gSeaGen

Carla Distefano

carla.distefano@lns.infn.it

A bit of history about gSeaGen

gSeaGen: a GENIE-based code to generate neutrinos for underwater/ice telescopes

Project started in 2012 in ANTARES: comparisons with GENHEN, migration to C++, using of modern and maintained neutrino interaction libraries

It was used since the first simulations to generate events in ORCA (GENIE is optimizes in the range 1-100 GeV, GENHEN does not run at energies lower than 10 GeV)

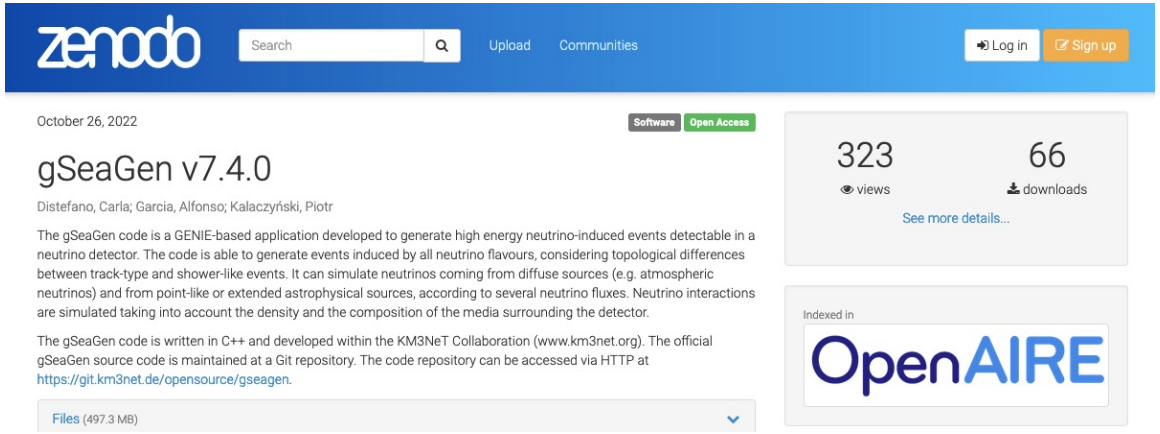
Today it is the reference generator of KM3NeT (ORCA and ARCA) thanks to HEDIS, the high energy extension of GENIE (implemented by Alfonso Garcia and available in GENIE 3.2.0)

gSeaGen is a public code:

Public git repository: https://git.km3net.de/opensource/gseagen

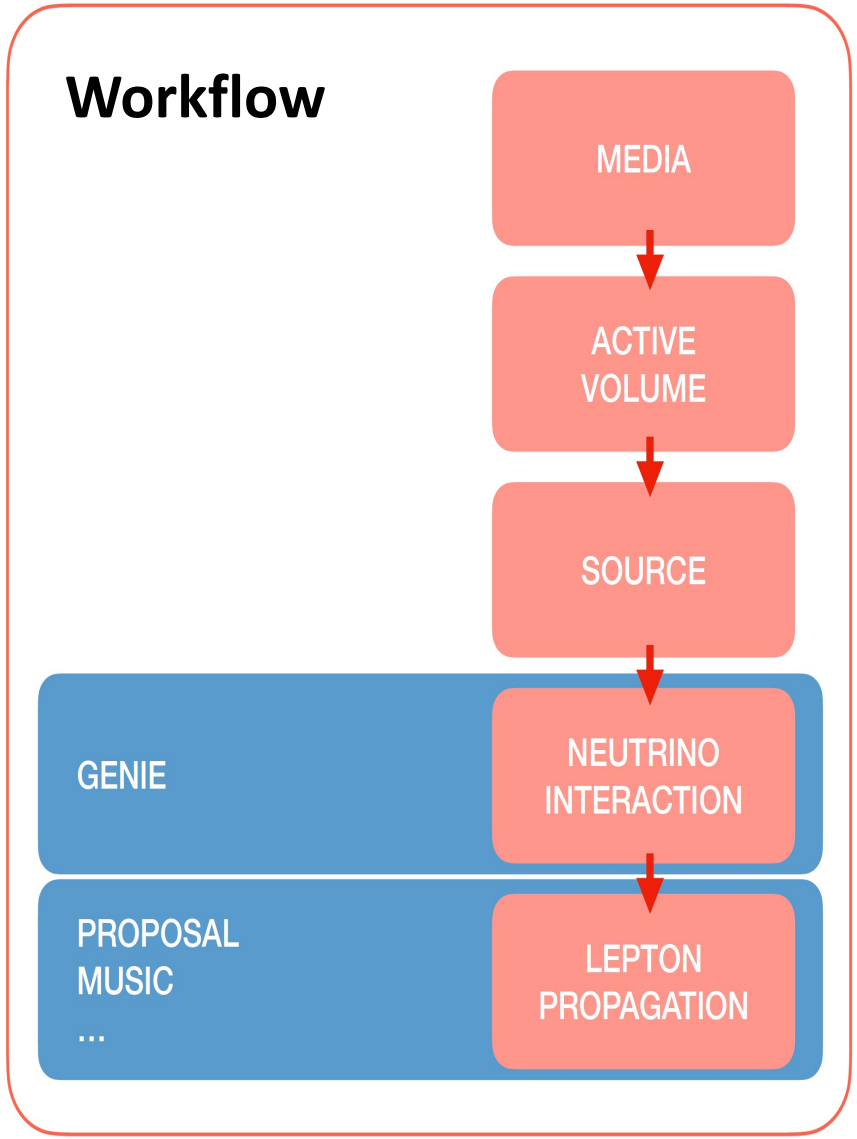
Published on ZENODO

Article on Computer Physics Communications 256, 107477, 2020
(updated in git https://opensource.pages.km3net.de/gseagen/gseagen.pdf)



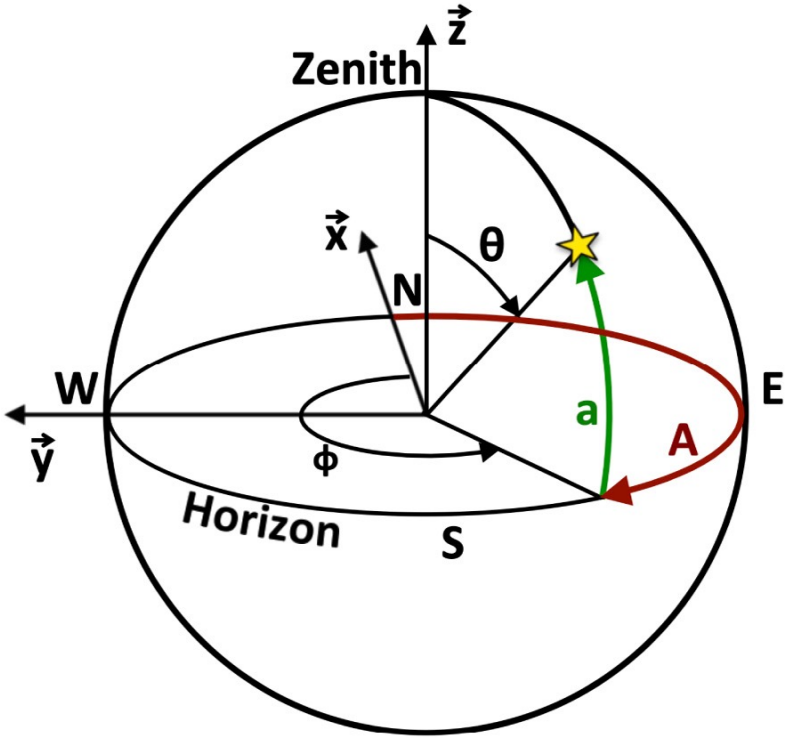
gSeaGen main features

- It simulates neutrino-induced events detectable by an underwater/ice neutrino telescope.
- It simulates all neutrino flavours, taking into account topological differences between track-type and shower-like events.
- It simulates diffuse fluxes, point-like and extended sources
- Minimal compiling requires only GENIE (and its external dependencies)…
- But it is linkable to external codes for the propagation (MUSIC, TAUSIC, TAUOLA, PROPOSAL,..)

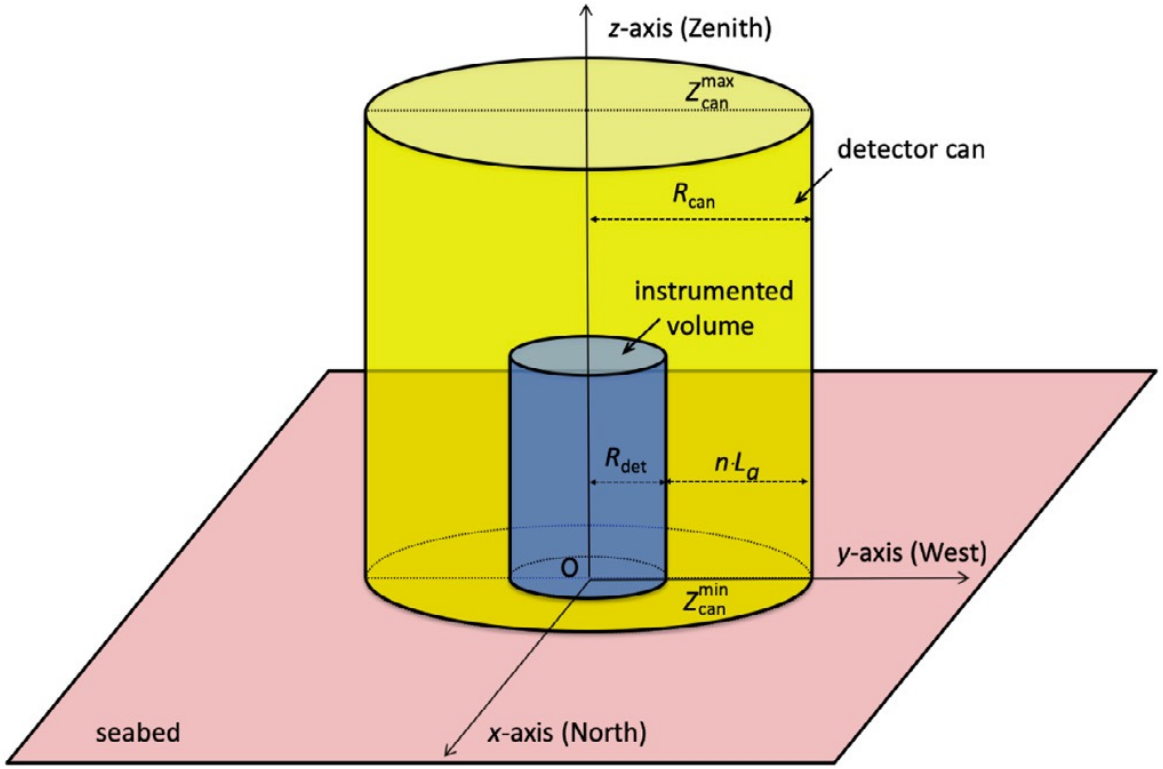


The coordinate system and the detector can

Cartesian system (but UTM system available) with an associated astronomical horizontal system



Detector can: only particles generated inside or reaching the can surface are saved



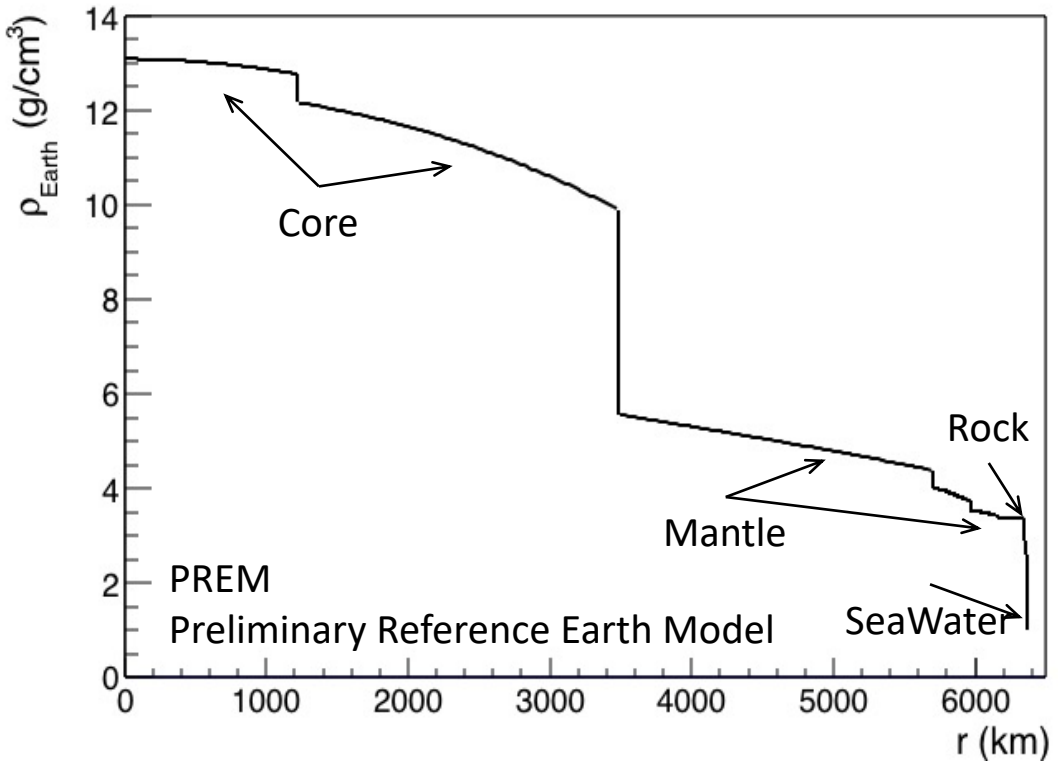
The target media

Four different target media are defined: **SeaWater**, **Rock** (GENIE interaction volume), **Mantle** and **Core** (calculation of P_{Earth}).

Default compositions for target media defined in gSeaGen.

SeaWater			
Element	Percent	Element	Percent
O^{16}	85.84	S^{32}	0.091
H^{1}	10.82	Ca^{40}	0.04
Cl^{35}	1.94	K^{39}	0.04
Na^{23}	1.08	Br^{80}	0.0067
Mg^{24}	0.1292	C^{12}	0.0028
Rock			
Element	Percent	Element	Percent
O^{16}	46.3	Na^{23}	2.36
Si^{28}	28.2	Mg^{24}	2.33
Al^{27}	8.23	K^{39}	2.09
Fe^{56}	5.63	Ti^{48}	0.57
Ca^{40}	4.15	H^{1}	0.14
Mantle			
Element	Percent	Element	Percent
O^{16}	45.22	Fe^{56}	5.97
Mg^{24}	22.83	Al^{27}	2.25
Si^{28}	21.49	Ca^{40}	2.24
Core			
Element	Percent	Element	Percent
Fe^{56}	90.0	Ni^{58}	10.0

Default densities:
Rock: 2.65 g/cm³ and SeaWater: 1.04 g/cm³)
Core and Mantle from the PREM model



The user has the possibility to change the composition for alla the target madia and density for SeaWater and Rock
→ systematics due to medium compositions, simulation of under-ice detectors…

The neutrino interaction volume

The interaction volume is the volume where a neutrino interaction could produce detectable particles.
The size of the interaction volume and the amount of each medium depend on the type of the neutrinos and interactions.

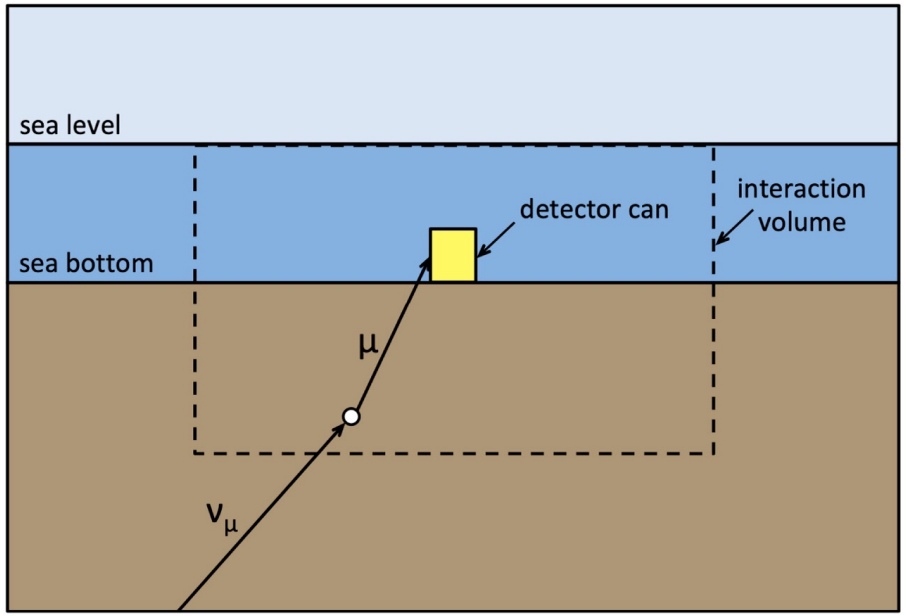
Track-type events
Muons and taus, produced by the neutrino interaction (muon and tau neutrinos in CC), may be detected even if the interaction vertex is outside the can.

The volume is a cylinder made by a layer of **Rock** and a layer of **SeaWater**, surrounding the can. The size is chosen based on the muon/tau maximum range in water and in rock, evaluated at the highest energy of simulated neutrinos.

Shower-like events
In case of electron neutrinos and muon or tau neutrinos interacting only in NC, the interaction produces shower-like events.

Such events may be detected only if they are generated inside the light sensitive volume.

The interaction volume is therefore defined as a cylinder coincident with the detector can and entirely made by **SeaWater**.



The interaction volume is built by the gSeaGen geometry driver (**GSeaGeometry**), using TGeoManager and it is the input geometry for GENIE (**GMCJDriver**)

The detector can is not part of the geometry but only a reference for the propagation

Generation of the arrival neutrinos

gSeaGen flux drivers (based on GENIE GFluxI)

Base class:

GSeaAtmoFlux (class GSeaAtmoFlux: public GFluxI)
- generates the neutrino direction (diffuse fluxes but also astrophysical point-like and extended sources)
- generates the neutrino vertex
- generates the neutrino energy according to power-law spectrum
- calculates the events weights

Real implementation:

Definition of the physical fluxes to weight the events

GSeaRealAtmoFlux (class GSeaRealAtmoFlux: public GSeaAtmoFlux): BARTOL, FLUKA, HONDA, ROOT FUNC
GSeaRealPointFlux (class GSeaRealPointFlux: public GSeaAtmoFlux): ROOT FUNC

Classes instantiated at run time and used to initialize the GENIE GMCJDriver

Generation of the neutrino direction

Diffuse fluxes: arrival direction (θ,ϕ) generated with an isotropic distribution → neutrino direction cosines

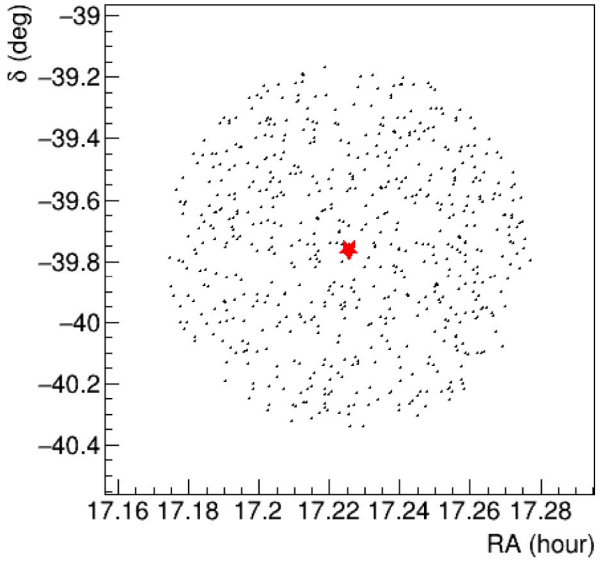
$$\begin{cases} v_x &= -\sin(\theta)\cdot\cos(\phi) \\ v_y &= -\sin(\theta)\cdot\sin(\phi) \\ v_z &= -\cos(\theta) \end{cases}$$

Point-like sources: arrival direction generated from source coordinates

$$\begin{aligned} \sin(a) &= \sin(\delta)\cdot\sin(\lambda)+\cos(\delta)\cdot\cos(\lambda)\cdot\cos(\mathrm{HA}) \\ \tan(A) &= \frac{\sin(\mathrm{HA})}{\cos(\mathrm{HA})\cdot\sin(\lambda)-\tan(\delta)\cdot\cos(\lambda)}+180^\circ \end{aligned}$$

$$\begin{cases} \theta &= 90^\circ - a \\ \phi &= 360^\circ - A \end{cases}$$

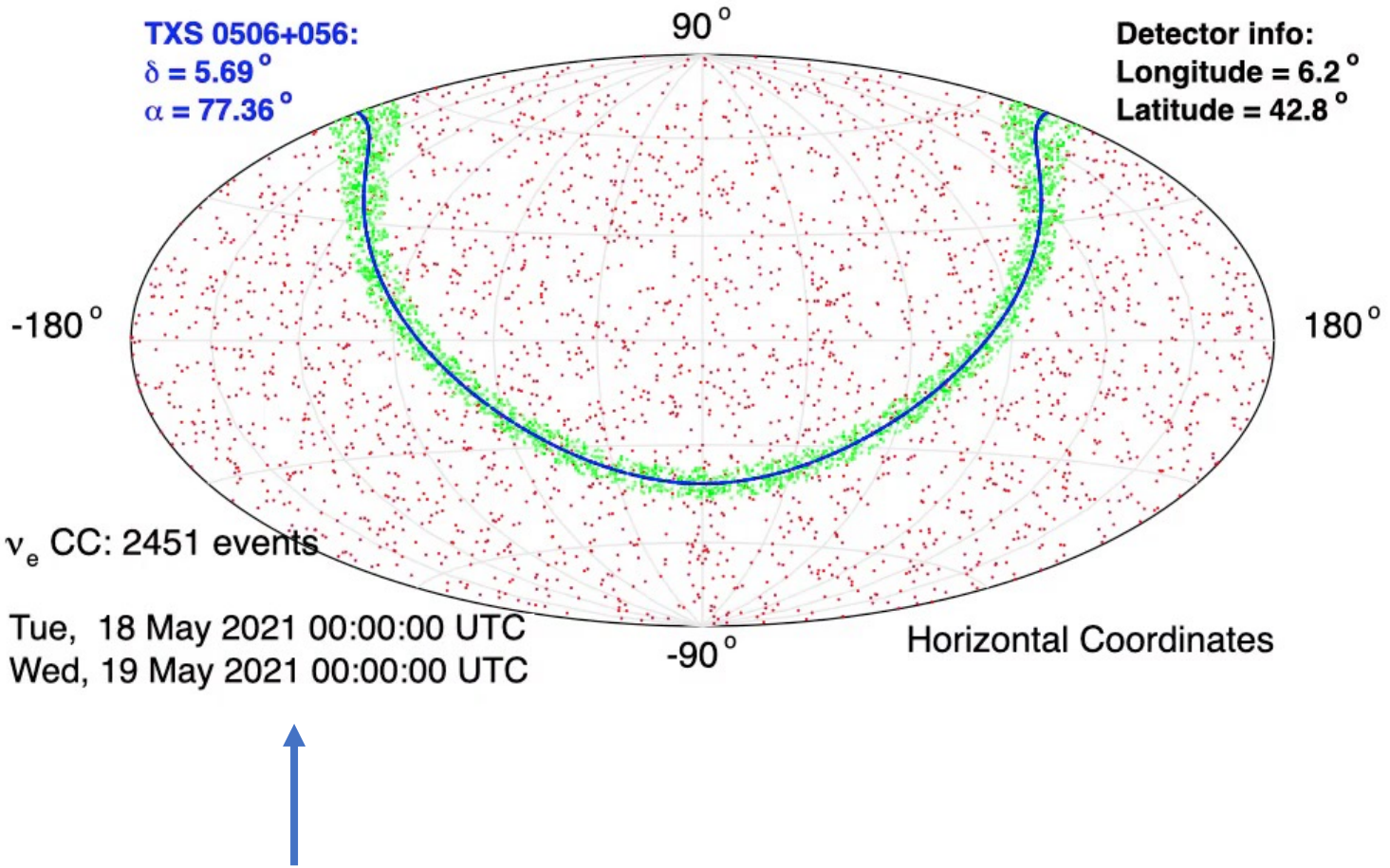
Extended sources: arrival direction generated in a circle centred in the source coordinates



All astronomical algorithms needed in the simulation of astrophysical sources have been implemented in a gSeaGen internal library (**GAstro**)

Generation of the neutrino direction

Alfonso Garcia



Could the point-mode be useful in DUNE to simulate SN explosions?

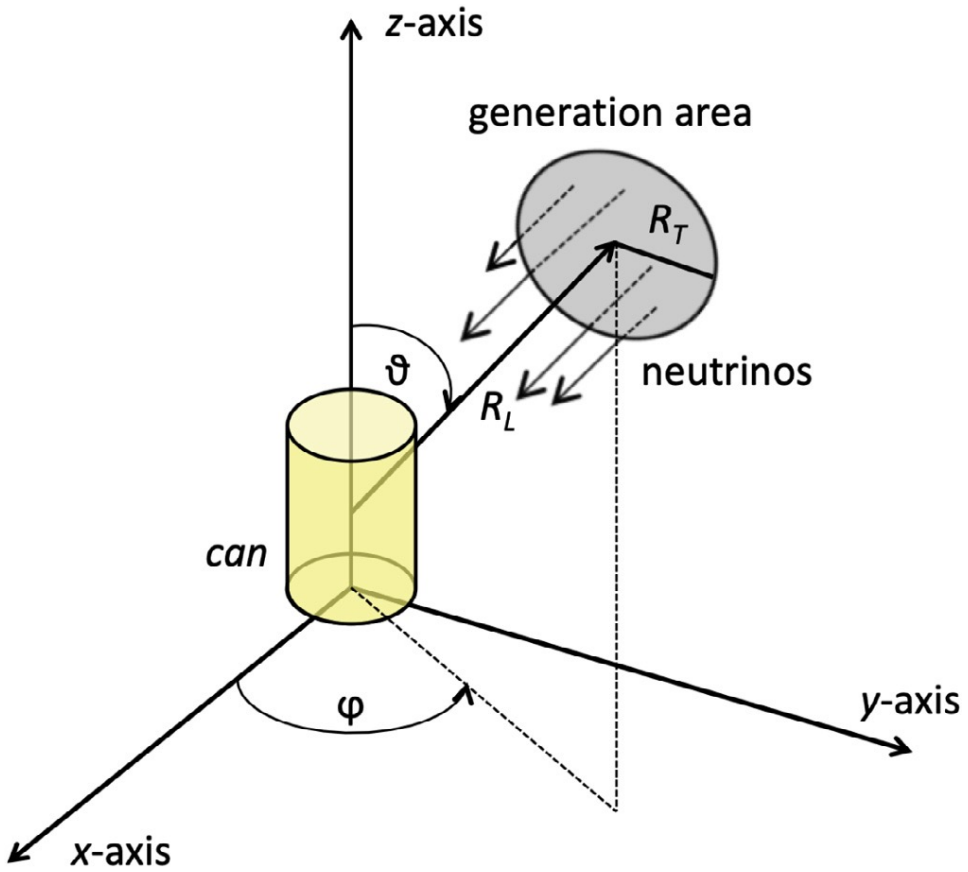
We are exploring the use of the VLE GENIE tune (GVLE18_01a_00_000) for KM3NeT detectors

Event time simulation in a given time interval is available for any kind of fluxes

Generation of the neutrino vertex

The neutrino vertex is generated uniformly on a surface (the generation area)

The generation area is tangent to a sphere of radius RL, centred at the detector can centre.



The radius RL is equal to the diagonal of the interaction volume so that the generation area is always outside this volume.

The radius of the generation area is:

$$R_T = D/2 + 100 \text{ m}$$

where D is the can diagonal

But it is possible:

- Set D equal to the interaction volume diagonal
- Set RT to a given value
- **Set the generation area as the projection of the can on to the plane perpendicular to the neutrino direction.**

Generation of the neutrino energy

Neutrinos are generated with energy following a power-law spectrum E^{-X}, where X is the spectral index input by the user

Event weights are used to weight the events according to the physical spectrum

$$w_{evt} = f(E,\theta,\phi)^{phys} / f(E,\theta,\phi)^{gen} = w_{gen} \cdot f(E,\theta,\phi)^{phys}$$

Event weight

$$w_{\text{gen}} = \frac{I_E \cdot I_\theta \cdot T_{\text{gen}} \cdot A_{\text{gen}} \cdot N_\nu \cdot E^X \cdot P_{\text{scale}} \cdot P_{\text{Earth}}(E, \cos\theta)}{N_{\text{Tot}}}$$

Generation weight

T_{gen} = 1 sec

P_{scale}: GENIE probability scale

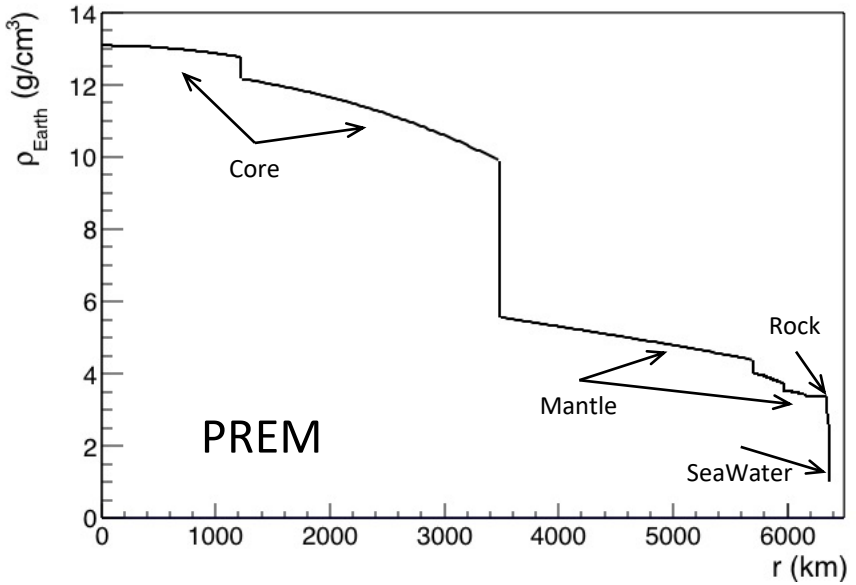
It takes into account the GENIE interaction weight scheme:

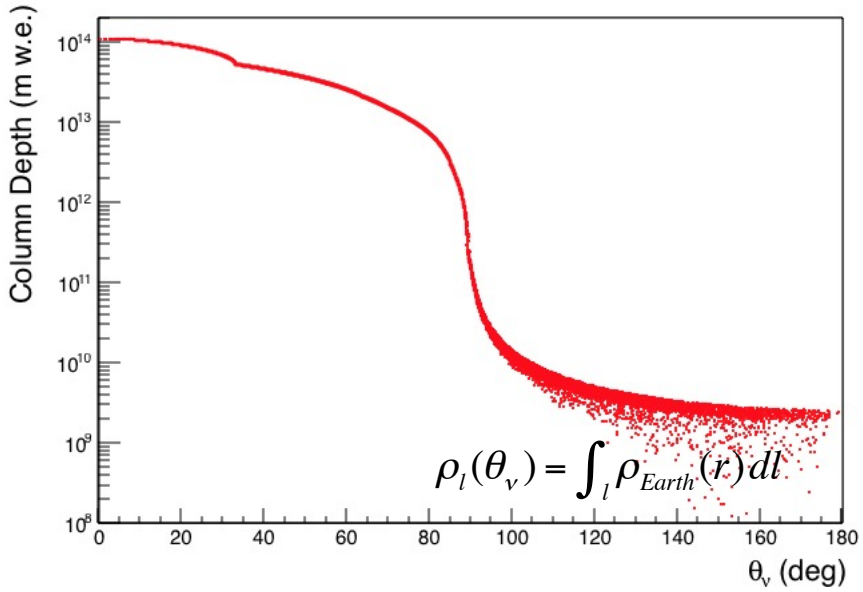
In GENIE 3.2.0 it is possible to enforce the interaction (i.e. all the generated neutrinos interact)

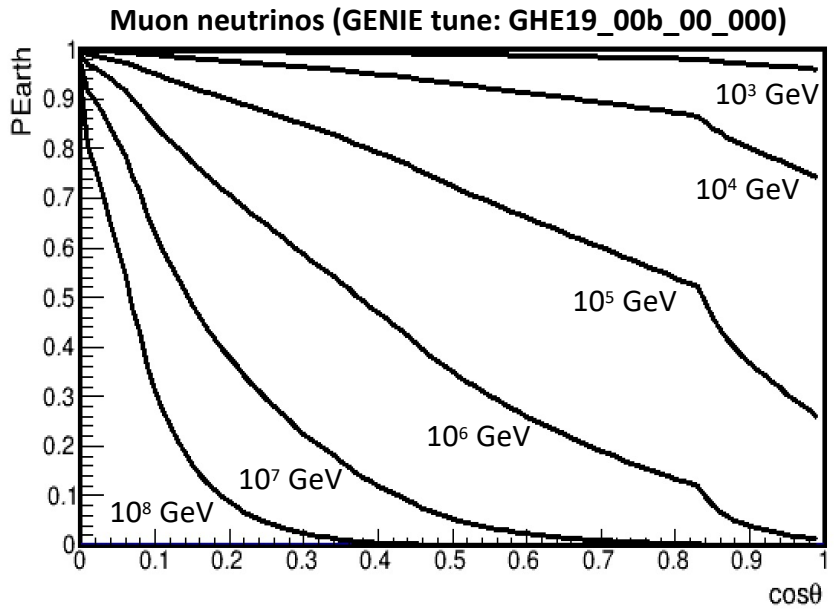
Transmission probability through the Earth

$$P_{Earth}(E_\nu, \cos\theta_\nu) = e^{-N_A \cdot \sigma_l(E_\nu,\theta_\nu) \cdot \rho_l(\theta_\nu)}$$

where $\sigma_l(E_\nu,\theta_\nu)$ is the average cross section per nucleon along the neutrino path $\rho_l(\theta_\nu)$ (i.e. taking into account the different layer compositions and density profile).







Simulation of the neutrino interaction

It is simulated with the standard GENIE **GMCJDriver**, initialized with:

- geometry file created by GSeaGeometry
- flux driver GSeaRealAtmoFlux (or GSeaRealPointFlux)

Code implemented in the **GGenerateEventGenie** (class GGenerateEventGenie: public GGenerateEvent)

GGenerateEvent: muon and tau propagation

- if the interaction vertex is inside the can, the event is saved in the output
- it not, muons and taus are propagated and stored in the output only if they reach the can surface

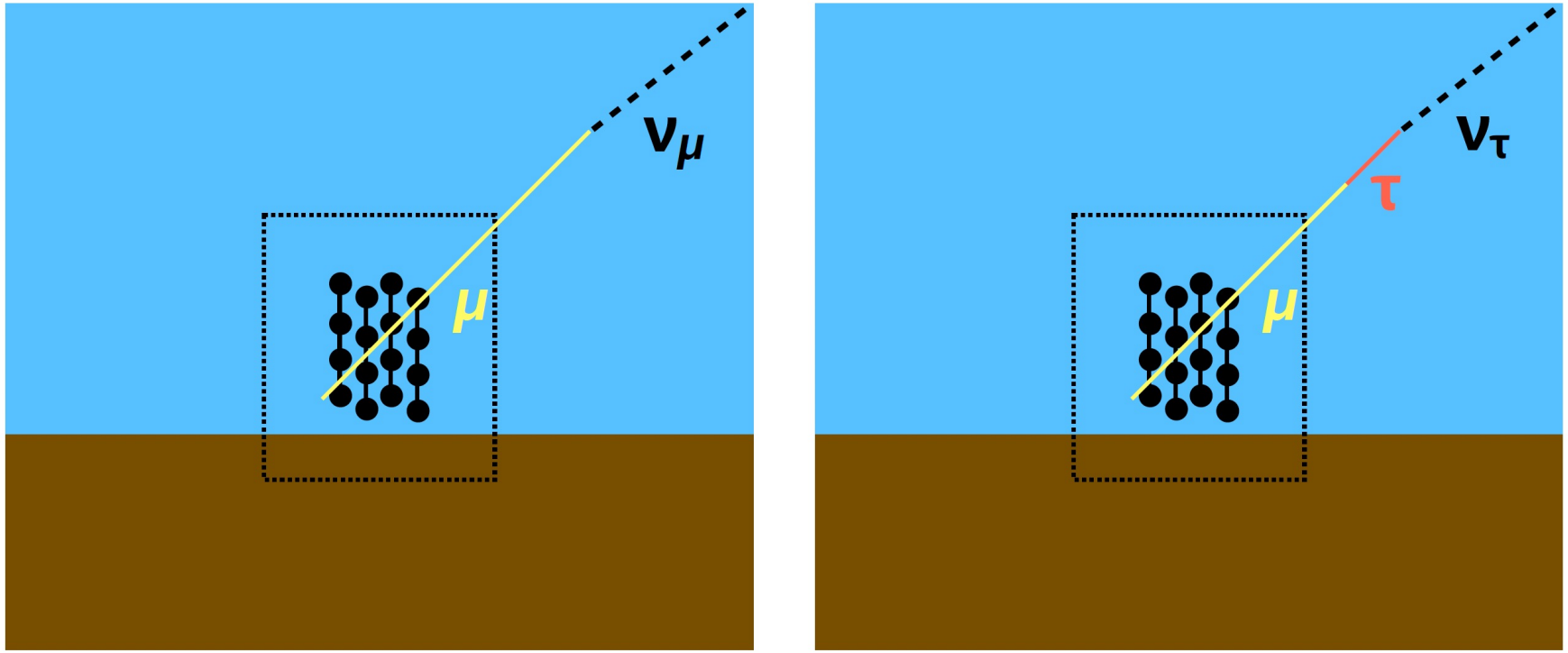
The possibility of implementing new generation classes is foreseen.

Output file (ROOT):

- native format using gSeaGen event class
- standard GENIE format (only at the interaction level, no leptons propagated up to the can)

Lepton propagation

- Muon propagation:
  - PROPOSAL linkable with 5.0.0 and later (tested up to 6.1.5). Recommented
  - MUSIC.
  - JPP (internal KM3NeT).
  - PropaMuon (native from gSeaGen). Not optimazed above 1 TeV
- Tau propagation and decay done with TAUSIC+TAUOLA: Planned implementation of PROPOSAL



Running gSeaGen

Git repository:
https://git.km3net.de/opensource/gseagen

Have a look at the README file: documentation, info about installing and usage, and examples

Set GSEAGEN pointing the the main directory

configure [options]
make

gSeaNuEvGen [options]

gSeaNuEvGen -n 1E6 -e 1,500 -t -1.,1. -f "dat/grn-ally-20-12-solmin.d[14]" --event-generator-list CC ….

gSeaNuEvGen -h → to see all available options

Using gSeaGen libraries

Organized in libraries

$GSEAGEN/lib/libGOutputWriters.so
$GSEAGEN /lib/libGSeaEvent.so
$GSEAGEN /lib/libGSeaNuDrivers.so
$GSEAGEN /lib/libPropaMuon.so

Include files are in

$GSEAGEN/src/OutputWriters
$GSEAGEN/src/SeaEvent
$GSEAGEN/src/SeaNuDrivers
$GSEAGEN/src/PropaMuon

1) Try to use the flux drivers in the DUNE code
2) Try to write a new implementation of the GGenerateEvent for the lepton propagation

GSeaNuDrivers: GAstro, GGenerateEvent, GGenerateEventGenie, GPEarth, GSeaAtmoFlux, GSeaGeometry, GSeaRealAtmoFlux, GSeaRealPointFlux

GSeaEvent: GBinParam