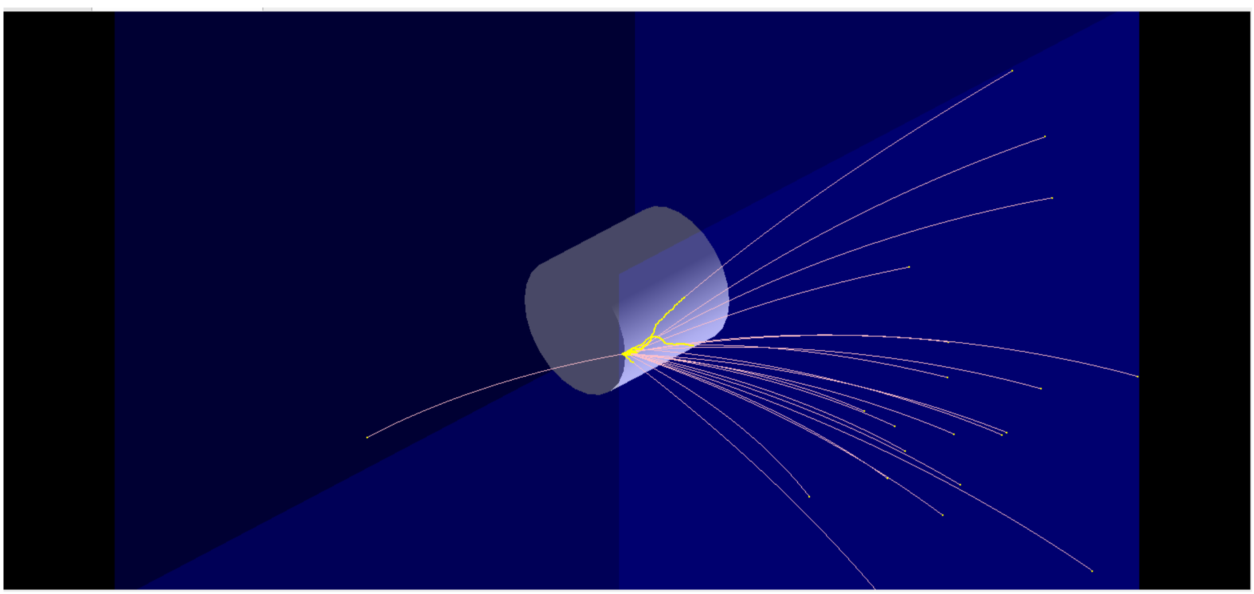
Geant4 Exercises 1

Part 1

Following is rendered image of 20 muons hitting the target with 20MeV energy, and muon ionisation and Bremsstrahlung processes deactivated in a magnetic field of 0.05 tesla.

Code used in simulator

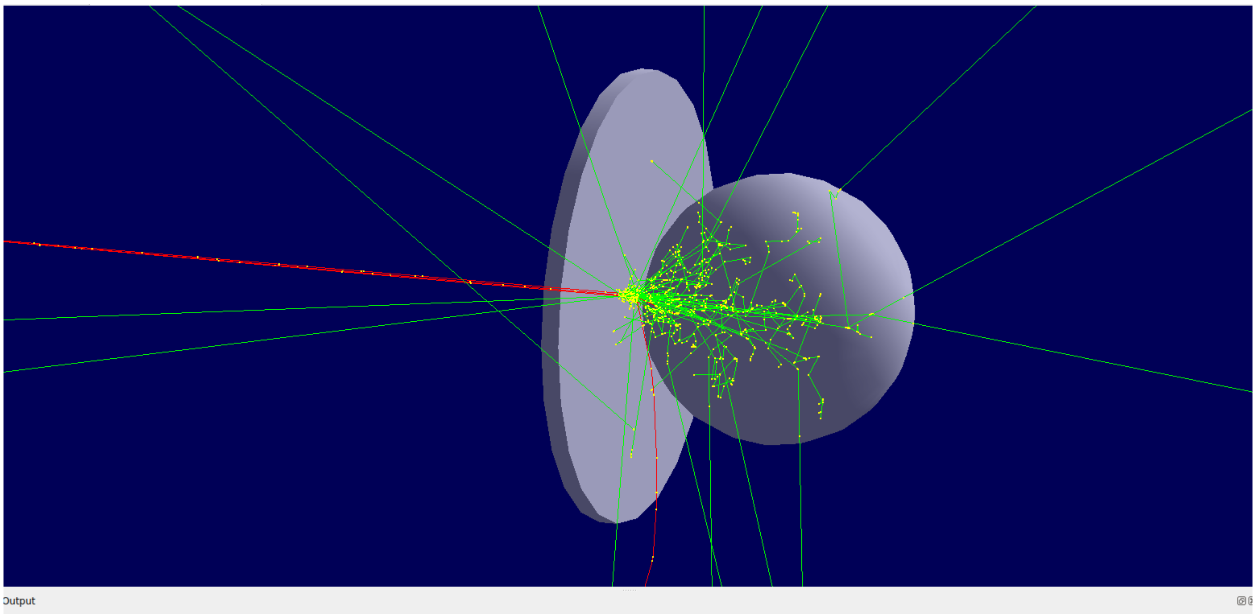
```
#Random seed 1234 is, by default
/globalfield/setvalues 0 0.05 0 tesla
/gun/particles mu-
/gun/energy 20MeV
/process/inactivate muIoni
/process/inactivate muBrems
/run/beamOn 20
```



Part 2

Adding an object

A new target object, the sphere, has been added behind the shield. The object is a sphere with aluminium as the material.



- Rendered image of the lead shield and aluminium target being bombarded by mu+ beam

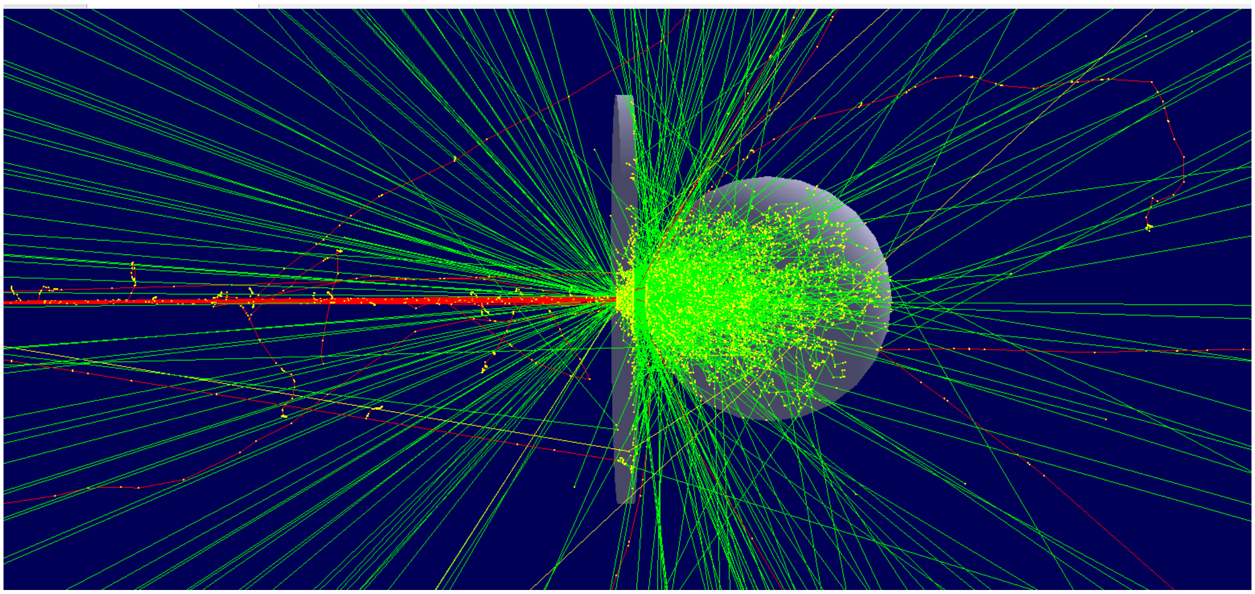
The code to make the modification in the setup is implemented in Detettor.cpp file. Basically, the G4Sphere module is included, and new material, aluminium, is added and applied to the code accordingly. The modified detector.cpp file is in the submitted work.

Making Object Energy Counter

A new Object is made Energy Counter by modifying action.cpp and detector.cpp file. The modified code is in the submitted work.
(Using Object ID 1)

Running the simulation on 200 Events

The program is rerun using 200 mu+ particles with 300 KeV energy in the magnetic field of 0.01 tesla.
#Random seed 1234 is, by default
/globalfield/setvalues 0 0.01 0 tesla
/gun/particles e-
/gun/energy 300MeV
/run/beamOn 200



-Render of Lead shielding from electron beam coming from left to right

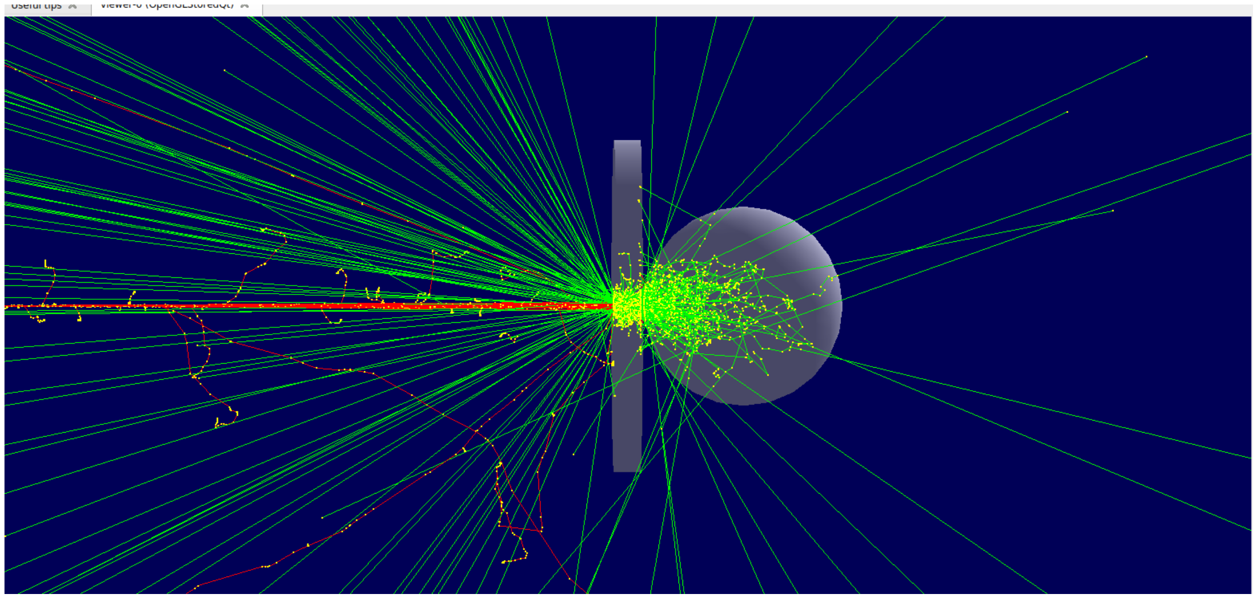
Remember first. A 2 cm thick lead shield protects a 30 cm radius aluminium target.
Output cvv file is included in the submitted work in a folder named experiment_with_thin shield and other relevant files.

Python Code and relevant cvv files are in folder named python_Analysis.

Varying Thickness and running Experiment again

Note - Currently I don't know how to vary thickness in the simulated environment itself, In analysis, I changed the thickness of the shield and then rerun the simulation.

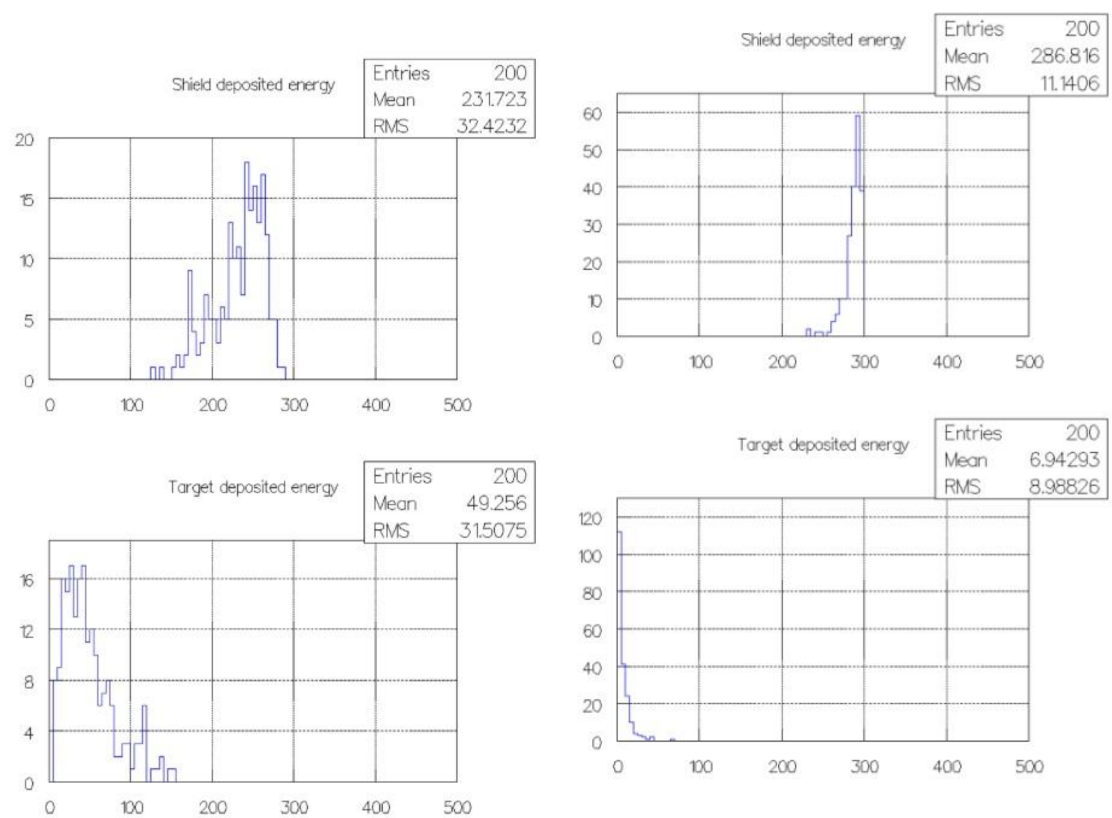
#Random seed 1234 is, by default
/globalfield/setvalues 0 0.01 0 tesla
/gun/particles e-
/gun/energy 300MeV
/run/beamOn 200



-Render of thicker lead shielding from electron beam.

Note - In the render not all 200 particles are depicted, But the ratio is maintained. It is evedent from render that increacing the thickness of shield increases its effictivness. We will do further analysis in python to solidify this statement.

.



-Shild absorb most of the energy of the beam(left) as the shild thickness is increased absorbed energy also increased(right)

Python code

```
import numpy as np

#function to create numpy array from csv file
def import_csv_as_array(file_path):
    return np.genfromtxt(file_path, delimiter=',')

#Defining 4 relevant csv file
thin_shield = import_csv_as_array('output_thin_Shield.csv')
thick_shield = import_csv_as_array('output_thick_Shield.csv')
object_case1 = import_csv_as_array('output_thin_Target.csv')
object_case2 = import_csv_as_array('output_thick_Target.csv')

#Ratio of energy deposit on object to shield
Energy_deposit_1 = object_case1/thin_shield
Energy_deposit_2 = object_case2/thick_shield

#converting numpy array to list
list_1 = Energy_deposit_1.tolist()
list_2 = Energy_deposit_2.tolist()

#defining the function to take average
def average(lst):
    return sum(lst) / len(lst)

#finding average of energy deposit ratio
avg1=average(list_1)
avg2=average(list_2)

print(f"2.0 cm lead shield {1-avg1:.4} ratio of energy, and 4.0 cm lead shield {1-avg2:.4} ratio of energy")
```

2.0 cm lead shield 0.761 ratio of energy, and 4.0 cm lead shield 0.9746 ratio of energy
(Ratio of energy means "average of the ratios of deposited energies in the two objects")

-Ritesh
s2456148