] "VMCStepLogger," https://github.com/AnthonySwain/VMCStepLogger, Date Accessed: 17-08-2023.

[3] "ALICE ZDC," https://alice-collaboration.web.cern.ch/node/34995, Date Accessed: 12-09-2023.

[4] "ALICE Experiment," https://www.home.cern/science/experiments/alice, Date Accessed: 11-09-2023.

[5] "ParticleSpecificBranchAliceO2," https://github.com/AnthonySwain/AliceO2/tree/ParticleSpecificMaps/Steer, Date Accessed: 05-09-2023.

[6] "MacroDevO2," https://github.com/AnthonySwain/AliceO2MacroDev, Date Accessed: 17-08-2023.

[7] "VecGeom," https://gitlab.cern.ch/VecGeom/VecGeom, Date Accessed: 17-08-2023.

[8] "O2Tuner," https://github.com/AliceO2Group/o2tuner, Date Accessed: 17-08-2023.

[9] "O2Tuner Optimisation Branch for this project," https://github.com/AnthonySwain/o2tunerRecipes/tree/voxels/ParticleSpecificMaps, Date Accessed: 05-09-2023.

[10] C. Bierlich, S. Chakraborty, N. Desai, L. Gellersen, I. Helenius, P. Ilten, L. Lönnblad, S. Mrenna, S. Prestel, C. T. Preuss, T. Sjöstrand, P. Skands, M. Utheim, and R. Verheyen, "A comprehensive guide to the physics and usage of pythia 8.3," 2022.