

## LArSoft - Bug #12197

### Geant4 output different when changing G4-unrelated configuration

04/07/2016 07:35 AM - Matthias Luethi

<b>Status:</b>	Closed	<b>Start date:</b>	04/07/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Gianluca Petrillo	<b>% Done:</b>	100%
<b>Category:</b>	Simulation	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	16.00 hours
<b>Occurs In:</b>	v04_36_00	<b>Co-Assignees:</b>	
<b>Experiment:</b>	MicroBooNE		

#### Description

I was checking if the raw wire signals with spacecharge turned on and off look as they were similar for the same event. Let me explain what I did and what I wanted to accomplish:

The main goal was to compare single muon tracks with and without spacecharge effects switched on, so we can quantify the effect of spacecharge on tracks. To accomplish this in a first stage I did the generation with this parameter switched on (so I do not get random seeds for the later geant4 stage) using the single\_muon\_gen.fcl.

services.SeedService: @local::per\_event\_seedservice

Then I do the detector simulation stage once with spacecharge switched on and off separately (one needs the SCEoffsets\_70kV.root file do do this), with the service above switched on in this fhicl file as well. Unfortunately when I look at the two first events in each set I generated, they look very different and certainly not like they are just a distorted version of each other (see the attached two png files). It seems that although I am using the seed service per event, geant4 does not take the same seed for simulation. Do you have any idea what I am doing wrong (I also put in the fhicl files if you have time to look at them)?

#### History

##### #1 - 04/07/2016 09:53 AM - Gianluca Petrillo

- Subject changed from Seed to Geant4 output different when changing G4-unrelated configuration
- Status changed from New to Assigned
- Assignee set to Gianluca Petrillo
- Occurs In v04\_36\_00 added
- Occurs In deleted (v0\_02\_00)

##### #2 - 04/07/2016 10:04 AM - Gianluca Petrillo

I could reproduce the issue.

Configurations are [A] without space charge and [B] with space charge.

Each of them is singularly perfectly reproducible and in full control of random seeds.

The very first particles of jobs [A] and [B] also match perfectly.

At a certain point, Geant4 random number generator is asked more random numbers in one job than in the other for the same particle, and the simulation diverges.

Space charge is supposed not to have any random component, and to affect only propagation of electrons to the wires.

It might be that randomness is used *during* the propagation, that might be affected by the position.

I have to dig deeper.

##### #3 - 04/07/2016 10:05 AM - Gianluca Petrillo

- % Done changed from 0 to 10

##### #4 - 04/07/2016 11:02 AM - Gianluca Petrillo

The issue is understood.

The transportation mechanism first applies attenuation to the electrons generated in the active volume, reducing their number.

The attenuation represents the loss of electrons while they follow the electric field through the liquid argon toward the wire planes. This reduction is proportional to the distance from the wire plane (x coordinate).

Then the electrons are divided in "clusters" and each cluster is subject to diffusion. The amount of diffusion is stochastic.

So, in summary: the space charge dislocation changes the amount of attenuation, that may change the amount of electron clusters, that would change the number of diffusion-related random numbers extracted.

Solution is in progress. There may be some further effect after this one that I have missed so far.

**#5 - 04/07/2016 11:03 AM - Gianluca Petrillo**

- % Done changed from 10 to 40

**#6 - 04/08/2016 06:11 PM - Gianluca Petrillo**

- Status changed from Assigned to Resolved

- % Done changed from 40 to 100

I have split the random number stream of LArG4 into three:

- generation (the engine used also by Geant4) ["GEANTSeed"]
- propagation: NestAlg, part of LArVoxelReadout ["PropagationSeed"]
- radiological decay: part of LArVoxelReadout ["RadioSeed"]

The idea is that changes in the propagation and in the radiodecay will not affect the generation.

**The changes are somehow breaking:** the control of LArG4 random seeds is now done with three configuration parameters (in brackets in the list above) rather than just with the old Seed parameter.

There should be no consequence when the configuration does not try to override seeds.

Of course, the new version is not able to replicate old runs, since the random streams will differ.

I have published the branch feature/gp\_Issue12197 for larsim repository, based on develop (v05\_08\_00).

I have verified that on two events the generation is the same (comparing log messages at DEBUG threshold), and that is all the verification I can do. A bit of physics validation is recommended.

I do not plan to port the change back in v04\_36\_00, where the feature request started from.

It is a hard work, it would compromise v. 4.36 jobs and it's not really worth the effort.

As a side effect, I have developed a small debug tool that prints the current call stack. It was useful to track who was calling random functions without the use of a debugger.

It is published in the branch feature/gp\_DebuggingTools in larcore.

**#7 - 06/17/2016 03:46 PM - Gianluca Petrillo**

- Status changed from Resolved to Closed

**Files**

single_muon_all_with_dist.fcl	17.6 KB	04/07/2016	Matthias Luethi
SCEoffsets_70kV.root	88.7 KB	04/07/2016	Matthias Luethi
single_muon_gen.fcl	3.94 KB	04/07/2016	Matthias Luethi
single_muon_all_sans_dist.fcl	17.6 KB	04/07/2016	Matthias Luethi
evd_uboone_R1_S0_E2_with_col.png	1.97 MB	04/07/2016	Matthias Luethi
evd_uboone_R1_S0_E2_sans_col.png	1.96 MB	04/07/2016	Matthias Luethi