NEST in LArSoft

Wed, 24-July-2013, small update 21-Aug

Eric, Matthew, Ben J

NEST is now in LArSoft (in a limited way)

- The main feature of NEST we will capture is the anti-correlation of Nphotons, Nelectrons.
- All NEST Photons produced w Birks recombo
- All of Matthew's multiple "sites" work to produce recombo with Box model is unincorporated as yet
- (To be fair, LArSoft also offers Box model now, thanks to Bruce Baller.)

NEST photons

- Matthew used 3 time constants to produce "quanta" in standalone NEST.
- We chuck NEST's times, and instead let OpFastScintillation's method throw photon birth times from 2 principle late and early constants
- NEST Photons are not tracked in LArSoft. Their response on the uBooNE PMTs is calculated from the look-up library Ben J created from a full uBooNE Scintillation photon-tracking simulation. This works for scintillation, photons given a photon voxel of origin.
- It is in this sense MicroBooNE-specific, as yet. If a photon lookup library existed for LBNE (35 ton) and paddle optical elements were in, we could make it LBNE (35 ton) specific.

NEST electrons

- "maxStep" set to 0.05 micron, which forces delta ray production
- The electrons are also not tracked. They are saved and then "drifted" in LArVoxelReadout, a stalwart class of LArSoft. Drifting, to remind, means a transverse and longitudinal spread is produced and a cloud of electrons is placed on the nearest wire with appropriate timing information.
- We abandon the placement of LArSoft's would be recombined, lifetime-attenuated electrons on wires, in favor of NEST's lifetime-attenuated-only electrons. Because it's already done the recombo.

Mechanics

- We make NestS1Light, formerly known as G4S1Light.cxx,.hh, which is a G4VProcess and thus has a PostStepDoIt(), and which creates the quanta, derive from OpFastScintillation, which is the erstwhile LArSoft fast-simulation.
- The inheritance gives us for free Ben's method to calculate the response on the optical elements (PMTs).

Mechanics: Packages touched

LArG4

LArNestInterface.h,cxx *to hold NumEl, NumPh, dE, dx... and pass among steppers*

NestS1Light.cxx,h *MS's code taken pretty much wholesale*

FastOpticalPhysics.cxx,h *BJPJ's code, adding possibility to instantiate NEST to produce scint photons.*

LArVoxelReadout *"drift" NEST's electrons, apply only lifetime correction*

LArVoxelReadoutGeometry *to force small steps to produce delta rays*

OpFastScintillation *BJPJ class from which NESTS1Light inherits, which provides the method to put the photons that survive onto the OpticalDetectors*

Simulation

LArG4Parameters_Service.h,cxx

simulationservices.fcl

to select NEST

```
fNESTOn    = pset.get< bool   >("EnableNEST", false  );
```

EventGenerator

prodsinglesNUANCE_uboone.fcl

job script

runtime parameters

- We add a fcl parameter EnableNEST, which is defaulted to false.
- Overridden at runtime (see `prodsingleNEST_uboone.fcl`) with services.user.LArG4Parameters.EnableNEST: true
- If true, EnableNEST forces NestS1Light instead of OpFastScintillation. It also forces to drift NEST's e's and attenuate only w lifetime. It also forces the 0.05 micron stepsize.

Steppers

- NestSILight's PostStepDoIt(G4step) comes first, LArVoxelReadout's ProcessHit (G4step) comes second.

- This is crucial here, because in NestSILight we hold onto the electrons and dE/dx produced in a singleton called LArNestInterface. Those e's are then seen and "drifted" in LArVoxelReadout.

LArNestInterface.h

```
class LArNestInterface
{

public:
  LArNestInterface();

   double CurrentStepEnergyDeposit(); // takes the value from the current G4 step
   int NumberIonizationElectrons() {return fNumIonElectrons;}; // returns the number of ionization
electrons created by the energy deposit
   int NumberScintillationPhotons() {return fNumScintPhotons;}; // returns the number of scintillation
photons created
   std::string G4Stepper() {return fLastG4Stepper;};
  TLorentzVector    Last4Location() {return fLastStep4Vec;};

   void SetNumIonEl(int &dum) {fNumIonElectrons = dum;};
   void SetNumSciPhi(int &dum) {fNumScintPhotons = dum;};
   void SetEnergyDep(double &dum) { fEnergyDeposit = dum;};
   void SetG4Stepper(std::string &dum) { fLastG4Stepper = dum;};
   void SetLast4Location(TLorentzVector &dum) { fLastStep4Vec = dum;};

   void Reset() {fEnergyDeposit=0.0; fNumIonElectrons=0; fNumScintPhotons=0; fLastG4Stepper="NA";
fLastStep4Vec.SetXYZT(0.,0.,0.,0.);} ;
   // clears the stored value of the energy deposition to be ready for the next step


 private:
   double fEnergyDeposit;
   int fNumIonElectrons;
   int fNumScintPhotons;
   std::string fLastG4Stepper;
  TLorentzVector fLastStep4Vec;

 };
```

LArNestInterface.cxx

All the normal implementation ...

... and then a free function outside the class, available to anyone to pull a reference to the one living, static instance of this class:

a singleton

```
LArNestInterface& getLArNest() {
 static LArNestInterface thisLNI;
 return thisLNI;
}
```

Validation

- Validation largely remains work to be performed
- I have cursorily checked that each NEST step produces a reasonable number of electrons and photons, and those are faithfully relayed to RecordPhotonsProduced and LArVoxelReadout.
- Matthew says to make the Nph/dE and Nel/dE plots for mip e's and confirm peaks at 29300 and 22000, respectively, for 500 V/cm.
  - Haven't done this yet, though looks to be in ballpark
  - The ana module to make these histograms needs writing.

Validation

- Would also be nice to make a plot that simply sums Ndetected photons to Nelectrons on wires over all wires and PMTs and confirm that that correlates more tightly to true particle energy than we get from LArSoft without NEST

Jonathan Insler doing this now.

  - This implies an ana method to loop over OptDetHits and RawDigits.
    - Also needs writing.
- Do this for different primaries: muons, e's, protons
- Compare w LArSoft's Box and Birks
- To be clear, I am happy (I prefer) someone else to step in and do some of this.

Conclusions

- NEST is in LArSoft
- All the work to show that NEST, as captured thus far, is superior/equal to LArSoft's vanilla photon/electron production/signal formation, remains.
- It's hard to imagine that further NEST details which we've yet to capture will matter too much to the physics.