Geant 4

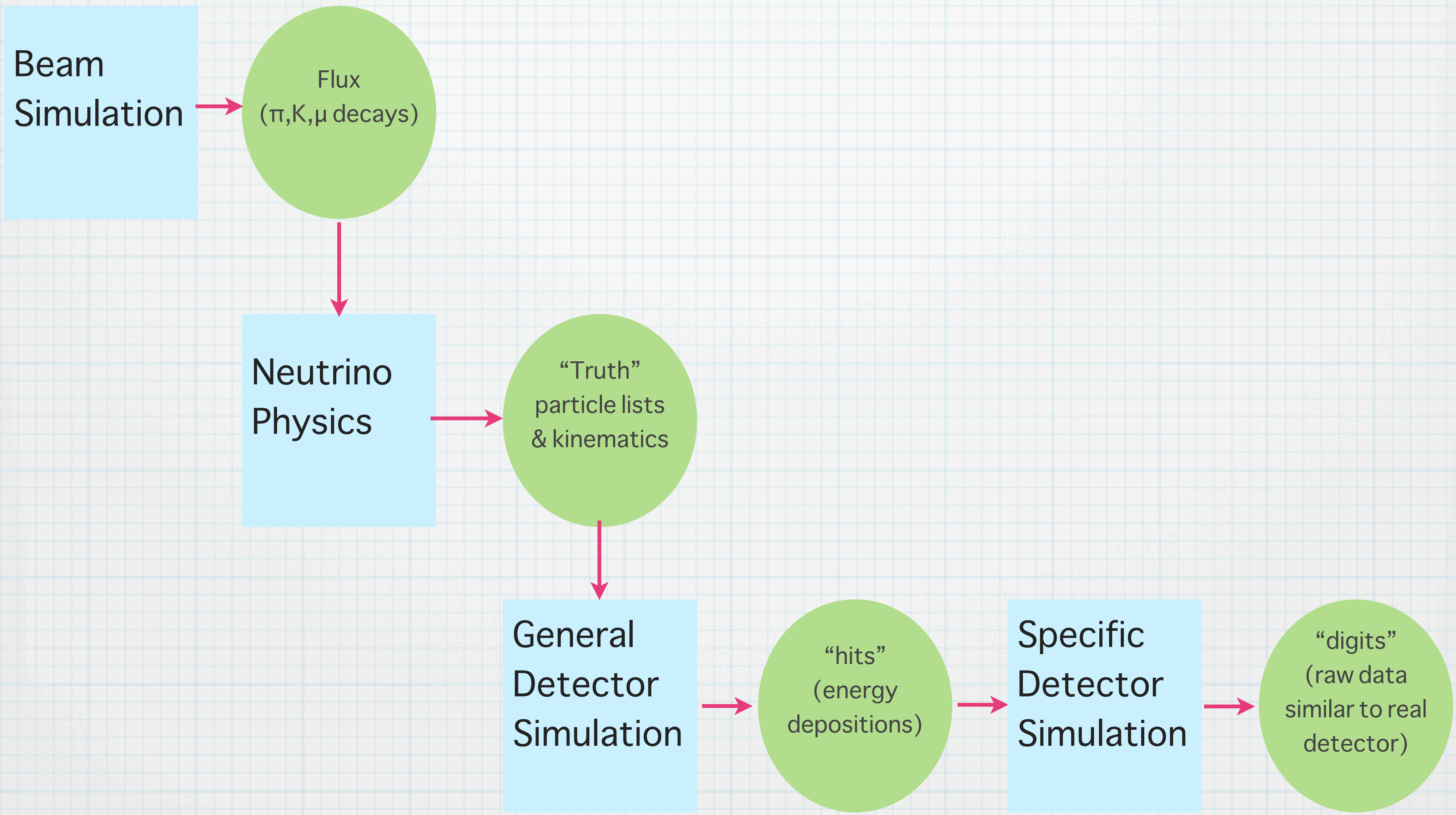
Simulation Tools in the Neutrino Experiments

Robert Hatcher

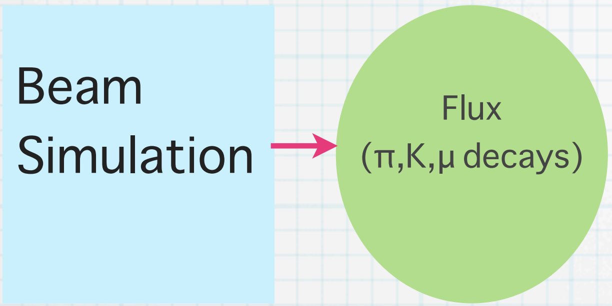
Fermilab Computing Division

2013-02-27

General Workflow & Products



Flux



Combines geometry w/ hadronic physics (initial production is especially important), energy loss, and decays. Initial particles are 120 GeV or 8 GeV protons

In NuMI and LBNE experiments the two primary tools are: G4NuMI (and derivatives) and FLUGG

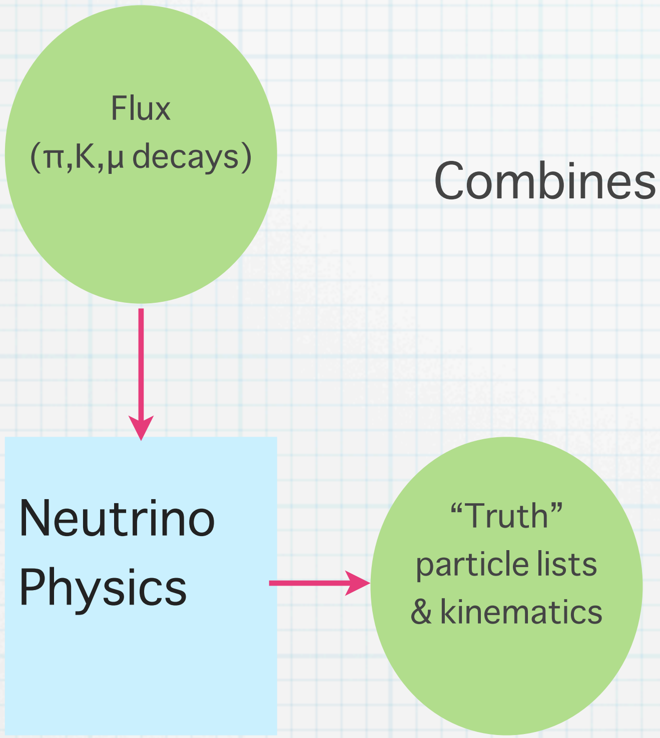
- The first is a purely Geant4 base simulation
  - (uses "QGSP" physics list out-of-the-box, can be changed w/ edit to source code and rebuilding the executable)

- The second uses the same G4 geometry but uses fluka for physics; glued together via a package called "flugg"

Booster simulation is currently inactive

Geant 4

Neutrino Physics



Combines flux, geometry w/ neutrino physics

Currently in use: GENIE and NEUGEN3 (MINOS is deprecating this in favor of GENIE)

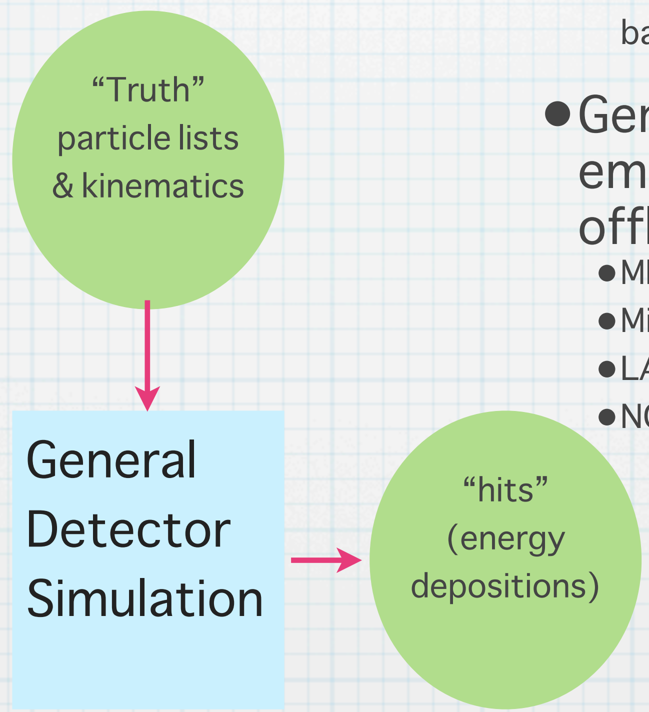
GENIE can be used stand-alone (Minerva) or embedded in the offline framework (NOvA/ LArSoft, eventually MINOS)

Geant 4

Bulk Properties including, Reinteractions, Decays, Energy Loss and Transport

Particles to be dealt with range from 100 GeV to 25MeV; most emphasis is on the sub-5GeV range. Particles include all species of hadrons, charged leptons and ions.

Only MINOS is magnetized, but the list of materials run the gamut from low density single element liquid Argon, to complex admixtures of materials and densities.



Basic particle physics combined with specific detector geometries.

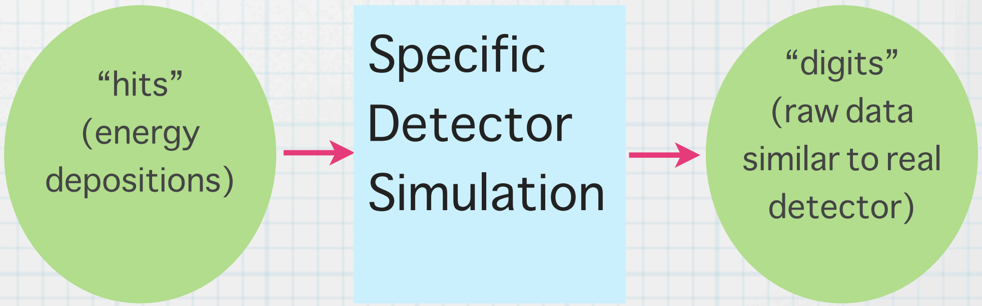
- Majority are Geant4 based
  - ART based systems use QGSP_BERT physics list by default, but trivially changeable at run time
  - MINOS uses Geant3 in its old fortran system, and ROOT's VMC interface to Geant3 in the C++ based system (should also work w/ Geant4)
- Generally these simulations are embedded into the experiment's offline framework
  - MINOS uses "loon" (homegrown built upon ROOT)
  - Minerva uses Gaudi
  - LArSoft (ArgoNeut, microBooNE, LBNE) uses ART
  - NOvA uses ART

Geant 4

Detector Specific Simulations

The stage of converting energy depositions into “raw digits” involves the specific technology of the detector. Only the LAr experiments can make much use of code sharing due to the significant differences.

Some experiments that involve light collection use Geant to do that part of simulation, either directly embedded in the framework, or as a separate simulation that provides distributions that can be sampled (mostly for speed and efficiency reasons).



Backup Slides
Details on ν Simulation

- Based on NOvA presentation (so a specific bias towards their tool chain)

General Notes on Simulations

- The simulation is a chain or series of steps each modeling a specific physical process
- Robert's Law of Simulations:
  - The MC is only as good as the weakest link in the chain; factorize them to make them tractable problems
- Kregg Arms' Rules of Monte Carlo Fight Club:
  1. The Monte Carlo is always wrong
  2. The Monte Carlo is ALWAYS wrong
  3. The Monte Carlo is always the best we have at the time
  4. ~~Only two guys to a fight~~
  5. ~~One fight at a time~~
  6. The analysis groups are collectively the only customer of

Geant 4

General Notes on Simulations

- The simulation is a chain or series of steps each modeling a specific physical process
- Robert's Law of Simulations:
  - The MC is only as good as the weakest link in the chain; factorize them to make them tractable problems
- Kregg Arms' Rules of Monte Carlo Fight Club:
  1. The Monte Carlo is always wrong
  2. The Monte Carlo is ALWAYS wrong
  3. The Monte Carlo is always the best we have at the time
  4. ~~Only two guys to a fight~~
  5. ~~One fight at a time~~
  6. The analysis groups are collectively the only customer of the Monte Carlo group
  7. The Customer is always right

Geant 4

General Notes on Simulations

- The simulation is a chain or series of steps each modeling a specific physical process
- Robert's Law of Simulations:
  - The MC is only as good as the weakest li[…] factorize them to make them tractable [...]
- Kregg Arms' Rules of Monte Car[…]
  1. The Monte Carlo is always wrong
  2. The Monte Carlo is ALWAYS wrong
  3. The Monte Carlo is always the best we have at the time
  4. ~~Only two guys to a fight~~
  5. ~~One fight at a time~~
  6. The analysis groups are collectively the only customer of the Monte Carlo group
  7. The Customer is always right

The trick it to know what needs improvement and what is feasible given the constraints of knowledge of the systems being modeled, people to develop the models and the available compute resources

General Notes on Simulations

- The simulation is a chain or series of steps each modeling a specific physical process
- Robert's Law of Simulations:
  - The MC is only as good as the weakest link in the chain; factorize them to make them tractable problems
- Kregg Arms' Rules of Monte Carlo Fight Club:
  1. The Monte Carlo is always wrong
  2. The Monte Carlo is ALWAYS wrong
  3. The Monte Carlo is always the best we have at the time
  4. ~~Only two guys to a fight~~
  5. ~~One fight at a time~~
  6. The analysis groups are collectively the only customer of the Monte Carlo group
  7. The Customer is always right

General Steps

simulation step	software	work product
Beamline	flugg{ fluka / Geant4 } GENIE	flugg.root gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other “generator”	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Photon ^Simple Transport	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

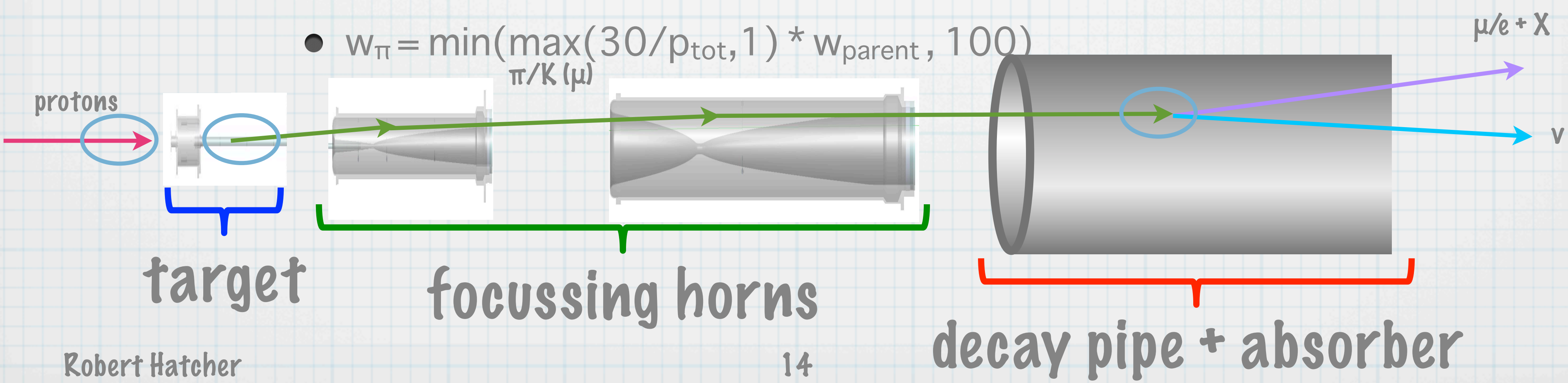
Geant 4

Beamline Simulation

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other “generator”	GENIE or CRY/SingleParticle	+ ν<MCTruth>,ν< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Photon ^Simple Transport	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

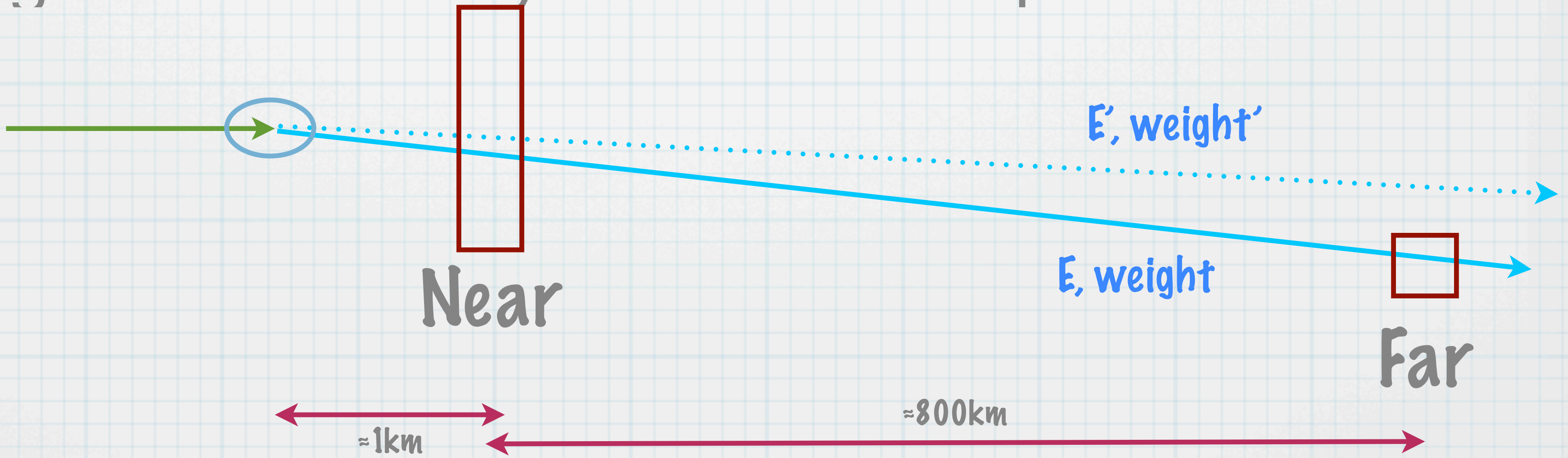
Beam Simulation (Flugg)

- Is it “fl-ugh” or “floog”?
- Geant4 geometry (C++) + Fluka interactions (fortran)
  - G4 geometry is quite detailed (and good match to as-built)
  - fluka interactions everywhere (not just target)
- Record decay, initial secondary production and initial proton info
  - initial protons have some position and angular distribution
  - 2ndary production models are active areas of study
- Uses importance weights and thresholds
  - $w_{\pi} = \min(\max(30/p_{tot}, 1) * w_{parent}, 100)$



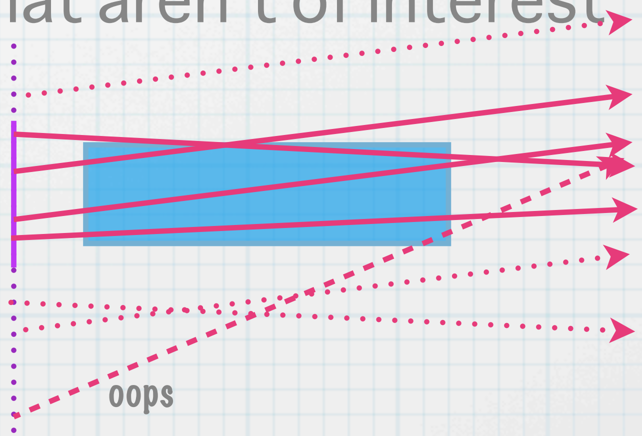
Decay Reweighting

- 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 ν energy will also depend on position
- Near and Far detectors subtend a different angular size → they see different spectra



Flux Window

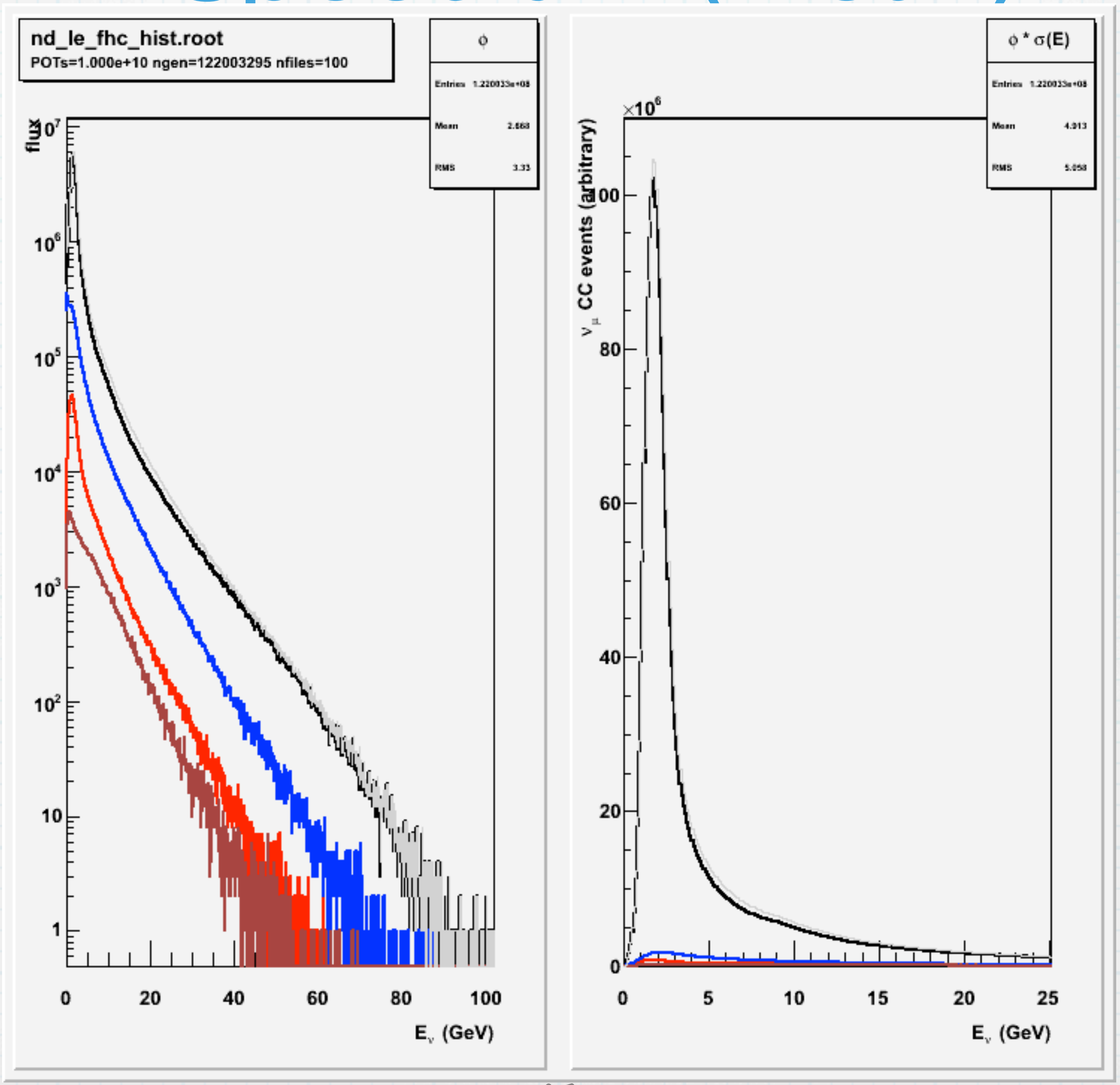
- A fictitious parallelogram in space from whence neutrino rays emanate
- needs to be sized:
  - large enough that all (to best approximation) relevant rays that might run through the geometry pass through the window
  - small enough to exclude rays that aren't of interest



ν Spectrum (Near)

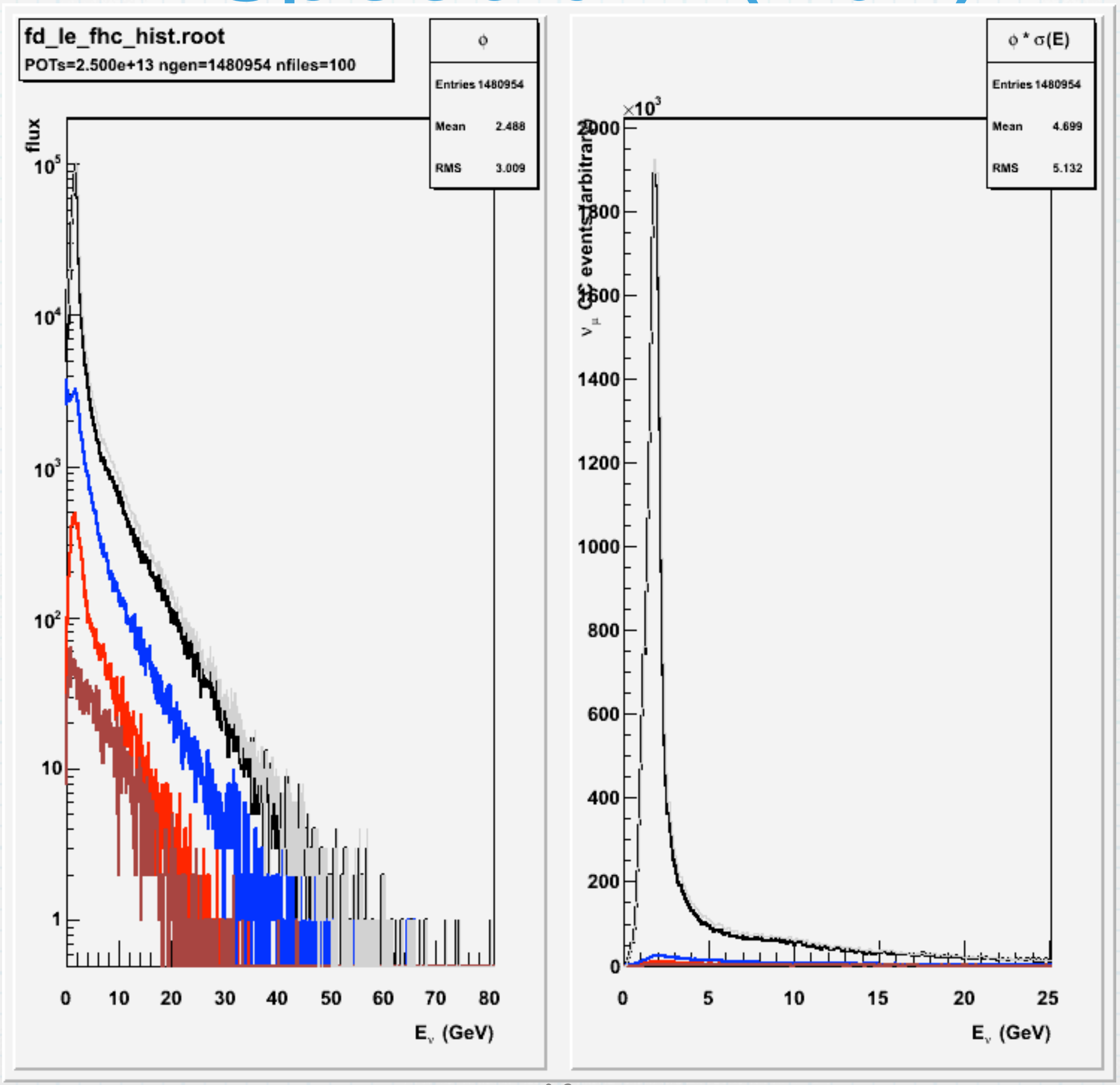
Geant 4

Flux

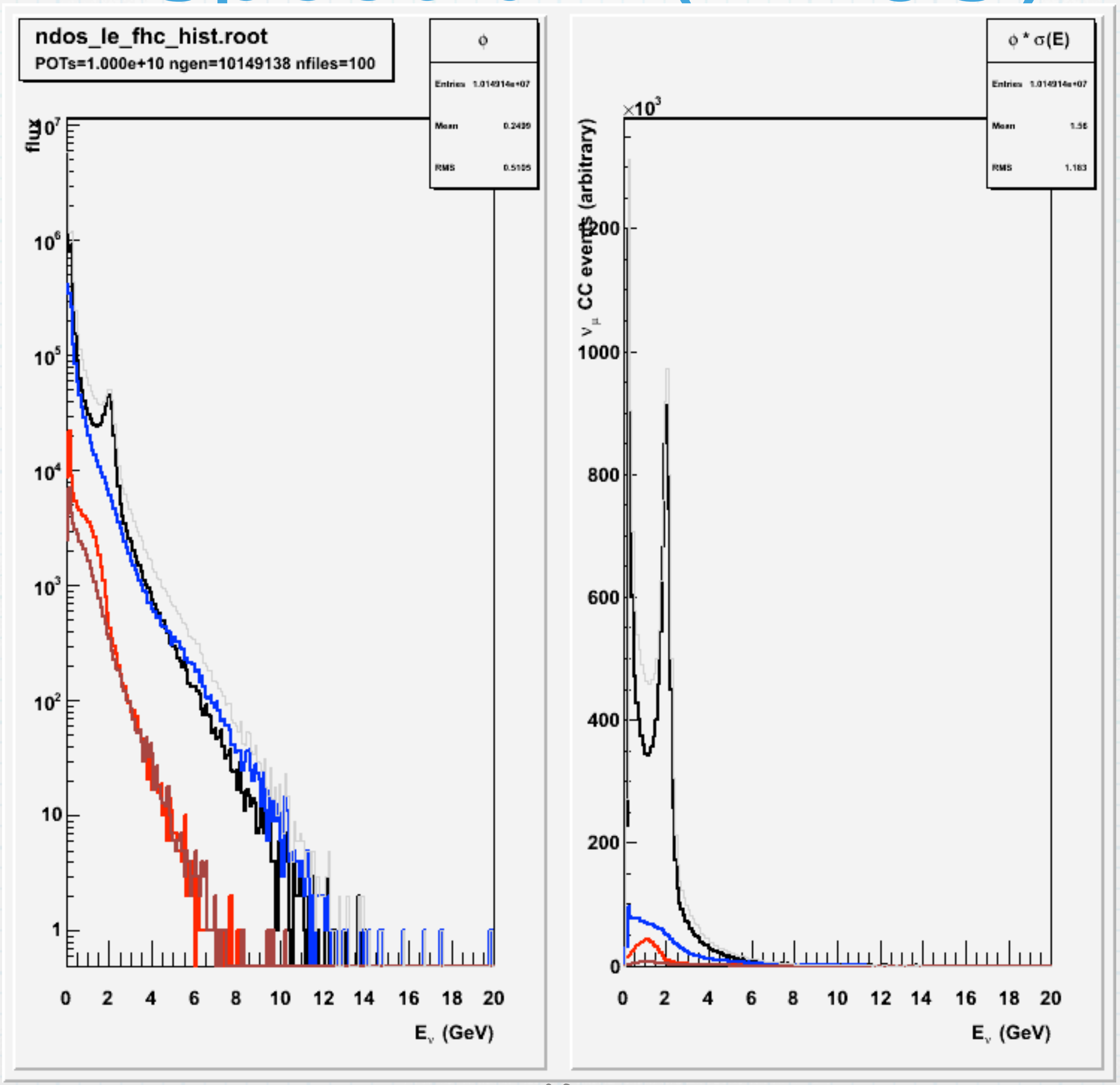


Flux × σ

ν Spectrum (Far)



ν Spectrum (NDOS)



Geant 4

Beamline Simulation

simulation step	software	work product
Beamline	flugg{ fluka Geant4 }	flugg.root
ν Interactions or other "generator"		
Energy Depositions		
Light Propagation		
DAQ Electronics	ReadoutSim	vector<RawDigit> + vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

Flugg files are primarily (weighted) ntuple entries representing decays of hadrons (pions,kaons) and muons that result in a neutrino. For simple plotting they contain info for energies and weights of nu rays running through a fictitious detector center; for all but the far detector the energy and weight spectrum varies across the detector.

Geant 4

Beamline Simulation

simulation step	software	work product
Beamline	flugg{ fluka / Geant4 }	flugg.root
ν Interactions or other "generator"		
Energy Depositions		
Light Propagation		
DAQ Electronics	ReadoutSim	vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

The information in the file must be re-evaluated to sample the "flux window". Think of neutrino rays passing through the window and intersecting the geometry. The window placement and orientation relative to the beam system is specific for each detector. The window must be of sufficient size to encompass all rays that might intersect the detector.

Beamline Simulation

Geant 4

simulation step	software	work product
Beamline	flugg{ fluka / Geant4 }	flugg.root
ν Interactions or other "generator"		
Energy Depositions		
Light Propagation		
DAQ Electronics		
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

Turn decays into (neutrino) rays.

Additional information is carried from the flux ntuple downstream in the chain to allow flux re-weighting of the original hadron production physics (cf. SKZP).

Beamline Simulation

Geant 4

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ ν<MCTruth>,ν< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Photon^Simple Transport	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

"Flux" ntuples

- Retain decay info which allows one to
  - re-evaluate for new detector positions
  - correctly correlate energy, position, direction
- Similar ntuple format used for pure Geant4 (and geant3) simulations of beamline
  - g4numi uses (almost?) same geometry as flugg
  - GENIE can read all 3 formats
- Carry 2ndary production info for reweighting
  - HARP, SPY, MIPP, NA49, NA61/SHINE

GNuMIFlux/GSimpleNtpFlux

- GENIE's GNuMIFlux
  - read an entry
  - pick random point on flux window (x,y,z)
  - calculate x-y weight, energy, p4
  - accept/reject based on weights ($wgt_{x\text{-}y} * wgt_{importance}$)
  - (possibly) push backwards along ray to (x',y',z0)
  - ⇒PathSegmentList created from this ray
  - cycling back to same entry won't give same ray
    - different window point ⇒ different weight, energy, trajectory
- GENIE GSimpleNtpFlux
  - simple ntuple format of flavor, position, direction, weight

Flux Validation

- Hard ... nothing directly measures flux
- Ongoing effort to compare secondaries w/ experiment as proxy for flux
- Comparison of alternatives
  - flugg vs. geant4 vs. geant3
- Possibilities for future test suites:
  - validation of coordinate transform + boost
  - ...

Geant 4

Beamline Simulation

simulation step	software	work product
Beamline	flugg { fluka / Geant4 }	flugg.root
	GENIE	gsimple.root
ν Interactions or other "generator"		...ux>
		...article>
Energy Depositions		
Light Propagation		...al>
DAQ Electronics		
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

GENIE's GNuMIFlux class interprets flugg (and similar) files, performing the sampling over a window for a given detector location relative to the beam. This can be computationally expensive, so it makes sense to do it separately and save the results.

Enter the ART Framework

Geant 4

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other “generator”	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Simple ^ PhotonTransport	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

Enter the ART Framework

Geant 4

simulation step	software	work product
Beamline	flugg { fluka / Geant4 }	flugg.root
	GENIE	gsimple.root
ν Interactio…	nova (ART framework)	art::Event (art::Run)
or other “genera…		…v< MCFlux> …TSum) …ector<MCParticle>
Energy Depositio…		…HitList> …icleList>
Light Propagati…		…tonSignal>
DAQ Electron…		…vDigit> …vTrigger>
Reco/Analysis	UserCode	…racks + histograms

[Sigh] It is unfortunate that the ART framework uses the term “**event**” in this context, instead of “**record**” [or “snarl”] because we in the neutrino field usually reserve the term “event” to mean an individual **neutrino interaction**. Overloading the term “event” often leads to confusion.

Neutrino Physics

Geant 4

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ ν<MCTruth>,ν< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Simple PhotonTransport	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

Neutrino Physics

Geant 4

simulation step	software	work product
Beamline	flugg { fluka Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Deposi		FLSHitList> ParticleList>
Light Propaga		<PhotonSignal>
DAQ Electr		<RawDigit> <RawTrigger>
Reco/An		ngs, Tracks

GENIE brings together an external flux, a user supplied ROOT geometry and its knowledge about neutrino cross sections to determine if individual neutrino rays pulled from the flux interact in the geometry. If they do, then it chooses a vertex location and the kinematics of the interaction (QE/resonance/DIS/etc,x,y...).

Geant 4

Neutrino Physics

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Deposi		SHitList> rticleList>
Light Propaga		otonSignal>
DAQ Electr		awDigit> awTrigger>
Reco/An		, Tracks

Given an initial state in the nuclear environment, GENIE also propagates particles out of that nucleus (undergo internuclear scattering and absorption). This results in a list of target / probe (nu) particles, intermediate states and final state particles.

Neutrino Physics

Geant 4

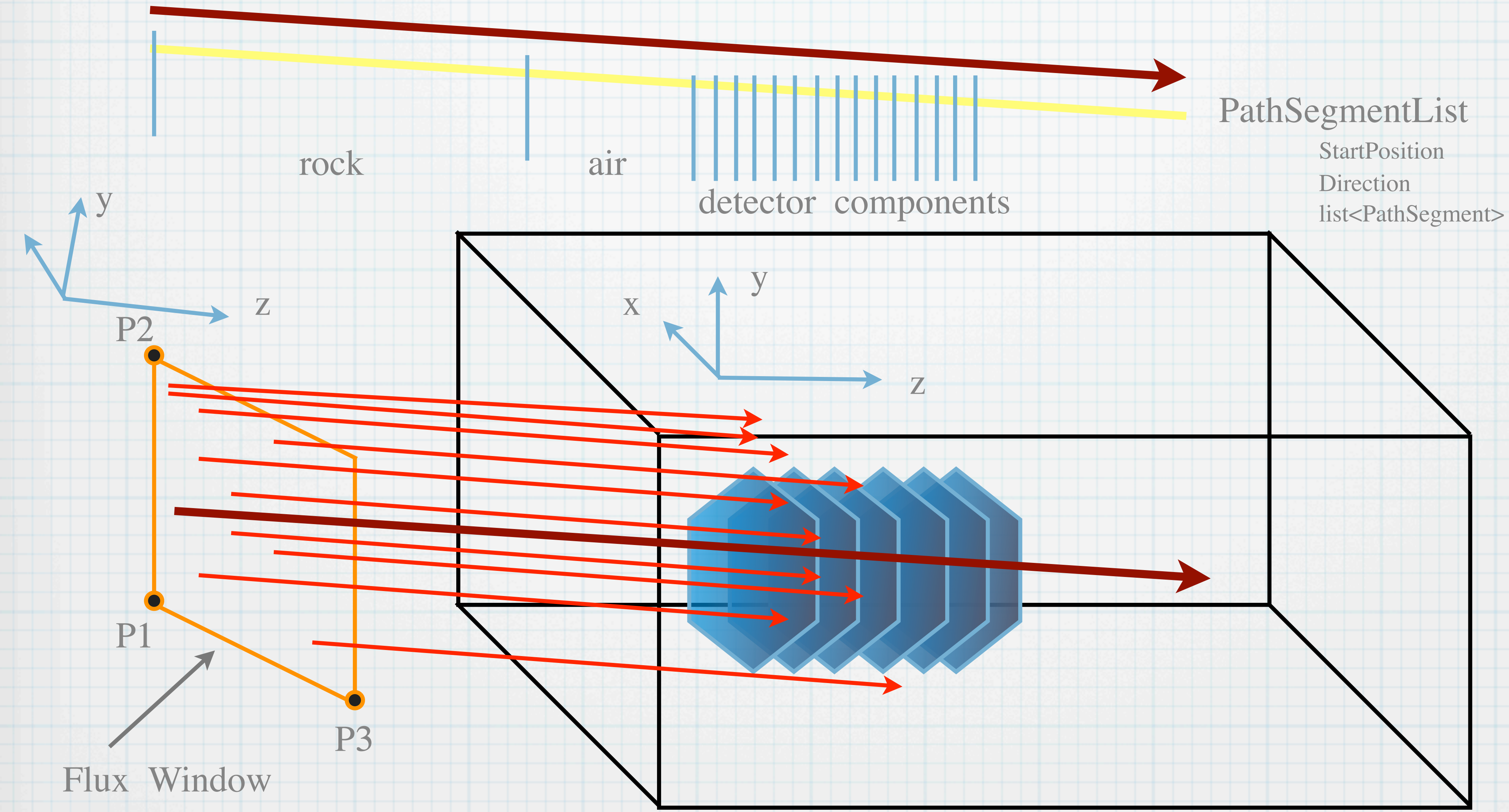
simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
ν Interactions or other "generator"	nova (ART framework) GENIE or CRY/SingleParticle	art::Event (art::Run) + v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Deposi		SHitList> leList>
Light Propaga		nSignal>
DAQ Electr		Digit> Trigger>
Reco/An		racks

GENIEHelper (NOvA,LAr) handles doing overlays (multiple interactions w/ offset times) and filling the ART record. It also copies the flux info to the ART record.

GENIE can also "mix" neutrino flavors from the flux generator.

Robert Hatcher

ν Rays and Geometry



Geant 4

PathSegments

- components: dist from ray start; step length; volume; medium; material; global position for entrance and exit
- Used to determine probability of interaction in segments and then choice of vertex position

Geant 4

Geometry Analyzer

- Uses ROOT geometry
  - that's what experiments generally use in their offline reconstruction (G4 linkage is looked on askance)
  - has the necessary straight-line "transportation"
    - all that is needed is volume enter/exit point along ν ray
- NOvA: some discussion about validating
  - cross check ROOT vs. G4 vs. simple by-hand cases
  - partly to validate GDML → ROOT / G4 representation

GENIE x-sections

- GENIE cross-sections: ~440 MB (15574 splines)
  - 271 for proton, 301 for neutron, 577 for each of 26 nuclei
    - currently need only ~14 of 26: C^{12}, N^{14}, O^{16}, Na^{23}, Al^{27}, Si^{28}, S^{32}, Cl^{35}, Ar^{40}, K^{39}, Ca^{40}, Ti^{48}, Fe^{56}, Ba^{137}
    - not: Mg^{24}, P^{31}, V^{51}, Mn^{55}, Fe^{54}, Fe^{57}, Fe^{58}, Ni^{59}, Cu^{64}, Sn^{119}, Pb^{207}
  - 500 knots, [0:200] GeV spaced logarithmically
    - 5% of points > 120; ND/FD MN flux has entries up to 104 GeV
    - could study effect of fewer knots
  - Loading from ROOT file (vs XML) speeds up loading, but doesn't appear to significantly change peak memory usage
- Study: # knots in spline vs. accuracy

Geant 4

X-Section Validation

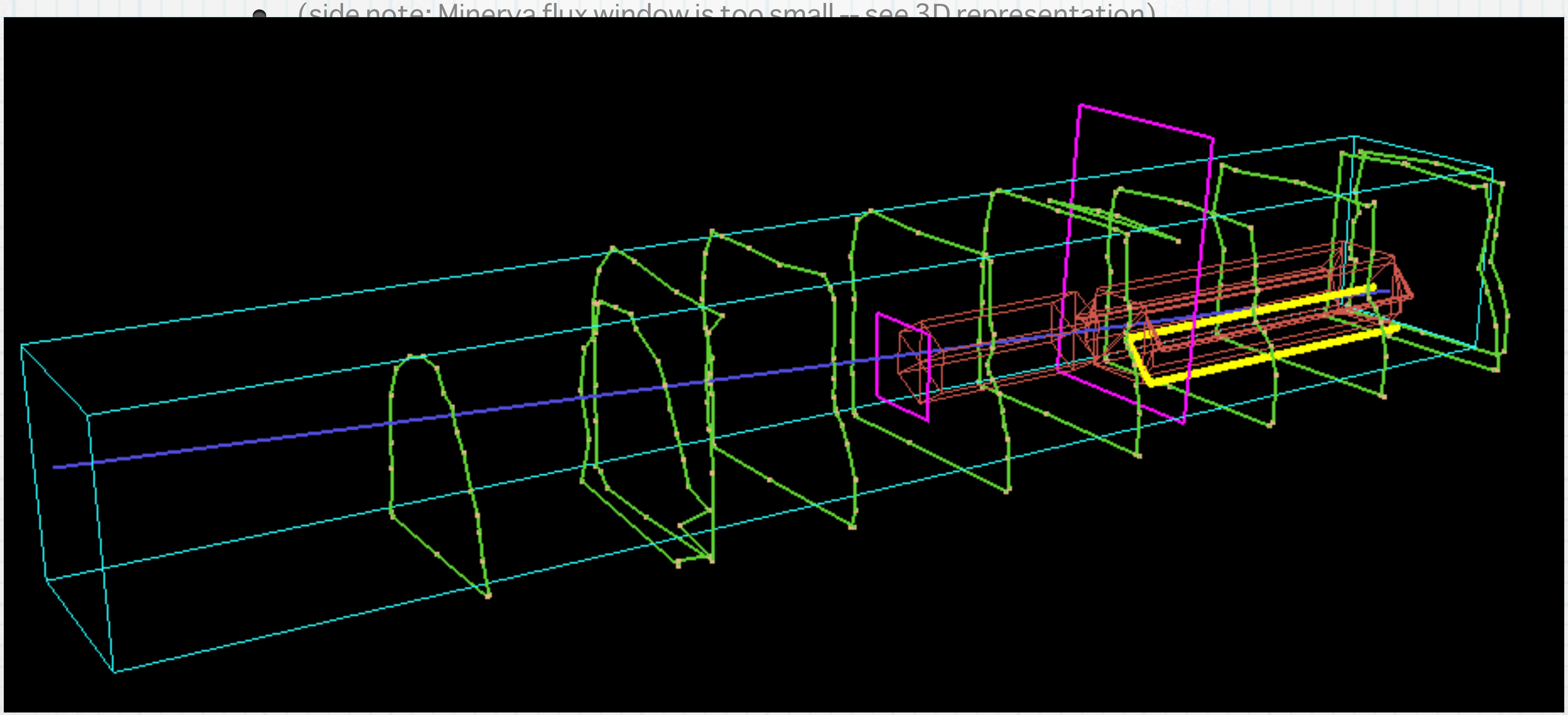
- GENIE collaboration has collected together extensive set of measurements
  - can compare to other neutrino and electron scattering experiments
  - total σ related to F_1 and xF_2, ultimately to parton distribution functions
  - some exclusive processes are theoretically better understood and thus can be a standard candle with single adjustable parameter (M_A in QE)
- MINOS/NOvA use 2 detectors to minimize

Internuclear Scattering

- Steve Dytman primary developer
  - ?data

Detector Environment

- MINOS Near Hall model is not very accurate
  - ...just a rectangular box
    - (side note: Minerva flux window is too small -- see 3D representation)



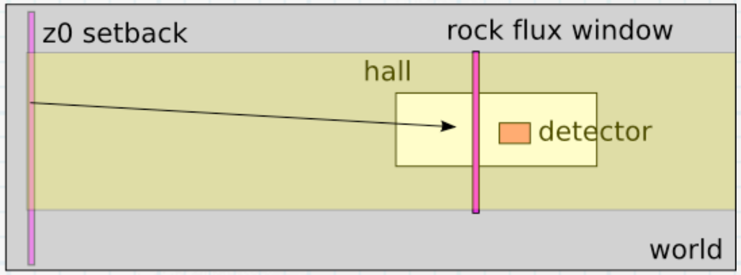
GMINOS "hall" | survey measurements | detector | flux window | beam centerline

Geant 4

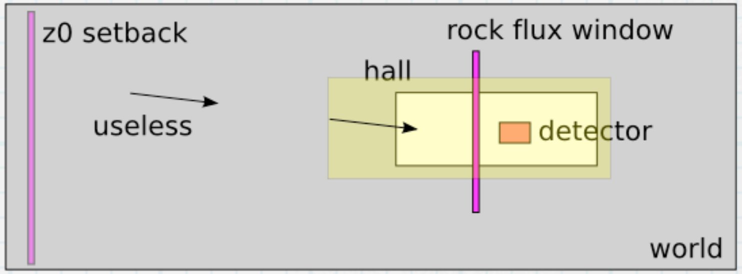
Choosing a Vertex "Outside the Box"

- 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
- Partially integrated into GENIEHelper, but not

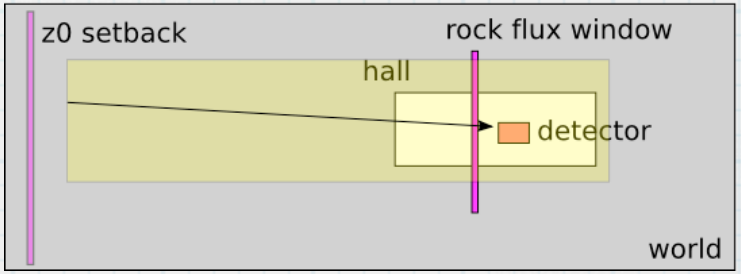
No GeomSelector



RockBox: 2 GeV

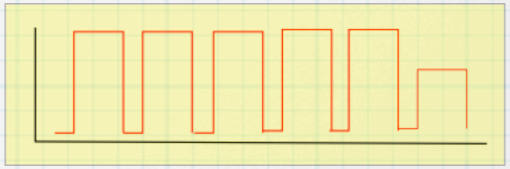


RockBox: 80 GeV



How to overlay

- Collect events
  - MINOS: pull from input sample files
    - Poisson distribution - nDetPerSpill + nRockPerSnarl for a given intensity
    - single use of detector events, randomize pulling from rock files (reuse, except once)
  - NOvA: generate events until used X POTs
- 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

Shower & Track Physics

Geant 4

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Simple PhotonTransport	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

final state only

Shower & Track Physics

Geant 4

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propaga		<PhotonSignal>
DAQ Electr		<RawDigit> <RawTrigger>
Reco/Ana		ngs, Tracks

final state only

Geant4 propagates the final state particles through its representation of the geometry. It knows about the physics of energy loss, particle scattering, decays, etc.

"Hits" are true energy depositions.
FLS = Fiber in (Long,Liquid) Scintillator

Robert Hatcher

Geant 4

Shower & Track Physics

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propaga…		…otonSignal>
DAQ Electr…		…awDigit> …awTrigger>
Reco/Ana…		…, Tracks

final state only

The output ParticleList includes shower intermediates and such, e.g. Michel electron truth info.

Photonics

simulation step	software	work product
Beamline	flugg { fluka / Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other "generator"	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Simple PhotonTransport ^	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

Electronics

simulation step	software	work product
Beamline	flugg { fluka / Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other “generator”	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Simple PhotonTransport ^	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

Geant 4

End of Simulation

simulation step	software	work product
Beamline	flugg { fluka / Geant4 }	flugg.root
	GENIE	gsimple.root
	nova (ART framework)	art::Event (art::Run)
ν Interactions or other “generator”	GENIE or CRY/SingleParticle	+ v<MCTruth>,v< MCFlux> (RunData, POTSum) MCTruth hold a vector<MCParticle>
Energy Depositions	Geant4	+ vector<FLSHitList> vector<ParticleList>
Light Propagation	Simple Photon^Transport	+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

End of Simulation

Geant 4

simulation step	software	work product
Beamline	flugg { fluka, Geant4 }	flugg.root
		gsimple.root
		art::Event (art::Run)
		+ v<MCTruth>,v<MCFlux> (RunData, POTSum)
		MCTruth hold a vector<MCParticle>
Energ		+ vector<FLSHitList> vector<ParticleList>
Ligh		+ vector<PhotonSignal>
DAQ Electronics	ReadoutSim	+ vector<RawDigit> vector<RawTrigger>
Reco/Analysis	UserCode	Clusters, Prongs, Tracks + histograms

The goal of a simulation is to create "fake data" that mimics what the real detector would give, but includes truth information.

Geant 4

Final Notes

- Many steps are computationally expensive
  - combinatorics are a bear (detectors, beams, mixing...)
  - wise to save and share common files
    - sometimes even (sensible) intermediate stages
    - if one is only interested in an early stage and files aren't available, then do only the necessary steps
      - if you're validating GENIE truth info and not using the hits or beyond, then only run through the generator stage.
- Make use of random seeds to "mix it up" without having to re-run earlier stages
  - especially the first GENIE ν interaction; reuse flux files
- As far as possible try to validate individual steps, not the whole chain at once