MANCHESTER
1824

The University
of Manchester

Tutorial 2: Understanding Basic Light Simulation

Andrzej Szelc

Using material from Alex Himmel's DUNE tutorials, found here:
https://cdcvs.fnal.gov/redmine/projects/dunetpc/wiki/Photon_Simulation_Tutorial

Intro

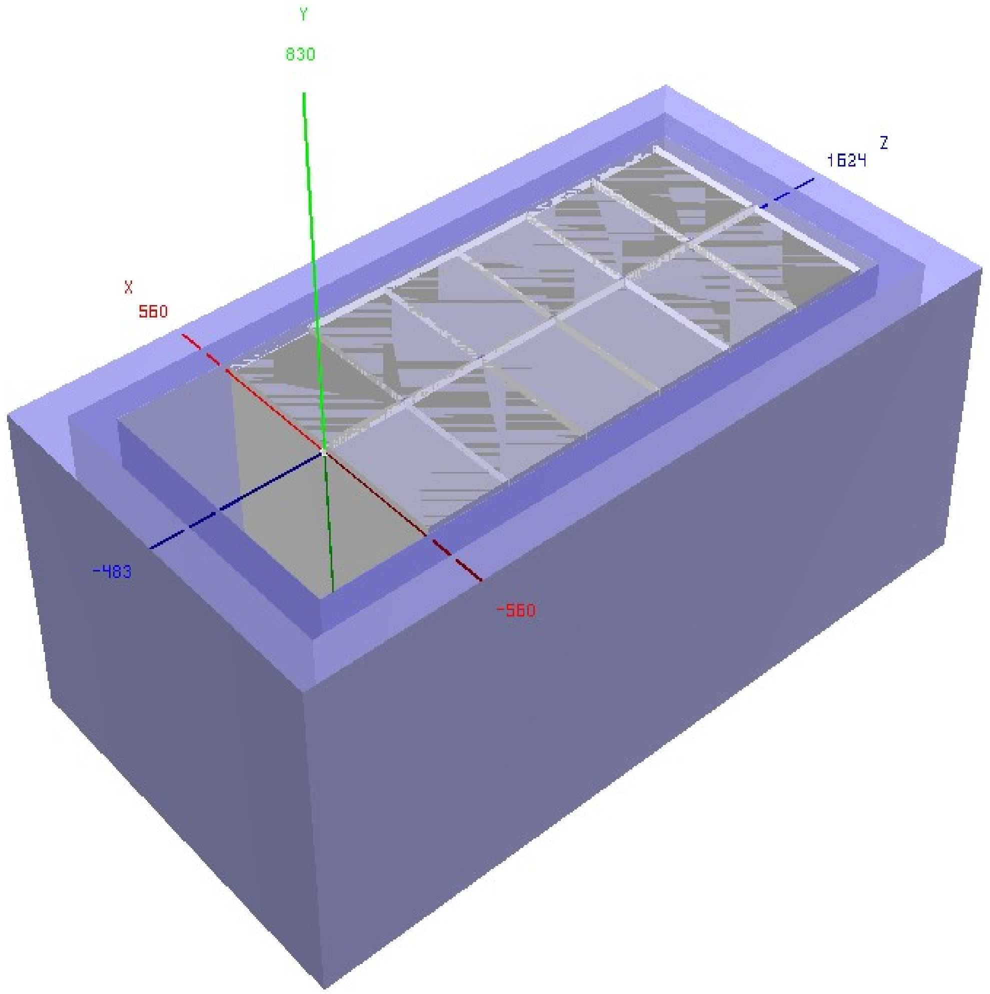
- In this tutorial we will look at simulating scintillation through the LArGeant Stage.
- We will take a look at how DUNE optical libraries look.
- We will see what the output of a particle travelling through the detector is.
- And then we will try to change its parameters somewhat to see what happens.

Reminder: Log in to Your Computing Account

- ssh -X -Y user01@18.231.121.254
- Password is written on the whiteboard
- Most files you will need will be in /home/andrzej/workshop_files/

The University of Manchester

Reminder nr. 2: we are working with the 1x2x6 DUNE geometry



Light Detectors are in the middle in sets of 10.

We will be playing with a version that has WLS covered foils on the cathodes (on both sides)

You will be able to play with different locations of generated files.

Note on using Optical Libraries

- Optical Libraries are quite large - ~1.8GB currently for DUNE. This causes all kinds of trouble, e.g. recently for grid jobs (each job needs to access this file).
- Optical library files approved for use live in the dune_pardata package.
  - You already have it setup most likely. Check with:
    ```
    ups active -aK+  | grep dune_pardata
    ```
- For the most recent versions we put them in to a thing called stash-cache, which means they **should** be found in /cvmfs/dune.osgstorage.org/
- It doesn't work out of the box alas, so we will use a hacky solution.

Libraries at this Workshop

- Can be found in: /home/andrzej/workshop_files/library/
- Do an "ls" there. There are two library files there: "v1" and "v4". There are also corresponding .gdml files. You will need to use the correct combinations.
- To use them do: `export FW_SEARCH_PATH=$FW_SEARCH_PATH:/home/andrzej/workshop_files/library/`
- This will tell LArSoft that this directory is a place where it can look for the files. You will still need to specify which ones though.

MANCHESTER 1824

The University of Manchester

Second step to making our lives easier

- Most of the files are crafted for this workshop and are in the /home/andrzej/workshop_files
- You can either copy them to your home directory, or:
- `export FHICL_FILE_PATH=$FHICL_FILE_PATH:/home/andrzej/workshop_files`
- This adds my directory to the path Larsoft uses to search for .fcl files.

Task 1, prep:

- You will need the file: `optical_tutorial_libanalysis.fcl`
- You will need a version of larsim from my directory with `/feature/andrzej_workshop` checked out and compiled. If it has not compiled, try:

```
$ source /home/andrzej/larsoft/localProducts_larsoft_v07_11_00_e17_prof/setup
$ mrbslp
```

The University of Manchester

MANCHESTER 1824

optical_tutorial_libanalysis.fcl

```
physics:
{

 producers:
 {
 }

 analyzers:
 {
   libana: @local::dunefd_photonlibraryanalyzer
 }

 analyzeIt:  [ libana ]

 #trigger_paths is a keyword and contains the paths that modify the art::event,
 #ie filters and producers
 trigger_paths: []

 #end_paths is a keyword and contains the paths that do not modify the art::Event,
 #ie analyzers and output streams.  these all run simultaneously
 end_paths:     [analyzeIt]
}
```

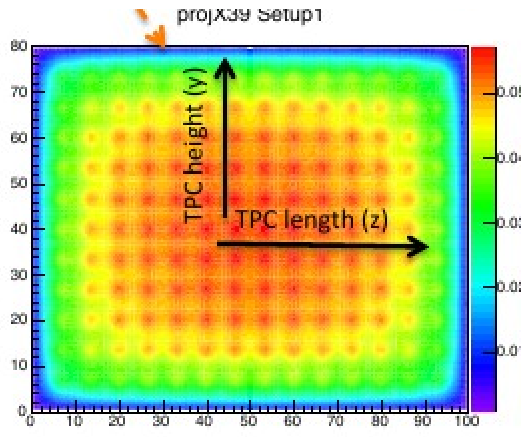
```
## DUNE geometry with foils and cryostat.
services.Geometry.GDML: "dune10kt_v4_1x2x6.gdml"
services.Geometry.ROOT: "dune10kt_v4_1x2x6_nowires.gdml"

services.PhotonVisibilityService.LibraryFile: "dune10kt_v4_1x2x6_withfoils_lib.root"
services.PhotonVisibilityService.StoreReflected: false
```

Task 1:

- Run `optical_tutorial_libanalysis.fcl`

  for both libraries. Make colz comparion plots of XProjection,YProjection, ZProjection.
- What is the difference?
- Change StoreReflected to “true”. Rerun the analsysis files – what has changed?



Tip 1:
-T filename.root
In the lar call overrides
the filename set in the .fcl file

Tip 2:
A bare bones .root script
to make a colz plot from this
file is in backup

Bonus Task:
Modify
PhotonLibraryAnalyzer
To make a 1d plot in YX
Plane. Or make it
Otherwise. Compare
StoreReflected: true vs false

Task 1, cont'd.

- If you managed to get the StoreReflected true/false files you should have noticed that there is more light closer to high x values.
- This is because these geometries include WLS covered foils on the CPA.
- This results in the extra light. This light should arrive at the CPA whilst visible.

This how such a foil produced for SBND Looks like.

Task 2 prep

- You will need the file: `optical_tutorial_sim_muons.fcl`
- This file will generate 2GeV muons at a certain place in your detector.
- The analyzer module will provide 4 TTrees with interesting information. We will take a look at what information resides there.

Task 2

```
# Define and configure some modules to do work on each event.
# First modules are defined; they are scheduled later.
# Modules are grouped by type.
physics:
{

 producers:
 {
   generator: @local::dunefd_singlep
   largeant:  @local::dunefd_largeant
   rns:       { module_type: "RandomNumberSaver" }
 }

 analyzers:
 {
   pmtresponse: @local::dunefd_simphotoncounter


 }
```

This is where we tell the analyzer which trees we want.

```
#
physics.analyzers.pmtresponse.MakeAllPhotonsTree: true
physics.analyzers.pmtresponse.MakeDetectedPhotonsTree: true
physics.analyzers.pmtresponse.MakeOpDetsTree: true
physics.analyzers.pmtresponse.MakeOpDetEventsTree: true
physics.analyzers.pmtresponse.MakeLightAnalysisTree: false
```

```
#
services.LArPropertiesService.ScintByParticleType:false
services.LArG4Parameters.UseLitePhotons: true
services.PhotonVisibilityService.ReflectOverZeroX: true
services.PhotonVisibilityService.StoreReflected: true
```

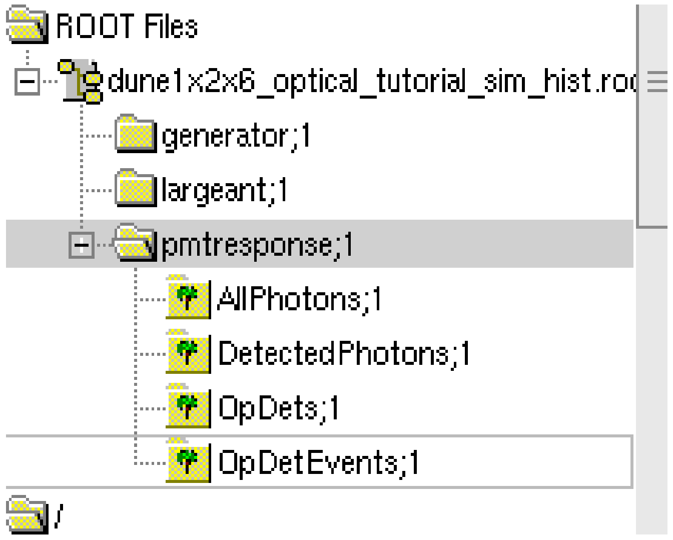
Use the lightweight version of photons.

Use symmetry to project half of library to both sides

Requesting Reflected Light

Task 2 cont'd

- Take a look at the `_hist.root` file you got as a result. It has four TTrees:
  - AllPhotons, all the emitted photons
  - DetectedPhotons, self explanatory
  - OpDets, (on per PD basis)
  - OpDetEvents (on a per event basis)



```
ROOT Files
  dune1x2x6_optical_tutorial_sim_hist.roo
    generator;1
    largeant;1
    pmtresponse;1
      AllPhotons;1
      DetectedPhotons;1
      OpDets;1
      OpDetEvents;1
/
```

- Take a look at the DetectedPhotons tree:
  - Do the OpChannel, Wavelength and Time plots look reasonable?
  - Try to measure the slow timing constant of argon.

The University of Manchester

Task 2.1

- You will need the file: `optical_tutorial_sim_marley.fcl`
- Launch a set of events. You can go larger than previously (a 100?) – it's much quicker.
- Make a plot of OpChannel on a per event basis.
  - Compare with the muon results. What is the number of events on a per-event basis? (remember to normalize)

    look at the timing distributions of the detected photons. Are they different? Any idea why?

Remember Tip 1:
-T filename.root
In the lar call overrides
the filename set in the .fcl file

MANCHESTER 1824

The University of Manchester

Task 2.2

- Let's go back to: `optical_tutorial_sim_muons.fcl`

```
#
services.LArPropertiesService.ScintByParticleType:false
services.LArG4Parameters.UseLitePhotons: true
services.PhotonVisibilityService.ReflectOverZeroX: true
services.PhotonVisibilityService.StoreReflected: true
```

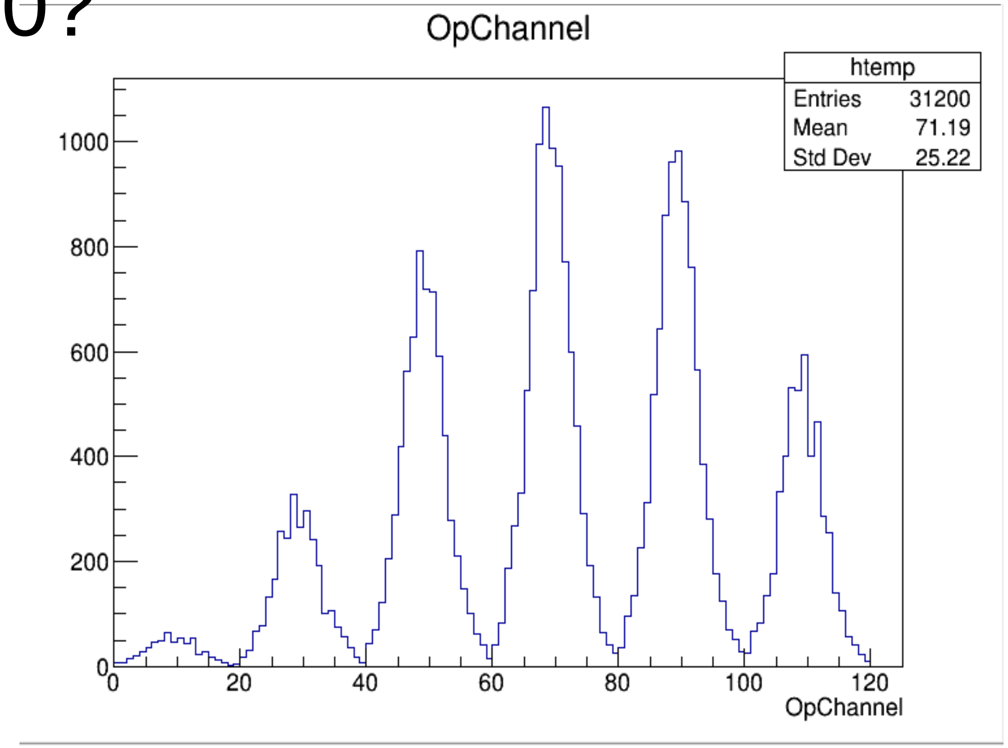
- Since we are using the Reflected light, some of the photons should be coming to the detectors at visible wavelengths. SimPhotonsLite don't store this info, so no wonder we didn't see them.
- Let's change the use OnePhotons and see if it helps.

Task 2.2

- That doesn't work. :(
- It's not you, it's LArSoft. There is a bug in the SimPhotonCounter that I was not able to fix in the last few days.
- For whatever reason only one type (standard/reflected) gets read in at a time (they are both saved). Tweaking requires modifying LArG4 – this can be a bonus task if you like.
- If you want to look at what happens for reflected light, a pre-generated file is here: `workshop_files/example_of_reflected_light_hist.root`
- Use that file to make a plot of wavelengths from the VUV and reflected light.

Task 3

- Go back to the muons file (again).
- We've launched them at -300 cm in X.
- This is pretty close to the anode planes.
- What happens to this plot if we move the muons back to -150? -20? +150?
- Make an overlay plot.



Conclusions

- If you got this far – congratulations, you are now able to run simple light simulation jobs and understand what is happening in them.
- There is of course a lot more that can be done with light, but that needs us to start looking at reconstruction of events, which will be next.
- One thing I did not cover was the construction of the optical library. This is a bit more complicated, but a reasonable tutorial can be found here: https://cdcvs.fnal.gov/redmine/projects/dunetpc/wiki/How_to_make_a_photon_library

Backup/Tips

Making some plots

- Visual way:
  - `root -l <my_file>_hist.root`
  - `new TBrowser()`
  - Find the name of your .root file in the list
  - Select pmtresponse, select DetectedPhotons, right click and select treeviewer.
  - You can plot any of the branches and apply cuts.

Making some plots (2)

- The script way.
  - Create a new file called myScript.C
  - In it:

```
void myScript()
{
TFile * fin = new
TFile("<myfile>_hist.root","READ");
TTree * mytree = (TTree *)fin-
>Get("pmtresponse/DetectedPhotons");
mytree->Draw("Time","");
}
```

  - Run: `root -l myScript.C`