Radioactivity Lab: Simulation User Guide

This simulation of the lab set-up was designed in Geant4, a C++ toolkit developed to simulate the passage of particles through matter. The simulation creates the geometry used in the radioactivity lab, and provides a visualisation of the Sr90 decay and the particle trajectories. In addition, it provides simulated data which you can use for comparison with your own experimental data. It is recommended that you play around with the various commands, e.g. by changing various aspects of the geometry or changing the particles produced, so you can see the effect of each part of the setup.

Basics

To begin a run, include the command `/run/beamOn n` in your macro file. This produces `n` primary particles with a predetermined type, energy and direction, which can be controlled by other commands in the same macro. After a run, if you have switched on the visualisation, all the particle trajectories are displayed simultaneously in the viewer as coloured lines. The colour of the line depends on the particle type according to the following table:

Particle	***Colour***
Electron	Red
Photon	Green
Zr90	Blue
Y90	Yellow

The simulation writes to the text file datarun_number.txt (found in the current directory from which you ran the executable) whenever particles interact with the silicon sheet. The following data is available in the text file (from left to right): total energy deposited in silicon, z-coordinate of the source and thickness of any absorbers present in the geometry (excluding the silicon). Each new row represents a different particle.

The default configuration

The default geometry initialised on start up consists of the wood work-bench, the Sr90 source (a strontium foil surrounded by casing) and the silicon detector complete with metal casing. The commands listed below can be used in the macro files to position the source and detector as required for each part of the experiment. The entire simulation region has dimensions 50 cm x 50 cm x 100 cm — any particles which leave this region are killed and are no longer tracked. The simulation is configured so that the `/run/beamOn n` command produces `n` strontium nuclei and decays them according to the beta decay chain seen in the lab script. A word of warning, if you run the simulation twice in the same directory you will overwrite any datarun_number.txt files that exist in that directory.

Commands

UI commands in Geant4 are used to initiate runs, configure new geometries and generally alter the workings of the program between runs. Commands have the following format

```
/directory/subDirectory/.../command [parameter1] [parameter2]
                                ...
```

with the number of parameters depending on the command. The parameters must be separated by single spaces. The commands that have been created specifically for this experiment are listed below. Geant4 also provides default commands which you can use for greater customisation — a full list of these can be found *here* [1]. In addition, specific commands for controlling the particle source can be found *here* [2].

Macro files (text files containing multiple commands which are executed one after another) have been provided which will enable you to collect simulated data for the lab experiments. To collect the data you need to update the macro file to match your lab experiment, execute the macro file using the command line and wait for it to finishing running. You should then import the data into Python where you can graph it and carry out analysis. To execute a macro file, run the simulation executable `lab.exe [-b] [filename.mac]`. The '-b' option runs the executable in batch mode, which switches off the visualisation and is therefore much faster. Note that any statement following a hash (#) in the macro file is not executed by the program, it is only displayed in the terminal.

As mentioned previously, the command `/run/beamOn n` generates `n` primary particles, propagates them, creates any required secondaries (such as decay products or ionisation electrons), and tracks all particles until they have left the simulation region. This is therefore a very important command.

All geometry modifying commands are preceded with the directory `/lab`. The commands in the directory `/action` control specific actions which take placing during a run.

/lab/

`/lab/worldMat [material]`

Change the material of the world volume (the simulation region).

`material` candidates: `G4_AIR (air) G4_Galactic (vacuum)`

`/lab/benchOn`

Create wood workbench.

`/lab/benchOff`

Remove wood workbench.

[1] `http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/Control/AllResources/Control/UIcommands/_.html`

[2] `http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/GettingStarted/generalParticleSource.html#g4particlegun-equivalent-commands`

`/lab/AlSheet [z_coord] [thickness]`

Create an Aluminium sheet on axis with specified z coordinate and thickness.

`z_coord` candidates: any real, default $= 0$, unit $=$ cm.

`thickness` candidates: any real ≥ 0, default $= 0$, unit $=$ cm.

`/lab/AlSheetDel`

Remove Aluminium Sheet.

`/lab/CuSheet [z_coord] [thickness]`

Create a Copper sheet on axis with specified z coordinate and thickness.

`z_coord` candidates: any real, default $= 0$, unit $=$ cm.

`thickness` candidates: any real ≥ 0, default $= 0$, unit $=$ cm.

`/lab/CuSheetDel`

Remove Copper Sheet.

`/lab/LeadSheet [z_coord] [thickness]`

Create a lead sheet on axis with specified z coordinate and thickness.

`z_coord` candidates: any real, default $= 0$, unit $=$ cm.

`thickness` candidates: any real ≥ 0, default $= 0$, unit $=$ cm.

`/lab/LeadSheetDel`

Remove lead Sheet.

`/lab/Source [x_coord] [y_coord] [z_coord] [angle]`

Create the Sr90 source including casings with the specified coordinates and orientation.

`x_coord` candidates: any real, default $= 0$, unit $=$ cm.

`y_coord` candidates: any real, default $= 0$, unit $=$ cm.

`z_coord` candidates: any real, default $= 0$, unit $=$ cm.

`angle` candidates: any real, default $= 0$, unit $=$ deg.

`/lab/SourceDel`

Remove Source.

`/lab/SiDet [x_coord] [y_coord] [z_coord] [angle]`

Create the Silicon sheet used for detection without casing with the specified coordinates and orientation.

`x_coord` candidates: any real, default $= 0$, unit $=$ cm.

`y_coord` candidates: any real, default $= 0$, unit $=$ cm.

`z_coord` candidates: any real, default $= 0$, unit $=$ cm.

`angle` candidates: any real, default $= 0$, unit $=$ deg.

`/lab/SiDetDel`

Remove Silicon sheet without casings.

`/lab/SiDetAndCasings [x_coord] [y_coord] [z_coord] [angle]`

Create the Silicon sheet used for detection with casing with the specified coordinates and orientation.

`x_coord` candidates: any real, default $= 0$, unit $=$ cm.

`y_coord` candidates: any real, default $= 0$, unit $=$ cm.

`z_coord` candidates: any real, default $= 0$, unit $=$ cm.

`angle` candidates: any real, default $= 0$, unit $=$ deg.

`/lab/SiDetAndCasingDel`

Remove Silicon sheet with casings.

`/action/` and `tracking`

`/action/killBackwardToggle`

Toggle if backward travelling particles are killed. By default this is false.

`/action/ignoreProductsFromSiToggle`

Toggle if particles produced in the silicon sheet are included in the output files. By default this is true.

`/tracking/storeTrajectory [bool]`

Boolean to store the particle trajectories during simulation. By default this is true. This must be switched on to use the visualisation.

`/action/setStoreDetectedOnly`

Set this to only store the events that are actually detected by the silicon detector.

`/action/killAngularRange [cosine angle to z-axis]`

Kill any particle for which the cosine of the angle it makes to the z-axis is less than the argument to this option.