

# Simulation Guide

Sarah Choate

## 1 Introduction

This is a guide for running simulations on both stand alone Genie as well as a full detector simulation using art/larsoft. The main focus of this project was to generate  $\nu_\tau$  simulations and compare with  $\nu_\mu$  and  $\nu_e$  simulations. Therefore the main focus of this document will be generating  $\nu_\tau$  simulations with brief asides on how to generalize for  $\nu_e$  or  $\nu_\mu$  simulations. The goal for both stand alone Genie and the full detector simulation is to produce exclusively charge current (CC) events. Due to the nature of the output ROOT files produced from the full detector simulation and which particles are saved, it is difficult to look at the particles produced and know if only CC events were produced. It is easier to tell what type of event was produced with stand alone Genie. For CC events, every event should have a charged lepton based on which flavor neutrino interacted ( $\tau^-$ ,  $e^-$  or  $\mu^-$ ) and its decay products, which can be found with their probabilities through the Particle Data Group (PDG). Throughout this document it will be assumed that we are only discussing the generation of CC events and that any commands discussed result in the production of only CC events.

The initial environment setup is the same across both methods. Begin by logging onto Fermilab. Setup the environment using the command

```
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
```

then setup the most recent version of dunesw. The most recent version can be found using the command

```
ups list -aK+ dunesw
```

and then most recent version can be setup using the command

```
setup dunesw $VERSION -q $QUALIFIER
```

be sure the qualifier is “*e20:prof*”. Additional references can be found in the readme of the github.

## 2 Stand Alone Genie

This process will include the initial neutrino interaction with the argon nucleus such that a charged lepton is produced, for  $\nu_\tau$  interactions, a charged  $\tau$  is produced, that is then decayed through one decay chain according to the PDG.

### 2.1 Command for Running

In general, the command for running Genie is

```
gevgen -n {number of events} -p {pdg of interacting particle} -t {pdg for target nucleus} -e {neutrino energy}  
-f {flux file} --cross-sections {cross section file} --event-generator-list "{type of events to generate}"
```

For  $\nu_\tau$  the pdg is 16 and for the argon nucleus the pdg is 1000180400. We set the energy for the neutrino consistently as 0,100 and the type of events we want to generated are CC events. Then, specifically for the  $\nu_\tau$ , the command is

```
gevgen -n {number of events} -p 16 -t 1000180400 -e 0,100 -f /path/to/file/flux_dune_neutrino_FD.  
root,nutau_fluxosc --cross-sections /path/to/file/gxspl-FNALsmall.xml --event-generator-list "CC"
```

Here, flux\_dune\_neutrino.FD.root is the root file which contains various flux data, nutau\_fluxosc is the specific file contained in the root file used for this  $\nu_\tau$  simulation, but it also contains information for both  $\nu_e$  and  $\nu_\mu$  fluxes. gxspl-FNALsmall.xml contains cross section information and can be used for  $\nu_e$  or  $\nu_\mu$  as is. Neither of these files are required to run stand alone Genie, as if they are not included Genie will simply calculate the values as part of the simulation, but it will make the simulation take longer to run.

### 2.2 Generating a Usable ROOT File

The ROOT file that is initially generated does not necessarily include all the information required for analysis of the simulation, it should be called something like *gntp.0.gep.root*. The type of file that is more likely to be useful is a roottracker format. The way to convert the file is through the command

```
gntpc -i {input filename} -f roottracker
```

This will produce a file which is called something like *gntp.0.gtrac.root* which is more appropriate for analysis we wanted to do.

### 2.3 Important Note

An important note about the resulting decay products. In a  $\tau^-$  decay, some of the expected decay products include charged and neutral pions. When calculating branching ratios, we found that photons usually showed

up with the expected decay products and that the branching ratio of channels with  $\pi^0$  was much lower than expected. This is because Genie counts  $\pi^0$  as two photons (so 2  $\pi^0$  would be four photons). So, when calculating branching ratios from Genie, do not try to count a  $\pi^0$ , count photons instead. An example of this is the decay channel  $\tau^- \rightarrow \nu_\tau + \pi^- + 2\pi^0$  would show up in Genie as  $\tau^- \rightarrow \nu_\tau + \pi^- + 4\gamma$ .

## 2.4 Running for $\nu_\mu$ or $\nu_e$

Looking at the general command, for  $\nu_\mu$  the pdg is 14 and for  $\nu_e$  the pdg is 11. The same flux file can be used but rather than calling `nutau_fluxosc`, for  $\nu_\mu$  it would be `numu_fluxosc` and for  $\nu_e$  it would be `nue_fluxosc`. The cross section file remains the same.

# 3 Full Detector Simulation

In order to produce a full simulation, there are four steps, only three of which were used for this analysis. First, the initial interactions are created using Genie, then the resulting particles are propagated through Geant4 allowing for more interactions, and finally the detector geometry and effects are added with Detsim. The fourth step that was not utilized here is the reconstruction step which allows for visualization of interactions as well as includes information about jets. These steps are all accomplished using various fcl files and the `lar` command. Full help and additional features not discussed here for the `lar` command can be found using `lar -h` in terminal.

## 3.1 FCL Files

This stands for Fermilab Hierarchical Configuration Language (FHiCL). These are files which contain information like particle types and detector geometry. They can be edited in order to utilize specific types of geometry or specific interaction types. In order to check what parameters are being applied to the simulation, the command `fhi-cl-dump name.fcl` can be used. This will print out all the information used in the fcl file without having to go through every included fcl file. For this project, it was necessary to edit fcl files as, by default, they produce both NC and CC events. This was problematic since CC events are much less likely than NC events, so the amount of events that would have needed to be generated in order to get a reasonable amount of CC events was too much.

### 3.2 Genie Step

Running the Genie simulation is conceptually the same as running stand alone Genie except here we use larsoft rather than gevgen. In order to run Genie, a fcl file is required. Again, this can be a default fcl file or one that has been edited. For this project, we used edited fcl files which are included in this repository in the folder “fcl\_edits”. The command to run Genie with art in general is

```
lar -n {number of events to generate} -c name-of-fcl-file.fcl
```

This will generate a file with the naming scheme *name-of-fcl-file-used\_gen.root*.

### 3.3 Geant Step

Once the initial events and interactions have been created with Genie, they are propagated using Geant4.

An additional fcl file is required to run this step called *standard\_g4\_dune10kt.fcl* which will not be edited.

The command to run is

```
lar -n {number of events} -c standard_g4_dune10kt.fcl name-of-genie-file_gen.root
```

which will produce a file with the naming scheme *name-of-fcl-file\_gen\_g4.root*.

### 3.4 Detsim Setp

Finally, the detector effects are added through the Detsim step. Again, an additional fcl file is required to run this step called *standard\_detsim\_dune10kt.fcl* which will not be edited. The command to run is

```
lar -n {number of events} -c standard_detsim_dunt10kt.fcl name-of-g4-file_gen_g4.root
```

which will produce a file with the naming scheme *name-of-fcl-file\_gen\_g4\_detsim.root*. This is the file which is used for analysis.