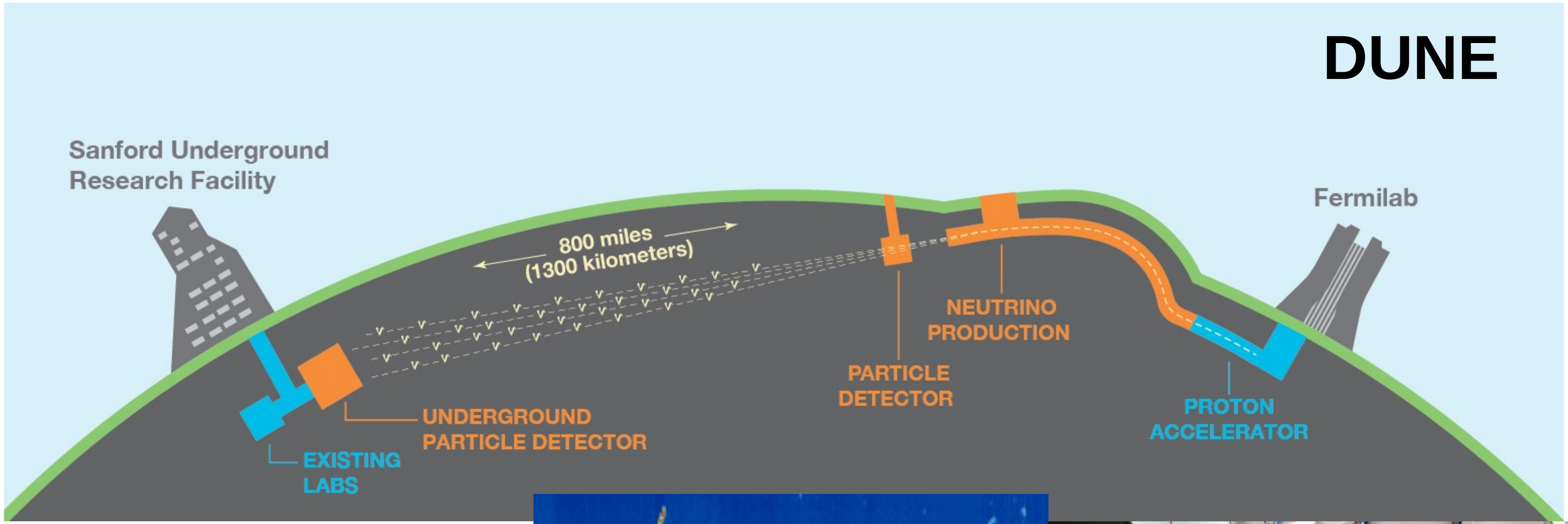
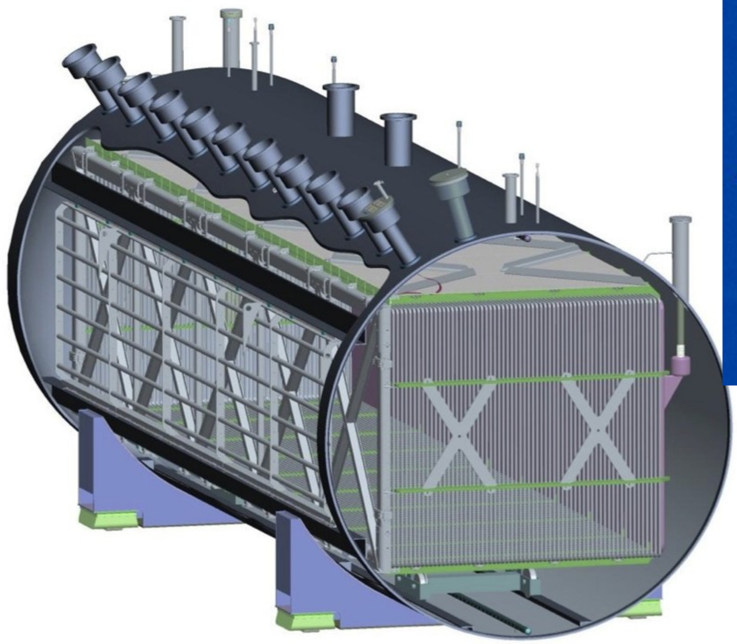
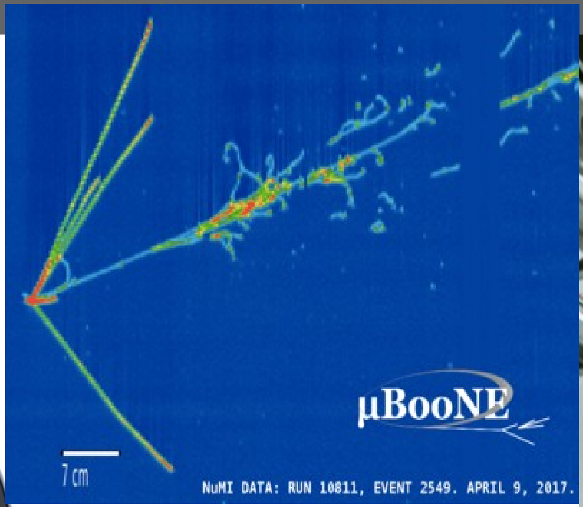
A Crash Course in LArSoft

Andrzej Szelc

LArTPC detectors for Neutrinos


DUNE
Sanford Underground Research Facility
Fermilab
800 miles (1300 kilometers)
NEUTRINO PRODUCTION
PARTICLE DETECTOR
PROTON ACCELERATOR
UNDERGROUND PARTICLE DETECTOR
EXISTING LABS



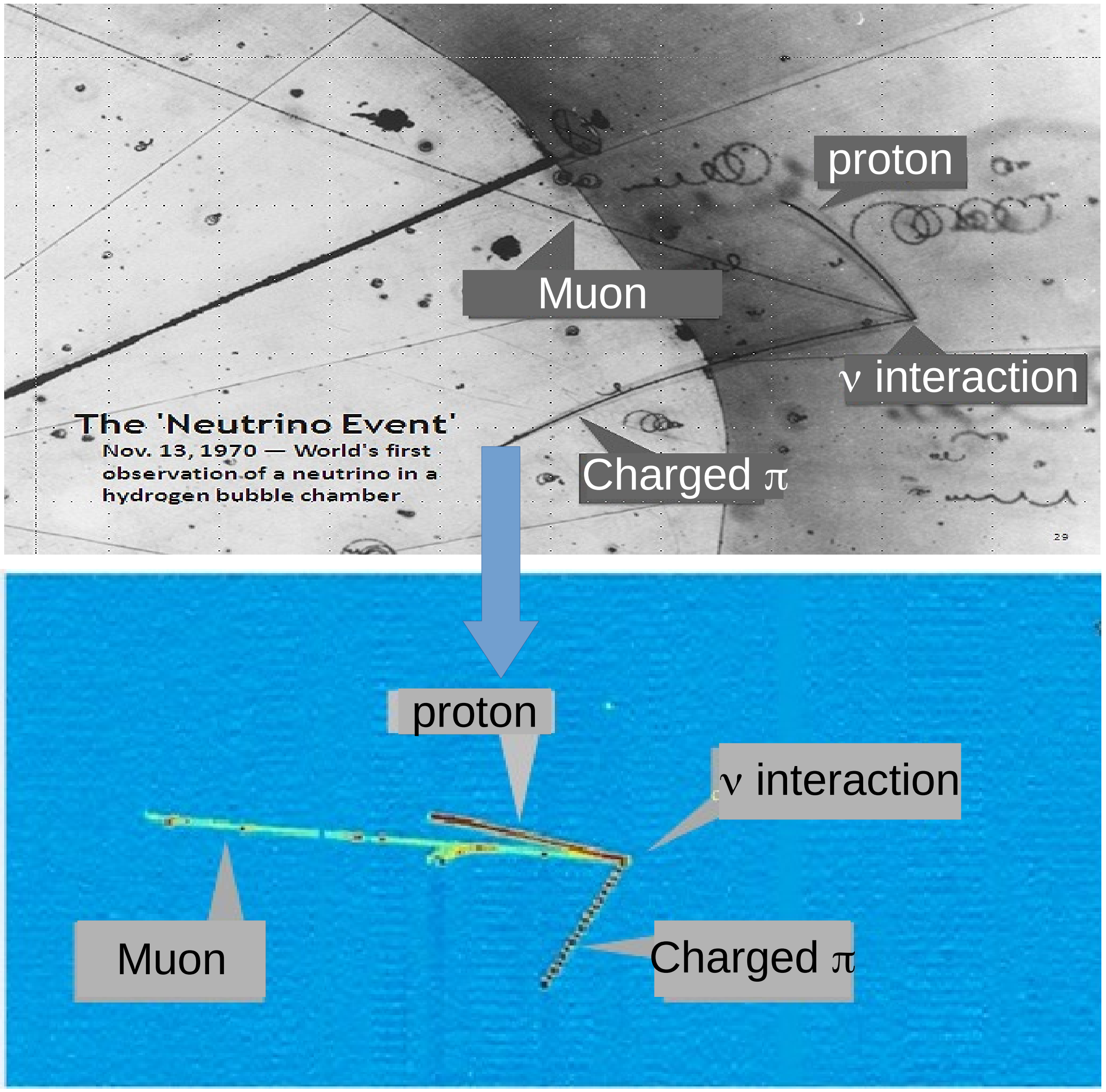


And a few others.

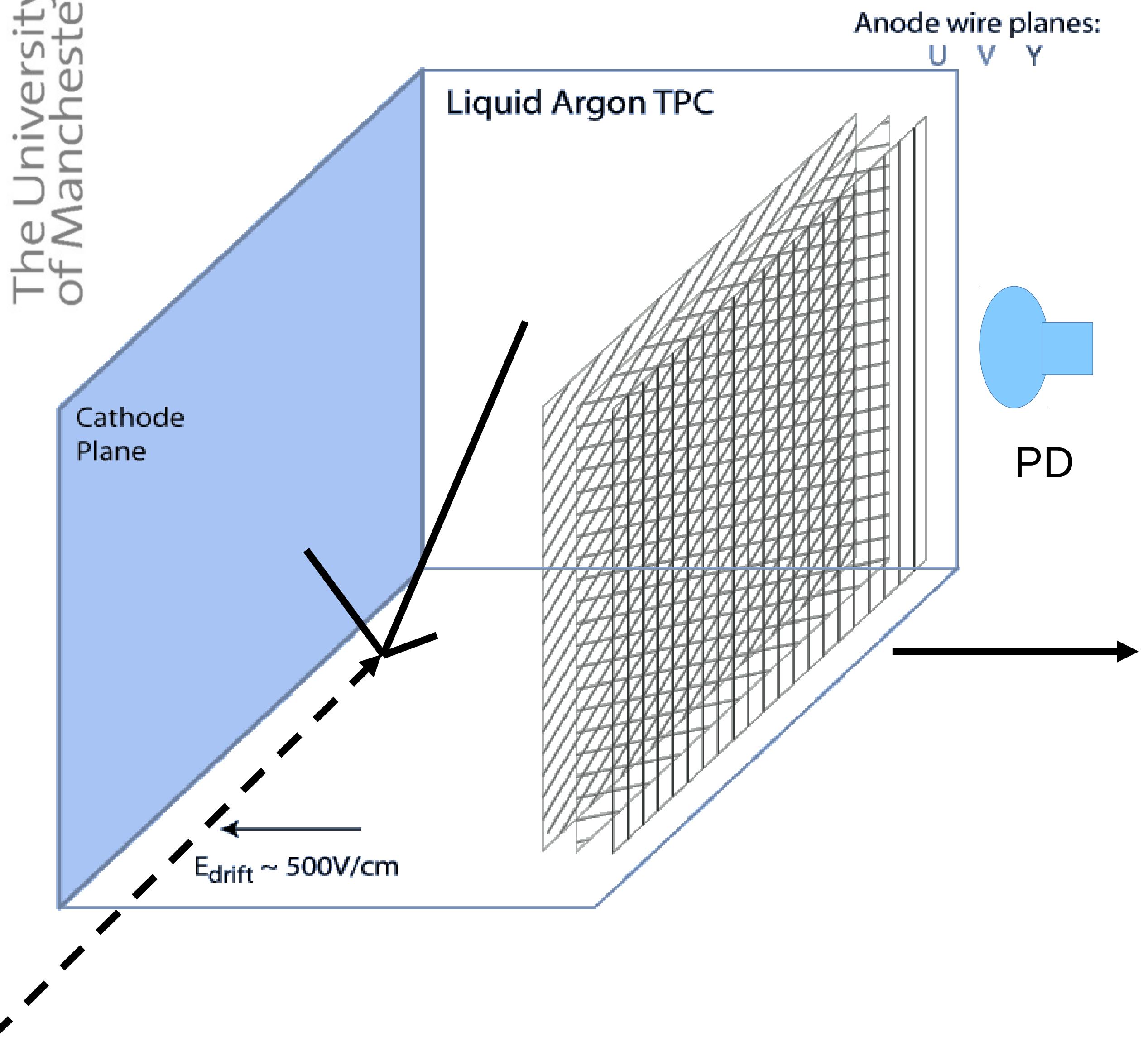
protoDUNE

Why Liquid Argon?

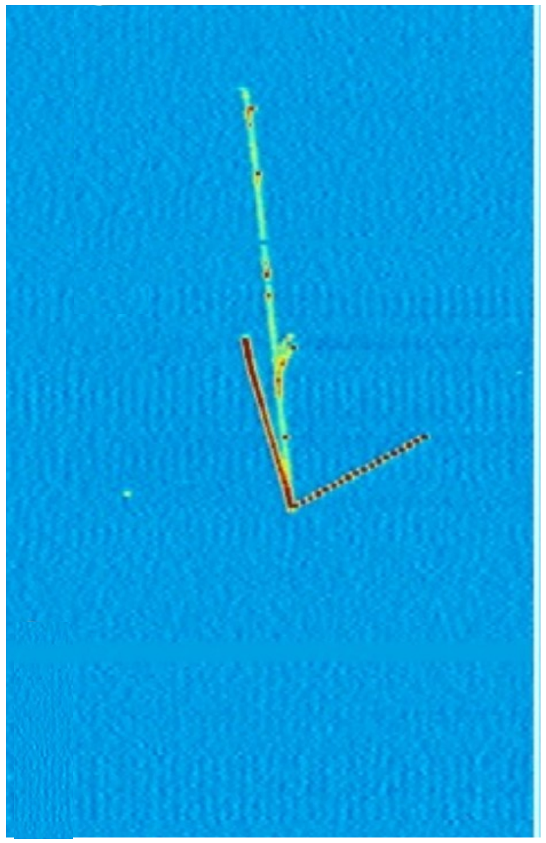
Bubble chamber quality of data with added full calorimetry.


proton
Muon
ν interaction
Charged π
The 'Neutrino Event'
Nov. 13, 1970 — World's first observation of a neutrino in a hydrogen bubble chamber
proton
ν interaction
Muon
Charged π

LArTPCs

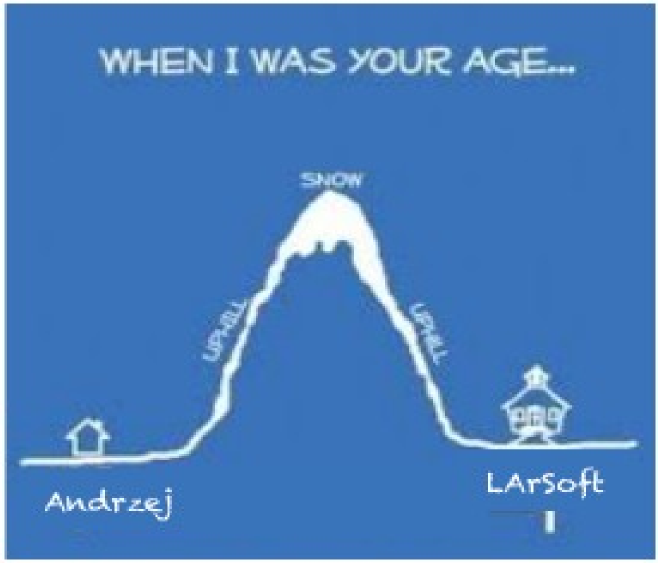


Anode wire planes:
U V Y
Liquid Argon TPC
Cathode Plane
PD
E_{drift} ~ 500V/cm

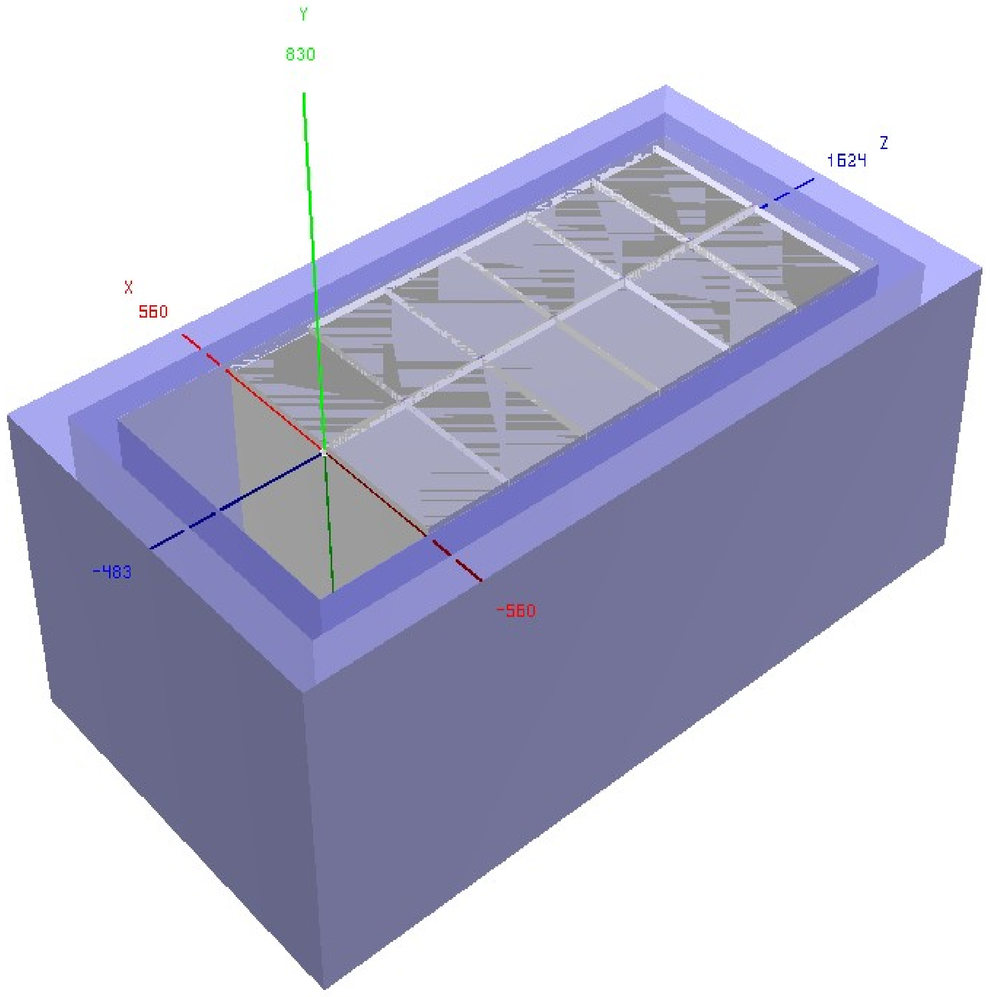


What is LArSoft

- LArSoft is a software framework that is used by several experiments. This is can make your life both easy and difficult at the same time.
- To run light simulation jobs in LArSoft, you have to have a rough idea of how run LArSoft (sorry).
- Getting LArSoft to run is relatively easy (back in my days...)
- That being said, it still has a learning curve. And it is steep.
- I will try to cover the bare minimum needed to get things working. I encourage you to take a look at e.g. the Paraguay workshop slides if you want to know (and do) more. http://indico.hep.manchester.ac.uk/conferenceOtherViews.py?view=standard&confId=5346



We will be (mostly) working with the 1x2x6 DUNE geometry



This is a reasonably small model of the DUNE full scale detector.

We can't actually run the simulation for the full detector (you'll understand later).

We have a large part of the framework set up for it, but it's not the fastest to run.

The University of Manchester

MANCHESTER 1824

Before we start: Where to Get Information?

- There will be links throughout the talk with specific documentation.
- First resource: http://larsoft.org/training Previous LArSoft talks and descriptions. Also, look here: https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki
- Most experiments have wiki pages on Redmine that are a good starting point:
  - https://cdcvs.fnal.gov/redmine/projects/uboonecode/wiki
  - https://cdcvs.fnal.gov/redmine/projects/sbndcode/wiki
  - https://cdcvs.fnal.gov/redmine/projects/lardbt/wiki
  - https://cdcvs.fnal.gov/redmine/projects/dunetpc/wiki
- These pages are editable once you have signed in to redmine and been added to the users group(s). These are your pages as well – if something doesn't work, tell the maintainers, or, even better: edit it yourself.

Before we start: Asking for Help

- Still stuck? Ask for help.
  - Mailing lists: larsoft@fnal.gov   SBND_SOFTWARE@fnal.gov microboone_analysis_tools@fnal.gov   lariatsoft@fnal.gov dune-reco@fnal.gov dune-proto-sp-dra@fnal.gov
  - Subscribing is easy - send an email to: listserv@fnal.gov

    leave subject empty, and in body write:

    subscribe <list> name lastname
- Good etiquette:
  - Make it easy for people to help you, e.g. create “Minimal, Complete, and Verifiable” example: http://stackoverflow.com/help/mcve .
  - This allows experts to reproduce your problem and find the fix quickly, as contrary to: “my code does not compile”. *What code? Where? What version?*
  - Often, you will find the solution yourself in the process. ;-)
  - If you haven't spent the time to understand your problem – why should the experts?

Before we start: Getting More Information

- Slack : http://slack.com/signin
- A combination of chat-group and blog. It's what all the cool kids use these days (apparently).
- Experiment – specific. Need to connect to the right one (and often need an invitation to join). Ones I know of are:
  - sbnd.slack.com
  - lariat-t1034.slack.com
  - dunescience.slack.com
  - shortbaseline.slack.com (SBN programme)
  - microboone.slack.com
- There may be others I don't know about.

Getting Information (almost abandonded version)

- LArForum: www.larforum.org
- This is an informal place to get help with larsoft software. Questions and answers are searchable or future reference.
- Connected between all experiments – large pool of experts to draw on.
- Not a bad place to connect amongst international groups and exchange experiences and ideas, especially if you're not on an experiment yet.
- Sign up and try it out: http://www.larforum.org/forum/viewforum.php?f=23
- Not as frequented as it used to be.

This talk

- I will give a really bare-bones, minimal, overview of how LArSoft works.
- I will **not** talk about things that I think are really useful to understand what's going on behind the scenes. Things like: UPS/UPD, Cmake, git, MRB. We will be using them, but we'll just hope nothing goes wrong.
- I **will** quickly talk about
  - What LArSoft is/does in general.
  - The ART event stack and producer/analyzer modules.
  - Configuring LArSoft using fhicl configuration files, and what you can do with them.
  - The LArSoft structure and the different repositories.
  - How to get LArSoft (CVMFS) repositories

What can LArSoft do (and how)?

- LArSoft is a software framework that allows running code (modules) responsible for **any** stage of liquid argon simulation or reconstruction.
- It is modular (hence modules) and can be configured at runtime (using FhiCL – more on that later).
  - You can run any and all modules in a single job. You can run the same module multiple times etc...
- It works with predefined objects that are stored on the “event stack” which is saved into a file.
  - You can run subsequent algorithms on these files as many times as you want
  - you can also split your jobs into stages.

LArSoft modules

- Three types of modules:
  - Producer: Can put new data products (objects) on the event.
  - Analyzer: Cannot put new data products (objects) on the event.
  - Filter: Decides whether to save or discard a given event.

- Key elements of a module:
- Filename: `LArG4_module.cc (all in one file)`
- `class LArG4 : public art::EDProducer { } //or EDAnalyzer or EDFilter`
- `DEFINE_ART_MODULE(LArG4)`
- `produces< std::vector<sim::SimPhotons> >(); //producer only`
- `evt.put(std::move(PhotonCol));   //producer only`

LArSoft/ART Rules
(stolen from Wes)

- **Thou shalt not modify data products (objects) “on” the event**
  - You may add things to the event, but once you have done so, they may not be changed
- **Thou shalt not have modules depend on other modules**
  - Developed a really cool function in your track-finder module you want to use somewhere else? Too bad.
- **Thou shalt only have modules interact with each other via the event**
  - Run through modules linearly, and always get previous results via the event
  - Though there is a truly evil way around that...

FHICL

- Fermilab Hierarchical Configuration Language, allows configuring jobs and modules on the fly.
- `fhicl::ParameterSet const& p` – this is how a C++ module can reference it.
- This is a parameter set:

  -
    ```
    standard_largeant:
    {
     module_type:               "LArG4"
     GeantCommandFile:          "LArG4.mac"
     DumpParticleList:          false
     DumpLArVoxelList:          false
     DebugVoxelAccumulation:    0
     VisualizeEvents:           false
     SmartStacking:             0 # non-0 turns it on. The 0x4 bit
                               # will shut off primary showering.
    }
    ```
  - This is an assignment of a parameter set:
    `dunefd_largeant:   @local::standard_largeant`
  - This now works:
    `dunefd_largeant.DumpLArVoxelList: true`

The University of Manchester

FHICL job file

```
#include "services_dune.fcl"
#include "singles_dune.fcl"

process_name: SinglesGen

// … //

physics:
{
 producers:
 {
   generator: @local::dune_singlep
   rns:       { module_type: "RandomNumberSaver" }
 }

 #define the producer and filter modules for this path, order matters,
 #filters reject all following items.  see lines starting physics.producers below
 simulate: [ rns, generator ]

 #define the output stream, there could be more than one if using filters
 stream1:  [ out1 ]

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

 #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:     [stream1]
}
```

Your awesome module definition

Your awesome module declaration

Make sure it runs

The University of Manchester

MANCHESTER 1824

Using .fcl files different use cases

- Generating a new ART/root file:

```
– Source:
  {
    module_type: EmptyEvent
    timestampPlugin: { plugin_type: "GeneratedEventTimestamp" }
    maxEvents:   1000000
    firstRun:    1           # Run number to use for this file
    firstEvent:  1           # number of first event in the file
  }
```

- Running a subsequent step on an existing .root file.

```
– Source:
  {
    module_type: RootInput
    maxEvents: 30000
    fileNames: ["gen_protoDUNE.root"]
  }
```

- Running multiple versions of a module.

```
– Producers:
   {
     largeant:  @local::dunefd_largeant
     pippo:     @local::dunefd_largeant
     rns:       { module_type: "RandomNumberSaver" }
   }
```

This is totally legal.
LArSoft will differentiat the modules by their labels.
You should probably change at least some parameters.

LArSoft – where can I find it?

- LArSoft by itself is detector agnostic (in principle)
- To get experiment specific code you need the detector specific repository (in our case it will be dunetpc).
  - https://cdcvs.fnal.gov/redmine/projects/dunetpc/repository/
  - https://cdcvs.fnal.gov/redmine/projects/uboonecode/repository/
  - https://cdcvs.fnal.gov/redmine/projects/sbndcode/repository/
  - https://cdcvs.fnal.gov/redmine/projects/lardbt/repository/
- The whole of LArSoft turns out to be more complicated.

MANCHESTER
1824

The University of Manchester

The LArSoft repository structure

LArSoft itself currently resides in multiple repositories:

- larcore
- lardata -> larcore
- larevt -> lardata -> larcore
- larsim -> larevt -> lardata -> larcore
- larreco -> larsim -> larevt -> lardata -> larcore
- larana -> larreco -> larsim -> larevt -> lardata -> larcore
- larpandora -> larsim -> larevt -> lardata -> larcore
- lareventdisplay -> larsim -> larevt -> lardata -> larcore
- larexamples -> larsim -> larevt -> lardata -> larcore
- lardataalg -> lardataobj -> larcorealg -> larcoreobj -> canvas

https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/_LArSoft_repositories_packages_and_dependencies_

A tool called MRB uses git and UPS to keep track of the dependencies between these repositories and makez sure you're good to go.

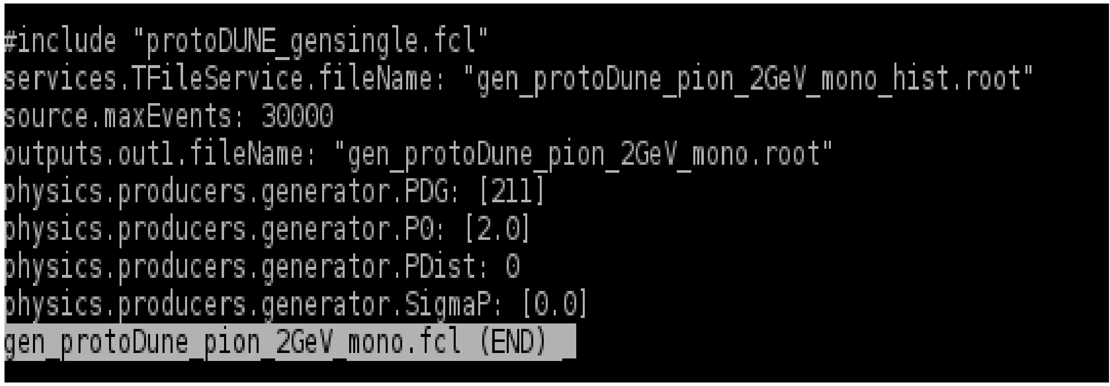
Often it will tell you that you're not.

How to find where something is?

- Configuration files (will be addressed later)
- LArSoft modules: include files will list a repository name, e.g.: `larsim/LArG4_module.cc`
- Don't need to download the code, can use environmental variables to look at code, e.g.
  - `ls $LARSIM_DIR/source/larsim/`
    Will have the code you need.
  - `env | grep LAR` gives you a list of suggestions (compare to previous page).

Dealing with .fcl files in multirep world

- More often than not, .fcl files will look like this:
- The "meat" is another file, and this one only defines a subset of parameters.
- How to find out what's going on?



```
#include "protoDUNE_gensingle.fcl"
services.TFileService.fileName: "gen_protoDune_pion_2GeV_mono_hist.root"
source.maxEvents: 30000
outputs.out1.fileName: "gen_protoDune_pion_2GeV_mono.root"
physics.producers.generator.PDG: [211]
physics.producers.generator.P0: [2.0]
physics.producers.generator.PDist: 0
physics.producers.generator.SigmaP: [0.0]
gen_protoDune_pion_2GeV_mono.fcl (END)
```

Tools exist that help you understand what's going on:

fhicl-expand prints out all of the included parameters.

find-fhicl tells you where a given .fcl file lives. (uB specific, but I'll show you how to get it in the tutorial.)

MANCHESTER 1824

The University of Manchester

How can you run LArSoft?

- cvmfs (CERN virtual machine file system)
  (next best thing after sliced bread)
- These are dynamically-loaded network drives that you can install on your server, desktop whatever.
  http://indico.hep.manchester.ac.uk/getFile.py/access?sessionId=26&resId=0&materialId=0&confId=5346
- They contain all of the needed dependencies to run LArSoft – you can use cvmfs to run larsoft on an almost new computer.

Summary

- LArSoft – not as scary as it looks but it does have a learning curve.
- Unfortunately we need to be able to run basic things before we continue with light simulation.
- In the quick tutorial that follows, we will set up a working directory, and launch some quick events.

Backup

LarSoft module key functions:

- `// read/write access to event`
- `void beginJob();`
- `void endJob();`
- `void reconfigure(fhicl::ParameterSet const& p);`
- `void beginRun(art::Run& run);`
- `void EndRun();`
- `void produce (art::Event& evt);  // or analyze, or filter`

ART

- ART is the framework that LArSoft runs on.
- art is an event-processing framework
  - Simulate, reconstruct, and analyze data on an event-by-event basis
  - The “Event” is an elementary art concept: the basic unit of data to be processed by art
- Why do you need a framework?
- https://web.fnal.gov/project/ArtDoc/Pages/workbook.aspx
  - more documentation. Art Workbook recommended.

Note on disk-spaces on FNAL machines

- FNAL machines have three types of storage:
  - “Blue-arc”. This is the standard kind of disk that you know – it is quick and nice and there is never enough of it. They are not visible from the grid.
    - /experiment/app/   this is where your working directories should be
    - /experiment/data/   this is where your local data files should end up
  - dCache – distributed disk drives. Slightly slower, but orders of magnitude more space available. They are visible from the grid.
    - /pnfs/experiment/scratch   lost of space, but gets deleted occasionaly. For temporary use – quick test etc....
    - /pnfs/experiment/persistent   less space, but stays until deleted. Note that mv from scratch to persistent doesn't work.
  - Tape storage – out of scope of this talk.

srcs/<repository>/ups/product_deps

```
# This @product_deps@ file defines dependencies for this package.
# The *parent* line must the first non-commented line and defines this product and version
# The version must be of the form vxx_yy_zz (e.g. v01_02_03)
parent sbndcode v06_55_00
defaultqual e14
# //...//
#
fcldir  product_dir job
gdmldir product_dir gdml
# table fragment to set FW_SEARCH_PATH needed
# to find gdml files:
table_fragment_begin
    pathPrepend(FW_SEARCH_PATH, ${UBOONECODE_DIR}/gdml)
    pathPrepend(FHICL_FILE_PATH, .:./job)
table_fragment_end
# With "product  version" table below, we now define depdendencies
# Add the dependent product and version
product          version
larsoft         v04_27_00
ubutil          v01_32_00
uboone_data     v01_01_00
cetbuildtools  v4_14_02 - only_for_build
• end_product_list
```

This is where you fix the problems

These are the dependencies

MRB
- basic trouble shooting (2)

- `mrbsetenv` //check whether all dependencies are ok
- `mrb i -j4` // compile and install
- `mrbslp` //set up your localProducts directory

This is where trouble happens.

- More often than not, you are either missing a product version that is a dependency – check whether it is available:
  - ups list -aK+ <productname>
  - echo $PRODUCTS
  - ls <product directories>   – make sure it is there.
- Or you have set up a version of a dependency that clashes with one that you need for one of your other packages:
  - ups active <productname>
  - ls srcs/<repository>/ups/product_deps - do they clash?
  - Try unsetup <productname>

MRB
- basic trouble shooting

- `mrb newDev`
- `source localProducts.../setup`
- `cd $MRB_SOURCEDIR`
- `mrb g -t <right version> uboonecode` //git clone repository
- `cd $MRB_BUILDDIR`
- `mrbsetenv` //check whether all dependencies are ok
- `mrb i -j4` // compile and install
- `mrbslp` //set up your localProducts directory

This is where trouble happens.

Put all of this together: (hands on tutorial - next)

- Set up LArSoft
- Setup mrb
- Get a repository
- Build
- Run a larsoft job using .fcl file:

```
lar -c
gen_protoDune_pion_2GeV_mono.fcl -n 10
```