Summer Student Report
Conversion between the GeoModel and Acts Frameworks

Carl Jonas Berggren

September 27, 2024

1 Introduction

The goal of my summer student project was to implement software for conversions of the geometric description of the ATLAS Muon Spectrometer between the **GeoModel** and **Acts** software tool kits.

1.1 Acts

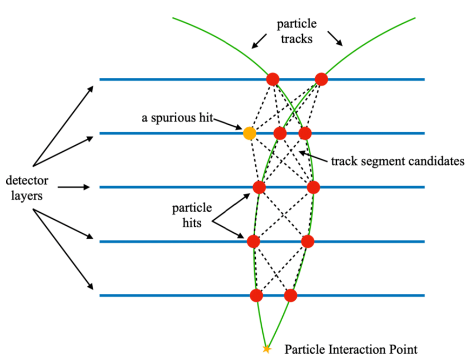

Figure 1: tracking problem [1]

A common tracking software (**Acts**) is a software tool kit designed for tracking of charged particles in high energy physics (HEP) experiments.

The signal output of a silicon pixel detector, for example, usually contains very limited but crucial information, that needs post processing to extract the useful information needed for physics analyses. The information provided as signal may simply be a detector element identifier, a time of arrival, and a hit location in the coordinates dictated by the location and orientation of the sensor itself.

The task of the tracking software is to take this information provided by the detector elements and the geometric description of the detector to reconstruct the path taken by the charged particle through the detector. This path can then be used to compute several important quantities like the initial η and ϕ of the particle and, using the magnetic field present in the detector, its momentum and charge. This is done by interpolating the hit locations (red dots in fig. 1) with a modeled trajectory. Due to the strong homogeneous magnetic field applied to most HEP experiments, this trajectory is usually assumed to be approximately helical. Complications one needs to consider are noise (orange dot), scattering of the particles, imperfect alignment of the detector parts, uncertainty in the measurements and the fact that events usually contain many hundreds of tracks, that may be close to each other.

The focus of my work is to allow for an accurate description of the detector geometry, meaning that we have an accurate description of where the red dots are in absolute coordinates. This essential for the track reconstruction to be performed successfully. This description needs to be not only precise, but also lightweight and quickly navigable.

1.2 GeoModel

GeoModel is a software toolkit designed for detailed and light weight detector description in high energy physics experiments. With its modular structure and database interface it allows for a lightweight and selective manipulation of the detector description for fast tracking.

2 Contribution

I implement a **GeoModelDetectorObjectFactory** that is capable of reading the **GeoModel** database, identify and extract the required geometric **GeoModel** objects and convert them to the appropriate **Acts** representation for the use in tracking applications. The converter has to transfer **GeoModel::GeoShape** objects to **Acts::Experimental::DetectorVolume** or to **Acts::SensitiveSurface** objects depending on the applications requirements.

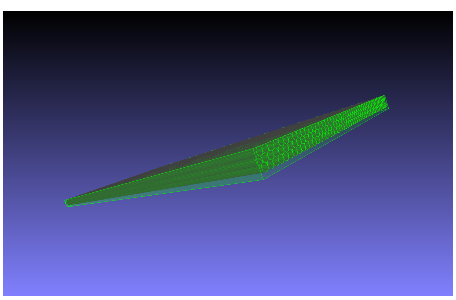

Figure 2: 3D model of the converted MDT module

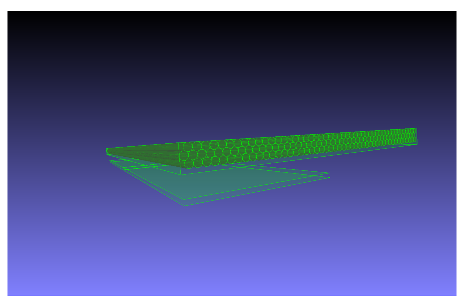

Figure 3: 3D model of the converted MDT module with its neighboring RPC

In my first contribution to the acts project (#3499) I implemented the conversion for the Monitored Drift Tubes (MDTs). The MDT modules are made up of individual tubes contained in a cuboidal volume. For the tracking application all tubes need to be converted to **SensitiveSurface** and the overarching cuboidal volume containing them needs to be converted to **DetectorVolume**. This **DetectorVolume** instance holds both the bounding box and the sensitive surface in a format that the tracking software can process. A 3D model of this object can be seen in fig. 2.

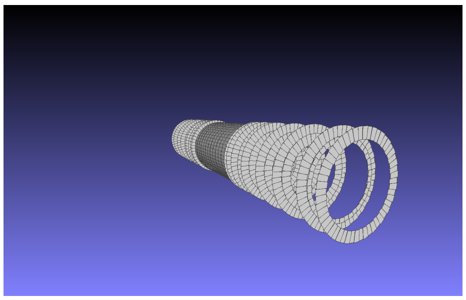

Figure 4: 3D model of the converted inner tracking detector

The next step consisted of generalizing the factory to be able to convert the active gas gaps in the Resistive Plate Chambers (RPCs) to **SensitiveSurface**. Since the **GeoModel** objects themselves don't contain any information about if and how they should be converted, this information needs to be passed manually as an argument. I therefor implemented a recursive search algorithm that receives a number of query parameters and searches for matching objects in the entire tree structure of the **GeoFullPhysVol** objects. A visualization of the conversion of both a MDT and RPC module using this search algorithm can be seen in fig. 3.

The converter needs to be also compatible with the ATLAS Inner Detector (ID). Therefor the factory converts the **GeoFullPhysVol** objects of the ID to **SensitiveSurface** as can be seen in fig. 4.

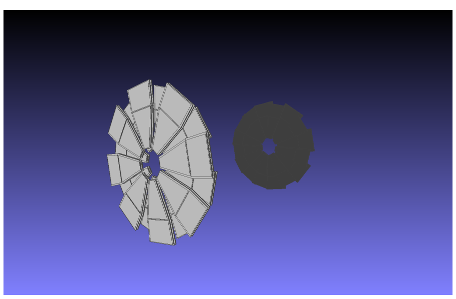

Figure 5: Visualization of the converted MicroMegas

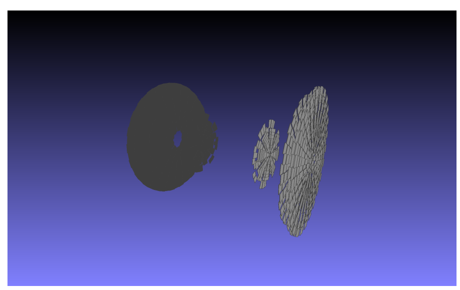

Figure 6: Visualization of the converted (s)TGCs

The contribution #3540 expands on this by adding the conversion capability for Thin Gas Chambers (TGCs) and MicroMegas. A special complication for the TGCs consisted in the fact that I could not use preexisting converters. In **GeoModel** TGCs are described using the **GeoSimplePolygonBrep** class for which I had to develop a dedicated converter. This converter is capable of converting **GeoModel** polygons with four or six vertices to **SensitiveSurface**.

The result of the conversion can be seen in fig. 6.

The conversion capability for the MicroMegas was already given through the expandable implementation I made for the other detector modules. The result of this conversion can be seen in fig. 5.

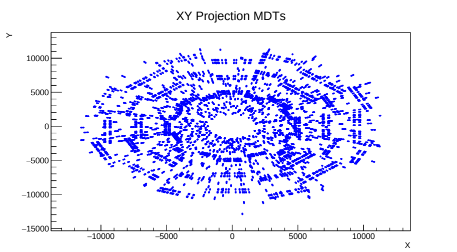

Figure 7: xy projection of propagation through MDTs

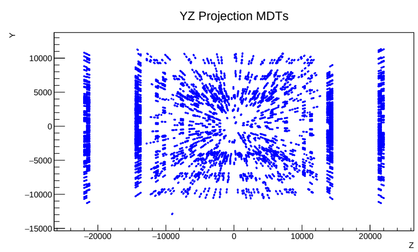

Figure 8: yz projection of propagation through MDTs

After adding these feature I refactored some of the code to add the capability to convert **GeoModel** objects to **Acts::Volume**. This was done for compatibility with the current refactoring of the geometry building in **Acts**. The contribution can be found in the pull request #3576. In the same contribution I developed a series of unit tests that check that my converters produce the expected results. I implemented unit tests also for converters written by other members of the Acts group.

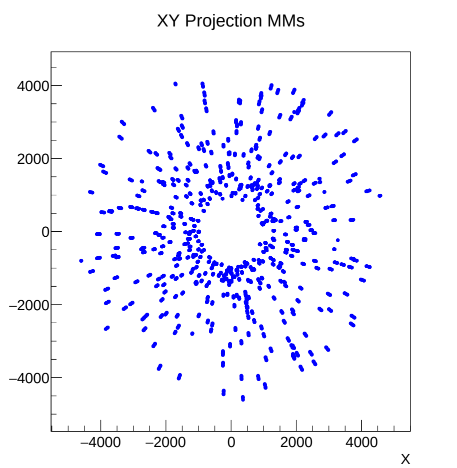

Figure 9: xy projection of propagation through MicroMegas

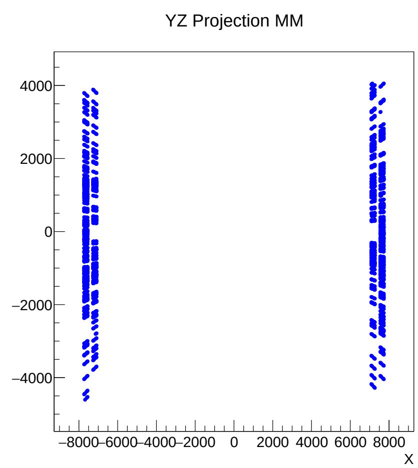

Figure 10: yz projection of propagation through MicroMegas

In the PR #3585 I implemented a generalization to the **DetectorObjectFactory**, that adds the capability to convert top level **FullPhysVol** to **DetectorVolume** even in the case where it doesn't contain any subvolumes.

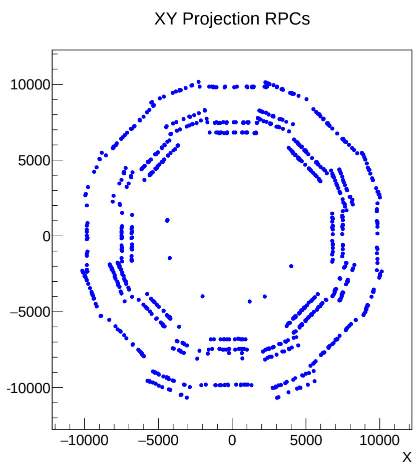

Figure 11: xy projection of propagation through RPCs

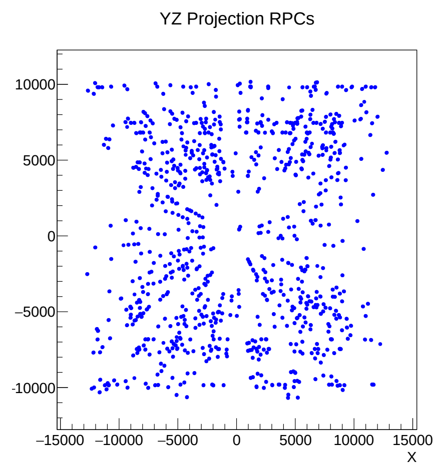

Figure 12: yz projection of propagation through RPCs

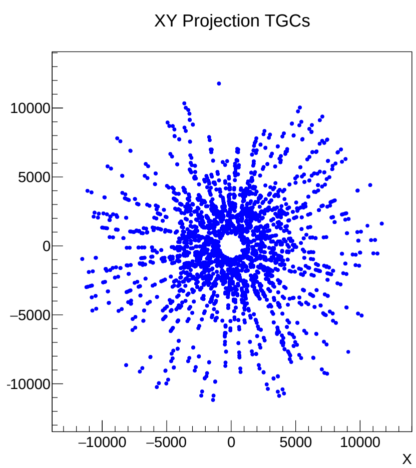

Figure 13: xy projection of propagation through (s)TGCs

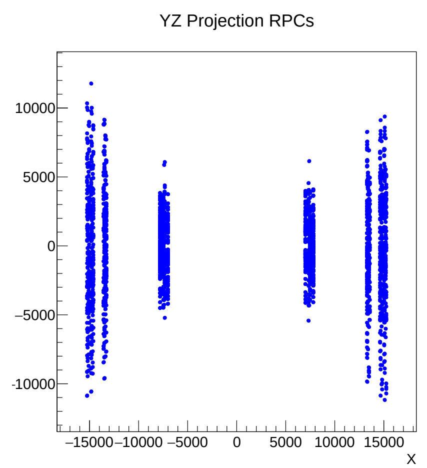

Figure 14: yz projection of propagation through (s)TGCs

In the PR #3592 I added a **GeoSubtractionConverter** to make the conversion work for all RPCs present in the ATLAS detector. Some of the RPCs are described using **GeoSubtraction** objects. These can't be converted directly since they don't have any analogous **Acts** representation. Therefore the converter decomposes the **GeoSubtraction** object recursively into its underlying shapes and convert the object being subtracted from to its analogous representation. At this stage the entire ATLAS muon detector can be converted from

GeoModel to **Acts**.

As a final test to check the proper conversion of the detector parts, I developed a propagation test. Simulated particles are generated going in arbitrary directions and propagated through the digital representation of the detector. The results for the different detector types can be seen in figs. 7 to 14.

3 Conclusion

In my summer student project I successfully implemented the conversion software needed for muon tracking to be performed using the **Acts** toolkit based on a **GeoModel** representation of the ATLAS detector. In addition I validated the software I implemented by developing a number of unit tests and performing a propagation test through the converted detector. Both testing approaches demonstrated the code to perform as expected. Further investigations left to make concern the streamlining of the fashion in which the **SensitiveSurface** vectors are handled.

Although I greatly enjoyed working with **Acts** the project heavily focused on software development and did not pose any significant physics-related questions, which limited my opportunity to apply many of the skills I learned based on my background in physics. I would have liked to engage more with the physics and algorithmic concepts underpinning the tools I was working with, such as tracking algorithms and particle and detector physics. Nonetheless I was amazed at this opportunity to develop my understanding of the production process for large scale software projects, developing code in a team and contributing to an open source project.

References

[1] Cenk Tüysüz, Carla Rieger, Kristiane Novotny, Bilge Demirkoz, Daniel Dobos, Karolos Potamianos, Sofia Vallecorsa, Jean-Roch Vlimant, and Richard Forster. Hybrid quantum classical graph neural networks for particle track reconstruction. *Quantum Machine Intelligence*, 3:29, 12 2021.