

Vorlesung: Prof. Dr. M. Klute, Dr. P. Goldenzweig

Übung: Dr. G. De Pietro

Assistenz: Dr. P. Ecker, J. Eppelt, L. Reuter, J. Saffer, D. Wong

Übungen zu Teilchenphysik I

Wintersemester 2024/25

Exercise 5

To be worked on until January 9, 2025

About GEANT

In experimental particle physics, Monte Carlo (MC) methods (simulations) are used to design detectors, understand their behaviour and compare experimental data with theory. MC production involves two steps: event generation and detector response simulation. In the first step, sets of outgoing particles produced by collisions of incoming particles in the accelerator, called *events*, are generated. The detector response to these particles is then simulated. The most widely used tool in particle physics for this purpose is GEANT4¹.

GEANT4 is a toolkit for simulating the passage of particles through matter. Its applications include particle, nuclear and accelerator physics, as well as medical and space science studies. GEANT4 includes tools for geometry handling, 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 of how the layout will affect the trajectories of particles in the experiment.
- “Tracking” is the simulation of the passage of a particle through matter. This takes into account possible interactions and decay processes.

¹<https://geant4.web.cern.ch>

- “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” involves recording the details of each run (a set of events) and setting up the experiment in different configurations between runs.
- The GEANT4 package provides a range of visualisation options, including OpenGL, and a user interface based on tcsh.

To simulate the events we will use GEANT4. We will control GEANT4 using Python via the interface provided with the `geant4_simulation` package. 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 to allow tracking of particles and energy deposition in the detector, and a simple interface to control the simulation. The parts of the interface needed 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 use of the Python package `geant4_simulation`, are provided in the form of Jupyter Notebooks in the following repository, familiar 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 **Exercise05** containing two Jupyter Notebooks, **Exercise05.Part1.ipynb** and **Exercise05.Part2.ipynb**, as well as the python package `geant4_simulation` required for this exercise.

If you are having trouble updating the repository using `Git -> Pull from Remote`, please switch back to the **Datenanalyse** image temporarily for this task only.

The exercises will be done in the two Jupyter Notebooks, which also contain all the descriptions below. From this point on, you will only need the Jupyter Notebook to complete the exercise.

In this exercise a short introduction is given in notebook **Exercise05.Part1.ipynb**, while a sampling calorimeter is simulated with GEANT4 in notebook **Exercise05.Part2.ipynb**.

²<https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/LanguageBindings/languageBindings.html>

Commands of the GEANT interface

This section gives some of the commonly used code snippets of the GEANT4 interface. These are also given as an introduction in the Jupyter Notebook. The following Python commands create a particle simulation and an event display using GEANT4. You can find them in the Jupyter Notebook as well.

- `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 types³. 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/2024/reviews/rpp2024-rev-monte-carlo-numbering.pdf>