Subject: Detector ID
Posted by Ina Syndikus on Wed, 07 Jun 2017 13:33:08 GMT
View Forum Message <> Reply to Message

I'm looking at the output file of my simulation and trying to find out which secondary particles my protons produce in the (active) volume of XB. One of the things that I looked at is TrackPoint.fDetectorID. Its value is always 38. Is this XB? How do I find out which detector has which number? Is there a list?
And when we talk about detector IDs: R3BXBallPoint has also a XBCrystalPoint.fDetectorID, which has the values 6,8,10 & 12. I assume the numbers stand for the 4 crystal shapes. Am I right?

Subject: Re: Detector ID
Posted by Dmytro Kresan on Mon, 12 Jun 2017 06:39:22 GMT
View Forum Message <> Reply to Message

I was not able to find any match with "TrackPoint" branch in R3BRoot. Did you mean "TraPoint"? In this case, those are the hits from Tracker. Matching of data to a detector is done using the name: R3BXBall (detector) - R3BXBallPoint (hits of particles), the name of the branch has to be changed to "XBallPoint" (XBCrystalPoint is currently used).

There is enum type in r3bdata/R3BDetectorList.h, but it is used for counting how many points in each detector an MC track has, and not for storing together with simulated hits.

fDetectorID is meant to be sub-system internal, and it looks like, in case of XBall, it is the Volume ID. This number changes, depending on the order you add your detectors to the simulation run in the macro. It is better to use R3BXBallPoint.fCrystalType (1,2,3,4) and .fCrystalNb (running index, I suppose).

Best regards,
Dima

Subject: Re: Detector ID
Posted by Ina Syndikus on Tue, 13 Jun 2017 19:10:47 GMT
View Forum Message <> Reply to Message

Sorry, my fault. The branch is indeed TraPoint. I assume the tracker is the SSD Detectors surrounding the target (which is called "TRACKER" when I include it in the r3bsim.C file)?

I have another question about the classes "XBallPoint" & "XBallCrystalHitSim". It seems that "XBallPoint" includes all the particles passing XB, including a lot of electrons. But what is with "R3BXBallCrystalHitSim"? It includes much less particles and seems to get rid of all the electrons. How is this done? Is there a cutoff (additional to the one for Geant)? And is the energy of the electrons summed to the energy of the other particles?

Subject: Re: Detector ID
Posted by Dmytro Kresan on Wed, 14 Jun 2017 08:01:05 GMT
View Forum Message <> Reply to Message

Quote: I assume the tracker is the SSD Detectors surrounding the target (which is called "TRACKER" when I include it in the r3bsim.C file)?

Yes, it is correct.

XBallPoint contains one object for every particle which has hit an active volume of a crystal. Typically many entries per crystal.

XBallCrystalHitSim is supposed to describe a hit in a crystal, which integrates all relevant Points. 1 hit per crystal.
Energy loss is summed up. Optionally you can switch on some uniform smearing, controlled with R3BXBall::SetNonUniformity(....) method.
Details of the algorithm: R3BXBall.cxx lines 258 - 280.

Best regards,
Dima