

ENERGY DEPOSITS IN SIMULATION

**Wesley Ketchum
FNAL**

MICROBOONE MOTIVATION

- **Trying to do detector variation simulations**
 - Detector variation systematics should ideally be determined by varying the same events, and seeing the effects of the desired variation
 - “Event” is not just neutrino interaction, but the simulation of the outputs of that interaction in LAr
 - “*GEANT4*” *part of our simulation*
 - Desired variations are **after** particle interactions
 - *Space charge variations*
 - *Diffusion*
 - *Purity*
 - *Light yield*
 - *Recombination model*

CURRENT PROBLEM

- **Current LArG4 combines particle interactions and signal propagation**
 - Step one: simulation interactions and determine energy deposited in a voxel
 - *This is the part we want to not vary*
 - Step two: determine number of electrons and photons, propagate them to wires/phototubes, and create SimChannel/SimPhotons objects
 - *This is the part that we want to vary!!!*

POSSIBLE SOLUTIONS

- Force the particle interaction bit to always be the same via random seeds
 - *Joseph tried this and had a hard time getting it to work*
 - *This is an avenue for exploration, for sure...but I worry it may be difficult if it's not obvious*
- Split the LArG4 module
 - *Store energy depositions in LAr in the first stage*
 - *Do propagation of electrons and photons in the second stage*
 - Vary all systematic effects here
 - *This is aligned with the LArG4 refactoring project*
- Pushing on second option, compatible with the v06_26_01 LArSoft branch (uboone MCC8)

THINKING AHEAD...

- **Requirements for detector variations**
 - Ability to recompute number of electrons and number of photons produced per voxel
 - *Recombination, spacecharge, photon yield model, etc.*
 - *IonisationScintillation algs should probably be art tool*
 - Ability to redo electron drifting
 - *Diffusion, purity, spacecharge changes*
 - Ability to redo photon propagation
 - *Prompt/late ratio, w/wo cherenkov light, different lookup libraries, etc.*

THINKING AHEAD FURTHER

- **Additional possible use cases**

- Ability to move energy depositions in space and time
 - *Space: detector studies*
 - *Time: event mixing*
 - *Easiest for time. Easy for space if fully contained. Difficult for space if particles exit*
 - → should each energy deposition contain particle exit info?
- Ability to move energy depositions into different detectors
 - *Could better inform TPC design choices (protoDUNEs vs. DUNE FDs)*
 - *Multi-detector systematic studies*
 - SBND vs. ICARUS, DUNE SP vs. DP, DUNE FD vs. ND
 - This is likely essential for understanding any DL-based reco approach
- Simulate/filter on topologies/"events" of interest

OUTLINE OF THE MICROBOONE WORK

1. Create energy deposition object to store in event
2. Modify LArVoxelReadout/LArG4 to store energy deposits
3. Standalone IonizationScintillation Alg (separate from LArG4)
4. Standalone ElectronDrift module to take energy deposits → SimChannels
5. Standalone PhotonPropagation via library module to take energy deposits → SimPhotons
6. Validate results are the same (within randomness from propagation, which should be small)
7. Modify FCLs for DetectorVariations to reflect new changes, and verify they work

REPOSITORIES

- Affects, currently, two repositories
 - *lardataobj* for the *EnergyDeposit* object
 - *larsim* for the changes in *G4* and new modules for propagation
- I've pushed MCC8-line compatible feature branches for both
 - *Currently based on uboonecode v06_26_01_11*
 - *feature/wketchum_LArG4Edeps_2*
 - *Note, see my /uboone/app/users/wketchum/larg4_refactor area*

STATUS:

CREATE ENERGY DEPOSITION OBJECT

- Bill Seligman wrote this for the LArG4 refactoring
- I modified it to include PdgCode
 - *Avoid lookup or current scintillation by particle type*
- https://cdcvcs.fnal.gov/redmine/projects/lardataobj/repository/entry/lardataobj/Simulation/SimEnergyDeposit.h?utf8=%E2%9C%93&rev=feature%2Fwketchum_LArG4Edeps_2
- I'm open to suggestions for modifications if they can be made today
 - *E.g. avoid repeated calculations of midpoint and step length?*
 - *Ordered lists of Edeps?*

Status:

Modify LArVoxelReadout/LArG4

- Done*
- New module has option (in LArG4 parameters) for storing these
 - *services.LArG4Parameters.FillSimEnergyDeposits: true*
 - *services.LArG4Parameters.InitialSimEnergyDepositSize: 750000*
- Still creating SimChannels in the normal way for now so we can do direct comparisons
 - *Can modify to not store this if/when we'd like*
- Typically, ~600,000 energy deposits per event
- Need to validate energy deposits outside TPC active volume
 - *Important for light yield ... I bet I'm not getting these yet ... still needs to be looked at by me...ideas welcome...*

STATUS:

IONIZATIONSCINTILLATION ALG

- Done
- Takes in energy deps, and LArProperties and SpaceCharge and LArG4 parameters and DetectorProperties to do all this
- Validation should follow from validating electron/photon stages
- See larsim/IonizationScintillation

STATUS:

ELECTRON DRIFT MODULE

- Done x 2*
- I made a module a while ago which tries to be smarter more efficient in doing this
 - *larsim/ElectronDrift/ElectronDrift_module.cc*
- Bill made a module too that more faithfully copies what LArG4 was doing in its function
 - *I made small changes to use new IonizationScintillation object, and some bug fixes*
 - *larsim/ElectronDrift/SimDriftElectrons_module.cc*
- Mine runs in $\sim 1/3$ the time, but seems biased downwards by $\sim 5\%$
- Bill's seems to repeat the previous code more faithfully...

STATUS: VALIDATION

- I have a gallery macro to compare total deposited charge and energy per SimChannel
 - */uboone/app/users/wketchum/dev_areas/larg4_refactor/compare_SimChannels.C*
 - *SimElectronDriftElectrons_module looks ok...*
- Additionally will be good to verify signal timing in SimChannels
 - *Need some additions for that*
- We are doing larger sample validation now
- No validation set for photons yet, but will follow similar scheme
 - *A place someone could start to work ...*

STATUS:

PHOTONPROPAGATION MODULE

- Module is there ... but totally unvalidated and unoptimized
 - *See larsim/PhotonPropagation*

WORKFLOW CHANGES

- Need to store EnergyDepositions after “LArG4” stage
- Add propagation module in detsim stage?
 - *And drop EnergyDeposits at the end of that fcl file?*
- Needs testing of memory usage more rigorously...
 - *So far it looks ok to me*

(PERSONAL) PROPOSAL (HERB/TRACY, SPEAK UP...)

- Include feature/wketchum_LArG4Edeps_2 in lardataobj and larsim for a v06_26_01 era release
 - *And could be merged to develop? I haven't tested that*
 - *The code could use cleanup, but I'm prioritizing the uboone work needed*
 - *We could wait on the ElectronDrift and PhotonPropagation stuff since it's still in a little flux*
 - MicroBooNE can stage those in uboonecode and merge to simulation later
 - But, it's all new and won't break anything...