ELTE MSc – Modelling Laboratory

Simulating the NEBULA detector in Geant4 – Report 2.

Balázs Pál
Eötvös Loránd University

March 9, 2021

Abstract

On the MSc course "*Computer Modelling Laboratory*" at ELTE, I've worked on a project in nuclear physics, where I studied the behaviour of the Japanese NEBULA detector when it was bombarded by neutron beams. For the simulation and analysis I've used the Geant4 general-purpose software, which is capable of producing state-of-the-art simulations and results in almost any field in nuclear- or particle physics.

1. Introduction

The learning curve of the Geant4 software is a relatively steep one. The user presented with an enormous amount of functionality at the beginning, and even the most basic example requires good programming skills for the user to understand it efficiently. Because of this, good chunk of the last week was spent by me learning to use Geant4. Specific aspects and other

2. Simulation of the NEBULA detector

My initial consideration was to install the software „smsimulator" and use it for the project to simulate the NEBULA detector faithfully to reality. Besides Geant4, it also requires the ROOT and ANAROOT applications to be installed and configured appropriately. Unfortunately it seems so that there is a current compatibility issue between the newest build of the ROOT software and smsimulator. Because of the narrow time frame I have for this project, I temporarily aborted the setup of smsimulator. Instead I've started to implement a simplified version of the NEBULA detector and a neutron beam to be shot at it.

Parts of the simulation was written by Dávid Pesznyák in his 2020 BSc thesis about the calorimetry of muons and neutrons[1]. I've used some of his Geant4 codes in order to construct a full simulation of my own. This implementation simplifies the NEBULA detector as a square prism, consist of several, smaller, also square prism-shaped scintillator rods. This is already sufficient enough to get approximate simulation results, but it could be improved later on.

3. Technical details

A typical Geant4 simulation consist of two main parts. The first is the definition of geometry and other general behaviour in case of specific events during a simulation, eg. the calculation or logging of specific quantities. Learning this part is generally the real and biggest challenge using Geant4.

The second main part in a full-fledged Geant4 simulation is the creation of specific command pipeline, which is automatically executed whether at the startup of an interactive session or whether executed as a „no-graphics" instance, called by the compiled binary of the current simulation. This is the easier half of the work, since the available commands are predefined by default they have a very simple, pseudo code-like syntax.

4. Automation

Besides slowly creeping up the learning curve of Geant4, this week I've concentrated a lot on the automation of the project pipeline too. When using a simulation software for scientific purposes, an important prerequisite for the smooth and fast progress is to run it with properly selected parameters and commands. Setting them up before each run is a long and tiresome job and because it requires active human supervision at almost any given level, it can be a source of plenty of problems. That's why a properly set up automation greatly speeds up the simulation process in almost all cases.

Command line automation can also be a great help in the meticulous configuration and installing of a simulation application and it can be used to set up the required software environment instantaneously. Considering these benefits, I fully automatized the first half of the pipeline, where every software and library is set and ready to use.

5. Plans for next week

The next step is to finalize the `run` scripts, that encodes the actual events during a simulation. This will handle the specifications of the neutron beams, number of events and others.

[1] http://atomfizika.elte.hu/akos/tezisek/szd/pesznyakdavid_BScszd.pdf