

Hands-On Session 10

Code without Solution: hands-on-10.tar.gz

Code with Solution: hands-on-10-solution.tar.gz

Exercise 10

The exercise shows one usage of the biasing classes.

We have a shield, made of concrete, which is inserted between the calorimeter and the screen of session 9. This shield is divided in 10 slices, using a replica.

We want to estimate the efficiency of the shield to neutrons, and characterize the neutrons which succeed in exiting the shield. As this one is doing its job, only a few neutrons can traverse this shield. The goal of the exercise is to bias the simulation in the shield to get more neutrons traversing.

Intro

The following items have been introduced

- Two biasing classes:
 - EDBiasingOperator
 - EDBiasingOperationSplitAndKill

in the detector construction, the **ConstructSDandField()** method has been modified to attach the biasing operator to the logical volume of the shield slices

- In the main program, a command line option `< b >` has been added to activate or not the biasing, this option controls the use or not of the [G4GenericBiasingPhysics](#) physics constructor

1. Compile and run the code in a nonbiased way (so called “analog” way)

./exampleED -b off

Look at how the neutron processes appear.

Shoot a few protons, to verify that the shield is indeed preventing most of particles to traverse

2. Run the code in biased mode

./exampleED -b on

Look at how the neutron processes appear now.

Shoot a few protons, you will see that now more particles can go through the shield

Part A

The [EDBiasingOperator](#) must send the [EDBiasingOperationSplitAndKill](#) biasing operation to be applied.

- In the code here, it does not. Make it returning the biasing operation using the proper method.
- Run the code in biasing mode to verify that now particles go through the shield.
- You will notice that the track weight has been added to the ntuple. Plot some quantities (eg: Ekin) not using and using the weight:¹

```
|| root [] Screen->Draw("Ekin");
|| root [] Screen->Draw("Ekin","weight");
```

Part B

We will reduce the flux backward, for this apply a « Russian Roulette » technique

Look at the corresponding section of the code on the [EDBiasingOperationSplitAndKill](#) class, which provides further hints to implement the killing

¹the second plot is the correct one