FLAG

Anton Taleiko Author, supervisors: Andreas Salzburger, Mika Huhtinen
University of Helsinki, Helsinki, Finland

Abstract
When particle detectors are used at high efficiency, they are exposed to radiation, which over time causes the detectors to wear out. The summer student project FLAG aimed to create radiation simulations of particle detectors to estimate their lifetimes to allow for better planning in logistics. The project involved two main parts: data conversion between the particle simulator Geant4 and the radiation simulator FLUKA, and simplification of existing particle detector models used in particle simulations to make the models light enough for FLUKA to handle in terms of complexity.

Keywords
CERN report; contribution; FLAG; radiation simulation; FLUKA; Geant4; ACTS; data conversion; model simplification; ATLAS.

1 Motivation

In high energy physics, different kinds of simulators, such as particle and radiation simulators, are used for different purposes. One use for radiation simulators is to simulate the ratiation environment, e.g. for estimating the lifetime of particle detectors. When particle detectors are used at high efficiency, they are exposed to radiation. One key problem for radiation simulators, when simulating this kind of environment, is to feed them with a correct geometry model. The summer student project FLAG (FLUKA-ACTS-Geant4) aimed to create these kinds of radiation simulations in the radiation simulation software FLUKA [1], and tackle the challenges that come with detector model conversion and simplification needed to make the input data compatible with it.

2 Implementation

Digital models of particle detectors, such as the Open Data Detector (ODD) [2] model, are already used in Geant4 particle simulations. However, these models are stored in GDML (a type of XML) format, while FLUKA takes input in a different format, called INP. Thus the goal was to convert the ODD model into INP format. Fortunately, a Python module called pyg4ometry [3] exists, which supposedly can convert files between these two formats. There were, however, complications. While Geant4 particle simulations simulate particles on a microscopic scale with detailed descriptions of the environment, FLUKA simulates radiation on a macroscopic scale and takes much simpler structures as input. While the ODD model could technically be converted to FLUKA's format, the simulations would not run in practice. For the project, this meant that another step had to be added: simplifying the ODD model. The goal was to create a structure simple enough for FLUKA to handle. Since the detectors are generally shaped like tubes around the center of the particle collisions that occur in them, a cylinder was chosen. The simplification would be done with the particle tracking software A Common Tracking Software (ACTS) [4] that would be used to record the materials in the ODD model and project them onto an imaginary cylinder. These projections would then be used to map the materials onto smaller areas that the cylinder was divided into. A schematic of the pipeline can be seen in figure 1.

With the given requirements, the pipeline the project aimed to create can be divided into the following steps:

1. Record the materials in the ODD model with the ACTS toolkit

2. Make a simplified model of the ODD detector in Geant4 format by creating cylinders with segments to project the materials on and create layers for the different materials that have been recorded within an area on the cylinder
3. Convert the simplified model into FLUKA's input format with the Python modeule pyg4ometry

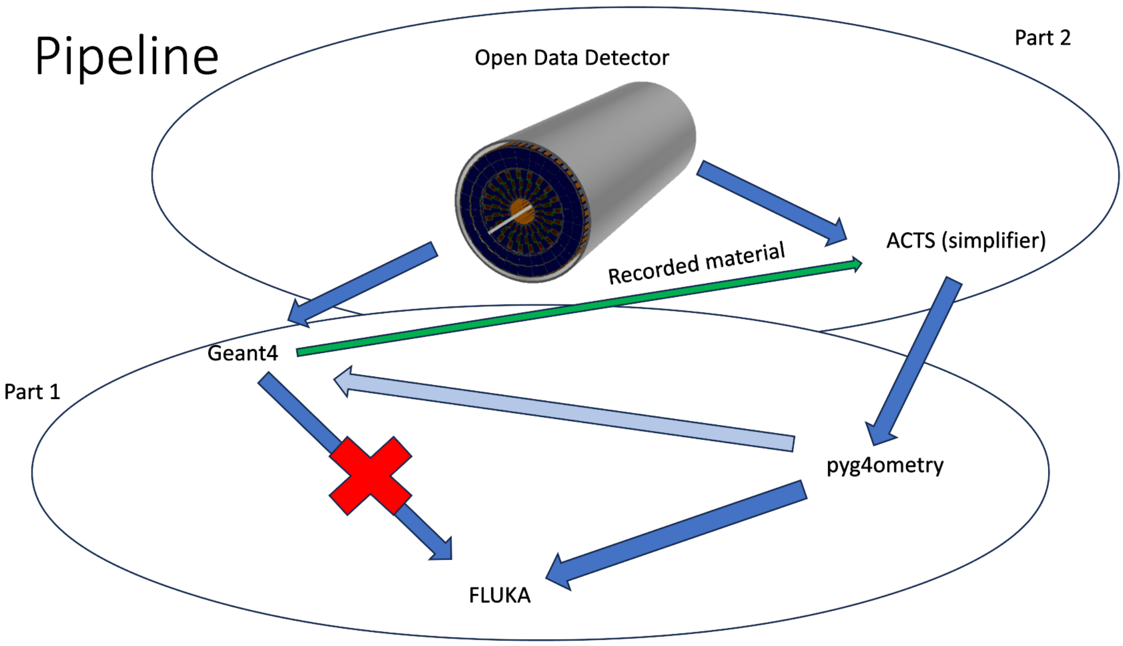

Fig. 1: A visualization of the FLAG pipeline.

In order for the whole pipeline to work it had to verified that pyg4ometry actually created correct structures. Because of this, step 3. was done first and is here referred to as part 1, while the simplification process in step 1. and 2. are referred to as part 2.

2.1 Part 1: Model conversion

To verify that the Python module pyg4ometry was producing sensible results, a material comparison procedure was used to verify that the amount of material was the same before and after the conversion from from Geant4 to FLUKA's input format. The model used for the verification was a simple cylinder, like the one in figure 2, that step by step was increased in complexity by increasing the number of cylinders, adding layers and lastly segments like in figure 3. The material comparison was done by sending rays from the center of the cylinder out towards the edges of the space that the cylinder was placed in. The amount of material encountered along the rays' paths was recorded both in Geant4 and FLUKA, and the results were compared in plots to verify that the conversion was successful. An example of such a plot of the results can be seen in figure 4, where the cylinders in figure 2 were used.

2.2 Part 2: Model simplification

Simplifying the ODD model into something that could be described as cylinders with layers and segments involved several steps: recording the materials in the structure using ACTS, mapping the materials to areas on a cylinder's surface that the cylinder had been divided into and generating a new simplified model that could be put through the conversion process from Geant4 format to FLUKA format. The different stages of this process are visualized in figure 5.

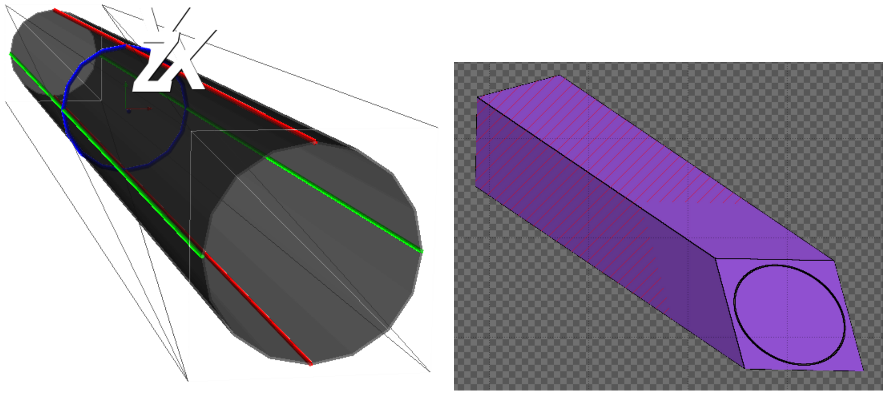

Fig. 2: A cylinder in Geant4 format visualized with pyg4ometry (left) and the same cylinder converted to FLUKA format (right).

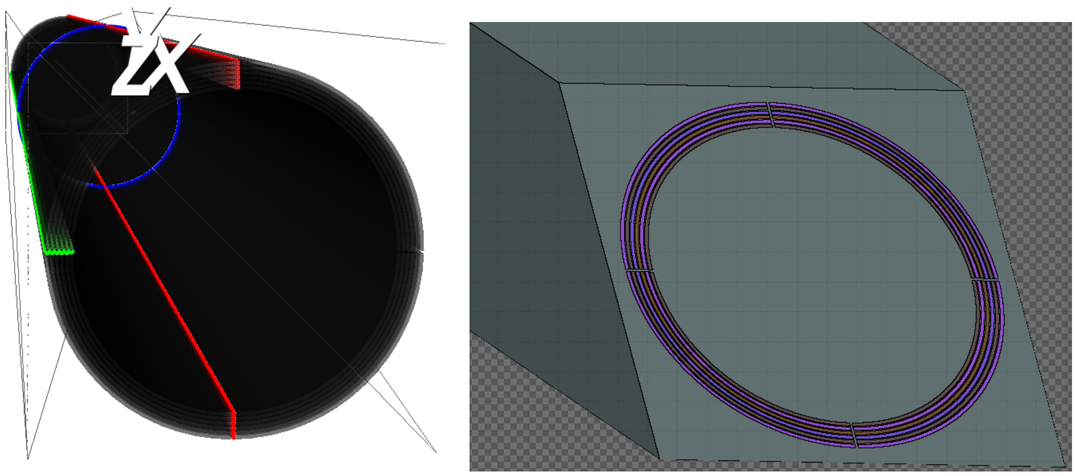

Fig. 3: A cylinder in Geant4 format with multiple layers and segments (left) and the same cylinder converted to FLUKA format (right).

3 Results

The conversion from Geant4 to FLUKA format using pyg4ometry has been successful so far. There are, however, oddities start to occur once the number of layers is increased beyond 6. Then, the material comparison plots start diverging. An example of this can be seen in figure 6, where 10 layers were used to make the difference more obvious. Based on the recording with one layer, the amount of material in the Geant4 seems to be correct, since the material scales linearly. The amount of material in FLUKA on the other hand seems to bee incorrect. This has to be investigated. One of the "sanity checks" that was done was not writing the Geant4 geometry to a file as a middle step and then reading it for the conversion. The results were the same when the Geant4 geometry was created in the conversion program itself before converting it to FLUKA format, hence the format conversion to GDML can be excluded as culprint.

The model simplification is still under work. The material recording is done and yields satisfying results. When a material material comparison similar to the one used in the conversion process is run, the materials in Geant4 format and the recording in ACTS match. An example of a comparison plot can be seen in figure 7. The material mapping is still in progress. It is still under discussion whether the cylinders with layers and segments will be structure simple enough for FLUKA to handle, or if the model needs to be simplified further by creating materials that are an average of the ones recorded in an

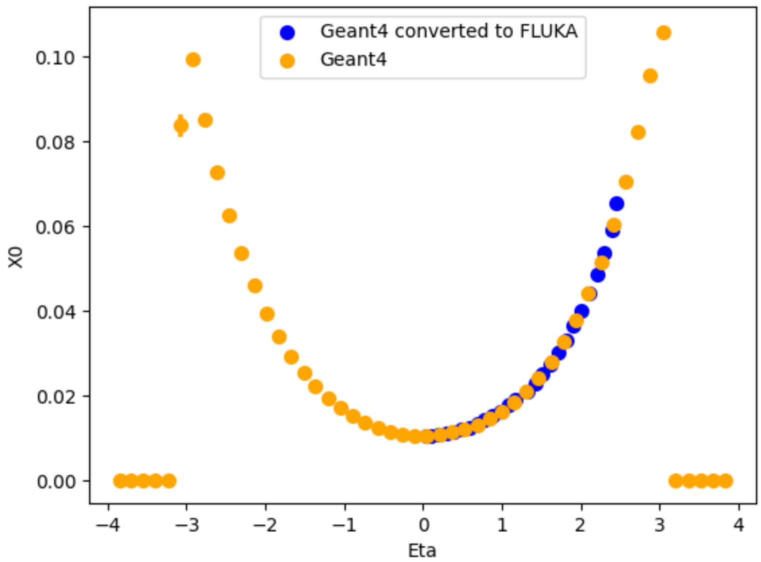

Fig. 4: A comparison plot that shows the amount of material recorded for different angles of the rays. The recording rays for the FLUKA geometry were only sent out with eta values between 0 and 2.5.

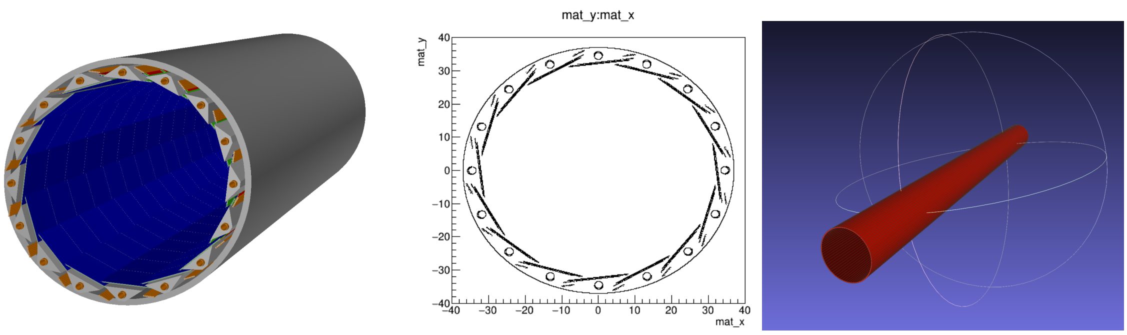

Fig. 5: The detector model at the three stages of the simplification process: the original ODD detector model (left), the material recording in ACTS (center) and the cylinder that the recorded materials are projected onto (right).

area. This is because as the cylinders, layers and segments increase in number, a new material has to be defined for every part of the cylinder. This may still be too much for FLUKA.

References

[1] FLUKA, https://fluka.cern/
[2] OpenDataDetector, https://gitlab.cern.ch/acts/OpenDataDetector
[3] pyg4ometry, https://pypi.org/project/pyg4ometry/
[4] ACTS, https://github.com/acts-project/acts

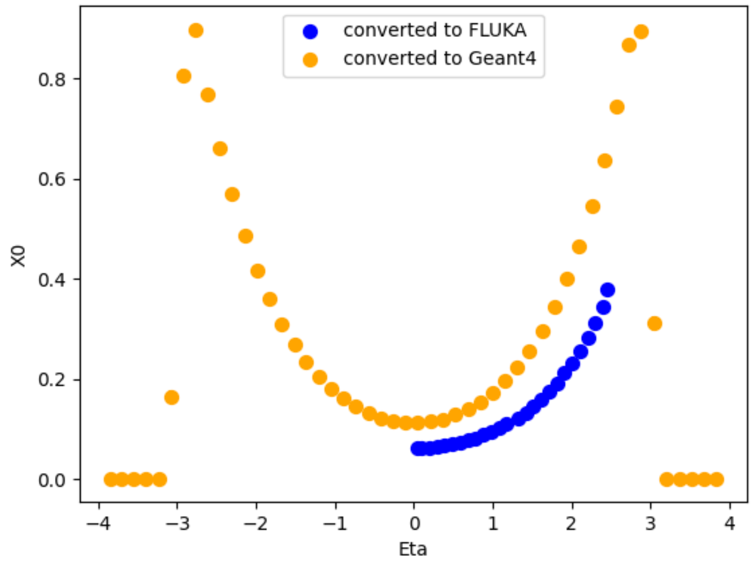

Fig. 6: When the number of cylinder layers is increased beyond 6, the material comparison plots start to diverge. An example where the number of layers is 10.

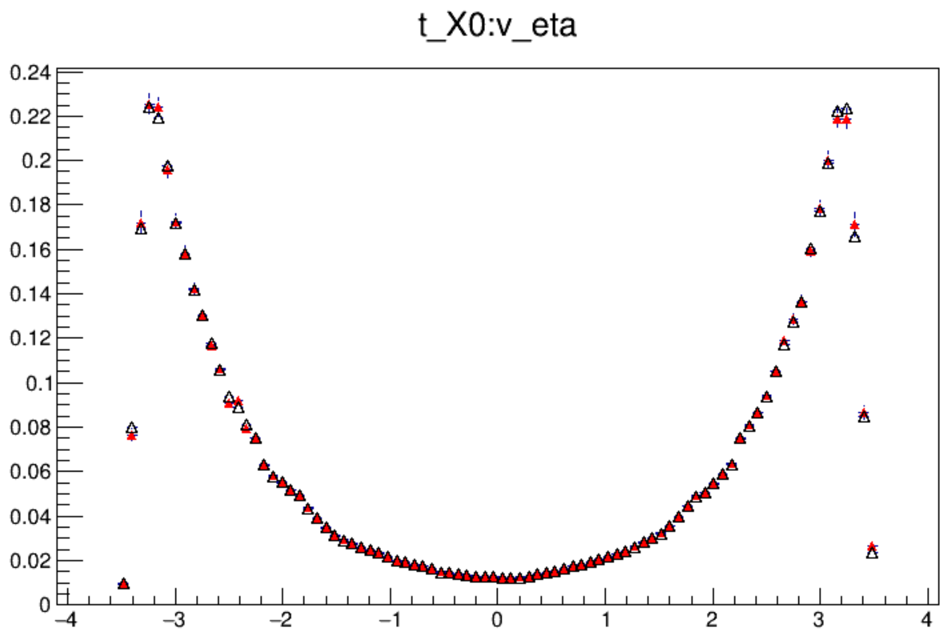

Fig. 7: A material comparison between a layer from the original ODD model and a corresponding material recording in ACTS.