NuTools & LArSim - LArSoft’s gateway to GENIE and Geant4

Robert Hatcher
Fermilab Computing Sector
Scientific Computing Division / Systems for Scientific Applications /
Scientific Computing Simulation / Physics & Detector Simulations
2016-06-23
What is usability, anyway?

**usability** \,
\,ˈyu-zə-ˈbi-lə-tē\ noun

“The extent to which a product can be used by specified users to achieve specified goals with **effectiveness, efficiency, and satisfaction** in a specified context of use.”

— ISO 9241-11

**effectiveness** how fully the final solution satisfies the original need
- can you get your idea implemented?
- e.g., signal processing, image processing, MVA...

**efficiency** how easy it is to get to that solution
- fitness of the tools
- learning curve
- maintainability

**satisfaction** by how many years working with it will shorten your life
Did I say, “maintainability”!

What maintainability has to do with all of this??

... it’s not usability... it’s just code maintainers’ business! Right?

Well... no:

- **LArSoft is a collaborative contributed project:**
  - *you* write it
  - *you* change, fix and extend code to new needs
  - *you* get frustrated when the code is unreadable
  - *your* effectiveness, efficiency and satisfaction are on the table

- **maintainability is (also) about**
  - *design* accommodating changes $\rightarrow$ *effectiveness*
  - *readable* and understandable code $\rightarrow$ *efficiency*
Outline

● Overview of neutrino simulations
● The a\text{rt} framework, LArSoft, NuTools
● GENIEHelper: Wrapping GENIE up
  ● modifying GENIE behavior
  ● brief mention of NuReweight
● LArG4: The Geant4 a\text{rt} module
  ● Physics Lists
  ● User Actions
  ● Suggestions for moving forward
● Summary
General Simulation Workflow & Products in Neutrino Experiments

We factorize the steps to make them tractable problems:
- Simulation of the beamline
- Simulation of the detectors
- Different energy scales
- Even detector simulations have large variation in needs due to a variety of technology

Flux ($\pi, K, \mu$ decays)

(Also secondaries off target to compare with e.g. MIPP)

Neutrino Physics (e.g. GENIE)

“Truth” particle lists & kinematics

General Detector Simulation

“hits” (energy depositions)

Specific Detector Simulation

“digits” (raw data similar to real detector)
General Simulation Workflow & Products in Neutrino Experiments

- We factorize the steps to make them tractable problems
- Simulation of the beamline
- Simulation of the detectors
- Different energy scales
- Even detector simulations have large variation in needs due to a variety of technology

Beam Simulation ➔ Flux (π,K,μ decays) ➔ Neutrino Physics (e.g. GENIE) ➔ “Truth” particle lists & kinematics ➔ General Detector Simulation ➔ “hits” (energy depositions) ➔ Specific Detector Simulation ➔ “digits” (raw data similar to real detector)
General Simulation Workflow & Products in Neutrino Experiments

We factorize the steps to make them tractable problems:
- Simulation of the beamline
- Simulation of the detectors
- Different energy scales
- Even detector simulations have large variation in needs due to a variety of technology

We simulate:
- Flux $\pi, K, \mu$ decays (also secondaries off target to compare with e.g. MIPP)
- Neutrino Physics (e.g. GENIE)
- "Truth" particle lists & kinematics
- General Detector Simulation
- "hits" (energy depositions)
- Specific Detector Simulation
- "digits" (raw data similar to real detector)
LArSoft Lines Of Code

circa 2016-04-22

$ cloc /grid/fermiapp/products/larsoft/larcore/v05_00_02/source/ /grid/fermiapp/products/larsoft/lardata/v05_03_00/source/ /grid/fermiapp/products/larsoft/nutools/v1_23_02/source /grid/fermiapp/products/larsoft/larsim/v05_02_00/source/ /grid/fermiapp/products/larsoft/larreco/v05_03_00/source/ /grid/fermiapp/products/larsoft/larana/v05_03_00/source /grid/fermiapp/products/larsoft/larpandora/v05_02_01/source/ /grid/fermiapp/products/larsoft/lareventdisplay/v05_02_00/source/ /grid/fermiapp/products/larsoft/larexamples/v05_00_07/source/ 1310 text files. 1308 unique files.

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>701</td>
<td>49671</td>
<td>43680</td>
<td>205556</td>
</tr>
<tr>
<td>C/C++ Header</td>
<td>570</td>
<td>15521</td>
<td>26100</td>
<td>35249</td>
</tr>
<tr>
<td>XML</td>
<td>8</td>
<td>41</td>
<td>72</td>
<td>757</td>
</tr>
<tr>
<td>Bourne Shell</td>
<td>8</td>
<td>70</td>
<td>78</td>
<td>142</td>
</tr>
<tr>
<td>Perl</td>
<td>1</td>
<td>19</td>
<td>6</td>
<td>74</td>
</tr>
</tbody>
</table>

SUM: 1288 65322 69936 241778

$ cloc /grid/fermiapp/products/nutools/v1_23_02/source /grid/fermiapp/products/larsim/v05_02_00/source/ 306 text files. 305 unique files.

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>162</td>
<td>9496</td>
<td>8762</td>
<td>44359</td>
</tr>
<tr>
<td>C/C++ Header</td>
<td>131</td>
<td>2379</td>
<td>3573</td>
<td>6695</td>
</tr>
<tr>
<td>XML</td>
<td>2</td>
<td>8</td>
<td>18</td>
<td>116</td>
</tr>
<tr>
<td>Perl</td>
<td>1</td>
<td>19</td>
<td>6</td>
<td>74</td>
</tr>
<tr>
<td>Bourne Shell</td>
<td>3</td>
<td>52</td>
<td>63</td>
<td>70</td>
</tr>
</tbody>
</table>

SUM: 299 11954 12422 51314

Simulation ~ 21% of total LOC

includes connection to GENIE & Geant4, G4 User Actions, as well as WireSim circa 2016-04-22
### art / root Lines Of Code

$ cloc [top of the head of the art repository, not included the 'doc' directory]

1436 text files. 1366 unique files. 497 files ignored.

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>377</td>
<td>6040</td>
<td>4174</td>
<td>36555</td>
</tr>
<tr>
<td>C/C++ Header</td>
<td>367</td>
<td>6781</td>
<td>7366</td>
<td>23649</td>
</tr>
<tr>
<td>CMake</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Bourne Shell</td>
<td>78</td>
<td>504</td>
<td>251</td>
<td>1365</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>96</td>
<td>570</td>
<td>1204</td>
</tr>
<tr>
<td>Bourne Again Shell</td>
<td>7</td>
<td>53</td>
<td>39</td>
<td>356</td>
</tr>
<tr>
<td>Python</td>
<td>1</td>
<td>13</td>
<td>32</td>
<td>230</td>
</tr>
<tr>
<td>XML</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>C Shell</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Perl</td>
<td>30</td>
<td>267</td>
<td>97</td>
<td>1210</td>
</tr>
</tbody>
</table>

SUM: 947 files, 14427 blank, 13243 comment, 68838 code.

$ ~/bin/cloc root

10966 text files. 10751 unique files. 1810 files ignored.

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>3869</td>
<td>253948</td>
<td>353006</td>
<td>1257809</td>
</tr>
<tr>
<td>C/C++ Header</td>
<td>3925</td>
<td>93108</td>
<td>128715</td>
<td>410475</td>
</tr>
<tr>
<td>CMake</td>
<td>72</td>
<td>649</td>
<td>679</td>
<td>4042</td>
</tr>
<tr>
<td>Bourne Shell</td>
<td>78</td>
<td>504</td>
<td>251</td>
<td>1365</td>
</tr>
<tr>
<td>HTML</td>
<td>192</td>
<td>2206</td>
<td>30</td>
<td>177</td>
</tr>
<tr>
<td>JavaScript</td>
<td>18</td>
<td>2602</td>
<td>994</td>
<td>13454</td>
</tr>
<tr>
<td>Fortran 77</td>
<td>4</td>
<td>351</td>
<td>0</td>
<td>12570</td>
</tr>
<tr>
<td>[others file types]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SUM: 9163 files, 397933 blank, 530773 comment, 1979673 code.
NuTools: shared code for LArSoft + NOvA + other art-based ν expts.

NuTools

- EventDisplayBase
- EventGeneratorBase
  - CRY
  - GENIE
    - GENIEHelper.h
  - GiBUU
- G4Base
  - ConvertMCTruthToG4.h
  - DetectorConstruction.h
  - ExampleAction.h
  - G4Helper.h
  - PrimaryParticleInformation.h
  - UserAction.h
  - UserActionFactory.h
  - UserActionManager.h
- IFDatabase
- MagneticField
- NuBeamWeights
- NuReweigh
  - GENIEReweight.h
  - ReweightLabels.h
  - art
    - NuReweight.h
    - ReweightAna_module.cc
- SimulationBase
  - GTruth.h
  - MCFlux.h
  - MCNeutrino.h
  - MCParticle.h
  - MCTrajectory.h
  - MCTruth.h

undergoing re-factorization of data objects vs art module code

LArSim

Purview of LArSoft collaboration

LArSim

- DetSim
- SimWire_module.cc
- EventGenerator
  - CORSIKA
  - CRY
  - GENIE
    - GENIEGen_module.cc
  - MuonPropagation
- LArG4
  - LArG4_module.cc
- MCChanger
- MCDumpers
- MCSTReco
- PhotonPropagation
- RandomUtils
- SimFilters
- Simulation
  - AuxDetSimChannel
  - LArG4Parameters
  - LArVoxel*
  - ParticleHistory
  - ParticleList
  - SimChannel
  - SimPhotons
  - <other stuff>
- TriggerAlgo

NuTools: shared code for LArSoft + NOvA + other art-based ν expts.

Robert Hatcher
Event Generation w/ GENIE

- **NuTools** evgb::GENIEHelper wraps up GENIE for *art*
  - lots of configurability {GENIE event types; top volume; flux ...}
    - sub-pages have additional information
  - `Sample()` fills \{ simb::MCTruth, simb::GTruth, simb::MCFlux \}
  - call repeatedly (per *art* record) until “pile-up” condition is met
  - API also provides:
    - ctor: initialize w/ pset + geometry (TGeoManager, filename, “mass”)
    - end-of-run: TotalExposure()

- **LArSim** evgen::GENIEGen_module
  - fetch geometry (ROOT TGeoManager, filename, “mass”); passes fhicl::ParameterSet to GENIEHelper
  - initialize pset from alternative random # seed service
  - `produce()` accumulates std::vector of simb::MCTruth, simb::GTruth, simb::MCFlux (+ art::Assns) and puts it in the art::Event
  - add “PassEmptySpills” parameter beyond GENIEHelper
  - fill sumdata::RunData, sumdata::POTSummary, sim::BeamGateInfo
  - create/fill lots of histograms
GENIEHelper

Quick Guide to FHCL parameter variables

Basic GENIEHelper Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POTPerSpill</td>
<td>pile-up parameter (either this or the next should be 0)</td>
</tr>
<tr>
<td>EventsPerSpill</td>
<td>pile-up parameter - generate fixed # of GENIE events per ART record</td>
</tr>
</tbody>
</table>
Into the weeds with GENIE
Beam Simulation

- **Geant4 geometry (C++) + [G4 or Fluka physics (fortran)]**
  - G4 geometry is quite detailed (and good match to as-built)
  - [fluka, if used, physics everywhere (not just target)]

- **Record decay, initial secondary production and initial proton info**
  - now: ancestors history from initial proton to decay products
  - 2ndary production models are active areas of study

- **Possibly uses importance weights and thresholds**
  - \( w_\pi = \min(\max(30/p_{\text{tot}}, 1) \times w_{\text{parent}}, 100) \)
  - threshold = 1 GeV or not (off-axis, i.e. NOvA)
Decay Reweighting

- The probability that a decay results in a neutrino ray that goes through any point depends on the relativistic boost at the decay point; the $\nu$ energy will also depend on position.
- Near and Far detectors subtend a different angular size $\rightarrow$ they see different spectra.
A fictitious parallelogram in space from whence neutrino rays emanate

needs to be sized:

- large enough that all (to best approximation) relevant rays that might run through the geometry pass through the window
- small enough to exclude rays that aren’t of interest
**GNuMIFlux/GDk2Nu vs GSimpleNtpFlux**

- **GENIE’s GNuMIFlux or GDk2Nu**
  - read an entry from ntuple of decay info
  - pick random point on flux window \((x,y,z)\)
  - calculate \(x\)-\(y\) weight, energy, \(p_4\)
  - accept/reject based on weights \((wgt_{x,y} \times wgt_{\text{importance}})\)
  - (possibly) push backwards along ray to \((x’,y’,z0)\)
  - \(\Rightarrow\) PathSegmentList created from this ray

- cycling back to same entry won’t give same ray
  - different window point \(\Rightarrow\) different weight, energy, trajectory

- **GENIE’s GSimpleNtpFlux**
  - simple ntuple format of flavor, position, direction, weight
  - provision for carrying extra info to allow limited hadron reweighting
  - some file level meta data (window position, total protons,...)
v Rays and Geometry

PathSegmentList

StartPosition
Direction
list<PathSegment>

P1
P2
P3

Flux Window

rock
air
detector
components
GENIE x-sections

- GENIE cross-sections distributed as UPS product

  - UPS sets $\text{GENEIEXSECFILE}$ ($\text{GENEIEXSECPATH/gxspl-FNALsmall.xml.gz}$)
  - details found in $\text{GENEIEXSECPATH/README}$
  - not differential x-sec; only as a function of neutrino energy
  - 500 knots, [0.01:400] GeV spaced logarithmically
    - 5% of points > 120; flux has long $E_\nu$ tail
    - could study effect of fewer knots - accuracy vs. size
  - file size ~750 MB (23584 splines)
  - 272 for proton, 302 for neutron, 590 for each of 41 nuclei
    - all 6 $\nu$ flavors
    - e.g. $^{12}\text{C}$, $^{14}\text{N}$, $^{16}\text{O}$, $^{23}\text{Na}$, $^{27}\text{Al}$, $^{28}\text{Si}$, $^{35}\text{Cl}$, $^{40}\text{Ar}$, $^{39}\text{K}$, $^{48}\text{Ti}$, $^{56}\text{Fe}$, $^{137}\text{Ba}$, ...
    - $\text{gxspl-FNALlarge.xml.gz}$ has 106 isotopes (needs unpacking)
  - For non-standard configurations UPS also distributes
    $\text{UserPhysicsOptions.xml}$ and/or $\text{EventGeneratorListAssembler.xml}$
    - add path to $\text{GXMLPATH}$ so GENIE finds it
Choosing a Vertex “Outside the Box”

- When a “topvol” isn’t set, GENIE considers the entire geometry.
- GeomSelectorRockBox trims the volume to the hall + minimum safety + a size proportional to the neutrino energy.
Overlay Pile-up

- Collect events
  - MINOS: pull from input sample files
    - Poisson distribution: \( n_{\text{DetPerSpill}} + n_{\text{RockPerSnarl}} \) for a given intensity
    - single use of detector events, randomize pulling from rock files (reuse, except once)
  - NOvA: generate events until used fixed POTs/Spill

- Distribute events in time over spill interval according to intensity profile
  - offset truth info times (StdHep/HepMC)
  - also offset corresponding hit times, if already propagated in GEANT
  - if combined particle list, adjust parentage indices
  - add any event kinematics/flux records to list for spill
    - good to have mechanism tying kin/flux to particle list
Why Can’t I … ?

“Okay, I’ve looked at the GENIEHelper wiki and I don’t see any way to change physics parameters (e.g. $M_A$, MEC values) or choose physics models from within a FHiCL file for event generation. What gives?”

- short answer: just because ...you don’t want the wrong answer...
- long answer: GENIE must be run in a consistent mode

- to decide whether a $\nu$ flux ray interacts at all and to pick event vertex
  - GENIE samples the material along the ray’s path for the amount of material transversed
  - $P_{\text{interaction}} = \text{the number of nuclei} \times \text{total cross-section}$
  - total cross-section splines are pre-calculated
    - this is computationally expensive!
  - controlled by GENIE UserPhysicsOptions.xml (and possibly EventGeneratorListAssembler.xml)
Why is it ... so slow?

- **Start up**
  - spline reading
    - Loading from ROOT file (vs XML) speeds up loading, but doesn’t appear to significantly change peak memory usage
  - geometry exploration
    - Use pre-calculated max path lengths
  - flux handling
    - limit number of files copied to local disk MaxFluxFileMB

- **Event Generation**
  - off-axis flux w/ Dk2Nu & GNuMI?
    - Use pre-transformed flux GSimpleNtpFlux
  - atmospheric flux
    - GENIE has ideas about how to improve this; lack manpower
  - rock event generation
    - tuning ... (also, is it the GENIE stage or G4 stage?)
Why is the memory footprint big?

- **Total Cross-Section Spline File** `gxspl1-FNALsmall.xml`
  - many knots (500 over E range of [0.01:400] GeV)
    - probably not due to energy range (only a few knots above beam E)
    - need a study “smoothness” vs. # of knots
  - many isotopes (not all needed for all detector geometries)
    - single XML file ... only read once, no provision for multiple files
    - teach GENIE how to do lazy loading ... on the to-do list 😞
    - also adding unnecessary isotopes hurts everyone
      - if not in x-sec spline file: GENIE generation gives up (needed to set max prob)
      - e.g. Beryllium was added to wire composition
        - ~2% of wire mass is Beryllium (added to originally pure Copper)
        - wires are ~0.0023% of the LAr mass in DUNE FD
What Else?

- NuReweight package - interface to GENIE functionality
  - new weights can be calculated for existing GENIE events by changing physics parameters (e.g. $M_A$) and (some) models.
    - need to reconstitute `genie::EventRecord`
    - ratio of differential cross-section
    - doesn’t change reco interference from “bkg” events
  - used to vary models in order to study systematics
  - adjust existing MC samples to new central values w/out regeneration
  - needs someone to take primary lead for maintenance
    - original author has left the field
    - reported issue w/ speed … might need restructuring
GENIE Questions?

- A brief pause ...
- Ask now or forever hold your peace ...
- No, not really ... ask me any time
Tools for Geant4 Use

- NuTools’ G4Base provides help, but less comprehensive
  - Different from generators (e.g. GENIE) for which a unified representation of “interactions” (MCTruth) exists; but that doesn’t hold for output products of Geant4 — energy losses, # photons ...
  - Serves as a tool kit (much like G4 itself)
  - G4Helper - basic setup in art framework of G4 fundamentals
  - DetectorConstruction - create G4 geometry from GDML file
  - ConvertMCTruthToG4 - "does what it says on the tin"
  - G4PhysListFactory (alternative to official standard G4) - choose G4 physics lists at run time
  - UserAction [ Factory ] - user, ah, actions at various points in G4 running
    - see followup pages

- LArSim
  - larg4::LArG4_module
  - register own larg4::PhysicsList for use w/ G4PhysListFactory
  - other LAr specific code
Let’s follow this guy... what could go wrong?
namespace g4b {

class UserAction {

public:

    UserAction() {};
    UserAction(fhicl::ParameterSet const& pset) { Config(pset); }
    virtual ~UserAction() {};

    // Override Config() to extract any necessary parameters
    virtual void Config(fhicl::ParameterSet const& /* pset */ ) {};

    // Override PrintConfig() to print out current configuration
    virtual void PrintConfig(std::string const& /* opt */ ) {};

    // The following is a list of methods that correspond to the available
    // user action classes in Geant 4.0.1 and higher.

    // G4UserRunAction interfaces
    virtual void BeginOfRunAction (const G4Run* ) {};
    virtual void EndOfRunAction (const G4Run* ) {};

    // G4UserEventAction interfaces
    virtual void BeginOfEventAction(const G4Event*) {};
    virtual void EndOfEventAction (const G4Event*) {};

    // G4UserTrackingAction interfaces
    virtual void PreTrackingAction (const G4Track*) {};
    virtual void PostTrackingAction(const G4Track*) {};

    // G4UserSteppingAction interface
    virtual void SteppingAction (const G4Step* ) {};

    // Does this UserAction do stacking?
    // Override to return "true" if the following interfaces are implemented
    virtual bool ProvidesStacking() { return false; }

    // G4UserStackingAction interfaces
    virtual G4ClassificationOfNewTrack
    StackClassifyNewTrack(const G4Track*) { return fUrgent; }
    virtual void StackNewStage() {};
    virtual void StackPrepareNewEvent() {};

    // allow self-identification
    std::string const & GetName() const { return myName; }
    void
    SetName(std::string const & name) { myName = name; }

private:

    std::string myName;  //< self-knowledge
};

} // namespace g4b

#include "G4BaseUserAction.hh"

namespace g4b {

} // namespace g4b
**larg4::LArG4_module**

- Enables fhicl option for geometry overlap checking
  - could extend to also turn off GDML Schema validation (G4Base option)

- Written so as to only add `larg4::ParticleListAction`
  - in principle one could, as NOvA does, allow users to append additional UserAction derived classes.
    - Why? something like `g4n::RockCutterAction`

- **produce()** does
  - extract `std::vector< art::Handle< std::vector<simb::MCTruth> > >` from given labels, or just everything it can find
  - feeds each to G4 with `fG4Help->G4Run(aMCTruth)`
  - for each, get back `std::vector<simb::MCParticle>`, make Assn to MCTruth
  - at end, puts into the `art::Event`, those +
    - `std::vector<sim::SimChannel>`
    - `std::vector<sim::SimPhotons[Lite]>`
    - `std::vector<sim::AuxDetSimChannel>`

could do with some trimming of #includes
Sounds innocuous so far ...

● Some possible issues hidden in LArSim `larg4::PhysicsList`
  ● non-standard; needs maintenance w/ G4 version changes
  ● limits options: probably some things in LArSim don’t work if one chooses a pre-defined G4PhysList from G4 base release
  ● dual parallel world geometries (in addition to the base geometry)
  ● singletons ... decouples code ... but thread safe headache down the road

Ugh

// Note that the name below MUST match the name used in the
// LArVoxelReadoutGeometry or OpDetReadoutGeometry constructor.
LArVoxelParallelWorldScoringProcess->SetParallelWorld("LArVoxelReadoutGeometry");
Parallel Geometries

- Force G4 to consider all boundaries during transport
  - as I understand it, LAr is “voxelized” into (300 μm)³ cubes [default]
  - thought to be there to limit step size, force localized e/γ depositions
    - if so, this is probably not the optimal means of doing this

- In the mean time:
  - odd step structure
  - lots of overhead in transportation process
  - increased memory footprint
  - makes geometry unvisualizable
  - harder to validate geometry

- Zeno’s... paradox... ensues

- eventually land on boundary surface
Alternative to Parallel Geometries

● If this really is the reason, then probably an misuse of G4

● Two possibilities for how to achieve the same effect:
  ● add physics process that limits the step size
  ● special tracking cuts
    ● limit step size for particular particles in particular volumes
Other Geant4 Issues

- Trade off in physics fidelity vs. speed for EM options
  - needs exploration

- One of the physics processes (Scintillation?) is a fork of a standard G4 class just to expose interface that reports number of photons, rather than stack/toss entries
  - latest G4 tag includes this interface (or equivalent)
  - this class can then be removed when that G4 version is adopted
    - improves maintainability ... makes it someone else’s job.

- Is radioactive decay of interest? ... add to physics list?
Testing, Testing... Is this thing on?

- By “upstream” vendors ...
  - GENIE: release validation (improving...big push this summer)
  - G4: release validation [Julia] - monitor “standard” setups
  - G4: physics/performance [Hans on LArIAT]
    - possibly propose to G4 to some tests targeted toward LAr concerns
    - similar to “simplified CMS” test

- By LArSoft Expt / Users ...
  - none of the above is guaranteed to cover regions of interest to Expt X
    - different mix of particles, energies, materials (okay, here mostly LAr, but few of the above tests are on LAr), physics processes, resolutions/sensitivities/thresholds
  - need tests that
    - ensure LArSim/NuTools hasn’t broken interface for a fixed G4 version
    - validate physics quantities when G4 itself changes
Backup Slides
Neutrino Beams

- **NuMI (Main Injector)**
  - LE & ME target/horn configurations
- **Booster Neutrino Beam**
- **LBNF** under design

**Tertiary Test Beams**

- Fermilab: LArIAT ‡ ✓
- CERN: DUNE proto-types
  - single & dual phase ✓ ✓

**Neutrino Detectors**

- MINOS [+] ‡
  - (Near & Far detectors - magnetized)
- ArgoNeuT ‡ ✓
  - (same small LAr detector in test beam / NuMI beam)
- MINERvA ‡
  - (fine grained & multi-target material)
- NOvA ‡
  - (Near & Far detectors - off-axis)
- SBND ✓
  - (Short Baseline Near Detector Expt, formerly LAr1ND)
- ANNIE
  - (to study neutron production in water using BNB ν)
- µBooNE ‡ ✓
- miniBooNE ‡
- ICARUS-T600
  - (to be refurbished & moved from Gran Sasso National Lab in Italy to serve as BNB Far Detector)
- DUNE ✓ + ✓
  - (Deep Underground Neutrino Experiment, formerly LBNE)
    - (Near & Far detectors + 35 ton + test beam single & dual phase prototypes at CERN)
• Beamline Simulations

Energy [GeV]

CC / 6E20 POT / kton / 0.1 GeV

ν

ν

5

10

15

On-Axis

A)

ν

14.6 mrad Off-Axis (NOFLUKA11)

A Simulation

MINOS, Minerva, NOvA

µBooNE, SBND, ICARUS,

Wilson Hall

DUNE

to Minnesota

to South Dakota

nuMI beam

Linac

Booster

muon

campus

Project X

Main Injector and Recycler

Tevatron Ring (decommissioned)

μBooNE, SBND, ICARUS,

Wilson Hall

DUNE

to Minnesota

to South Dakota

nuMI beam

Linac

Booster

muon

campus

Project X

Main Injector and Recycler

Tevatron Ring (decommissioned)
A bit about events: topologies

Events & Backgrounds:
- $\nu$ “detector” (LAr vs. cryostat),
- $\nu$ in surrounding “rock”/“dirt”
- cosmics (single $\mu$, multi-particle)
- radiological sources
- spallation in the rock
- astrophysical (many very small energy depositions)
- nucleon decay ($p \rightarrow K^+\nu$ & $p \rightarrow K^0\mu^+$)

$\nu$ induced events:
- “pile-up” distributed in time (10$\mu$sec vs. drift time) & space (different from colliders)

- basic $\nu$ topologies:
  - CC nu-mu
  - CC nu-e
  - NC (inc pi0)

Much of the physics is in distinguishing these.
As G4Steps are taken, e- drift is calculated within LArG4, not using G4 routines; Transport of photons from G4Step to PMT is sampled from a lookup library (which might originally have been generated with Geant4)
**Some Benchmarking Results**

**LArLATSoft:** single charged particle \((<E>=780\text{ MeV}, \sigma=700\text{ MeV})\)

**Timing:** `art` timer 1000 events (except reco: 100, only \(\pi\))

<table>
<thead>
<tr>
<th>Module</th>
<th>Timing ((\pi^\pm)) sec/evt</th>
<th>T((e^\pm))</th>
<th>T((\mu^\pm))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geant4</td>
<td>0.140</td>
<td>0.388</td>
<td>0.088</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17.8% (1.10%)</td>
<td>39.2%</td>
</tr>
<tr>
<td>ROOT I/O</td>
<td>0.252</td>
<td>0.245</td>
<td>0.203</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32.1% (1.97%)</td>
<td>24.7%</td>
</tr>
<tr>
<td>DetSim (WireSim)</td>
<td>0.394</td>
<td>0.358</td>
<td>0.339</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50.1% (3.08%)</td>
<td>36.1%</td>
</tr>
<tr>
<td>Reco (89% TrackMaker)</td>
<td>(~12)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(93.85%)</td>
<td></td>
</tr>
</tbody>
</table>

Note: Geant4 will scale linearly with complexity of events; reconstruction due to combinatorics will not!
Running DUNE Simulation

DuneTPC: 4 APA FarDet “workspace” (g4.9.6p03)
Timing: `art` timer (100 events)

<table>
<thead>
<tr>
<th>Module</th>
<th>Timing</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENIE Gen</td>
<td>0.1268 sec / evt$^\dagger$</td>
<td>0.05%</td>
</tr>
<tr>
<td>Geant4</td>
<td>3.842 sec / evt</td>
<td>1.55%</td>
</tr>
<tr>
<td>DetSim (WireSim)</td>
<td>44.1075 sec / evt</td>
<td>17.81%</td>
</tr>
<tr>
<td>Reco</td>
<td>199.4674 sec / evt</td>
<td>80.52%</td>
</tr>
<tr>
<td>MergeAna</td>
<td>0.1656 sec / evt</td>
<td>0.07%</td>
</tr>
</tbody>
</table>

Note: Geant4 will scale linearly with complexity of events; reconstruction due to combinatorics will not!  
$^\dagger$ Not including x-sec load time (~80s)
The \texttt{art} framework

\texttt{art v1.17.07 -q debug:e9:nu}
\begin{itemize}
  \item \texttt{sctpksupport v1.10.01 -g current}
  \item \texttt{messagefacility v1.16.22 -q debug:e9}
  \item \texttt{fhiclcpp v3.12.09 -q debug:e9}
  \item \texttt{cetlib v1.15.04 -q debug:e9}
  \item \texttt{cpp0x v1.04.13 -q debug:e9}
  \item \texttt{boost v1.57.0a -q debug:e9}
  \item \texttt{gcc v4.9.3}
  \item \texttt{sqlite v3.08.10.02}
  \item \texttt{root v5.34.32 -q debug:e9:nu}
  \item \texttt{clhep v2.2.0.8 -q debug:e9}
  \item \texttt{fftw v3.3.4 -q debug}
  \item \texttt{gsl v1.16a -q debug}
  \item \texttt{pythia v6.4.28d -q debug:gcc493}
  \item \texttt{postgresql v9.3.9 -q p2710}
  \item \texttt{python v2.7.10}
  \item \texttt{mysql_client v5.5.45 -q e9}
  \item \texttt{xrootd v3.3.4d -q debug:e9}
  \item \texttt{libxml2 v2.9.2 -q debug}
  \item \texttt{cppunit v1.12.1c -q debug:e9}
  \item \texttt{gccxml v0.9.20150423}
  \item \texttt{tbb v4.4.0 -q debug:e9}
\end{itemize}