GENIE User Forum
20/11/2019

GMCJDriver
Neutrino telescopes

Alfonso Garcia

Nikhef

GMCJDriver

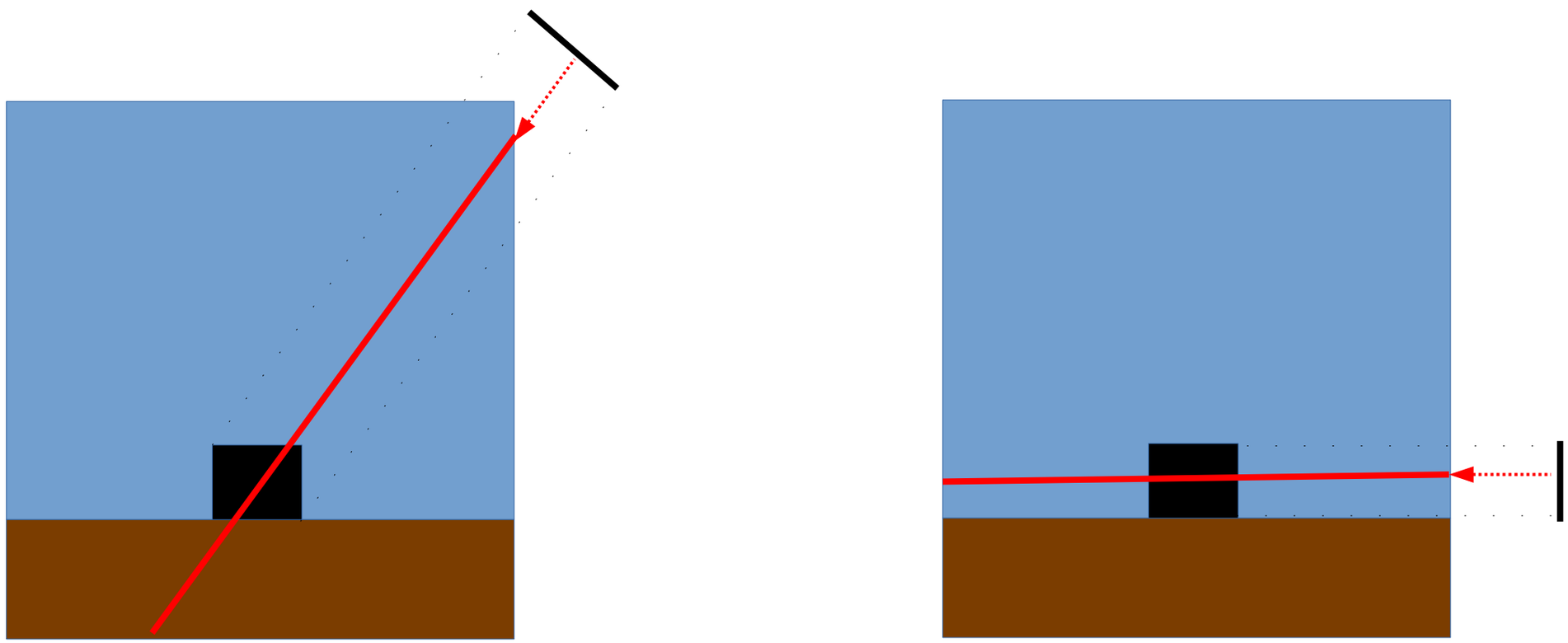
- This class allows event generation with complex flux and geometries.

- Who does it work:
  - Compute maximum probability of interaction in a particular energy range: $P_{int}^{max}(E_{min},E_{max})=1-\exp[-N_A sig(E_{min},E_{max}) pl_{max}]\sim N_A sig(E_{min},E_{max}) pl_{max}$
  - Shoot neutrino with energy E in geometry.
  - Compute its $P_{int}/P_{int}^{max}[E]$
  - Use accept/reject method to decide whether the neutrino interacts.
  - Weight event with $P_{int}^{max}[E]$.

- This method is really efficient if the differences between path lengths are similar inside the geometry.

Neutrino telescopes

- Neutrinos are generated isotropically around the geometry.
- Cylinder is used as default geometry for these experiments.
  - We ‘loose’ an important fraction (~25% assuming 2*Diameter=Height).
- Would it be possible to force all neutrinos to interact?



New GMCJDriver

- Add additional option → ForceInteraction
  - It should be called before Configure (same behavior as for ForceSingleProbScale).

```
//____________________________________________________________________________
void GMCJDriver::ForceInteraction()
{
// Use a single probability scale. That generates unweighted events.
// (Note that generating unweighted event kinematics is a different thing)
//
  fForceInteraction = true;

  LOG("GMCJDriver", pNOTICE)
    << "GMCJDriver will generate weighted events forcing the interaction. ";
}
```

- This option will disable the computation of P_{int}^{max}.

- Workflow.
  - Shoot neutrino with energy E in geometry.
  - Compute its P_{int}=1-exp[-$N_A$sig(E)pl]
  - Force the interaction along the trajectory of the neutrino.
  - Weight event with P_{int}

- The position of the vertex is computed using same method as before:
  - Map P_{int} for different elements and pick one accordingly.
  - Generate vertex in one of the materials containing the selected element.

Extra

- To compute the P_{int}^{max} for different energies the following binning is used:
- Not optimal when energy range is 10^2-10^{10} GeV.
  - Move to logarithmic scale.

OLD

```
double de   = fEmax/300.;//djk june 5, 2013
double emin = 0.0;
double emax = fEmax + de;
int n = 1 + (int) ((emax-emin)/de);

PDGCodeList::const_iterator nuiter;
PDGCodeList::const_iterator tgtiter;

// loop over all neutrino types generated by the flux driver
for(nuiter = fNuList.begin(); nuiter != fNuList.end(); ++nuiter) {
  int neutrino_pdgc = *nuiter;
  TH1D * pmax_hst = new TH1D("pmax_hst",
           "max interaction probability vs E | geom",n,emin,emax);
```

NEW

```
int NENE = 200;
double emin = 0.1;
double dE = (TMath::Log10(fEmax) - TMath::Log10(emin)) /(NENE-1);

double Ebins[NENE];
for(int i=0; i<NENE; i++) Ebins[i] = TMath::Power(10., TMath::Log10(emin) + i * dE);

PDGCodeList::const_iterator nuiter;
PDGCodeList::const_iterator tgtiter;

// loop over all neutrino types generated by the flux driver
for(nuiter = fNuList.begin(); nuiter != fNuList.end(); ++nuiter) {
  int neutrino_pdgc = *nuiter;
  TH1D * pmax_hst = new TH1D("pmax_hst",
           "max interaction probability vs E | geom",NENE-1,Ebins);
  pmax_hst->SetDirectory(0);
```