The Liquid Argon Offline Software Package: LArSoft

Eric Church

Yale University, PO Box 500, MS309, Fermi National Accelerator Lab, Batavia, IL, USA, 60510-5011

Brian Rebel

PO Box XYZ, MS309, Fermi National Accelerator Lab, Batavia, IL, USA, 60510-5011

Bonnie Fleming

Yale University, PO Box XYZ, Physics Department, Yale University, New Haven, CT, USA, 12345-1234

Abstract

The software package for the simulation, reconstruction, and analysis of Liquid Argon TPC (LArTPC) experiments at Fermilab is called LArSoft. It is a general purpose package currently in use by the ArgoNeuT, MicroBooNE and LBNE collaborations. Any LArTPC can make use of its algorithms as long as the particular experiment supplies a properly formatted description of its detector geometry and electronics response.

1. Introduction

The software package for the simulation, reconstruction, and analysis of the ArgoNeuT, MicroBooNE and proposed LBNE/LAr40 experiments is called LArSoft. Additionally, LAr1, which is a proposed 1 kiloton LAr40 precursor, also uses LArSoft for its design studies. LArSoft is a general purpose package for LArTPC experiments. Any Liquid Argon Time Projection Chamber (LArTPC) can make use of its algorithms as long as the particular experiment supplies a properly formatted description of its detector geometry and electronics.

A liquid argon detector and a sophisticated software toolkit would be enormously powerful in analyzing event topologies typical of neutrino interactions, but which in traditional legacy technologies are difficult to parse and make sense of, and for which complete reconstructions remain elusive. Figure 1 shows just such an event. The large water Cerenkov detectors, even with high photocathode coverage, only see the Cerenkov rings on their walls; meanwhile, liquid argon detectors see every nuance of the event.

Email addresses: `echurch@fnal.gov` (Eric Church), `brebel@fnal.gov` (Brian Rebel), `bonnie.fleming@yale.edu` (Bonnie Fleming)

In this paper we describe LArSoft's functionality, with a particular emphasis on the Simulation and Reconstruction techniques it encompasses. Analyses in which LArSoft has been used are also discussed.

2. Framework and Tools

2.1. ART and Externals

LArSoft is built on the Analysis and Reconstruction Toolkit (ART) framework designed and maintained by the Fermilab Computing Division (CD) for intensity frontier experiments. The CD group that produces ART is the Computing Enabling Technologies (CET) group. Currently, the Mu2e, NOvA, and LArSoft collaborations use this framework. Using ART means that support for I/O, job configuration, and data provenance are supplied by the CD, freeing LArSoft developers to focus on simulation, reconstruction, and analysis. The intensity frontier experiments are in the process of developing an MOU with CD to formalize support of the framework for the lifetime of each experiment using it. The compilable code used for LArSoft is written in C++. ART's job scripting language is the FNAL CD in-house language fhicl. The approximately twenty external packages needed for LArSoft (ROOT, Geant4, Boost, Python, LHAPDF, GENIE, etc.) are distributed by CD as UPS binaries. This distribution scheme integrates well with ART on Fermilab computers operating with the prescribed Fermi Scientific Linux operating system. Successful off-site implementations of LArSoft with other Linux flavors exist at institutions that have carefully replicated the Fermilab scheme.

2.2. Repository, Build System, Compute Farm

The underlying ART framework code is available and maintained in CD [?] repositories; the LArSoft simulation and reconstruction code exists in a svn [?] repository. The package allows for frequent and easy user updates of the code along with more carefully controlled changes to the framework. As with C++ and ART, post-doctoctoral researchers and students using svn acquire useful skills for modern era software tasks.

LArSoft uses the SLAC-originated build system SoftRelTools (SRT [?]). SRT is an easy-to-use package which, with the user conforming to a few simple rules, builds private dynamic libraries for the user's desired packages against the full public build release. This feature in which the analyzer's private build areas run seamlessly with the public release is the most attractive feature of SRT. The public release is pulled from the head release in the repository, via a cron job, and built every night.

LArSoft uses redmine [?] for its project management. Redmine holds LArSoft's wiki for user-contributed reports and documentation, its Document repository for technotes and presentations and meeting minutes, its code review systems, and most importantly its code repository. Redmine is a neatly organized central project clearinghouse for users and administrators, and it is used to keep track of all facets of LArSoft.

A compute farm for parallel processing is available to LArSoft users, supported by FNAL CD. The condor [?] job submitter and node allocator is accessed by a well-defined prescription that is documented in detail on the LArSoft central redmine [?] website. Currently about 100 worker nodes in the FNAL intensity frontier cluster are available; a few thousand more nodes on the wider FNAL grid can also be accessed. This should provide ample computing as MicroBooNE and LAr40 enter a phase of high statistics Monte Carlo event generation and reconstruction. The Fermilab intensity frontier computing farm benefits from CD support. Collaboration users and developers from all three collaborations can be quickly added by system managers using simple scripts that currently give access to a relevant computing node.

3. Simulation

The LArSoft simulation interfaces with standard external packages. The GENIE [?] event generator models neutrino interactions, the CRY [?] package simulates cosmic ray interactions, and Geant4 [?] models the detector response. As the readout electronics differ for each experiment using LArSoft, each experiment must provide a detailed simulation of its electronics. This electronics simulation is currently well-modeled for ArgoNeuT, and is modeled for MicroBooNE with the circuit's LaPlace transform, and a good first approximation place-holder exists for LAr40.

LArSoft simulation jobs are modular in nature, with each module reading input data objects and writing output data objects to the event. A typical simulation job is shown in figure **??**.

The event generation, performed in the module known as LArG4, is in an advanced stage of development and is already used to perform TPC simulations for the three main LArSoft experiments. Of the three main experiments using LArSoft, LAr40's version of LArG4 uses a slightly stripped down geometry, with the detector's instrumentation still in development.

A working simulation chain for both the optical and TPC systems is in place and will be described later in this chapter. Further development and validation for both systems is ongoing.

3.1. Geometry

Geometries in LArSoft are currently defined in the Geometry Detector Markup Language (GDML) scripting language for the MicroBooNE, ArgoNeuT and proposed LAr40 liquid argon detectors. The GDML description is supplied via a text file and defines a nested hierarchy of volumes and descriptions of contributing materials. The MicroBooNE detector geometry is created via a series of scripts that define the size and shape of the world, cryostat, and TPC volumes and perform the placement of repeated elements such as TPC wires and PMT assemblies. Aspects of the event generation and detector simulation steps seek specifically named volumes and associate particular properties with them – these will be described in relevant sections that follow. LAr40's design calls for multiple TPCs, and so LArSoft generically allows an extra layer in the hierarchy to accommodate this. ArgoNeuT and MicroBooNE have precisely one module.

3.2. Nice Pictures of the detectors

Figure **??** shows the Geant4 implementation of the MicroBooNE detector geometry in LArSoft. The top picture shows the entire detector hall region surrounding the cryostat. The lower left picture shows the interior layout of the active detectors (TPC+PMTs) in the cryostat, and the lower right picture shows details of the TPC.

3.3. Optical Simulation

The Photon Propagation module for MicroBooNE is a PMT-system specific module. ArgoNeuT has no optical system. LAr40's acrylic bar system is in the proposal stage and is not yet developed in LArSoft. The electronics simulations for the optical system are experiment specific and remain under development.

3.4. Event Generation

With the geometry defined, we next discuss the steps necessary for generating a sample of simulated events. An event generator forms the first step of the standard simulation chain, and produces a set of Monte Carlo truth particles, which are then propagated through the LArG4 detector simulation. The EventGenerator package in LArSoft contains interfaces to several event generation modules, some of which are internal to LArSoft, and others which are external packages.

3.4.1. GENIE

Beam neutrino interactions can be simulated using the GENIE package. The flux description is sampled from a specified flux file and interactions are generated in a volume of specified name from the GDML geometry description. The neutrino flavors, various beam parameters and target information can also be specified.

3.4.2. CRY

Cosmic ray interactions are simulated using the CRY package. The latitude and altitude of the detector are specified, and interactions due to cosmic rays above a certain energy threshold are produced over a specified volume which has its center at the origin of the TPC volume.

3.4.3. SingleParticle

LArSoft additionally incorporates a particle gun event generator called SingleParticle. This allows the user to specify a number of particles to generate by their PDG ID and desired kinematics. Kinematics can either be specified exactly or sampled from a Gaussian or uniform probability distribution.

3.4.4. Optical

For studies of the optical systems and construction of the fast optical simulation library, LArSoft incorporates a Light Source event generator. This event generator simulates an isotropic light source of a specified size, shape and intensity. The light source can be either static, or moved through the detector on an event by event basis. For a moving light source, positions and properties are defined either systematically by dividing a volume into rectangular voxels and stepping through them one by one, or via a set of light source descriptions supplied via a text file. The description of the light source is stored in the event for use in further analysis or library building.

3.5. Propagating Particles Through the Simulation

The truth particles generated in the event generator step are passed to a Geant4 based detector simulation called LArG4. The geometry GDML file is parsed to create a Geant4 detector description using the built-in GDML parser, which interfaces to LArSoft via the DetectorConstruction LArSoft class.

3.6. Physics Lists

The user can specify which physics processes to enable using LArG4's configurable physics list system, which allows the specification of enabled physics constructors on a job by job basis. At the most basic level, a physics constructor is comprised of a list of Geant4 physics processes and the particles to which these processes apply. Available physics constructors inherit from the ... LArG4 thus becomes extensible with new physics processes which can be enabled or disabled on a job by job basis, without requiring a rebuild as would be standard with a hard coded physics list. The default list of enabled processes in LArSoft is equivalent to those enabled by the QGSP_BERT Geant4 physics list.

3.7. Photon Propagation

MicroBooNE contains 30 PMTs with TPB coatings. Wavelength shifting in the TPB coating is included as a physics process, though is not required in the current sensitive detector scheme. A certain fraction of photons incident on the TPB are absorbed according to a measured TPB absorption spectrum. Then a Poisson-sampled number of photons are emitted with frequencies sampled from the TPB emission spectrum, with a mean of N times the number absorbed with N set to 1 for current studies.

Optical physics simulations in LArSoft can take one of two forms: full or fast simulation, the latter of which involves several simulation modules and so will be described separately in section . The full simulation involves stepping every optical photon individually through the detector volume. Typical photon yields for an event can be in the 10^{meh} range, hence this is a very computationally intensive procedure. The OpticalPhysics physics constructor has been specially adapted for LArSoft and includes scintillation and Cerenkov production, Rayleigh scattering, reflections at boundaries, absorption at boundaries and in the bulk, and wavelength shifting physics processes. The configurations of these are described below. Selected full optical simulation results can be found in section **??**.

Scintillation production is configured with a photon momentum spectrum of 9.7 ?? 1 eV and a yield of 24,000 photons per MeV of energy loss, incorporating both a fast and a slow component, which can be scaled by a quenching factor specified for each scintillating particle. These quenching factors, and the possibility of utilizing a more systematic specification of the quenching per particle, require further investigation. Cerenkov photons are produced with yields and energies corresponding to the standard Frank-Tamm spectrum of Cerenkov radiation. Rayleigh scattering and photon-absorption process are enabled, where the scattering length is specified at 90 cm for all wavelengths and the absorption length is set to 2000 m (approximately infinite) for all wavelengths, for the purposes of preliminary studies. The vast majority of photons produced are 128-nm scintillation photons, so we do not expect neglecting the wavelength dependence of these parameters to cause a significant problem. A simplified reflectivity model is used, whereby each type of boundary in the detector is supplied with a wavelength dependent total reflectivity and specular/diffuse reflection fraction. For preliminary studies only the steel/argon boundaries at the edge of the cryostat are reflective, with all other surfaces, including wires, field cage, etc, being opaque. The steel/argon boundaries have a total reflectance of 25

LAr40 proposes another light collection system, with its own wavelength shifting technology, and it won't be elaborated upon here.

3.8. Detector Instrumentation

LArSoft utilizes a TPC, of course, and then, as mentioned previously, also has the flexibility to collect light. We discuss the generic TPC here and then, MicroBooNE's optical system. LBNE LAr40's optical system remains at an early proposal stage, and we don't elaborate on it.

3.8.1. TPC

The data object calculated by LArG4 simulation which relates to the TPC detector is a three dimensional map of charge deposits from each track over the liquid argon TPC volume, called a LArVoxelList. The drift of this charge in the electric field towards the wire planes is then simulated while still in the LArG4 module. The map is generated by dividing the TPC volume into cube shaped regions (voxels), each of which is assigned a unique ID, and attaching a Geant4 sensitive detector to these regions. During the particle stepping simulation, the energy loss by any particle in such a region is recorded along with the relevant track ID and voxel ID. In order to improve simulation performance, the voxelization scheme is subsumed within a Geant4 parallel geometry. In this way, only particles which will produce scintillation deposits need to be re-stepped at the voxel boundaries, and other particles, for example, optical photons, can take much longer steps in the argon volume.

3.8.2. PMTs

Photomultiplier tube assemblies, as described in Chapter 9, are included in the MicroBooNE geometry. The LArG4 output object for both optical simulations is a PMTHitCollection, which is a collection of the 4-positions and 4-momenta of each photon that has stepped across the boundary of each PMT sensitive volume. The Geant4 sensitive detector that produces these collections is called a PMTSensitiveDetector, and is attached to any volume in the GDML job configuration. The geometry with the appropriate name, as supplied in the LArG4 PMTSensitiveDetector stops and kills the track of any optical photon stepping into the relevant volume and stores its kinematics in the relevant PMT hit, indexed by PMT ID. Optical systems also exist within a parallelized geometry so that the PMT assemblies are, by default, insensitive to other stepping particles.

There are two possible detection schemes which can be utilized. Either a sensitive detector can be attached to the PMT lens, and all wavelength shifting processes in the TPB plate accounted for within the simulation, or the sensitive detector can be attached to the TPB plate itself, and the efficiency of the overall assembly can be accounted for with multiplication by a global efficiency factor. Currently we utilize the second strategy, with the global assembly efficiency set to 0.03, as determined by our calculations. We plan to measure this efficiency for each assembly in a test stand before installation. The method whereby the sensitive detector is attached to the PMT lens itself has the advantage of accounting for the full radial dependence efficiency of detection for light striking the TPB coated plate, but the disadvantage that the absolute conversion efficiency and output spectrum of TPB as well as the precise TPB deposition over the area of the lens must be known.

3.9. Mechanics of the Simulation

3.9.1. The UserAction interfaces

Mechanics of stepping and interrupting the G4 tracking. **How much of this, or next subsubsection do we really want?**

3.9.2. Material Properties Handling

Material properties required by Geant4 such as Birks constant, liquid argon absorption spectra, reflection spectra at steel / argon interfaces, and several others, cannot be supplied within the GDML geometry specification. Hence there exists a separate material property loading service called the MaterialPropertyLoader. that reads parameters from a supplied configuration and loops through the parsed geometry after detector construction. When a volume of the relevant material is found, the set of properties read from the configuration for this material are attached. The configuration format for the parameters file is likely to change in future versions of LArSoft...

3.9.3. Electron Drift and TPC Electronics Simulation

Once the map of charge across the volume has been simulated using LArG4, the drift of this charge to the wire planes and the simulation of an expected wire response is performed. The electrons are then drifted, also in LArG4. The charge drift based on properties supplied by the LArParameters service, which supplies the global properties of the liquid argon relevant to TPC operation in LArSoft. The drift velocity is calculated from the argon temperature and applied electric field and the charge yield per MeV of energy deposit is reduced via recombination in accordance with Birks law. Diffusion parameters are supplied directly... **which are ...** The charge in each voxel is divided into several parts of equal size, referred to in the drift simulation as clusters(not to be confused with reconstruction clusters), which are each drifted individually in the simulation.

3.9.4. Signals on wires, deconvolution

Any cluster reaching the wire plane is used to produce a simulated Electron object which contains information about the nearest TPC wire, the arrival time, the amount of charge in the cluster, and the voxel and Monte Carlo track that originated the ionization electrons. Electron objects are then used to simulate a TPC wire response. An excellent working model exists incorporating wire geometry, pulse shaping, electronics noise and ADC sampling and scaling effects. We elaborate on the mechanics here.

Physically, the electron cluster, is convolved with the electric field response, which is a ramp (sawtooth) of about a μsec width as it approaches the induction (collection) plane. That signal is convolved with the particular electronics response of the wires. Those two convolutions obviously can be done in either order. That is, they commute. In LArSoft, therefore, we convolve the field response with the electronics response, and store that complex result in two histograms in a Root file. This is dubbed the response kernel. The second convolution, in fact, is done as a multiplication in Fourier space, per the identity that a convolution in time is equivalent to the inverse transform of the product of the two Fourier transforms. Details here must be carefully attended to. But the advantage is that one avoids the large, nested loops in a convolution and avails of Root's built-in, efficient Fast Fourier Transform package. At this stage a signal is formed. It only remains to digitize and convert to the ADC scale appropriate to the electronics. This vector, one per wire, is stored. It represents the end of the simulation. The first stage of the reconstruction, therefore, will be to read this up and go through the steps necessary to recover the electron cluster. We detail such steps elsewhere.

3.9.5. Fast Optical Simulation Modules

Since the stepping of every photon generated in an event is a computationally intensive process, an alternative, fast library sampling simulation is in development. Currently this exists in a prototype stage, the details of which are described below.

The optical fiducial volume is divided into optical voxels, and the scintillation light production intensity and time profile in each voxel is recorded in a PhotonVoxelList object described in Section . The sensitive detector configuration is somewhat different to that applied to the charge voxels, since the isotropic light production is a result of a very specific physics process. This means we can improve efficiency by not requiring stepping particles to know about the presence of optical voxels, but rather supply a modified scintillation process, FastOpScintillation process acts like the G4Scintillation process, but rather than producing OpticalPhotons as secondary particles, it fills the PhotonVoxelList with the map of scintillation photon production across the detector volume. The photon voxels are defined separately to charge voxels, since they will undergo a separate size optimization.

A separate module, called PhotonPropagation contains a set of PMTHitCollections giving the expected final 4-positions and 4-momenta of detected photons for a very intense light source at each voxel in the detector. By comparing the intensity of the scintillation light source present in each voxel in the event to the intensity of the light sources used to generate the library file, and then applying Poisson fluctuations, a number of photons to sample for each PMT in the geometry is calculated. A set of photons are sampled from the library file, their detection time is smeared by the time profile of scintillation in each voxel, and by combining the expected responses from the light in each voxel, a PMTHitCollection is generated representing the expected detector response for the event.

The photon library is both geometry and voxelization scheme specific, and changes to either require a full regeneration of the library, which is a one-time computationally intensive job. The LightSource event generator is used to supply the intense sources of 9.7 eV photons for library generation, and all optical photons are stepped using the full optical physics simulation. The PMTHitCollections produced by LArG4 for light sources placed in each voxel, as well as the light source intensity, are stored in the library file using a module called PhotonLibraryBuilder, which is a part of the Photon-Propagation package. The library building and sampling simulation chain are shown in **??**.

4. Reconstruction

LArSoft benefits from the experience of multiple experiments. Each LArTPC provides essentially the same basic information after accounting for small detector differences. Thus, algorithms developed for one experiment can be directly used by another, as long as the differences in geometry are properly accounted for in the algorithms. The reconstruction chain proceeds in the following steps

1. First, the raw energy depositions (digits) are calibrated into signals on wires.
2. Next, hits are formed from the regions containing wire signals that are above a tunable threshold.
3. Hits are then grouped into clusters.
4. Clusters are classified to be projections from either 3D tracks or showers. Tracks and showers inherit from the class prong, in the usual C++ sense.
5. Vertices are located using prongs that are shown to originate from a common point.
6. Finally, prongs and vertices are associated into events.

The LArSoft reconstruction chain is complete in that initial algorithms for each step currently exist. More effort will be required to demonstrate that every, or at least most of the desired, event topologies can be handled in an automated way in the reconstruction chain. Detailed studies of various event classes are in progress to understand the performance of each reconstruction algorithm.

The reconstruction algorithms in LArSoft have benefited from several advances in image analysis techniques developed over the past decade. For example, the algorithm used to cluster groups of energy depositions together is directly taken from the heavily-cited work of Ester, Kriegel, Sander, and Xu's [] Density Based Spatial Clustering of Applications with Noise (DBSCAN). The two-dimensional (2D) vertex, or more accurately line-endpoint, finding algorithms are based on a corner finding technique used for locating edges and corners in photographic images. Initial particle tracking is performed using Hough line finding techniques.

Tracks, showers, and vertices are three-dimensional (3D) objects. At step 4 above, one begins to need to use the information from the 2D wire plane projections to reconstruct the 3D object from which it originates. Currently the 2D reconstruction codes are robust; the 3D versions are less mature. The following sections summarize the current state of the most important components of the reconstruction code.

Figure **??** gives an indication of the current capabilities of LArSoft using real data from ArgoNeut. Figure 126 and Figure 127 show an example of a fully reconstructed simulated event using the MicroBooNE implementation of LArSoft.

4.1. Calibrating the signals

4.2. 2D modules: hits,clusters,endpoints,houghlines

Need a paragraph here on FFTHitFinder.

Clustering is performed according to the DBSCAN procedure. The basic DBSCAN procedure as implemented in LArSoft accepts three parameters a minimum cluster size MinPts and two characteristic clustering scales ϵ_1 and ϵ_2, which dicate .... The following procedure is performed iteratively. First we start from a hit that has not already been associated with a cluster or discarded as noise. The algorithm then finds all hits that exist within the boundary of an ellipse with axes..... with this hit at the center. All hits are then collected in this neighborhood into the cluster. Visiting each added hit, another ellipse is centered on this hit, and encompasses further hits within the cluster. When the neighborhood of every hit in the cluster has been checked, all hits that have been added are counted. If they number more than a threshold MinPts, then the collection of hits is stored as a cluster object, and the hits are removed from the clustering sample. If the threshold is not reached, the hit is discarded as noise and removed from the clustering sample. Once all hits have been either drawn into clusters or discarded as noise, the algorithm is complete.

(Figure 128 Un-clustered(left) and clustered(right) hits in the event display.)

Two dimensional tracking is performed using a Hough line finding method. The Hough space spans two dimension identified by ..... A point in the Hough space represents a straight line at angle ... from the origin of the real space. Hence a point in real space can be represented as a curve in the Hough space, which marks all straight lines which pass through the real point.

Figure 129 Schematic representation of the Hough line determination method.

The Hough line finder method takes all 2D hits that are contained in a cluster, each of which is a single point in a two dimensional real space, and transforms them into Hough curves. The map of (...) space is now comprised of a set of many curves, and at each point one can calculate a curve density. The point with the highest density of curves represents the straight line with the most good passes through all the hits in a cluster. In the interest of finding a single solution stable against random fluctuations, smoothing is applied to the density map using a Gaussian smoothing kernel. Only Hough lines above a critical hit density threshold are counted, and several lines can be generated per cluster up to a maximum line count. The resolution in each of the ... and dimensions for the algorithm can also be specified.

Vertexing and Vertex Matching. Defining the precise endpoints of tracks is not easy with a simple clustering algorithm like DBSCAN, or with a Hough line finding method. Hence LArSoft incorporates specialized endpoint finding algorithms. A Harris vertex finding algorithm is used to find vertices from the hits that form DBSCAN clusters. The algorithm begins with a two dimensional set of hits extracted from DBSCAN clusters. A Gaussian derivative is used to determine the gradient of the hit density at each pixel in the 2D view of hits. A Harris vertex is then defined as a region where there is a strongly falling gradient of hit density in two orthogonal directions (forming a “corner”). The strength of the vertex is calculated from the magnitude of the aforementioned gradients. With non-maximal suppression enabled, if several vertices are found within some user defined distance window, only the strongest vertex is kept and the others are discarded. This algorithm identifies a set of regions which are candidates for vertices independently of the line finding process.

A matching procedure is then applied, whereby Hough lines are matched to Harris vertices. If a vertex exists within a user defined window around the endpoint of a reconstructed Hough line, it becomes associated with that vertex. A Harris vertex which has two or more matching Hough lines is defined as a strong vertex, whereas one which is inconsistent with at least two Hough lines is defined as a weak vertex. The strength of a "strong" vertex is redefined as the original Harris vertex strength multiplied by the lengths of its associated Hough lines. The algorithm to find vertices defined by intersecting tracks or prongs that point back to a common 3D point to within some pointing error, has been implemented to first order, but refinement of this procedure remains an outstanding task. The current corner vertices serve as seeds for this search.

We will wish to be able to compartmentalize sub-events within a particular time window. For example, we will want to categorize a stopping cosmic muon track (with its Michel electron track and vertex) apart from the beam window-triggered neutrino event (with all its tracks/clusters/vertices, etc.).

4.3. 3D modules: spacepoints, showers, kalmantracks, vertices

Three dimensional tracking is presently under intense development. Here one looks to see whether track projections, or Hough lines, from the 2D clusters already defined in the individual TPC wire planes can be matched and interpreted as projections from a real 3D track in the TPC. Two 3D tracking algorithms exist in LArSoft.

A Kalman filter-based tracking package exists in LArSoft and is mature. ArgoNeuT's experience with it suggests it will be useful for defining straight as well as multiply scattered tracks, and we know from ICARUS results that we will benefit in measurement of momentum resolution. Further, pointing errors "come for free" with a Kalman filter, and the 3D Vertexing algorithm mentioned above will take these as its input. See **??** below for more discussion of Kalman tracking of muons.

A working shower finding algorithm is in place, but remains under development. The current implementation finds two dimensional showers in each view individually and projects in the standard way to 3D. The coordinates are given by equations **??**. We have successfully generalized the 2-plane algorithm in use by ArgoNeuT into the 3-plane algorithm required in MicroBooNE and LAr40. See 5.2.3 for analysis of showers in 3D.

Calorimetry follows by straightforwardly looping over prongs (tracks or showers) and summing up the properly-calibrated signals of each hit in each prong. Corrections are applied for the prong's angle with respect to the wire and for saturations in energy loss for particles which produce ionization electrons that drift long distances (Birk's law).

4.4. particle ID

Particle identification follows from the dE/dx determination resulting from this method. See Figure 4 for dE/dx-based particle ID.

In the uBooNE TDR there's a lot of chatter about would-be PMT particle ID from late-to-early light ratio. I'm inclined to think this is material not appropriate for this NIM.

4.5. Truth Level Reconstruction

Tools exist to understand the perfect way in which to gather the reconstruction objects. *Could put the π^0 cheater 3D and 2D event display pictures here.* This aids in understanding where algorithms need work.

4.6. Event Display

LArSoft contains an event display that is indispensible to understanding neutrino events in liquid argon. Reconstruction objects and raw information may be added and removed. Events can be specified for viewing with MC truth info overlaid, if applicable. Zooming features allow insight into the tiniest features around vertices.

The display has been tested with Monte Carlo and ArgoNeuT data, and the code has been written with ease of generalization to MicroBooNE and LAr40 in mind.

4.6.1. ArgoNeuT's plots/event-displays

Figure 5 shows an example event display for an ArgoNeuT data event.

4.6.2. MicroBooNE's EVD

MicroBooNE's 3-plane event display is shown in Figure 6. One notes the muon and other ...

5. Analysis

5.1. ArgoNeuT: Data

5.1.1. Lifetime Measurement

A novel technique to measure electron lifetime in liquid argon was used in Argoneut. Data was taken from a sample of through-going muons **????** and the signal height was plotted versus the measured drift time. See Figure 7. Then, in bins of drift time the signal sample was fit to a landau distribution. See Figure 8. The "most probable value" that results is plotted versus time once again and τ is extracted in the fit for $Q = Q_0 e^{\frac{-t}{\tau}}$. The lifetime τ is typically ≤ 1 msec for MicroBooNE.

5.1.2. μ CC Inclusive Analysis highlights

A full analysis which produced the first cross section measurement on Argon was recently published in [**?**]. This analysis used LArSoft to automatically select and reconstruct $\overline{\nu}_\mu$ Charged Current inclusive events. The differential cross-sections versus polar angle and Energy of the resulting muon are shown in Figure 9 averaged over the NuMI beam energy.

In short, the analysis proceeds per the chain in Figure 3. A through-going muon filter – "filters" being a generically applicable boolean operation used on input data streams – removes all events determined not to be ν_μ CC events originating in ArgoNeuT. LArSoft hit-finding, clustering, tracking, and vertexing, not to mention analysis modules for collecting the relevant diagnostics and results along the way, are all employed.

5.2. MicroBooNE: Simulation

5.2.1. Spacepoint Finding

MicroBooNE uses all the same reconstruction techniques as discussed in the previous ArgoNeuT subsection with one important enhancement. That is, MicroBooNE is a three-plane detector, where ArgoNeuT has only two drift planes. This implies a number of improvements and complications. It also highlights an over-arching principle of LArSoft, which is that all code should be written detector agnostically. Code that runs on 2-plane detectors must be written generally enough to work for 3-plane detectors.

Three planes means that the hits in one plane are to some degree redundant to those in the other two planes. This has implications for space point finding. Namely, one can imagine algorithms that require hit "triplets" to all have consistent timing information when tracks are consistent with the hypothesis that all the wires are crossed with a non-zero angle. When tracks on the other hand are close to parallel to the wires in one plane, the requirement on the existence of those hits can be relaxed or abandoned altogether. As yet, the currently employed algorithm treats each plane democratically, requiring a hit in each plane, though the width on some hits are allowed to be quite wide in time.

5.2.2. Kalman Filter-Based Tracking

We estimate that MicroBooNE will contain on the order of 10% of the muons which result from ν_μ Charged Current beam interactions. These muons are candidates for fitting by a Kalman filter.

Kalman Filter techniques to fit tracks to 3D points in liquid argon TPCs are discussed at [**?**]. The Kalman machinery is explained in [**?**]. LArSoft uses code from a package called Genfit [**?**], which is freely available at sourceforge.net [**?**]. That code was repurposed within the LArSoft framework to exist as its own module in the reconstruction chain. It takes Spacepoints as its input, with errors specified that are sub-millimeter in the drift direction, and are on the order of the wire spacing in the other two dimensions. Measured momentum errors are specified to be sub-50 MeV/c, though below a few hundred MeV/c and above the length of a fully contained muon in MicroBooNE this is a wildly optimistic estimate. The filter uses this information as a measure of the input noise in the system. The Kalman Genfit machinery proceeds to walk the track from measured spacepoint to spacepoint, building the covariance matrix as it goes that connects the Kalman "state" vector elements. The fitter does quite well with fully contained muons. The plot of the momentum resolution for simulated muons is shown at Figure [**?**].

Perhaps here we insert the plot that shows the Kalman momentum measurement vs the summed dE/dx for one plane, when I get to making it. Perhaps we should also, or instead of the idealized plot, show the results for the useMC=F switch, which as yet are decidedly gloomier.

5.2.3. Showers

One of MicroBooNE's main science goals is to learn if MiniBooNE's neutrino mode excess [**?**] below 500 MeV/c is a potential nuclear effect and not in fact from neutrino oscillations. That would mean MiniBooNE was seeing gammas, and not electrons. LArSoft possesses the tools to make this distinction. See Figure 11.

5.2.4. CRY

Dunno if Christina and Roxanne have results we can use here.

5.3. LBNE: LAr40 Analyses

While we've seen in Figure 2 that the LAr40 geometry is mature and we've asserted that the electronics exist at a reliable place-holder level, it remains also true that simulations in an instrumented detector are not yet robust.

Nevertheless, Monte Carlo truth studies may be performed. One of these studies that has been performed in the LArSoft machinery is a proton decay background study. We give a brief overview of the LAr40 study here.

5.3.1. LBNE pdk study

In proton decay studies in large liquid argon TPCs that are sited underground, not so deep as to completely eliminate cosmogenic background, we have very particular concerns. Namely, we worry about any background that mimics the so-called LAr TPC "golden mode:" $p \to K^+\overline{\nu}$. This reaction is $B - L$ conserving and is particularly germaine for Supersymmetry-inspired theories of proton decay [**?**]. The background we worry about is the one in which K^0_Ls produced by energetic cosmic muons in the rock then propagate into the liquid argon, slipping through any edge-detection requirement, and then charge exchange to the K^+, some of which may have a momentum close to the 342 MeV/c that is characteristic of the golden mode K^+ momentum.

We begin this study with muons that have been generated in a separate simulation and are propagated down to the cavern at the so-called "800L" site of DUSEL. LArSoft picks up the muons in a 300m by 300m square there, just over the detector. A variety of Geant4 settings are turned on in order that LArG4 can give reliable results. Namely, μ photoproduction by way of G4MuonInteraction is turned on in order to even be sure that we might generate K_L^0s. We turn on charge exchange for K^0s, as well. Then follow a variety of techniques to try to get around the rare statistics issues. Those techniques are the following. We bias-up the G4MuonInteraction cross section by a factor of five. We create 1500 times the K_L^0s and neutrons and Lambdas, as well, and weigh them accordingly by $\frac{1}{1500} \cdot \frac{1}{5}$, roughly, every time any one of these neutrals is created. Further, we ignore all muons which interact more than 5m into the rock from the liquid argon, as that represents about 7 K_L^0 interaction lengths in rock. We also ignore all would-be pernicious K_L^0s that are created in coincidence with the μ if that μ enters into the liquid argon. The notion is that these cosmogenic events will be easily veto-able with an efficient, in-time activity cut for tracks entering from the outside and pointing into the region of the suspected proton decay. Figure 12 shows the kinetic energy distribution of neutrons from muon spallation which make it into the liquid argon. We will not concern ourselves with neutrons below 20 MeV which will be benign with respect to creating proton decay backgrounds, thus we show the cut-off at that energy above which we may comparethe shape and integrated rate to other studies.

5.3.2. Jen's handscanning?

In a LAr40-like detector, in which one is trying to characterize signal efficiencies and pin down backgrounds, we can go a long way toward these goals just by viewing events. We organize "handscan" analyses and assign the hand scan analyzers ("handscanners") some number of simulated events, say, ν_μ Charged Current events and $\nu_m u$ Neutral Current events, whose misidentification as Charged Current events is a background the study wants to ascertain. The goal is to ask the handscanner to return his/her judgment of the event "flavor" in the absence of knowing the Monte Carlo Truth information. We may then fold in our understanding that a fuller analysis will bring tracking and calorimetry, etcetera, to bear on the events and thus have an even stronger ability to categorize and separate these events, beyond a mere eye for the proper topology. The handscanning contributes to an understanding of how to codify this process. We tabulate in **??** the results of a scan of 7 handscanners scanning various event flavors.

6. Summary

Here we summarize the paper.

References

[http://www-microboone.fnal.gov]

[https://www.jyu.fi/fysiikka/en/gla2011/Proceedings]

[Chapter 12 of MicroBooNE TDR.]

[1]

[M. Soderberg *et al.* [ArgoNeuT Collaboration] “Analysis of a Large Sample of Muons with the ArgoNeuT Detector,” in preparation.]argo-arxiv

Figure 1: In this Neutral Current ν_μ event a π^0 is produced along with a proton. The π^0 decays quickly to two photons in such a way that in the boosted frame (the lab frame) the γs have a small opening angle. (a) The software package WCSim [**?**] can only find one ring. (c) The Liquid argon detector, on the other hand, shows the two γs with a separable opening angle. This event display is for a MicroBooNE-like detector. We discuss the LArSoft Event Display in more detail in section 4.6, but, briefly, one sees the event here as it projects onto the collection plane and the induction planes, respectively, top to bottom. The signal on a wire is seen in the very bottom panel of (c).

(a)

../NNN2011/mtrl/imgs/WCSIM-ovrlap-rings.png

(b)

../NNN2011/mtrl/imgs/LAr-uB-ovrlap-rings.png

Figure 2: The LAr40 detector is shown here as built from the gdml scripting language in LArSoft. As specified in the LAr40 design [**?**] there are 2x3x18 Anode Plane Assemblies shown here in each of two 20 ktonne LAr volumes. Each cryostat is surrounded in rock whose composition is typical of that found at the DUSEL 800 feet level.

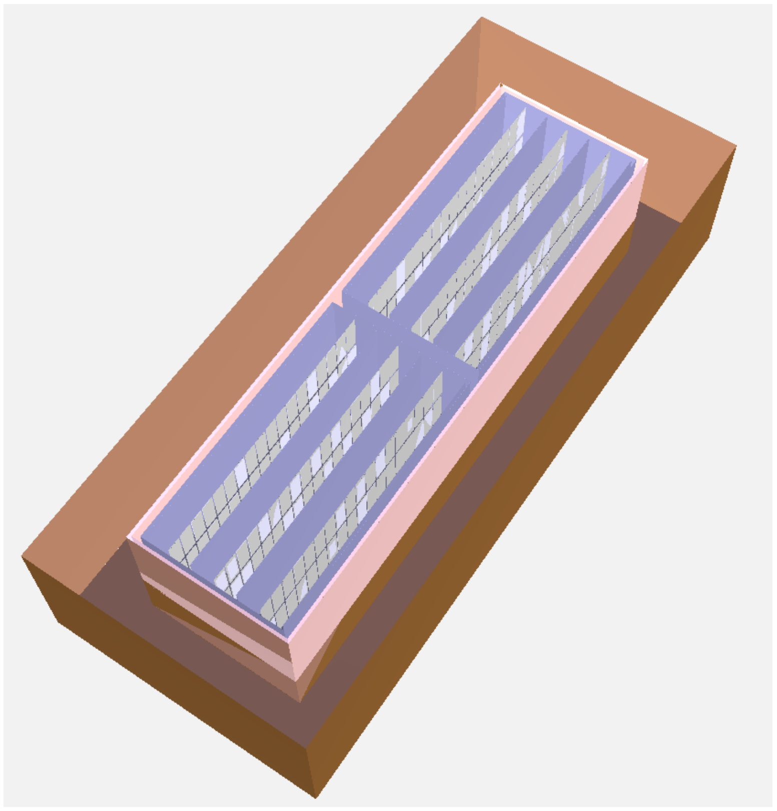

Figure 3: This flow chart shows the objects created along the reconstruction chain in LArSoft.

```
../GLA2011/proceedings/LArSoft-Recon-Flow-Soderberg.pdf
```

Figure 4: Particle ID in ArgoNeuT with LArSoft [**?**]. Top: ArgoNeuT Induction plane views of two events. Bottom: the dE/dx of the hits in chosen tracks in the two events. The likely muon track of the upper left event has the dE/dx of its hits (black points) calculated and superimposed over the MC dE/dx hypotheses of pions (fuchsia) and muons (green) in the lower left image. The likely proton track at the bottom of the upper right CCQE event image has the dE/dx of its hits superimposed that for MC hypotheses of proton (red), kaon (cyan), pions (fuchsia) and muon (green).

`../../uBooNE/LArSoft/pub/other/kal/argo-pID.png`

Figure 5: A data event in ArgoNeuT, viewed with the event display in LArSoft. Among other emitted particles, two π^0s are evident.

```
../NNN2011/mtrl/casestudy/larfigures/ArgoNeuT_event.jpg
```

Figure 6: A simulated CC event in MicroBooNE with LArSoft.**Replace with better event.**

../../uBooNE/LArSoft/pub/evd/uBooNE_CC_evd.png

Figure 7: ArgoNeuT data: signal height vs. drift time for muons.

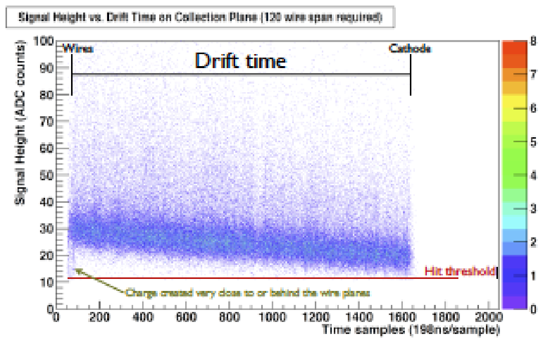

Figure 8: ArgoNeuT's mean signal height in bins of drift time, fit to a Landau distribution. The most probable value "MP" for each time bin is then plotted versus drift distance to extract the electron lifetime. This is done periodically during a run for calibration purposes.

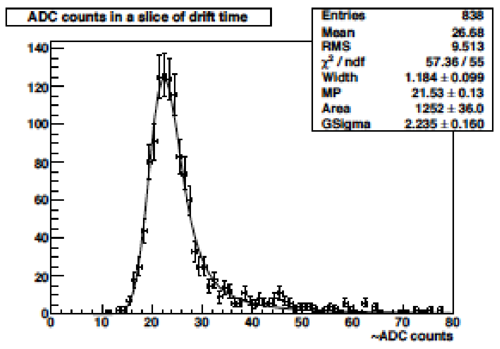

Figure 9: ArgoNeuT's measurement of ν_μ Charged Current inclusive differential cross-sections [**?**]. LArSoft was used to select and automatically reconsruct and analyze the events.

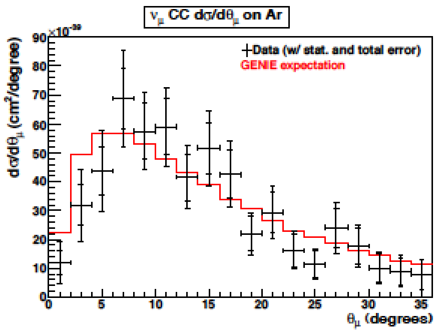

Figure 8.46: The CC ν_μ differential cross section in muon angle on an argon target. The differential cross sections are reported "per argon nucleus". Total and statistical-only error bars are shown.

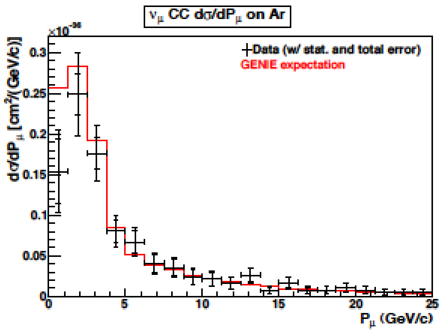

Figure 8.47: The CC ν_μ differential cross section in muon momentum on an argon target. The differential cross sections are reported "per argon nucleus". Total and statistical-only error bars are shown.

Figure 10: This is the Kalman filter's reconstructed momentum vs. true momentum for idealized space points in a MicroBooNE muon simulation. Muons were generated at five true momenta values.

`/Users/church/uBooNE/LArSoft/pub/t3dk/t3dkResn_lego.png`

Figure 11: This is the e/γ separation in the MicroBooNE simulation.

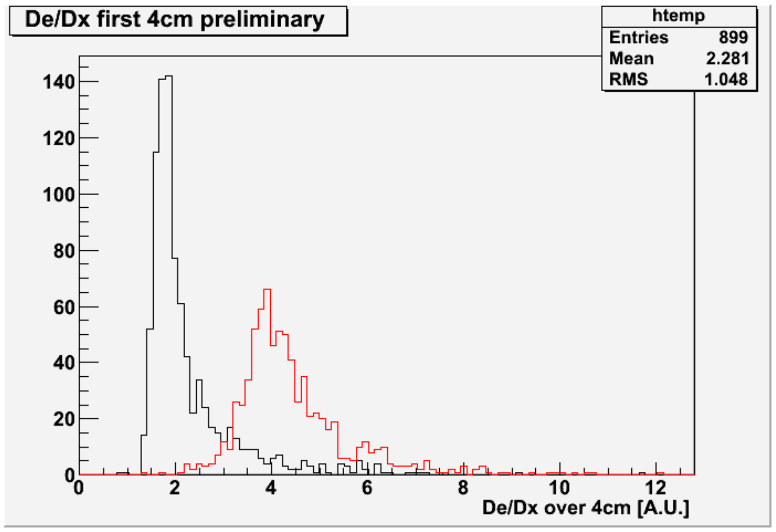

Figure 12: LBNE simulation. Kinetic Energy of neutrons which make it into TPC from primary cosmic muon photoproduction. Neutrons are produced at a far greater rate in muon spallation than the more serious K_L^0s and therefore afford an easier handle on normalising the proton decay backgrounds.

/Users/church/lbne/analysis/pub/nsInTPCFromWMuNuc.png