Subject: Re: Event filter after simulation phase
Posted by MG on Fri, 24 Apr 2020 21:05:23 GMT

I got a bit stuck again.   I've implemented custom Stack based on the example, along with a simple filter on track momentum. What I've found is that the tracks are indeed removed, but hits from these tracks registered in detector are not, they just have their trackID set to -2. I would like to remove these hits as well.

I've tried updating the loop in FairStack::UpdateTrackIndex method, adding

if(-2 == point->GetTrackID())
   hitArray->RemoveAt(iPoint);

at the end, after the indices are set properly and also modified the loop to go backwards, so I don't break the indexing by removing items:

for (Int_t iPoint = nPoints - 1; iPoint >= 0; --iPoint)

This produces interesting results, which I've found after printing the size of hitArray before and after the call to RemoveAt. Basically, it works correctly up to some point and then stops removing items, even though the index is smaller than the size of the array.

Example:
Quote:
index 13 trackID -2
size before 14
size after 13
index 12 trackID -2
size before 13
size after 12
index 11 trackID -2
size before 12
size after 11
index 10 trackID 7
index 9 trackID -2
size before 11
size after 11
index 8 trackID -2
size before 11
size after 11

This behavior continues through many events and then it crashes with segmentation violation at some point (like 800 events in). It seems to always appear after the first case of trackID not being -2, as if only removing items from the end of CloneArray worked. My guess is that I'm breaking something with memory, because the branch is already registered and filled.

Is there a better way to do this?

Thanks!