KIT-Fakultät für Physik
Institut für Experimentelle Teilchenphysik

Karlsruher Institut für Technologie

Vorlesung: Prof. Dr. T. Ferber
Übung: Dr. T. Chwalek
Assistenz: O. Lavoryk, M. Molch, M. Mormille, R. Quishpe

Übungen zu Teilchenphysik I

Wintersemester 2023/24

Exercise 3 To be worked on until November 16, 2023

Introduction to GEANT, Part 1

About GEANT

In experimental particle physics, Monte Carlo (MC) methods (simulations) are used in order to design detectors, understand their behaviour, and compare experimental data to theory. MC production proceeds in two steps: event generation and detector response simulation. In the first step, sets of outgoing particles produced in the collisions of incoming particles in the accelerator called *events*, are generated. Then, the detector response to these particles is simulated. The most widely used tool for this purpose in particle physics is GEANT4[1].

GEANT4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include particle, nuclear, and accelerator physics, as well as studies in medical and space science. GEANT4 includes tools for handling geometry, tracking, detector response, run management, visualisation, and a user interface:

- "Geometry" is a description of the physical layout of the experiment. This includes the detectors, absorbers, etc. and considerations how the layout will affect the trajectories of particles in the experiment.

[1] https://geant4.web.cern.ch

- “Tracking” is the simulation of the passage of a particle through matter. This takes into account possible interactions and decay processes.
- “Detector response” is a record of when a particle passes through a given volume of a detector and approximates how a real detector would respond.
- “Run management” includes recording the details of each run (a set of events) as well as setting up the experiment in different configurations between runs.
- The GEANT4 package offers a number of options for visualisation, including OpenGL, and a user interface, based on tcsh.

For the simulation of the events we will use GEANT4. We will steer GEANT4 using Python via the interface provided with the `geant4_simulation`.This package is a wrapper for the Geant4 python bindings `Geant4Py`[2]. It provides the necessary code to perform simple detector simulations with GEANT4. This includes the description of various geometries needed for the exercises, some additional classes that allow the tracking of particles and energy deposits in the detector and a simple interface to steer the simulation. The parts of the interface necessary to work on the exercises are briefly described in the corresponding section of the Jupyter Notebook provided for the exercise.

The exercises, as well as the complete source code and some introductory code showing the usage of the Python package `geant4_simulation`, are provided in the form of Python 3 Jupyter Notebooks in the following repository known from the previous exercises:

`https://gitlab.etp.kit.edu/Lehre/tp1_forstudents`

Log on to the jupytermachine and choose the **Geant4** image. If you have already loaded an image, you might have to change it as described in the first exercise. As usual, after updating the repository, you will find a new folder `Exercise03` containing the Jupyter Notebook, as well as the python package `geant4_simulation` required for this exercise.

The exercise will be done inside the Jupyter Notebook where all descriptions given below are contained as well. So from this point on you only need the Jupyter Notebook to work on the exercise.

[2] `http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/Appendix/pythonInterface.html`

Commands of the GEANT interface

In this section, some of the commonly used code snippets of the GEANT4 interface are given. These are given in an introductory fashion in the Jupyter notebook as well. The following Python commands will create a particle simulation and an event display with GEANT4. You will also find them in the Jupyter Notebook.

- g4 = g4sim.ApplicationManager()

  create a new instance of the ApplicationManager class.

- g4.set_geometry(<filename>)

  set the geometry to the one described in the file:
  ./geant4_simulation/geometry/<filename>.py.

- g4.set_physics_list('QGSP_BERT')

  set the 'physics lists' we will need in the corresponding exercise. The physics list specifies all particles and interactions considered in the simulation. It is mandatory for all simulations, you do not need to modify it and the relevant physics lists for each exercise are given in the exercise.

- g4.initialize()

  initialise the GEANT4 kernel.

- g4.set_particle(<PDG code>)

  set the type of the primary particle. The <PDG code> is an integer number defined by the Particle Data Group to specify different particle types[3]. For example, <PDG code>=22 is a photon.

- g4.set_energy(<energy>)

  set the energy of the primary particle in units of GeV.

- g4.start_run()

  start the simulation. You can also provide an argument which will define the number of simulated events, the default is 1. Also by default, the output of the simulation is printed as text, if you want a graphical representation you need to use visualize=True.

[3] https://pdg.lbl.gov/2022/reviews/rpp2022-rev-monte-carlo-numbering.pdf