

# gSeaGen: the KM3NeT GENIE-based code for neutrino telescopes

Git release: v7.4.3-test12

Carla Distefano  
carla.distefano@lns.infn.it

May 6, 2024

## Abstract

The gSeaGen code is a GENIE-based application developed to efficiently generate high statistics samples of events, induced by neutrino interactions, detectable in a neutrino telescope. The gSeaGen code is able to generate events induced by all neutrino flavours, considering topological differences between track-type and shower-like events. Neutrino interactions are simulated taking into account the density and the composition of the media surrounding the detector. The main features of gSeaGen are presented together with some examples of its application within the KM3NeT project.

**Keywords:** Astroparticle Physics, High Energy Neutrinos, Monte Carlo Event Generator, Neutrino Telescopes, Neutrino Oscillations, KM3NeT, GENIE

## 1 Introduction

The gSeaGen code [1] simulates events induced by all flavours of neutrinos and detectable by a neutrino telescope, using GENIE [2] to simulate the neutrino interactions. In particular, it allows electron, muon and tau neutrino events to be generated. Depending on the steering parameters it can produce events from multiple (anti)neutrino flavours within one execution of the program (e.g.  $\nu_\mu + \bar{\nu}_\mu$ ); simulate charged current (CC) and/or neutral current (NC) interactions; and simulate neutrinos arriving isotropically or from point or extended sources. The first versions of gSeaGen were limited to generating neutrinos up to 5 TeV due to GENIE's range validity. The new modules HEDIS and GLRES [3], implemented in GENIE version 3.2.0, compute neutrino differential cross sections up to 1 EeV. This allowed to extend the use of gSeaGen to generate neutrino interactions of any flavour at very high energies [4]. The gSeaGen code, originally developed as a generator of neutrino events, can also be used to propagate atmospheric muons generated with CORSIKA [5] from the sea level up to the detector surface [6].

The code was developed for KM3NeT [7]. Although applicable both to underwater and under-ice neutrino detectors, the code terminology, as well as the description presented in this paper, follow those of underwater detectors.

## 2 The coordinate system

The coordinates in gSeaGen are Cartesian with the positive  $z$ -axis pointing to the local zenith, the positive  $x$ -axis pointing north and the positive  $y$ -axis as needed to make a right-handed coordinate system. The origin is defined as being the water/rock boundary. In the event that the origin of the gSeaGen coordinate system does not coincide with that of the detector system, it is possible to carry out the translation with the running option `-origin` (see Sec. 10).

To define astrophysical source positions, an associated horizontal coordinate system is defined, with the azimuth measured eastward from the North, as shown in Fig.1. The relation

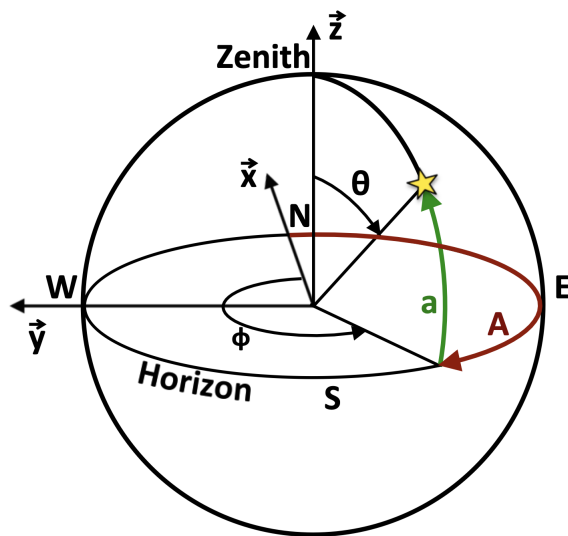


Figure 1: The Cartesian and the associated astronomical horizontal systems as defined in gSeaGen. The relation between the source altitude  $a$  and azimuth  $A$  coordinates and the angles  $\theta$  and  $\phi$ , defining the arrival direction of the emitted neutrinos is shown.

between the source altitude  $a$  and azimuth  $A$  coordinates and the angles  $\theta$  and  $\phi$ , defining the arrival direction of the emitted neutrinos, is then:

$$\begin{cases} \theta = 90^\circ - a \\ \phi = 360^\circ - A \end{cases} \quad (1)$$

In large telescopes such as those of KM3NeT, the positions of the detection units are often defined in the UTM coordinate system and the position and direction of the particles tracked inside the detector must also be expressed with respect to it. The user can activate the UTM system in gSeaGen with the running option `-utm` (see Sec. 10). The UTM coordinate system has the x-axis pointing to the UTM Easting ( $\phi_{\text{UTM}} = 0^\circ$ ) and the y-axis pointing to the UTM Northing ( $\phi_{\text{UTM}} = 90^\circ$ ), with the  $\phi_{\text{UTM}}$  angle increasing in the counter-clockwise sense in the x-y-plane of the detector. Besides, the UTM Northing points slightly towards the West geographic direction and it is rotated with respect to the geographic North by the convergence angle  $\alpha$  calculated from the geographical coordinates. Therefore directions in the gSeaGen UTM coordinate system are defined by the angles:

$$\begin{cases} \theta_{\text{UTM}} = \theta \\ \phi_{\text{UTM}} = \phi + 90^\circ + \alpha \end{cases} . \quad (2)$$

### 3 The detector active volume

The detector active volume is defined as a cylinder of water, the *can*, having the axis along the the z-axis and surrounding the instrumented volume as sketched in Fig. 2. The user can set the can defining the z-level of the can bottom  $Z_{\text{can}}^{\text{min}}$  ( $Z_{\text{can}}^{\text{min}} \geq 0$ ), the z-level of the can top  $Z_{\text{can}}^{\text{max}}$  ( $Z_{\text{can}}^{\text{max}} > Z_{\text{can}}^{\text{min}}$ ) and the can radius  $R_{\text{can}}$ .

gSeaGen can also generate a can by automatically reading a file containing the detector geometry. In this case the code defines a cylinder representing the detector's instrumented volume. The can is then built by enlarging the instrumented volume by  $n$  times the light absorption length,  $L_a$ , input by the user. Presently only ANTARES and KM3NeT geometry files are accepted by gSeaGen.

When the bottom of the can lies on the sea-bed  $Z_{\text{can}}^{\text{min}} = 0$ , it is possible to extend the interaction volume below the can by adding a layer of rock with the option `-bedrock` (see Sec. 10). The *bedrock* layer is considered part of the can in the event generation.

The can represents the detector horizon to the light emitted by the neutrino-induced particles. All the events leading to Cherenkov radiation in the can, i.e. events interacting inside the can or producing a muon reaching the can surface, are stored in the output file. The particle tracking inside the can and the Cherenkov light emission are performed in the next steps of the simulation chain by independent programs to study the detector response.

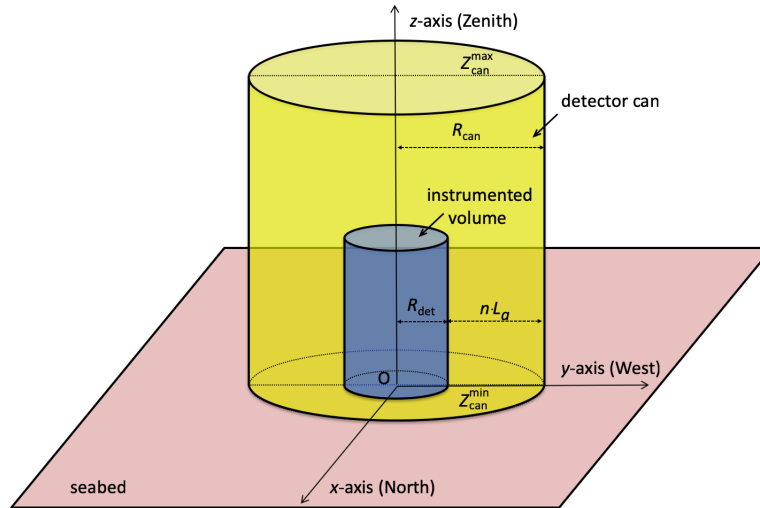


Figure 2: Definition of the detector can.

Table 1: Default compositions for target media defined in gSeaGen.

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

## 4 The target media

Four different neutrino target media are defined: *SeaWater*, *Rock*, *Mantle* and *Core*. The first two are used to define the interaction volume (see Sec. 5), while the others enter in the calculation of the transmission probability of neutrinos through the Earth (see Sec. 7). The default compositions for each target medium are listed in Tab. 1. The densities for SeaWater and Rock are set to  $\rho_s = 1.04 \text{ g/cm}^3$  and  $\rho_r = 2.65 \text{ g/cm}^3$  respectively. For Mantle and Core media, the density profile provided by the Preliminary Reference Earth Model (PREM) [8] is used.

The user can change the composition of all target media and the density for SeaWater and Rock, e.g. to study the systematic uncertainties due to medium compositions and also consider under-ice detectors.

## 5 The neutrino interaction volume

The *interaction volume* is the volume surrounding the detector in which a neutrino interaction can produce detectable particles. It is defined at runtime and is used to simulate the interaction of all generated neutrinos. The interaction volume is built using the TGeoManager ROOT class [9] and is used to configure the GENIE geometry driver. The interaction volume is defined as a cylinder surrounding the can and is made of sea water and rock, whose

composition and density are defined using the target media SeaWater and Rock, respectively.

The size and nature of the interaction volume depends on the topology of the generated neutrino events. In the case of electron neutrino CC interactions and all neutrino flavour NC interactions, a particle shower is produced. Such events, here referred to as *shower-like events*, may be detected only if the neutrino interacts inside the light-sensitive volume. In this case the interaction volume is defined as a cylinder coincident with the can and entirely made of seawater.

Muons produced in the neutrino interaction may be detected also if the interaction vertex is outside the can. This happens in the case of muon neutrinos interacting in CC, for which the muons are the primary leptons produced by the interaction. For this kind of event, referred to as a *track-like event*, the interaction volume is built taking into account the muon maximum range in water and rock, evaluated at the highest simulated neutrino energy. The interaction volume is then a cylinder made of a layer of rock and a layer of seawater, with the separation surface coincident with the sea bottom (see Fig. 3). The volume radius is equal to the can radius plus the muon maximum range in water. The layer of rock has a height equal to the muon maximum range in this medium. If up-going events are not simulated the rock layer height is set to zero. The layer of sea water has a height equal to the can height plus the maximum muon range in water and it can not exceed the detector site depth. If down-going events are not simulated, the sea water layer height is set equal to that of the can.

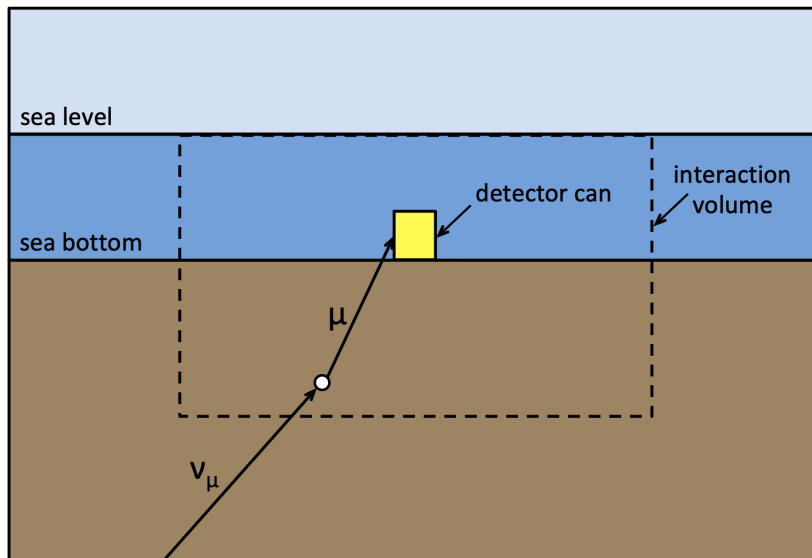


Figure 3: Schematic view of the interaction volume for track-like events.

Track-like events are also generated by CC-interacting tau neutrinos. The definition of the interaction volume depends in this case on the generated energy regime. Using standard GENIE tunes (e.g. G18\_02a\_00\_000) the maximum valid neutrino energy is 5 TeV. At these energies tau neutrinos do not induce long-track tau events. However, muons could result as product of the tau decay. For this reason when CC tau neutrino interaction events are simulated, the interaction volume is defined on the basis of the muon range as in the case of CC interacting muon neutrinos. In case of GENIE tunes allowing the simulation of neutrinos up to 1 EeV (e.g. GHE19\_00b\_00\_000), the calculation of the interaction volume is done

accounting for the decay length and products of the outgoing lepton as specified with option `-tau` (see Sec. 10)

The GLRES module implemented in GENIE allows to simulate Glashow resonant events [10]. Most of the studies of this interaction channel have been focused on trying to observe fully or partially contained showers in the detector coming from the decay of a W boson into hadrons [11]. To simulate these events the interaction volume can be restricted to the convex hull of the detector as hadrons will travel only a few meters. However, when the W boson decays into a muon or a tau the interaction volume must be significantly larger. gSeaGen can compute different interaction volumes depending on the decay channel as specified with option `-tau` (see Sec. 10) [4].

## 6 Generation of neutrino events

A flux driver class implementing the standard GENIE GFluxI class [12] has been developed within gSeaGen and used to configure the GENIE event generation driver. The interface generates the neutrino energy and the track information needed to simulate the interaction.

The gSeaGen flux driver is able to generate neutrinos coming from diffuse sources (e.g. atmospheric neutrinos) or from point-like and extended astrophysical sources. The neutrino energy is always generated according to a power law spectrum. However, the code provides for each event a weight that can be used to re-weight the generated neutrino events to the chosen astrophysical or atmospheric neutrino spectrum.

The neutrino generation algorithm implemented in the gSeaGen flux driver is described in this section.

### 6.1 Generation of the neutrino time

The time at which the neutrino reaches the detector is by default always set to zero for all the generated events. However, it is possible to randomly generate the event time within a user-defined time interval using with the running option `-time` (see Sec. 10).

### 6.2 Generation of the neutrino direction

The generation of the neutrino direction depends on the kind of simulated neutrino flux. In case of diffuse fluxes, the neutrino arrival direction, defined by the  $\theta$  and  $\phi$  angles, is randomly extracted according to an isotropic distribution and the neutrino direction cosines  $(v_x, v_y, v_z)$  are computed according to:

$$\begin{cases} v_x &= -\sin(\theta) \cdot \cos(\phi) \\ v_y &= -\sin(\theta) \cdot \sin(\phi) \\ v_z &= -\cos(\theta) \end{cases} \quad (3)$$

The simulation of neutrinos coming from an astrophysical point-like source is activated using the option `-point` (see Sec. 10 and 11.5). In this case, the neutrino direction is generated along the apparent source trajectory in the sky due to the Earth's rotation. The trajectory is calculated using the source equatorial coordinates (declination  $\delta$  and right

ascension RA) and the geographic coordinates of the detector, provided by the user (see Sec. 11.5).

The local sidereal time (LST) is calculated at the neutrino event time if a generation time interval is defined, otherwise it is uniformly extracted between 0 and 24 hours. The corresponding source hour angle is calculated as  $HA = LST - RA$ . The code then computes the horizontal coordinates of the source, altitude  $a$  and azimuth  $A$ , according to

$$\begin{aligned} \sin(a) &= \sin(\delta) \cdot \sin(\lambda) + \cos(\delta) \cdot \cos(\lambda) \cdot \cos(HA) \\ \tan(A) &= \frac{\sin(HA)}{\cos(HA) \cdot \sin(\lambda) - \tan(\delta) \cdot \cos(\lambda)} + 180^\circ, \end{aligned} \quad (4)$$

where  $\lambda$  is the detector latitude<sup>1</sup>. Source position may be given also in galactic coordinates and the transformation to the equatorial reference system is done by gSeaGen.

As an example, neutrinos coming from the supernova remnant RX J1713.7-3946 direction (RA:  $17^{\text{h}} 13^{\text{m}} 33.1^{\text{s}}$ ,  $\delta$ :  $-39^\circ 45' 44''$ ) have been simulated. The generated horizontal coordinates at the KM3NeT-ARCA site (latitude: N  $36^\circ 17' 48.34''$ , longitude: E  $15^\circ 58' 42.25''$ ) are plotted in Fig. 4.

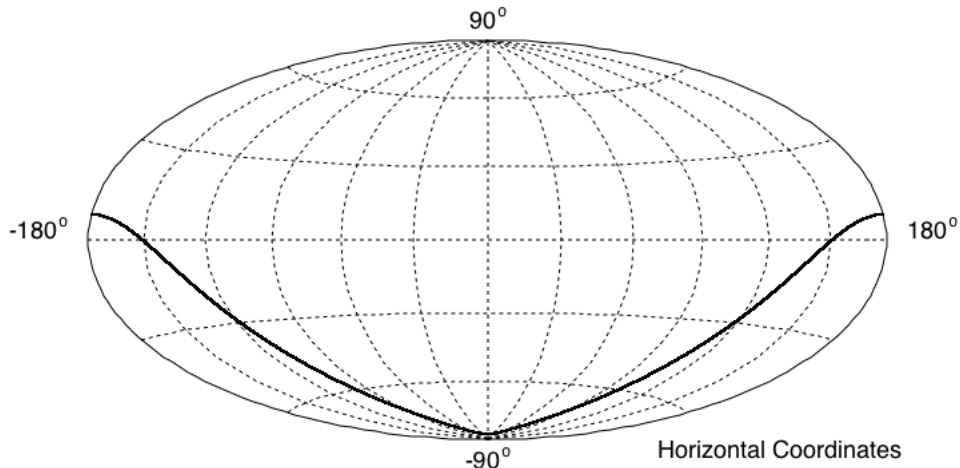


Figure 4: Horizontal coordinates of neutrinos coming from the supernova remnant RX J1713.7-3946 direction generated with LST between 0 and 24 hours at the KM3NeT-ARCA site.

Once the source position in the sky is generated, the neutrino arrival direction angles are computed using Eq. 1 and then transformed into the neutrino direction cosines with Eq. 3.

Finally, gSeaGen gives the possibility to simulate neutrinos coming from an extended astrophysical source with a small angular radius  $R_s$  ( $R_s \lesssim 10^\circ$ ). The generation proceeds as in the case of a point-like source with astronomical coordinates coincident with the extended source centre. Then, the neutrino directions are randomly generated according to a uniform distribution within a disk of radius  $R_s$ .

<sup>1</sup>The difference between geographical and geocentric latitudes is not taken into account since the corrections in the coordinates are negligible with respect to the angular resolutions of neutrino telescopes of tenths of a degree. Corrections for parallax are also neglected since they are irrelevant for the distant astrophysical neutrino sources [13].

For example, the arrival direction of neutrinos coming from RX J1713.7-3946 has been randomised in a circle of  $R_s = 0.6^\circ$ , corresponding to the source extension. The equatorial coordinates of the generated neutrinos are plotted in Fig. 5.

Sources of arbitrary topology can be simulated by re-weighting the flux within the generated disk (see Sec. 7).

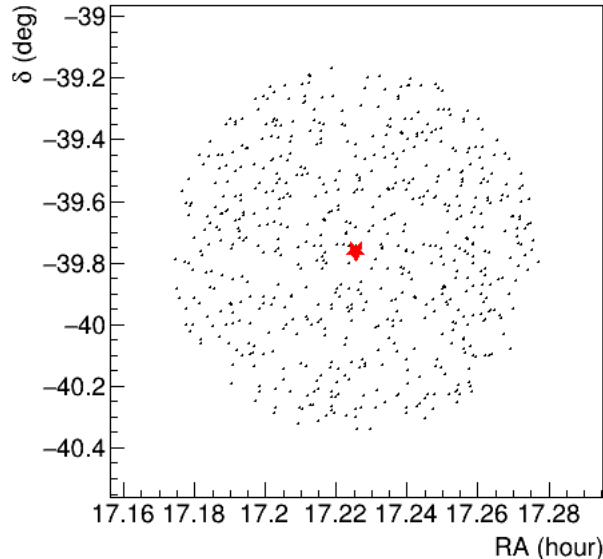


Figure 5: Equatorial coordinates of neutrinos generated from the extended source RX J1713.7-3946 with  $R_s = 0.6^\circ$ . The red star represents the coordinates of the source.

All astronomical algorithms needed in the simulation of astrophysical sources have been implemented in a gSeaGen internal library based on Ref. [13].

### 6.3 Generation of the neutrino vertex

As in the standard GENIE neutrino flux drivers, the neutrino vertex is generated on a circular surface with radius  $R_T$ . The *generation area* is tangent to a sphere of radius  $R_L$ , centred at the detector can centre (see Fig. 6). The radius  $R_L$  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:

$$R_T = D/2 + 100 \text{ m}, \quad (5)$$

where  $D$  is the detector diagonal. Using option `-rt`, the user has the possibility to modify the radius of the generation area setting  $D$  equal to the interaction volume diagonal or  $R_T$  to a given value (see Sec. 10).

A circular generation area with a fixed radius disfavours particular directions when the detector is strongly asymmetric. Thus, an alternative method has been developed to spread neutrino vertices within the surface defined by the projection of the can on to the plane perpendicular to the neutrino direction. This increases the number of events at the can by



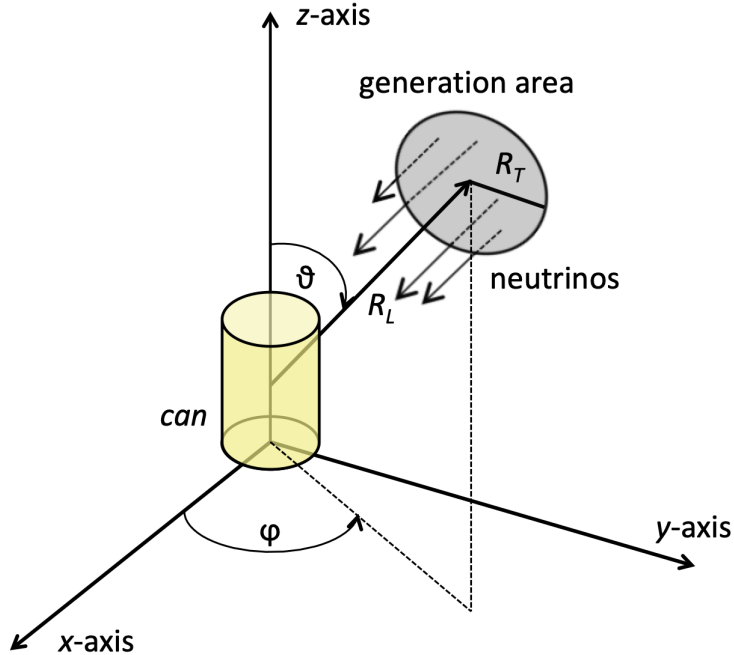


Figure 6: Definition of the generation area for the neutrino vertices. For a given neutrino direction, the generation area is a circle, with radius  $R_T$ , which is tangent to a sphere of radius  $R_L$  centred at the detector can centre.  $R_T$  and  $R_L$  are chosen such that the generation area is always outside the interaction volume and so that, for every given direction, the “shadow” of the generation surface covers the entire detector can. See text for more details.

30% or more, depending on the can shape. When this method is activated (with option `-rt`), the generation area is not constant but depends on the neutrino direction.

## 6.4 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. gSeaGen gives the possibility to bin the energy spectrum in equal divisions in  $\log_{10}(E)$ . The number  $N^i$  of incoming neutrinos in each bin, ranging from  $E_{\min}^i$  to  $E_{\max}^i$ , is given by:

$$N^i = N_{\text{Tot}} \cdot \frac{\int_{E_{\min}^i}^{E_{\max}^i} E^{-X} dE}{\int_{E_{\min}}^{E_{\max}} E^{-X} dE}, \quad (6)$$

where  $N_{\text{Tot}}$  is the total number of neutrinos in the whole energy range from  $E_{\min}$  to  $E_{\max}$ . When muon inducing events are simulated, the interaction volume is scaled according to the maximum muon range in rock and water, calculated at the highest neutrino energy in the bin (see Sec. 5).

Since interaction probabilities are very small numbers, GENIE scales up the probabilities to reduce the number of trials. This is reflected in the weights of interacting neutrinos as discussed in Sec. 7. The probability scale is defined as the maximum possible total interaction probability, i.e. the probability at the maximum neutrino energy (corresponding

to the maximum cross section) and for the maximum possible neutrino path length in the interaction volume [2]. When the energy range is binned, the interaction probability is scaled up at each bin, increasing the statistics for lower neutrino energies. In this way, the simulation can span different orders of magnitude in energy, as usually required in the simulation of neutrino telescopes.

## 6.5 Neutrino interaction and events at the detector

Once the neutrino energy, position and direction are generated, the neutrino interaction is simulated using the GENIE event generation driver class GMCJDriver [2]. If the neutrino interacts inside the can, all particles in the final state are stored in the output file. In the case of track-type events, the interaction could take place outside the can and still produce a visible lepton in the detector. In this case, all induced particles are rejected except the muons and taus produced at a distance from the can lower than their maximum range in water. These particles are then propagated and stored in the output file if they reach the can surface. By default, gSeaGen propagates muons using its internal muon propagator code PropaMuon, described in Appendix A. Alternatively, it is also possible to use external propagators such as MUSIC [14] and PROPOSAL [15] (recommended), if enabled at compile-time.

No internal propagator is available for tau. When using low energy GENIE tunes (e.g. G18\_02a\_00.000), outgoing tau from CC interactions may be decayed on the spot by GENIE itself, which uses PYTHIA6 [16] to compute the kinematics of the decay products. Tau decay is activated by modifying the corresponding flags in the GENIE configuration file `$GENIE/genie_xsec/CommonDecay.xml` (see [12] for major details). This method is a good approximation at energies below 1 TeV but is not valid at higher energies. An external tau propagator is then required when using modules HEDIS and GLRES. Currently gSeaGen may be linked to TAUSIC [14] and TAUOLA [17] to propagate taus and computes the kinematics of the tau decay products. If taus decay during the propagation, gSeaGen propagates any outgoing muons up the detector can.

## 7 Computation of the event weight

During the simulation, each event is assigned a weight  $w_{\text{evt}}$  in order to shape the generation to the expected neutrino flux. The event weight is the product of the generation weight  $w_{\text{gen}}$  and of the value of the neutrino energy spectrum  $f(E, \theta, \phi)$ :

$$w_{\text{evt}} = w_{\text{gen}} \cdot f(E, \theta, \phi). \quad (7)$$

The generation weight is:

$$w_{\text{gen}} = \frac{I_E \cdot I_\theta \cdot T_{\text{gen}} \cdot A_{\text{gen}} \cdot N_\nu \cdot E^X \cdot P_{\text{scale}} \cdot P_{\text{Earth}}(E, \cos \theta)}{N_{\text{Tot}}}, \quad (8)$$

where:

- $I_E$  is the energy phase space factor, equal to the generation spectrum  $E^{-X}$  integrated over the whole simulated neutrino energy range.

- $I_\theta$  is the angular phase space factor, depending on the type of neutrino source:

$$I_\theta = \begin{cases} 2\pi \cdot ((\cos \theta)_{\max} - (\cos \theta)_{\min}) & \text{for diffuse flux (sr)} \\ 1 & \text{for point/extended sources} \\ & \text{(dimensionless)} \end{cases} \quad (9)$$

- $T_{\text{gen}}$  is the generation time set to one second, also even when a time interval is defined. Weighing the events with the  $w_{\text{evt}}$ , the rate of events from the expected neutrino flux is then obtained.
- $A_{\text{gen}}$  is the area of the generation surface expressed in  $\text{m}^2$ . It has the constant value  $\pi R_T^2$  when the generation area is defined as a circle with a fixed radius. When the user activates the generation on the can projection (see Sec. 6.3),  $A_{\text{gen}}$  depends on the neutrino direction and is calculated for each event by the code.
- $N_\nu$  is the number of generated neutrino types.
- $E^X$  is the reciprocal of the generation spectrum evaluated at the generated neutrino energy.
- $N_{\text{Tot}}$  is the total number of simulated incoming neutrinos.
- $P_{\text{scale}}$  is the GENIE interaction probability scale [2] (see Sec. 6.4).
- $P_{\text{Earth}}$  is the transmission probability through the Earth evaluated as:

$$P_{\text{Earth}}(E, \cos \theta) = e^{-N_A \cdot \sigma(E) \cdot \rho_l(\theta)}, \quad (10)$$

where  $\sigma(E)$  is the total CC cross section per nucleon (accounting for the different layer compositions). This ignores the (small) effect of NC interactions, which will reduce the neutrino energy without absorbing it. The tau regeneration is also ignored since it is not relevant at the present neutrino energies and will be implemented in the future when the GENIE extension to very high energies will be available [3].  $\rho_l(\theta)$  is the column depth along the neutrino path inside the Earth up to the interaction vertex. It is computed as the line integral  $\rho_l(\theta) = \int_L \rho_{\text{Earth}}(r) dl$ , where  $L$  is the neutrino path at the angle  $\theta$  and  $\rho_{\text{Earth}}(r)$  is the Earth density profile according to the PREM model [8] (see Tab. 2). The column depth and the transmission probability for different neutrino energies calculated by gSeaGen are shown in Fig. 7 and 8, respectively.

## 8 Calculation of the systematic weights

The known accuracy of the input simulation parameters can be propagated with GENIE to provide uncertainties related to neutrino interactions. For each input physics quantity  $P$ , a systematic parameter  $x_P$  is introduced. Tweaking it modifies the corresponding physics parameter  $P$  as follows:

$$P \rightarrow P' = P(1 + x_P \cdot \delta P/P), \quad (11)$$

Table 2: Earth density profile according to the PREM model [8]: the variable  $\chi$  is the normalised radius  $\chi = r/R_E$ , where  $r$  is the distance from the Earth centre and  $R_E = 6371$  km is the Earth radius. SiteDepth is the detector site depth. The third column indicates the target medium associated to each layer.

Density (g/cm <sup>3</sup> )	Radius (km)	Composition
$13.0885 - 8.8381\chi^2$	$0 < r < 1221.5$	Core
$12.5815 - 1.2638\chi - 3.6426\chi^2 - 5.5281\chi^3$	$1221.5 < r < 3480$	Core
$7.9565 - 6.4761\chi + 5.5283\chi^2 - 3.0807\chi^3$	$3480 < r < 5701$	Mantle
$5.3197 - 1.4836\chi$	$5701 < r < 5771$	Mantle
$11.2494 - 8.0298\chi$	$5771 < r < 5971$	Mantle
$7.1089 - 3.8045\chi$	$5971 < r < 6151$	Mantle
$2.691 + 0.6924\chi$	$6151 < r < 6346.6$	Mantle
2.900	$6346.6 < r < 6356$	Mantle
$\rho_r$	$6356 < r < R_E - \text{SiteDepth}$	Rock
$\rho_s$	$R_E - \text{SiteDepth} < r < R_E$	SeaWater

where  $\delta P$  is the estimated standard deviation of  $P$ . The calculation of the systematic errors in GENIE is based on an event re-weighting strategy. A description of the full re-weighting scheme is reported in [12].

The evaluation of the systematics has been implemented in gSeaGen, using the GENIE class GReWeight [12]. The implementation accepts single parameters or a list of them as input. In the latter case, the code treats all parameters at the same time and calculates the global systematic weight. If the calculation is activated, the systematic weights  $w_{\text{sys}}$  are written in the output file. The modified distributions are obtained by multiplying the event weights by  $w_{\text{sys}}$ .

It is also possible to compute the systematic weights using the standard GENIE applications. These applications need as input the native GENIE format output file, available running gSeaGen with the option `-w` (see Sec. 10).

## 9 Installing the code

The gSeaGen source code can be downloaded from [zenodo.org/record/3715310](http://zenodo.org/record/3715310) [18]. Its compiling requires a Linux-like operating system and some basic tools, such as the gcc compiler suite, make and PERL. A minimal installation of gSeaGen requires only GENIE<sup>2</sup> as external package.

A number of environmental variables must be defined:

- GSEAGEN pointing at the top level gSeaGen directory;
- GENIE pointing at the top level GENIE directory;
- ROOTSYS pointing at the top level ROOT directory.

<sup>2</sup>The installation of GENIE is described in its user manual [12] and in its web site <http://www.genie-mc.org>.

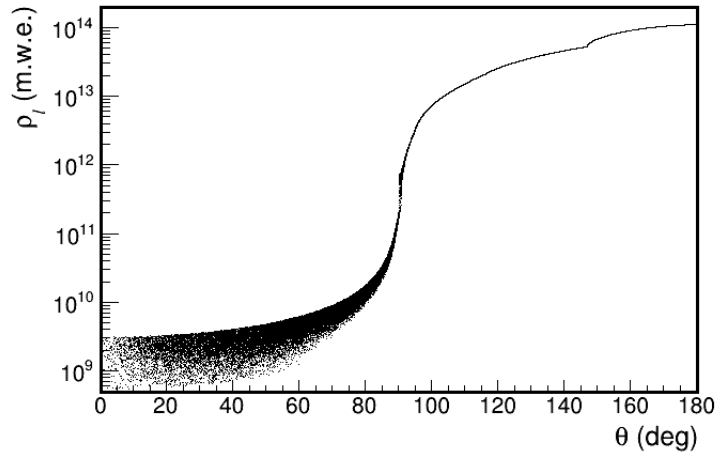


Figure 7: Column depth as a function of the neutrino arrival direction according to the PREM model [8]. The spread visible for downward going neutrinos is due to the range of interaction vertex depths from the same arrival direction.

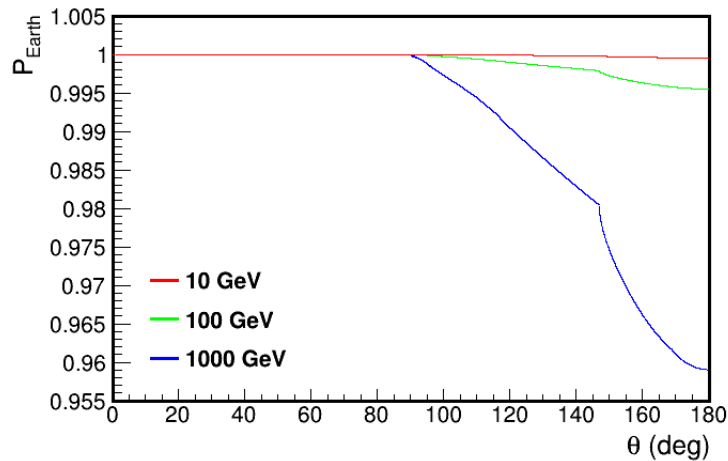


Figure 8: Transmission probability through the Earth  $P_{\text{Earth}}$  as a function of the neutrino arrival direction and for different neutrino energies.

gSeaGen has been written using the GENIE Auk production release series and tested with GENIE version 2.12.10. It is also possible to link gSeaGen with the new Bear release series (GENIE v3.0.0 and later). In this case the GENIE environmental variables for compiling are:

- GENIE pointing at the top level GENIE Generator directory;
- GENIE\_REWEIGHT pointing at the top level GENIE Reweight directory.

When the environmental variables are set, a minimal installation of gSeaGen is obtained typing:

```
$ cd $GSEAGEN
$ ./configure
$ make
```

The configuration script allows enabling/disabling features and specifying paths to external libraries such as the MUSIC and PROPOSAL propagators. The command

```
$ ./configure --help
```

lists the configuration options. The configuration script also creates the shell scripts `setenv.sh` and `setenv.csh`, helping the user to prepare the environment to run the code. These scripts automatically defines also the environment variable `GXMLPATH` pointing to `$GENIE/genie_xsec` if the directory exists (see [12] for major details).

## 10 Usage

To set the environment configuration required to run `gSeaGen`, the `setenv` script must be executed:

```
$ source $GSEAGEN/setenv.csh (in tcsh or csh shells)
$ source $GSEAGEN/setenv.sh (in bash shell)
```

The environmental variables required by GENIE and the ROOT version used during the compilation step are also set. Finally, the user can run the code with the following syntax:

```
$ gSeaNuEvGen [OPTIONS]
```

Running the code with the option `-h` (i.e. `$ gSeaNuEvGen -h`), the code writes the list of the available options:

- `-a generation_spectral_index [default: 1.4]`  
Specifies the generation spectral index.
- `-b n_of_bins [default: 1]`  
Specifies the number of energy bins in equal divisions in  $\log_{10}(E)$ .
- `-bedrock bedrock_height [default: 0]`  
Specifies the height of the bedrock just below the sea bottom (in meters).
- `-c "CAN:Zmin,Zmax,Rad" [default: "CAN:0.000,659.862,309.450"]`  
Specifies the detector can size where `Zmin` and `Zmax` are bottom and top `Z` coordinates of the can and `Rad` is its radius in meters.
- `-C`  
Force the interaction volume to be the detector CAN. This option has an effect only in case of muon and tau CC interaction. In the other cases the interaction volume is already the CAN.
- `--coord "UNIT:longitude,latitude" [default: RAD:0.74700,0.10763]`  
Sets detector site latitude and longitude in radians (`UNIT=RAD`) or in degrees (`UNIT=DEG`). The longitude is positive eastwards from the meridian of Greenwich, and negative to the West.

`--cross-sections XSecName [default: gxspl-FNALsmall.xml]`  
 Inputs the name of an XML file with pre-computed cross-section values used for constructing splines for GENIE<sup>3</sup>.

`-d depth [default: 2475]`  
 Specifies the site depth in meters.

`-e min_energy,max_energy [default: 1,1000]`  
 Specifies the neutrino energy range in GeV.

`--event-generator-list list_name [default: Default]`  
 List of event generators to load. Valid settings are the XML block names appearing in `$GENIE/config/EventGeneratorListAssembler.xml`.

eg. `--event-generator-list CC`  
 eg. `--event-generator-list NC`

`-format` Specifies the output file format. [default: et]  
 Possible choices are:  
 - et --> native gSeaGen ROOT output file  
 - gen --> native GENIE ROOT output file

It is possible to write the output file in different formats among those available, e.g.  
`-format et,gen`

`-f "simul:flux[nu_code],..."`  
 Specifies the types of neutrinos to be generated and the neutrino fluxes used to weight the events (see Sec. 11.2 for details).

`-format` Specifies the output file format. [default: et]  
 Possible choices are:  
 - et --> native gSeaGen ROOT output file  
 - gen --> native GENIE ROOT output file

It is possible to write the output file in different formats among those available, e.g.  
`-format et,gen`

`-n n_of_neutrinos`  
 Specifies the number of incoming neutrinos.

`-o output_event_file_prefix [default: gseagen]`  
 Sets the prefix of the output event file. The output filename is built as:  
`[prefix].[run_number].root`.

`-origin X0,Y0,Z0 [default: 0.,0.,0.]`  
 Sets the origin position coordinates (in meters) in the detector rest frame.

---

<sup>3</sup>Pre-computed cross-sections for GENIE are available on <http://www.genie-mc.org>.

`-point source_info`  
 Activates the generation from point-like and extended sources and specifies the source information (see Sec. 11.5 for details).

`-prop prop_code [default: PropaMuon]`  
 Sets the muon propagation code. Possible choices are: PropaMuon (internal code), MUSIC and PROPOSAL if enabled at compile-time.

`-r run_number [default: 100000000]`  
 Specifies the Monte Carlo run number.

`-rt [default: can]`  
 Set option to define the generation area radius  $R_T$ . Possible choices are:

- 1) `-rt can`  
 the generation surface will be a circle covering the detector can projected onto a plane perpendicular to each neutrino direction;
- 2) `-rt vol`  
 the generation surface will be a circle covering the interaction volume projected onto a plane perpendicular to each neutrino direction;
- 3) `-rt proj`  
 the generation surface will be the detector can projected onto a plane;
- 4) `-rt value`  
 the generation surface radius with  $R_T = \text{value}$  (in meters).

`--seed random_number_seed [default: 12345]`  
 Specifies the random seed number.

`-tau decay_mode`  
 Choose the decay mode of tau. Needed for HEDIS NUTAU-CC channel  
 Possible choices are:

- 1) `-tau muon`  
 tau will decay into muon.
- 2) `-tau shower`  
 tau will decay into shower.

`-t min_costheta,max_costheta [default: 0.,1. (up-going neutrinos)]`  
 Specifies the angular range of the neutrino arrival directions.

`-time`  
 Specifies the generation time interval.  
 The general syntax is: `'-time time'format:StartTime,StopTime`  
 where the `time'format` string specifies the format for `StartTime` and `StopTime`.

Notes:  
 - Possible values for `time_format` are:



TS: StartTime and StopTime are in Unix Time Stamp.  
MJD: StartTime and StopTime are in Modified Julian Date.  
DATE: StartTime and StopTime are in date in format (time in GMT)  
dd-mm-yyyy;hh:mm:ss

- Value for time\_format is not mandatory. In this case the code will set by default:  
time\_format = TS

`--tune generator_tune [default: G18_02a_00_000]`  
Specifies the generator tune, required only for GENIE version  $\geq 3.00.00$ .

`-utm`  
Use UTM detector system [default: false]

`-wgt [default: 1]`  
Weight the events according to interaction probability to increase statistics (GENIE option). Possible choices are:  
1) `-wgt 0`  
events are not weighted;  
2) `-wgt 1`  
events are weighted according to a maximum probability.

## 11 Input Files

### 11.1 Neutrino cross section files

When generating neutrino interaction events, most CPU-cycles are spent by GENIE on calculating neutrino interaction cross sections. For this reason, GENIE gives the possibility to use pre-computed cross section data and estimate the cross section by numerical interpolation [12]. Adopting this approach is mandatory in gSeaGen. By default the code searches the file `gxsp1-FNALsmall.xml` and if this file does not exist, it checks for the existence of the file `<generator_tune>.xml`, where `generator_tune` the name of the used GENIE tune.

The user can specify a different name for the pre-computed cross section file with the option `--cross-section`. Being an input for GENIE, file path-name is not required if the file is saved in the directory pointed by environmental parameter `GXMLPATH` [12].

Official pre-computed cross section files for several GENIE tunes are available from web page [https://scisoft.fnal.gov/scisoft/packages/genie\\_xsec/](https://scisoft.fnal.gov/scisoft/packages/genie_xsec/) or from the cvmfs repository in `/cvmfs/fermilab.opensciencegrid.org/products/genie/externals/pochoarus-genie_he_data/`.

### 11.2 Neutrino flux files

The user can define the type of neutrino and the neutrino fluxes used to weight the events with the option `-f` at run time. This does not affect the energy of the generated neutrinos since the power-law of the generation is controlled by options `-e` and `-a`. The general syntax of option `-f` is:

```
-f "simul:flux,...[nu_code],flux...[nu_code];simul:... ",
```

where the tag `simul` specifies the flux model, `flux` is the name of the file containing the flux values and `nu_code` is the neutrino type in the PDG code [19] (12:  $\nu_e$ , -12:  $\bar{\nu}_e$ , 14:  $\nu_\mu$ , -14:  $\bar{\nu}_\mu$ , 16:  $\nu_\tau$ , -16:  $\bar{\nu}_\tau$ ).

The neutrino flux can be provided with an ASCII file, containing the value of the neutrino spectrum at different neutrino energies and arrival directions. In order to get a continuously weighted flux, the code interpolates the tabulated values at the generated neutrino energy and direction. Possible values for the `simul` tag are BARTOL, FLUKA and HONDA that allow to read directly in the formats available on the web sites of Ref.s [20, 21, 22], respectively. For example:

```
-f "BARTOL:f210_3_z.kam_num[14] ",  
-f "FLUKA:sk_numu02.dat[14] ",  
-f "HONDA:grn-ally-20-12-solmin.d[14] ".
```

Different neutrino types or neutrinos and anti-neutrinos can be simulated simultaneously, e.g.:

```
-f "BARTOL:f210_3_z.kam_num[14],f210_3_z.kam_nbm[-14] ",  
-f "HONDA:grn-ally-20-12-solmin.d[14,-14] ".
```

It is possible to define more than one file for each neutrino code. Fluxes from different files will be added, e.g.:

```
-f "BARTOL:fmin10_0401z.kam_nue,f210_3_z.kam_nue[12] ".
```

The neutrino flux can also be provided using a user-defined function of the neutrino energy. In this case the tag `simul` has value `FUNC1` and `flux` is a mathematical function (C++ syntax) depending on `x`, which represents the neutrino energy in units of GeV, and defining the neutrino flux in units of  $\text{GeV}^{-1}\text{m}^{-2}\text{s}^{-1}\text{sr}^{-1}$ , e.g.:

```
-f "FUNC1:1E-5*pow(x,-2)[14] ".
```

Two-dimensional functions depending on `x` and `y` can be provided to define a dependence on the neutrino arrival direction. In this case the tag `simul` has value `FUNC2` and the variable `y` represents  $\cos(\theta)$ . Similarly three-dimensional functions depending also on the variable `z` (representing the angle  $\phi$  in radians) can be provided by setting the tag `simul` to `FUNC3`.

Furthermore, it is possible to combine different neutrino fluxes by defining more than one tag `simul`. Fluxes corresponding to the same neutrino code will be combined, e.g.:

```
-f "BARTOL:f210_3_z.kam_num[14];FUNC1:1E-5*pow(x,-2)[14] ".
```

Finally, values for `simul` and `flux` are not mandatory, e.g.:

```
-f "[14] ",  
-f "[12,14] ",  
-f "[16,-16] ".
```

In this case the code sets by default `simul=FUNC1` and `flux=1E-5*pow(x,-2)`.

New fluxes will be implemented in the future to give the user the opportunity to simulate new sources of astrophysical interest. Regardless of the specified flux, it is possible to re-weight the generated sample with a different flux using Eq. 7.

### 11.3 Maximum muon range file

In the case of muon and tau neutrinos undergoing CC interactions, the interaction volume is defined according to the maximum muon range in sea water and in rock as described in Sec. 5.

By default, gSeaGen uses the muon ranges computed with MUSIC [14] and contained in the file `$GSEAGEN/dat/muon_rmax_music.dat`. Users can also provide their own file. The file should have three columns corresponding to the following items of information:

- $\log_{10}(E)$ , where  $E$  is the muon energy in GeV,
- $\log_{10}(R_r)$ , where  $R_r$  is the range in rock in m.w.e.,
- $\log_{10}(R_w)$ , where  $R_w$  is the range in sea water in m.w.e.

The muon energy range in the file must be large enough to cover all possible energies of generated muons, while there are no constraints on the number of rows and on the energy binning. To define the new muon range input file, the user has then to define the environmental variable `GMURNGFL` pointing to the file with max muon ranges, e.g.:

```
GMURNGFL = GSEAGEN/dat/muon_rmax_music.dat.
```

### 11.4 Target medium composition file

To override default target medium compositions defined in Sec. 4, the user has to define the environmental parameter `MEDIACOMP`, pointing to the file defining the new compositions. The input file must be in XML format where the root element must be `media_comp`. The compositions are defined inside the tag `param_set`. Its attribute `media` specifies the name of the target medium. Possible values for `media` are: `SeaWater`, `Rock`, `Mantle` and `Core`. Each chemical element is given through the tag `param`, where the attribute `name` indicates the chemical element name and the content defines the value of the mass fraction. The list of elements from which the user can select to define a composition is: `Al`, `Br`, `C`, `Ca`, `Cl`, `Fe`, `H`, `K`, `Mg`, `N`, `Na`, `Ni`, `O`, `S`, `Si`, `Ti`. The mass number must not appear in the value of the attribute `name`, since it will be automatically set to the most abundant isotope. For example, the file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<media_comp>
  <param_set media="SeaWater">
    <param name="O"> 0.8879 </param>
    <param name="H"> 0.1121 </param>
  </param_set>
  <param_set media="Rock" density="2.65">
    <param name="O"> 0.533 </param>
    <param name="Si"> 0.467 </param>
  </param_set>
</media_comp>
```

will re-define the `SeaWater` and `Rock` compositions respectively with pure water and standard rock, while default compositions for `Mantle` and `Core` will be used. The user has the

possibility to set the density value (in  $\text{g}/\text{cm}^3$ ) for SeaWater and Rock media, adding the attribute density to the tag param\_set. For example it is possible to use the SeaWater medium to simulate ice by writing:

```
<param_set media="SeaWater" density="0.92">.
```

The attribute density is ignored for Mantle and Core media, for which the PREM density profile is always used. For another example, the user can have a look at the file \$GSEAGEN/dat/MediaComposition.xml, which explicitly defines all the compositions with their default values.

## 11.5 Astrophysical source info file

To activate the simulation of an astrophysical source, the user has to use the option `-point`, followed by information about the source. By default, a point-like source is simulated but an angular radius can also be set to simulate an extended circular source. It is possible to input just the source declination (`-point "DEC:Declination"`, with Declination in degrees): the right ascension will be automatically set to zero. If point/extended mode is activated, only tag FUNC1 for the `-f` option is available in the present version.

The user also has the possibility to input a file containing more information about the source (e.g. `-point "FILE:AstroSource.xml"`), such as the source equatorial or galactic coordinates and the source angular radius for an extended source. The input file must be in XML format where the root element is `astro_source`. Source information is set inside the tag `param_set`, where its attribute `source_name` specifies the name of the simulated source. Each piece of information is provided using the tag `param`, where the attribute name indicates the source parameter and the content defines the corresponding value. For example, giving the following file as input:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<astro_source>
  <param_set source_name="Dec-60">
    <param name="Declination"> -60. </param>
    <param name="RightAscension"> 0. </param>
  </param_set>
</astro_source>
```

is equivalent to running the code with the option `-point "DEC:-60"`. The complete list of all possible parameters is reported in Tab. 3. For example, the file \$GSEAGEN/dat/AstroSource.xml defines information for the star Antares and generates events during the whole of 2016.

## 11.6 Default parameter values input file

The default values of some input parameters can be changed by defining the environmental variable `DEFPARAM`. The variable must point to the input file containing the list of input parameters whose default value has to be changed. The input file is in XML format where the root element must be `default_params`. Each piece of information is provided using the tag `param`, where the attribute name indicates the parameter name and the content defines the new default value. For example, giving the following file as input:

Table 3: Complete list of input astronomical source parameters. If both equatorial and galactic coordinates are set, the equatorial ones will be used.

Parameter	Description
Declination	declination in deg
RightAscension	right ascension in deg
GalacticLatitude	galactic latitude in deg
GalacticLongitude	galactic longitude in deg
Equinox	equinox defining the galactic coordinates: J2000 or B1950
Radius	source angular radius in deg (point-source: Radius=0.)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<default_params>
  <param name="NBin">      5      </param>
  <param name="Alpha">    2      </param>
</default_params>
```

the default number of energy bins is changed from 1 to 5 and the generation spectral index is changed from 1.4 to 2. The complete list of all possible parameters for which it is possible to change the default value is reported in Tab. 4. The use of this input file changes the default values of the parameters which can however be modified using the running options described in Sec. 10.

## 11.7 Systematic parameter input file

The calculation of the systematic errors related to the neutrino interaction models is activated by defining the environmental variable SYSTLIST. The variable must point at the input file containing the list of physics parameters that can be varied and the associated systematic parameters (see Sec. 8). The input file is in XML format where the root element must be `syst_param`. The physics parameters are defined inside the tag `param_set` which accepts the attribute `set_type`. Setting `set_type="list"` defines a list of parameters. In this case, the code treats all parameters simultaneously and will calculate the global systematic weight. Each parameter to be considered for the calculation of the systematic weight is inserted with the tag `param`, where the attribute name indicates the physics parameter and the content defines the value of the corresponding systematic parameter, e.g.:

```
<param_set set_type="list">
  <param name="MFP_pi"> +1.0 </param>
  <param name="MFP_N">  -1.0 </param>
</param_set>.
```

There is no limit to the number of lists that the user may define. The “list” systematic weights will be output in the array `WSys` in the same order in which the lists are defined (see Sec. 12).

Setting `set_type="list;mirror"`, a second list with the same physics parameters and opposite systematic parameters is created. The two global weights will be output consecutively, e.g.:

Table 4: Complete list of names of the input parameters for which it is possible to change the default value, reported in the second column. The third column gives the unit of measure which must be used in the input file. Finally, as a reference, the corresponding running option is shown in the last column.

Parameter	Default	Unit of measure	Running option
RunNu	100000000	-	<code>-r</code>
RanSeed	12345	-	<code>--seed</code>
Can	0.000,659.862,309.450	meters	<code>-c</code>
Origin	0.,0.,0.	meters	<code>-origin</code>
SiteLatitude	0.74700	radians	<code>--coord</code>
SiteLongitude	0.10763	radians	<code>--coord</code>
SiteDepth	2475.	meters	<code>-d</code>
EvRange	1.,1.E3	GeV	<code>-e</code>
CtRange	0.,1.	-	<code>-t</code>
NBin	1	-	<code>-b</code>
Alpha	1.4	-	<code>-a</code>
RTOpt	can	-	<code>-rt</code>
HBedRock	0.	meters	<code>-bedrock</code>
PropCode	PropaMuon	-	<code>-prop</code>
WeightOpt	1	-	<code>-wgt</code>
EvFilePrefix	gseagen	-	<code>-o</code>
InpXSecFile	gxspl-FNALsmall.xml	-	<code>--cross-sections</code>
EventGeneratorList	Default	-	<code>--event-generator-list</code>
UseUTMSystem	false	-	<code>-utm</code>

```

<param_set set_type="list;mirror">
  <param name="MFP_pi">    +1.0      </param>
  <param name="MFP_N">    -1.0      </param>
</param_set>

```

is equivalent to writing:

```

<param_set set_type="list">
  <param name="MFP_pi">    +1.0      </param>
  <param name="MFP_N">    -1.0      </param>
</param_set>

```

```

<param_set set_type="list">
  <param name="MFP_pi">    -1.0      </param>
  <param name="MFP_N">    +1.0      </param>
</param_set>

```

Setting `set_type="parameter"` defines single physics parameters. In this case, the code will calculate the systematic weights for each parameter, separately. The results will be output in dedicated tags: `WSysGEN_ParName`, where `ParName` is the name of the systematic parameter.

As before, each physics parameter is input with the tag `param`, where the attribute name indicates the parameter name. In this case the content defines the number of the systematic parameter tweaking dial values between  $-1$  and  $1$ . For example, when setting:

```

<param_set set_type="parameter">
  <param name="MFP_pi">    5      </param>
  <param name="MFP_N">    3      </param>
</param_set>

```

the code will compute the systematic weights for the systematic parameter values  $-1$ ,  $-0.5$ ,  $0$ ,  $+0.5$ ,  $+1$  for the physics parameter `MFP_pi` and for values  $-1$ ,  $0$ ,  $+1$  for `MFP_N`.

An example can be found in the file `$GSEAGEN/dat/SystParams.xml`. For the complete list of the physics parameters considered in GENIE, see table 9.1 of Ref. [12].

## 12 Output File

The native `gSeaGen` event output is a ROOT file containing two “trees”:

- Header: reporting the general info about the simulation;
- Events: containing the neutrino induced particles reaching the detector can.

The user has the possibility to choose between two different tree definition (`EventTree` and `SimpleTree`) through the run option `-tree` option.

## 12.1 The EventTree format

The two ROOT trees are defined using gSeaGen data classes: Header is instance of the GenParam class and Events an instance of the GSeaEvent class. Details on gSeaGen classes are given in the Doxygen documentation linked in the README file.

The gFileConvert app, provided to convert EventTree type files to other formats, can be used as an example to read this type of file and write a code to process the files. Source files of gFileConvert can be found in  $\$GSEAGEN/src/tools/gFileConvert$ .

## 12.2 The SimpleTree format

The two ROOT trees are defined using simple variables. Reading this file is easier as it does not need to load any library. However, its writing slows down the running of the code. It is possible to convert a file written in the EventTree format into the SimpleTree one using the app gFileConvert. Following a description of the two trees.

The Header tree contains the following parameters:

- **gSeaGenVer** (*string*): gSeaGen version;
- **GenieVer** (*string*): GENIE version;
- **RunTime** (*string*): running time;
- **RunNu** (*int*): run number;
- **RanSeed** (*int*): random generator seed;
- **EventGeneratorList** (*string*): list of simulated interaction channels;
- **InpXSecFile** (*string*): cross-section spline file;
- **NTot** (*double*): total number of generated incoming neutrinos;
- **EvMin, EvMax** (*double*): generated neutrino energy range (GeV);
- **CtMin, CtMax** (*double*): generated neutrino arrival  $\cos\theta$  range;
- **Alpha** (*double*): generation spectral index;
- **NBin** (*int*): number of energy bins;
- **Can1, Can2, Can3** (*double*):  $Z_{\text{can}}^{\text{min}}$ ,  $Z_{\text{can}}^{\text{max}}$  and  $R_{\text{can}}$  of the can (m);
- **HRock, HSeaWater, RVol** (*double*): interaction volume: height in rock and water and radius (m);
- **SiteDepth** (*double*): site depth (m);
- **SiteLatitude, SiteLongitude** (*double*): detector latitude and longitude (rad);
- **Agen** (*double*): area of the generation surface ( $\text{m}^2$ );
- **RhoSW, RhoSR** (*double*): SeaWater and Rock densities ( $\text{g}/\text{cm}^3$ );



- **TGen** (*double*): generation time (s);
- **PropCode** (*string*): muon propagation code;
- **GenMode** (*string*): simulated livetime mode (DIFFUSE or POINT);
- **SourceFile** (*string*): input file name with source information (POINT mode);
- **SourceName** (*string*): name of the simulated source (POINT mode);
- **Declination, RightAscension** (*double*): source declination and right ascension (rad) (POINT mode);
- **SourceRadius** (*double*): source angular radius (rad) (POINT mode);
- **MJDStart, MJDStop** (*double*): source simulated interval in MJD (POINT mode only).

The Events tree contains the following parameters:

- **Evt** (*int*): event Id number (incremented sequentially);
- **PScale** (*double*): GENIE interaction probability scale (see Sec. 6.4);
- **TargetA, TargetZ** (*int*): mass number and atomic number of the target nucleus;
- **InterId** (*int*): interaction type Id, according to GENIE enumeration;
- **ScattId** (*int*): scattering type Id, according to GENIE enumeration;
- **Bx, By** (*double*): Bjorken kinematic variables x and y;
- **LST** (*double*): Local Sidereal Time (rad) (POINT mode);
- **MJD** (*double*): Modified Julian Day, if generated (POINT mode);
- **VerInCan** (*int*): 1 if the interaction vertex is inside the can, 2 if it is inside the bedrock and 0 outside.
- **WaterXSec** (*double*): neutrino cross section per nucleon in pure water ( $\text{m}^2$ );
- **WaterIntLen** (*double*): neutrino interaction length in pure water (m);
- **PEarth** (*double*): transmission probability through the Earth;
- **ColumnDepth** (*double*): column depth, entering in the calculation of PEarth (m.w.e.);
- **XSecMean** (*double*): average cross section per nucleon ( $\text{m}^2$ ) along the neutrino path, to be considered in the calculation of PEarth;
- **GenWeight** (*double*): generation weight ( $\text{GeV m}^2 \text{ s sr}$  for diffuse fluxes and  $\text{GeV m}^2 \text{ s}$  for point-like/extended sources);
- **EvtWeight** (*double*): event weight;

- **E\_nu** (*double*): neutrino energy (GeV);
- **Pdg\_nu** (*int*): neutrino PDG code;
- **Vx\_nu**, **Vy\_nu**, **Vz\_nu** (*double*): coordinates of interaction vertex in meters;
- **Dx\_nu**, **Dy\_nu**, **Dz\_nu** (*double*): neutrino track direction cosines;
- **T\_nu** (*double*): time of neutrino interaction (ns);
- **E\_pl** (*double*): primary lepton energy (GeV);
- **Pdg\_pl** (*int*): primary lepton PDG code;
- **Vx\_pl**, **Vy\_pl**, **Vz\_pl** (*double*): coordinates of interaction vertex (m);
- **Dx\_pl**, **Dy\_pl**, **Dz\_pl** (*double*): primary lepton track direction cosines;
- **T\_pl** (*double*): time of neutrino interaction (ns);
- **NTracks** (*int*): final number of particles in the event;
- **Id\_tr** (*int*[*NTracks*]): particle Id number;
- **E\_tr** (*double*[*NTracks*]): particle energy (GeV);
- **Pdg\_tr** (*int*[*NTracks*]): particle PDG code;
- **Vx\_tr**, **Vy\_tr**, **Vz\_tr** (*double*[*NTracks*]): coordinates of final particle position (m);
- **Dx\_tr**, **Dy\_tr**, **Dz\_tr** (*double*[*NTracks*]): particle track direction cosines;
- **T\_tr** (*double*[*NTracks*]): time of passage through the final position with respect to the interaction time (ns);
- **NSysWgt** (*int*): number of lists of systematic parameters;
- **WSys** (*double*[*NSysWgt*]): systematic weights for each list of parameters, in the same order in which the lists are defined;
- **NSysWgt\_ParName** (*int*): number of the tweaking dial values between  $-1$  and  $1$  for the parameter ParName;
- **WSys\_ParName** (*double*[*NSysWgt\_ParName*]): systematic weights for each tweaking dial value for the parameter ParName.

## 13 Logfiles

In addition to the file containing the generated events, gSeaGen writes logfiles showing some information on the simulation.

Table 5: List of parameters reported in the binstat logfile.

Parameter	Description
Bin	Bin number
EneMin, EneMax	Bin energy range (GeV)
NEvBin	Number of neutrinos to simulate in the bin
NEvBinScal	Real number of simulated neutrinos in the bin
NEvBinGen	Number of events generated and written in the output file
RT,RL	Generation area radii $R_T$ and $R_L$ as defined in Sec. 6.3 (meters)
X,Y,Z	Coordinates of the centre of the sphere with radius $R_L$ (meters)
RVol, HSeaWater, HRock	Radius and height in water and rock of the interaction volume (meters)
ProbScale	Interaction probability scale

### 13.1 The .binstat file

This file contains information on the number of events generated and definitions of the generation area and interaction volume for each energy bin. The list of parameters and their description is reported in Tab. 5.

When the generation area is defined by the projection of the can on to the plane perpendicular to the neutrino direction, it is not a circle and changes event by event. In this case, the -100 value is reported in the binstat file for the  $R_T$  radius.

The binstat file is created at the beginning of the running, reporting the values of all parameters except for NEvBinGen and ProbScale. For these the initial value is set to zero and updated at the end of the simulation of each energy bin.

## 14 Tools

In addition to the event generator, the gSeaGen project contains a series of tools that are not needed to run the main code but which can be useful for reading output files, making checks on event generation and more. All tools are implemented in the \$GSEAGEN/src/tools directory and installed in the bin together with the main code at compile time. In this section we report a list of these tools together main info.

### 14.1 gCalcAstro

gCalcAstro checks the transformation of the astronomical coordinates of a source in the detector coordinate system (both Cartesian and UTM). The calculation is done for given detector coordinates and for a list of sources, giving as input their J2000 equatorial coordinates and the epochs at which to calculate the position from a CVS file like this:

```
date,time,object,RA-J2000,DEC-J2000,azimuth,zenith,phi,theta
2007-10-04,03:03:03,SIRIUS,06:45:08.9,-16:42:58,141.723,68.557,126.352,111.443
2007-10-04,03:03:03,CANOPUS,06:23:57.1,-52:41:45,161.187,99.325,106.886,80.675
...
```

where RA-J2000 and DEC-J2000 are the J2000 equatorial coordinates, azimuth and zenith are the horizontal coordinates and phi and theta the source coordinates in the detector system. All the coordinates are in degrees. Starting from RA-J2000 and DEC-J2000, gCalcAstro calculates RA and DEC at the requested epoch and then azimuth, zenith, phi and theta.

The geographical coordinates of the detector can be defined by the user who can alternatively choose one of the predefined sites:

- ARCA: Latitude = 36.2915, Longitude = 15.9752 DEG
- ORCA: Latitude = 42.8002, Longitude = 6.0219 DEG
- ANTARES: Latitude = 42.799, Longitude = 6.166 DEG

For each of these sites a benchmark source coordinate is available in `$GSEAGEN/src/tools/gCalcAstro/data` and used by default, activating automatically the UTM system.

When the UTM coordinate system is used, the code prints the angles `Azimuth_UTM` and `Zenith_UTM` instead of the astronomical horizontal coordinates, to be compliant with the standard checks done in KM3NeT. These angles are calculated according to:

$$\begin{cases} \text{Azimuth\_UTM} &= 90^\circ - \text{Azimuth} + \alpha \\ \text{Zenith\_UTM} &= \text{Zenith} \end{cases}, \quad (12)$$

where  $\alpha$  is UTM convergence angle.

For details about the usage, see the help running `gCalcAstro/cross -h`.

## 14.2 gCalcCross

`gCalcCross` extracts the cross sections per nucleon from the XML cross section files used by GENIE to simulate the neutrino interactions. The utility calculates cross sections for the targets defined in `gSeaGen` (i.e. `SeaWater`, `Rock`, `Mantle` and `Core`) or for a target composition defined by the user. `gCalcCross` can be used as an alternative to `gspl2root` [12], the standard GENIE utility to convert XML cross section files into a ROOT format, which cannot extract cross sections for mixtures. `gCalcCross` is also part of functionality tests of `gSeaGen` that uses these cross sections to calculate the PEarth probability entering in the calculation of the generation weights.

`gCalcCross` generates a ROOT output file containing the objects described in Table 6. For details about the usage, see the help running `gCalcCross -h`.

The `gCalcCross` package contains the `gPlotCross` (python3) script to make plots, comparing cross sections for the same neutrino type in the different target media or cross sections for the same target but different `gSeaGen` versions linked to different GENIE versions. For details about the usage, see the help running `gPlotCross -h`.

## 14.3 gCalcPEarth

`gCalcPEarth` makes a test on the calculation of the transmission probability through the Earth, entering in the calculation of the generation weights. For a given GENIE tune, the app extracts the CC total cross sections from the related XML spline file and calculates the transmission probability as a function of the neutrino direction. The probability is calculated

Table 6: ROOT objects contained in the gCalcCross output file. TGraph objects containing cross sections and interaction lengths are stored for each neutrino type, target medium and interaction list combinations. The interaction length is not calculate for Mantle and Core (density not constant).

Type	Name	Content
TText	gSeaGenVer	gSeaGen version
TText	GenieVer	GENIE version
TText	GenTune	GENIE tune, e.g. G18_02a_00_000
TText	targets	list of targets, e.g. SeaWater,Rock,Mantle,Core
TText	neutrinos	list of neutrino types, e.g.nu_mu,nu_mu_bar
TText	interactions	interaction lists, e.g. CC,NC
TH1D	SeaWaterComp	SeaWater mass composition
TH1D	RockComp	Rock mass composition
TH1D	MantleComp	Mantle mass composition
TH1D	CoreComp	Core mass composition
...		
TGraph	nu_mu_xsec_CC_SeaWater	cross section
TGraph	nu_mu_xsecE_CC_SeaWater	cross section divided by energy
TGraph	nu_mu_lint_CC_SeaWater	interaction length
...		

for different neutrino energies and for all the neutrino flavours. The package contains the gCalcPEarth (python3) script to make plots. For details about the usage, see the help running `gCalcPEarth -h` and `gPlotPEarth -h`.

## 14.4 gFileConvert

gFileConvert converts the native gSeaGen EventTree type output files in other formats as the SimpleTree (see Sec.12) and in the KM3NET and ANTARES formats (available only to members of these collaborations). The app can be used as an example to read this type of file and write a code to process the files.

## 14.5 gGeometry

This gGeometry utility, still under development, has the purpose of verifying the building of the GENIE ROOT geometry file defining the neutrino interaction volume. Currently it is possible to plot the geometry files typing

```
gPlotGeo <geo\_name>.root
```

e.g.

```
gPlotGeo geometry-mu-1.root
```

The geometry files are temporarily created by gSeaGen during the neutrino event generation. The plots must therefore be made during the gSeaGen running.

## 14.6 gSeaGenCheck

gSeaGenCheck generate a series of benchmark simulations done with gSeaGen. Common inputs are defined in file `$GSEAGEN/dat/DefaultParams.xml` and used as default for all the benchmark simulations. The complete set of benchmark simulations is defined in file `$GSEAGEN/src/tools/gSeaGenCheck/README.md`.

To generate the benchmark simulations, run script `gMakeCheck` inputting the name of the desired benchmark simulation, e.g.

```
gMakeCheck NUEL
```

the default output directory is `$GSEAGEN/benchmarks/` but it may be changed inputting the new path as second argument, e.g.

```
gMakeCheck NUEL ./
```

Once the simulations have been generated, benchmark plots can be made with the code `gSeaGenCheck`. By defaults the code searches for the files of all the benchmark simulations generated for the current gSeaGen version and saved in the main directory `$GSEAGEN/benchmarks/`, skipping plots for those not found. The default output directory is again `$GSEAGEN/benchmarks/`. It is possible to changes the default input and output directories and select a short list of benchmark simulations using the running options. See the help running `gSeaGenCheck -h`.

## 14.7 gWeightCorsika

The `gWeightCorsika` app corrects the generation and the event weights when gSeaGen is used to propagate CORSIKA muons activeting the recycle. More info are in [6].

# 15 Applications

The gSeaGen code has been developed to provide the possibility to use maintained neutrino interaction codes/libraries. The code has been used as a cross-check for GENHEN [23], the reference neutrino simulation code for the ANTARES experiment [24], written in FORTRAN, and presently used also by the KM3NeT Collaboration for the ARCA detector [7] simulation.

As an example, muon neutrinos have been generated up to the present GENIE maximum valid energy of 5 TeV. The spectrum of generated events has been weighted according to the Bartol atmospheric muon flux model [25]. The results are shown in Fig. 9.

GENHEN is not optimised at energies below  $\sim 100$  GeV and can not be used to simulate interactions below 10 GeV. Moreover, it does not allow for the evaluation of systematics due to neutrino interaction models. For these reasons, the gSeaGen code was used by the ANTARES Collaboration in the final data analysis to measure atmospheric neutrino oscillation parameters [27].

These limitations of GENHEN at lower energies are especially important for the KM3NeT Collaboration, hence gSeaGen has been chosen as the reference code for the simulation of the ORCA detector which is optimised to detect neutrinos of a few GeV and is aiming at measuring the neutrino mass hierarchy. The gSeaGen code was used to optimise the detector geometry and to estimate the expected performances published in the KM3NeT Letter of

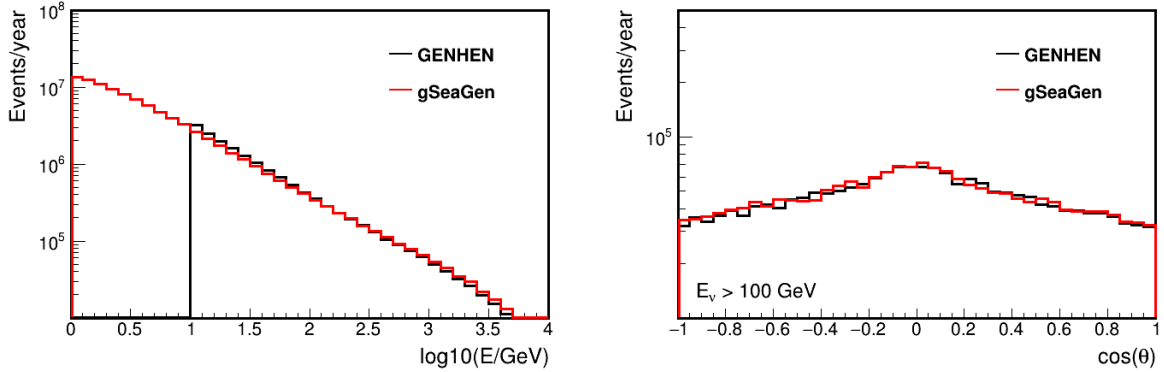


Figure 9: Energy spectrum (top) and angular distribution (bottom) of atmospheric muon neutrinos producing detectable events. Generated events are weighted according to the Bartol atmospheric flux [25]. The simulation was carried out with gSeaGen v6r1 [18] using GENIE version 2.12.10 [26]. Results from the standard neutrino event generator GENHEN [23] are shown for comparison. Note the difference in events per year in the 10-100 GeV range, due to GENHEN being optimised for events with energy above 100 GeV.

Intent [7]. At present, the code is used to carry out the Monte Carlo simulations for the ORCA data analysis [28].

Furthermore, both DeepCore and Upgrade detectors, which are the existing and the future infills of the existing IceCube neutrino telescope [29], also aim at measuring neutrino oscillations in the GeV range. These detectors are located in glacial Antarctic ice and sit above a layer of bedrock. While the code default values and naming convention specify an in-water detector, gSeaGen will also be suitable for modelling these experiments, with an adaptation of medium properties.

It is planned to extend the use of gSeaGen to the KM3NeT-ARCA and ANTARES high energy analyses when the GENIE extension at the PeV scale will be available [3]. Finally, gSeaGen can be used for GVD [30], a next-generation neutrino telescope being constructed in Lake Baikal. gSeaGen should be well-suited to generate interactions at energies from a few TeV up to 100 PeV for GVD.

## 16 Conclusions

The gSeaGen code, written in C++ and based on the GENIE neutrino interaction code, has been developed within the KM3NeT Collaboration. The gSeaGen code generates all flavour neutrino interaction events detectable in a Cherenkov neutrino telescope. It can simulate neutrinos coming from diffuse sources (e.g. atmospheric neutrinos) and from point-like or extended astrophysical sources, according to several neutrino fluxes.

The gSeaGen code provides the possibility of changing the composition and the density of the neutrino target medium surrounding the detector, allowing the study of systematic uncertainties due to medium composition and also considering both under-water and under-ice detectors.

Presently, gSeaGen is the reference code for the simulation of the KM3NeT detectors.

## A The PropaMuon code

The PropaMuon library is a three-dimensional Monte Carlo code for muon propagation through a generic medium. The code simulates the energy losses and the angular deviations of muons taking into account ionisation and radiative processes (bremsstrahlung, direct electron pair production and nuclear interactions).

The muon energy losses are expressed as the sum of the ionisation and radiative losses:

$$\frac{dE}{dx} = \left(\frac{dE}{dx}\right)_{\text{ion}} + \left(\frac{dE}{dx}\right)_{\text{rad}}. \quad (13)$$

The fluctuations associated with the ionisation are small and the energy loss mechanism is treated as continuous. The mean value is evaluated with the Bethe-Bloch formula [19]. PropaMuon can consider any medium composition if ionisation coefficients are provided for each element of which it is made. At present only elements listed in Tab. 1 are considered.

The radiative energy loss is subdivided into a “soft” part, treated as continuous, if the energy fraction  $v$  transferred from the muon to the photon is below a certain threshold ( $v < v_{\text{cut}}$ , here  $v_{\text{cut}} = 10^{-3}$ ), and a “hard” part ( $v \geq v_{\text{cut}}$ ) treated stochastically [31]. Given an element with atomic mass  $A$ , present in the mixture, the radiative energy loss is:

$$\begin{aligned} \left(\frac{dE}{dx}\right)_{\text{rad}} &= \left(\frac{dE}{dx}\right)_{\text{soft}} + \left(\frac{dE}{dx}\right)_{\text{hard}} \\ &= \frac{N_A}{A} E \int_0^{v_{\text{cut}}} \frac{d\sigma(v, E)}{dv} dv + \frac{N_A}{A} E \int_{v_{\text{cut}}}^1 \frac{d\sigma(v, E)}{dv} dv. \end{aligned} \quad (14)$$

The total energy loss in the medium is evaluated as:

$$\frac{dE}{dx} = \sum w_i \cdot \left(\frac{dE}{dx}\right)_i, \quad (15)$$

where  $(dE/dx)_i$  is the energy loss in the  $i$ th element having a weight fraction  $w_i$ .

Total cross section, mean free path and probability for “hard radiation” are:

$$\sigma_{\text{hard}}(E) = \int_{v_{\text{cut}}}^1 \frac{d\sigma(v, E)}{dv} dv, \quad (16)$$

$$\lambda_{\text{hard}}(E) = \frac{A}{\sigma_{\text{hard}}(E) N_A}, \quad (17)$$

$$P_{\text{hard}}(E) = \frac{dx}{\lambda_{\text{hard}}(E)}. \quad (18)$$

Differential cross sections for radiative processes are computed using the Petrukhin and Shestakov formula [32] for bremsstrahlung, the Kokoulin and Petrukhin formula [33] for direct electron pair production and the Bezrukov and Bugaev formula for the nuclear interaction [34].

Given a muon with energy  $E$ , the code extracts the hard radiation interaction path according to:

$$P(r) = \frac{1}{\lambda_{\text{hard}}} \exp(-r/\lambda_{\text{hard}}) \quad (19)$$



and selects the radiative process considering that:

$$\lambda_{\text{hard}} = \lambda_{\text{hard}}^{\text{brem}} + \lambda_{\text{hard}}^{\text{pair}} + \lambda_{\text{hard}}^{\text{nucl}}. \quad (20)$$

Then, the code computes the continuous energy loss (ionisation + soft radiation) and simulates the multiple scattering up to the interaction point. For the multiple scattering, the Lynch and Dahl formula [35] is used.

At this point, the code extracts the fraction  $v$  of the energy lost by the muon in the selected “hard radiation” process from a numerical solution of the equation:

$$P = \frac{\int_v^1 \frac{d\sigma}{dv} dv}{\int_{v_{\text{cut}}}^1 \frac{d\sigma}{dv} dv}. \quad (21)$$

where  $P$  is a random number between 0 and 1. The muon angular deviation due to the selected “hard radiation” process is simulated following the approximations proposed in Ref. [36]. The entire procedure is then repeated until the muon is at rest or input depth is reached.

The muon range simulated with the PropaMuon code in pure water and in standard rock is shown in Fig. 10. For comparison, the ranges simulated with MUSIC and PROPOSAL and the Continuous Slowing Down Approximation (CSDA) range [37] are also shown.

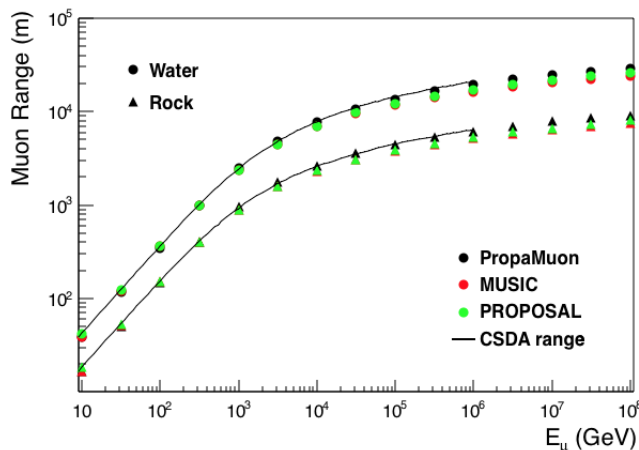


Figure 10: Mean muon range simulated in water and standard rock with PropaMuon and for comparison with MUSIC and PROPOSAL propagators, linkable to gSeaGen. The CSDA range is also shown (values taken from Ref. [37]).

## References

- [1] S. Aiello, et al., gSeaGen: The km3net genie-based code for neutrino telescopes, Computer Physics Communications 256 (2020) 107477. doi:<https://doi.org/10.1016/j.cpc.2020.107477>. URL <https://www.sciencedirect.com/science/article/pii/S0010465520302241>

- [2] C. Andreopoulos, et al., The GENIE Neutrino Monte Carlo Generator, *Nucl. Instrum. Meth. A* 614 (2010) 87–104. arXiv:0905.2517, doi:10.1016/j.nima.2009.12.009.
- [3] A. Garcia, A. Heijboer, High-energy neutrino event simulation at NLO in Genie for KM3NeT and other observatories, in: 36th International Cosmic Ray Conference (ICRC 2019) Madison, Wisconsin, USA, July 24–August 1, 2019, 2019. arXiv:1908.10077.
- [4] A. G. Soto, C. Distefano, P. Kalaczyski, on behalf of the KM3NeT collaboration, Upgrading gSeaGen: from mev to pev neutrinos, *Journal of Instrumentation* 16 (09) (2021) C09008. doi:10.1088/1748-0221/16/09/C09008.  
URL <https://dx.doi.org/10.1088/1748-0221/16/09/C09008>
- [5] D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, T. Thouw, CORSIKA: A Monte Carlo code to simulate extensive air showers (2 1998).
- [6] T. K. Collaboration, gSeaGen code by km3net: an efficient tool to propagate muons simulated with corsika.  
URL [https://opensource.pages.km3net.de/gseagen/CORSIKA\\_processing\\_with\\_gSeaGen.pdf](https://opensource.pages.km3net.de/gseagen/CORSIKA_processing_with_gSeaGen.pdf)
- [7] S. Adrián-Martínez, et al., Letter of intent for KM3NeT 2.0, *J. Phys. G: Nucl. Part. Phys.* 43 (2016) 084001.
- [8] A. M. Dziewonski, D. L. Anderson, Preliminary reference Earth model, *Physics of the Earth and Planetary Interiors* 25 (1981) 297–356. doi:10.1016/0031-9201(81)90046-7.
- [9] I. Antcheva, et al., Root — a c++ framework for petabyte data storage, statistical analysis and visualization, *Comput. Phys. Commun.* 180 (12) (2009) 2499 – 2512. doi:<https://doi.org/10.1016/j.cpc.2009.08.005>.
- [10] S. L. Glashow, Resonant scattering of antineutrinos, *Phys. Rev.* 118 (1960) 316–317. doi:10.1103/PhysRev.118.316.  
URL <https://link.aps.org/doi/10.1103/PhysRev.118.316>
- [11] A. M.G., et al., Detection of a particle shower at the glashow resonance with icecube, *Nature* 591 (2021) 220–224.
- [12] C. Andreopoulos, et al., The GENIE Neutrino Monte Carlo Generator: Physics and User Manual (2015). arXiv:arXiv:1510.05494.
- [13] J. Meeus, *Astronomical Algorithms*, Willmann-Bell, Incorporated, 1991.
- [14] P. Antonioli, et al., A Three-dimensional code for muon propagation through the rock: Music, *Astropart. Phys.* 7 (1997) 357–368. arXiv:hep-ph/9705408, doi:10.1016/S0927-6505(97)00035-2.
- [15] J. H. Koehne, K. Frantzen, M. Schmitz, T. Fuchs, W. Rhode, D. Chirkin, J. Becker Tjus, PROPOSAL: A tool for propagation of charged leptons, *Comput. Phys. Commun.* 184 (2013) 2070–2090. doi:10.1016/j.cpc.2013.04.001.
- [16] T. Sjöstrand, S. Mrenna, P. Skands, Pythia 6.4 physics and manual, *Journal of High Energy Physics* 2006 (05) (2006) 026. doi:10.1088/1126-6708/2006/05/026.  
URL <https://dx.doi.org/10.1088/1126-6708/2006/05/026>

- [17] N. Davidson, et al., Universal interface of tauola: Technical and physics documentation, *Computer Physics Communications* 183 (3) (2012) 821–843. doi:<https://doi.org/10.1016/j.cpc.2011.12.009>. URL <https://www.sciencedirect.com/science/article/pii/S0010465511003973>
- [18] C. Distefano, gSeaGen v6r1 (Mar. 2020). doi:10.5281/zenodo.3715310. URL <https://doi.org/10.5281/zenodo.3715310>
- [19] M. Tanabashi, et al., Review of particle physics, *Phys. Rev. D* 98 (2018) 030001. doi:10.1103/PhysRevD.98.030001. URL <https://link.aps.org/doi/10.1103/PhysRevD.98.030001>
- [20] BARTOL fluxes web site. URL <http://www-pnp.physics.ox.ac.uk/~barr/fluxfiles>
- [21] FLUKA fluxes web site. URL <http://pcbat1.mi.infn.it/~battist/neutrino.html>
- [22] HONDA fluxes web site. URL <http://www.icrr.u-tokyo.ac.jp/~mhonda/nflx2014/index.html>
- [23] D. J. Bailey, Monte carlo tools and analysis methods for understanding the antares experiment and predicting its sensitivity to dark matter, Ph.D. thesis, University of Oxford, United Kingdom (2002). URL <http://antares.in2p3.fr/>
- [24] M. Ageron, et al., ANTARES: the first undersea neutrino telescope, *Nucl. Instrum. Meth. A* 656 (2011) 11–38. arXiv:1104.1607, doi:10.1016/j.nima.2011.06.103.
- [25] G. D. Barr, et al., A Three - dimensional calculation of atmospheric neutrinos, *Phys. Rev. D* 70 (2004) 023006. arXiv:astro-ph/0403630, doi:10.1103/PhysRevD.70.023006.
- [26] GENIE web site. URL <http://www.genie-mc.org/>
- [27] A. Albert, et al., Measuring the atmospheric neutrino oscillation parameters and constraining the 3+1 neutrino model with ten years of ANTARES data, *JHEP* accepted for the publication (2018). arXiv:arXiv:1812.08650v3.
- [28] J. Hofstaedt, et al., Atmospheric neutrinos detected with the first KM3NeT detection units of ARCA and ORCA, in: *Proceedings, 36th International Cosmic Ray Conference ICRC 2019, PoS(ICRC2019)910*.
- [29] A. Ishihara, et al., The IceCube Upgrade - Design and Science Goals, in: *Proceedings, 36th International Cosmic Ray Conference ICRC 2019, PoS(ICRC2019)1031*.
- [30] A. Avrorin, et al., Baikal-GVD: status and prospects, *EPJ Web of Conferences* 191 (2018) 01006. doi:10.1051/epjconf/201819101006. URL <https://doi.org/10.1051/epjconf/201819101006>

- [31] P. Lipari, T. Stanev, Propagation of multi-teV muons, *Phys. Rev. D* 44 (1991) 3543–3554. doi:10.1103/PhysRevD.44.3543.  
URL <https://link.aps.org/doi/10.1103/PhysRevD.44.3543>
- [32] A. Petrukhin, V. Shestakov, *Can. J. Phys.* 46 (1968) S377.
- [33] R. Kokoulin, A. Petrukhin, Proceedings of the international conference on cosmic rays, Hobart, Australia, August 16-25, 1971, Vol. 4, 1971, p. 2436.
- [34] L. Bezrukov, E. Bugaev, *Sov. J. Nucl. Phys.* 33 (1981) 635.
- [35] G. Lynch, O. Dahl, *NIM B58* (1991) 6.
- [36] A. Van Ginneken, Fluctuations of muon energy loss and simulation of ionization cooling, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment.* 362 (2) (1995) 213 – 223.
- [37] Atomic and Nuclear Properties of Materials.  
URL <http://pdg.lbl.gov/2015/AtomicNuclearProperties>