Chapter 4

The simulation

The simulation performed for this thesis required the utilisation of a few different software, all selected following the ND collaboration recommendations [62]. In order to simulate the neutrino interactions with matter we used the Montecarlo generator GENIE, which has been officially adopted by the entire DUNE collaboration (Section 4.1). The geometry of SAND and the Near Detector complex were produced in form of GDML files, using *dunendggd*, a geometry generation tool entirely based on GGD (General Geometry Description). The present SAND's ECAL geometry in particular has been produced during this thesis (Section 4.2). The propagation of the particles generated by the neutrino interactions in the geometry was done using edep-sim, a simulation tool based on Geant4 (Section 4.3). Finally the digitization and reconstruction of the signal produced in the ECAL and STT was done using C++ code, based on previous simulations performed for the KLOE detector (Sections 4.4 and 4.5).

4.1 GENIE: MC neutrino generator

GENIE is a ROOT-based neutrino Montecarlo (MC) generator, designed using object-oriented methodologies and developed in C++ [63]. The present version provides comprehensive neutrino interaction modelling in a range that spans from about 100 MeV to a few hundred GeV, with the long term goal of expanding down to around 1 MeV and up to $\sim$ 1 PeV.

GENIE has been adopted by the majority of neutrino experiment including the Numi neutrino beam-line at Fermilab, and is the first attempt at a canonical neutrino event generator. Such a tool is essential in the design and execution of the new generation of neutrino experiments, including DUNE, that aim to reach new levels of precision. The universality of GENIE in

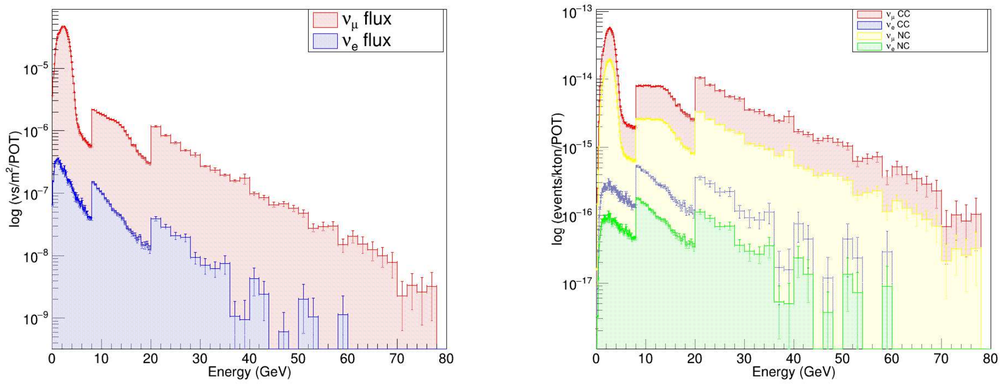
the neutrino experimental field, in particular, solves many problems that the previous MC generator had, due to the fact that they were developed independently for each experiment (for example GENEVE, NEUT, NeuGEN). This limitation made the generators fragile and in many cases incapable of keeping up with the state of the art in terms of theoretical models and experimental data.
GENIE also responds to the necessity of having consistent modelling over the wide energetic range of interest for present and future neutrino experiments, where perturbative and non-perturbative nucleon interactions and many scattering mechanisms are relevant. Often cross section, hadronization and nuclear models have different ranges of validity and need to be pieced together in order to cover all of the available phase space. There are also cases outside these ranges of validity for which new models need to be developed with the additional problem of the general lack of data in the ranges of modern accelerator neutrino experiments. In many cases the simulations are still tuned using data from bubble chamber experiments that date back to the 70's and 80's.

4.1.1 GENIE usage for the ND collaboration

GENIE takes as input a neutrino flux and a geometry representation, and gives as output a file with information regarding the process simulated and the particles in the final state. In particular the output file contains the list of particles that exit the struck nucleus after scattering interaction, hadronization and final state interactions. This file is then handed to edep-sim for propagation.
The fluxes files used by the ND collaboration and in the scope of this thesis are taken from the LBNF beam simulation under the guidance of the beam group. These files come in a variety of formats such as:

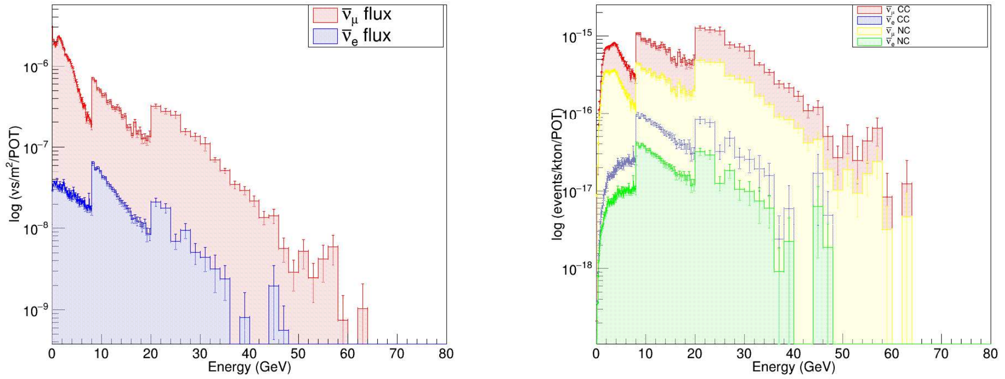
- *dk2nu* files which record information about the decay of hadrons and muons that generate the neutrinos in the beam line, while also providing a flux driver to integrate into GENIE;
- *GSimpleNtpFlux* files, a NTuple format contained into GENIE for storing flux ray information with minimal formatting;
- *ROOT histograms* containing informations regarding the flux neutrino spectrum for all flavours, at the ND (before oscillation) and FD (after oscillation) and the respective CC and NC event rates as a function of energy;

For the simulation performed in this thesis the latter format has been used. In particular the files are the ones produced with the optimized 3-Horn Design by the LBNF collaboration and presented in the Beam Optimization Review of October 2017 [64]. The unoscillated neutrino and antineutrino spectra and NC/CC interaction rates are shown in Fig 4.1 and Fig. 4.2 respectively.
Note that the calculation of cross sections for individual sub-processes are pre-calculated as a function of neutrino energy and stored in XML formatted splines. The ones used in this thesis have been partially retrieved from the previously existing ones recommended by the ND collaboration and partially generated using GENIE's internal spline generator.



(a) (b)

Figure 4.1: Simulation histograms for the NuMi neutrino flux produced with the optimized 3-Horn Design by the LBNF collaboration and presented in the Beam Optimization Review of October 2017: (a) Muon and electron neutrino logarithmic fluxes as a function of energy (GeV) in units of $\nu s/m^2$/POT. (b) Muon and electron neutrino CC and NC logarithmic event rates as a function of energy (GeV) in units of events/kton/POT. [64]



(a) (b)

Figure 4.2: Simulation histograms for the NuMi anti-neutrino flux produced with the optimized 3-Horn Design by the LBNF collaboration and presented in the Beam Optimization Review of October 2017: (a) Muon and electron anti-neutrino logarithmic fluxes as a function of energy (GeV) in units of $\nu s/m^2$/POT. (b) Muon and electron anti-neutrino CC and NC logarithmic event rates as a function of energy (GeV) in units of events/kton/POT. [64]

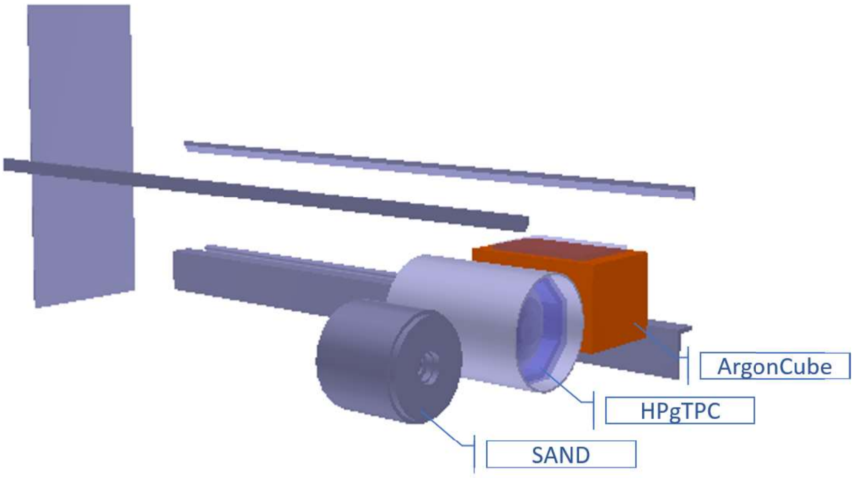


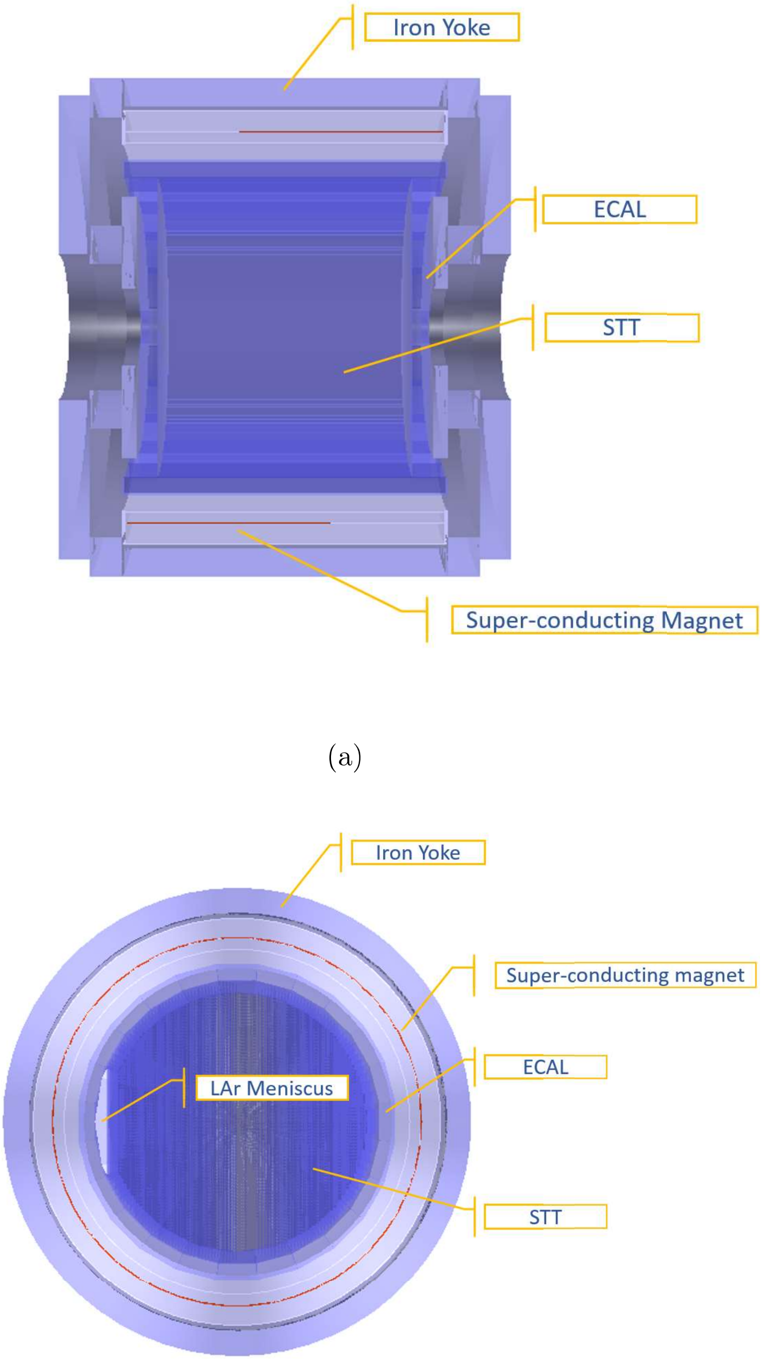
Figure 4.3: Image of the ND hall geometry generated with *dunendggd* and drawn using the OPL ROOT graphic tool.

4.2 GGD: geometry generation tool

The General Geometry Description (GGD) is a python based software system used to generate GDML files containing descriptions of constructive solid geometries [65]. The geometry conventions are the ones used by ROOT and Geant4.

The DUNE ND collaboration has developed its own geometry generation tool called *duneggdnd*, entirely based on GGD [66]. In Fig. 4.3 an image of the ND hall geometry generated with *dunendggd* and drawn using the OGL ROOT graphic tool is shown. It contains the LAr detector ArgonCube, HPgTPC and SAND. The origin of coordinates is chosen to be the location where the beam enters the hall which is assumed to be 574m from the beam's origin, directly on the beam axis.

SAND's geometry descriptions have been produced by the Near Detector collaboration, both for the STT configuration described in Chapter 3 and for the alternative 3DST tracker. A graphical representation of the STT design developed with *dunendggd* is shown in Figure 4.4, with its main components highlighted: the iron yoke, the magnet, the ECAL, the STT and the LAr meniscus. Both designs are in a phase of development and tuning, making new tweaks to the geometries a frequent occurrence. The present geometries for the two active components of the detector (i.e. the ECAL and the STT) will be described more in detail in this section.



(a)

(b)

Figure 4.4: Front (a) and lateral (b) cross section of the SAND geometry produced with *dunendggd* anddrawn with the ROOT graphical tool OGL.The main components of the detector are highlighted: the iron yoke, the magnet, the ECAL, the STT and the LAr meniscus.

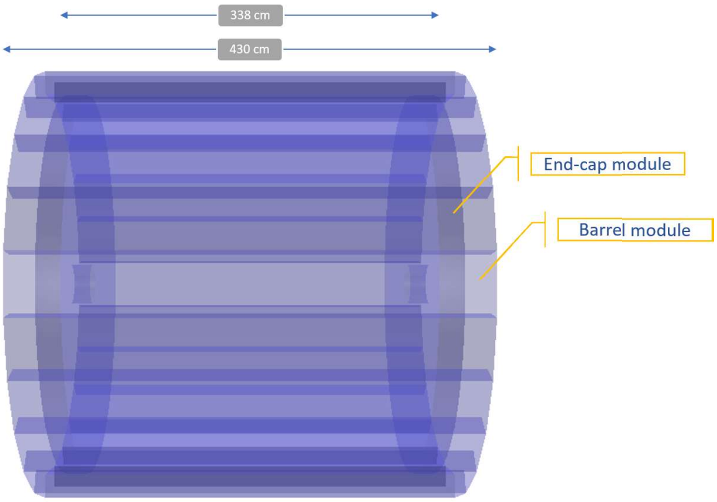


Figure 4.5: Image of the SAND's ECAL's geometry generated with *dunendggd* and drawn using the OPL ROOT graphic tool. The distance between the two inner faces of the endcaps (338cm) and the length of the barrel section (430 cm) are highlighted.

4.2.1 SAND's ECAL: geometry

For the purposes of this thesis, a new geometry for SAND's electromagnetic calorimeter has been produced. As described in Section 3.2.2 SAND's ECAL is a sampling calorimeter made of lead and scintillating fibre divided into a barrel section and two end-cap sections. The barrel calorimeter is a cylinder segmented into twenty-four trapezoidal modules 4.3 m long, 23 cm thick and bases of 52 and 59 cm. The two end-caps are divided into forty-five vertical modules of variable width and height.

A series of approximations, based on previous Fortran simulations made for the KLOE detector, have been made in the geometry. The calorimeter modules have been constructed from alternating slabs of plastic scintillator and lead, rather than from individual fibres. Both the end-cap and barrel modules are composed of 209 slabs of each kind, the Pb slabs being 0.04 cm thick and the scintillating ones being 0.07 cm thick.

The second simplification consists in the fact that, while the barrel modules are simulated with their real dimensions and shape, the end-caps are approximated to be two hollow cylinders of inner diameter 41.6 cm and outer diameter 400 cm. The "bend" that characterises the real end-cap is not reproduced, and no further separation into modules is simulated. The end-cap segmentation, as well as the simulation of the individual Photo-Multiplier

cells is left to the digitisation step of the simulation.
In Figure 4.5 a graphical representation of the entire calorimeter is shown, while the individual barrel and end-cap modules are shown in Figure 4.6.

4.2.2 SAND's STT: geometry

The Straw Tube Tracker is currently in a design stage and its configuration is subject to frequent modifications. In this section we describe the geometry used in the simulations performed during this thesis. The current official configuration might not correspond completely.
The STT is composed by 4 different modular components:

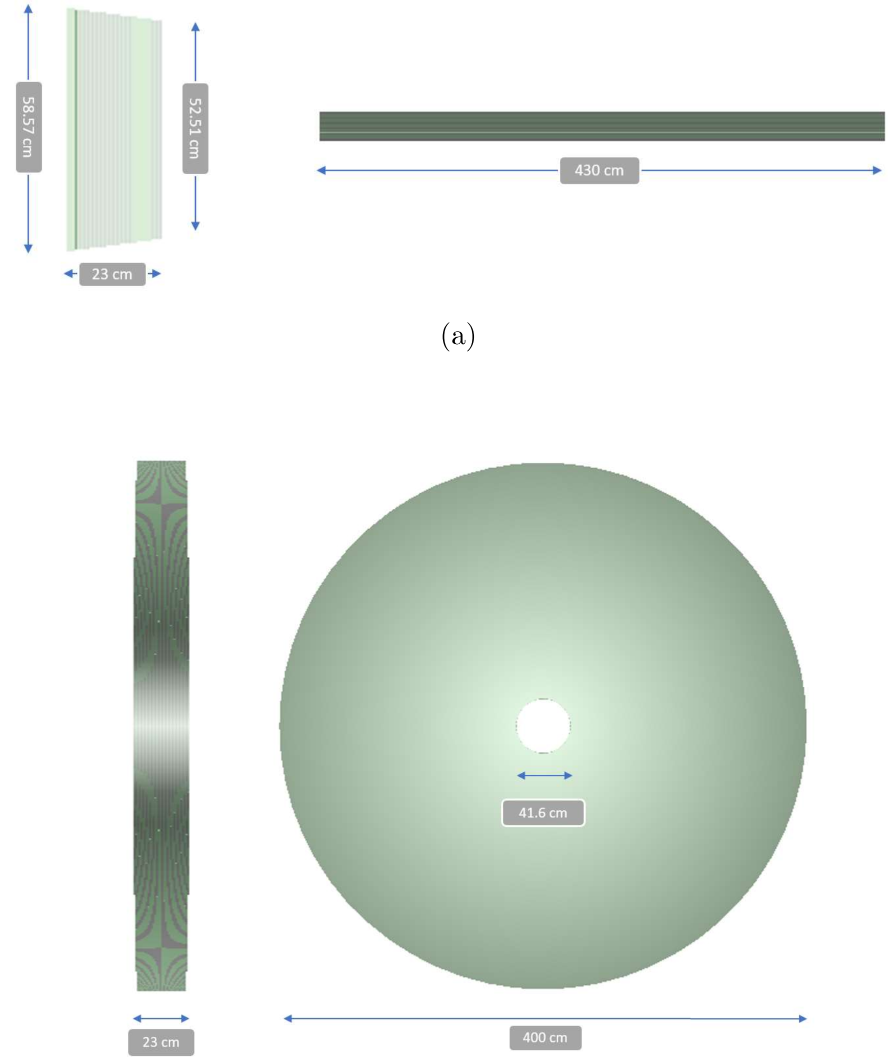
- The Liquid Argon meniscus, complemented by two straw tube XXYY planes placed on the contact surface between the LAr and the STT module region;

- The standard polypropylene modules, composed of a C_3H_6 target slab, a XXYY straw tube tracking plane and a 150 foils TR radiator;

- The graphite modules composed of a C target slab and a XXYY straw tube tracking plane;

- The slab-less modules, which do not offer any target slab and only include the straw tubes planes and the TR radiator;

The dimensions and composition of the modules' internal elements are coherent with the ones reported in Section 3.2.3, with the only omitted components being the support structures.
The STT is composed of 90 total modules plus the LAr meniscus, which is positioned at the front of the detector. Proceeding downstream the first 85 modules are alternated between the 78 polypropylene ones and the 7 graphite ones. Specifically, after the first three modules, which are polypropylene, the graphite modules are one every thirteen. The last 5 modules are slab-less.
While the thickness of the modules are consistent throughout the tracker region, their XY plane area varies so that the corners always touch the internal walls of the calorimeter modules and the smallest possible amount of empty space is left.



(a)

(b)

Figure 4.6: Front and lateral cross section of a ECAL barrel module (a) and a ECAL endcap module (b). The geometries have been produced with *dunendggd* and drawn with the ROOT graphical tool OGL. The components in green represent the scintillator slabs, while the the components in grey represent the Lead slabs.

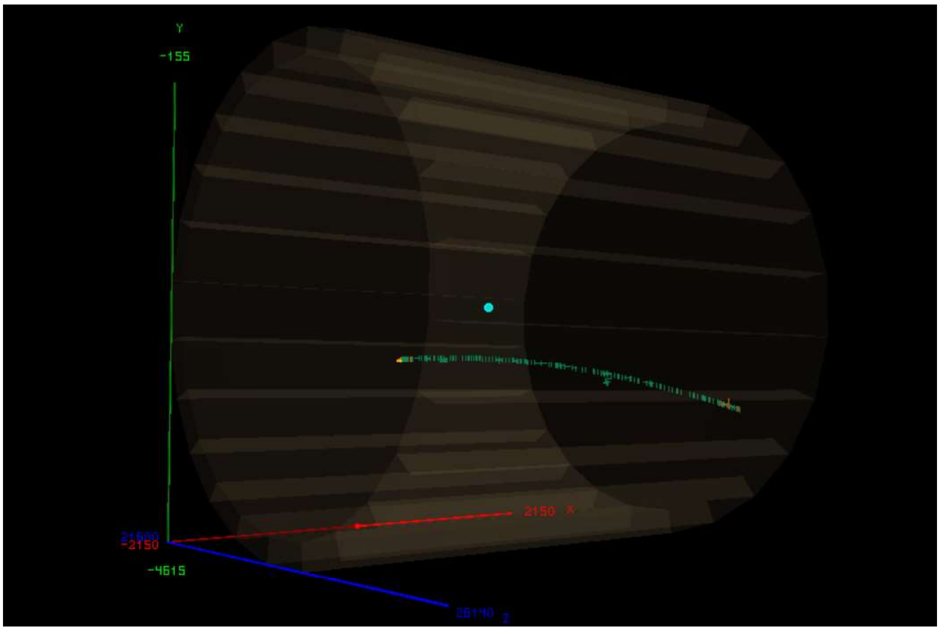


Figure 4.7: Image of a ν_μ CC interaction generated in SAND's ECAL's geometry with GENIE and propagated using edep-sim. The green dots represent the energy deposition hits of the exiting muon. The image has been produced using edep-sim's internal viewer: edep-disp.

4.3 Edep-sim: charged particle propagation

The energy deposition simulation (edep-sim) is a wrapper around the GEANT4 particle propagation simulation [67]. Given a particle kinematics input file and a GDML or ROOT geometry file, edep-sim is capable of simulating the particles' propagation.

Edep-sim supports a variety input file formats, including both NEUT and GENIE output files. In particular GENIE files are read in the "rootracker" format, which is a GENIE event ROOT tree standard evolved from work performed with the purpose of integrating the GENIE simulations with the nd280, INGRID and 2km detector-level MC generators. In particular during this thesis the t2k variant of the format was used.

Edep-sim is also capable of generating and propagating beams of particles in a more standard way using Geant4 command macros. This capability was also used in the scope of this thesis.

The edep-sim output is a standard ROOT file containing two keys: a TGeoManager object with the simulated geometry, and a TG4Event tree (an edep-sim specific class) containing the event information. The TG4Event

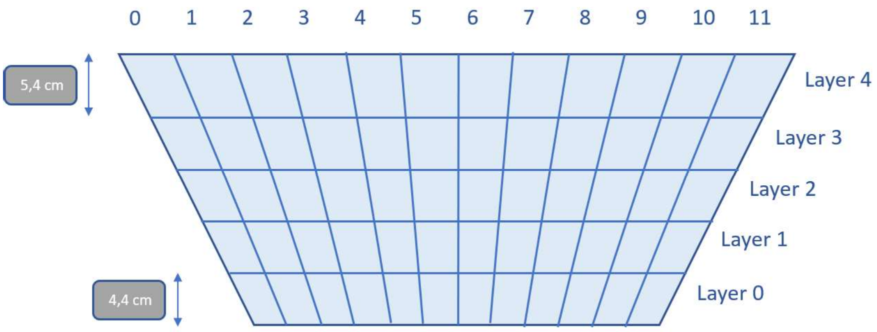


Figure 4.8: Graphical representation of the segmentation of one of the barrel calorimeter modules. The module is divided into five vertical layers: the first four from the bottom are 4.4 cm thick and the last is 5.4 cm thick. Each layer is further divided into 12 horizontal cells.

class contains the event number, the run number, information about the propagated primary particles, the particle tracks and the energy deposition. In Figure 4.7 a graphical representation of a neutrino event generated using GENIE and propagated using edep-sim in SAND's simulated geometry is shown. It is a ν_μ CC interaction, generated in one of SAND's ECAL front modules. The figure was produced using edep-disp, edep-sim's internal event viewer.

4.4 Signal digitization

For signal digitization we mean the simulation of the signal production in the active components of the detector, in accordance with the energy deposition of the particles. In the ECAL this implies the segmentation of the calorimeter modules into cells and the simulation of the photo-electron production in the photo-multiplier tubes. In the STT it requires assigning the hits and their energy deposition to the correct straw tube.

4.4.1 SAND's ECAL: digitization

The photo-electrons produced in the ECAL fibres by passing charged particles, are collected by photo-multiplier tubes connected at both ends of the calorimeter modules. These PMs define a segmentation in the signal collection into layers and cells. We simulate this division via a signal digitization C++ program that takes the information contained in the edep-sim output trees and, given the position of the hits, assigns the energy deposition to the

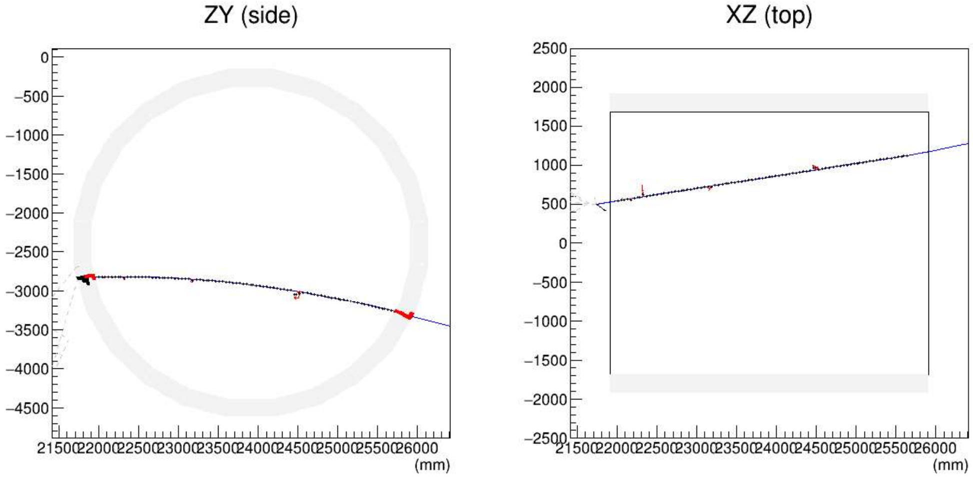


Figure 4.9: Graphical representation of the digitized signal produced by the particles generated in a ν_μ CC interaction event, having vertex in one of the front calorimeter modules. The cells hit by the particles are coloured in red and black, with the red ones having a higher energy deposition. The particle tracks are also highlighted, with the muon being coloured in blue, the neutrons in black and the photons in red. The left panel presents a projection on the ZY plane and the right one on the XY plane. Both are scaled according to the global ND hall coordinates.

correct calorimeter cell. The program has been developed based on previous simulations performed by the KLOE collaboration, and adapted during the scope of this thesis the the new calorimeter geometry.
The barrel calorimeter modules are divided into five horizontal layers. Starting from the smaller base, layers number 0 to 3 contain 40 Pb slabs and 40 scintillator slabs each, for a total of 4.4 cm in thickness. Layer 4 is 5.4 cm thick, containing 49 slabs of each kind. The layers are further divided into 12 trapezoidal cells, each having equally large bases. The cells are numbered from left to right as shown in Figure 4.8. Each module is also assigned a number id from 0 to 23, starting from the top module and proceeding counter-clockwise.
The two end-cap modules are similarly divided into four layers having the same thicknesses of the barrel modules' ones. Each layers contains 45 rectangular cells all having the same area. The enumeration of the cells and layers is analogous to the one for the barrel modules, while the two end-caps are given the id's 30 and 40.

Once the calorimeter is properly segmented we extract the information contained in the edep-sim output files, regarding the hits of the propagated particles contained in the active scintillator slabs. Each hit is automatically assigned by edep-sim to the module in which the slab is contained. Given the hit position we then allocate it in the cell whose centre is closest in terms of:

$$r^2 = (x_{hit} - x_{cell})^2 + (y_{hit} - y_{cell})^2 + (z_{hit} - z_{cell})^2 \tag{4.1}$$

Proceeding this way for each simulated event, each cell is assigned a collection of hits, each being associated to a certain time and energy deposition. Each hit is then assigned to the closest PM (i.e end of the calorimeter module). The energy deposition on the photo-cathode is simulated using the attenuation formula:

$$E_A = p_1 \times \exp\left(-\frac{d}{alt1}\right) + (1 - p_1) \times \exp\left(-\frac{d}{alt2}\right) \tag{4.2}$$

where d is the distance between the center of the cell and the closest photo-cathode, $p_1 = 0.35$, $alt1 = 50$ cm and $alt2$ is 430 cm for planes 0 and 1, 380 cm for plane 2 and 330 cm for planes 3 and 4. For each MeV of energy deposited on the photo-cathode on average 25 photo-electrons are produced. The number of p.e. for each hit is then extracted from a Poisson distribution having $25 \times E_A$ as its most probable value. The ADC (analogue to digital converter) signal assigned to the PM for each event is proportional to the total number of photo-electrons produced on the photo-cathode. In our simple simulation the two numbers coincide.

A graphical representation of the energy deposition divided between the calorimeter cells is shown in Figure 4.9. The event displayed is the same as the one shown in Figure 4.7.

To each photo-electron is also assigned a time of production. This is simulated as being obtained via time to digital converters (TDC) applied to the PMs at both ends of the cells. Each cell is thus associated with two time values from the TDC's: t_{TDC1} and t_{TDC1}. The individual photo-electron time $t_{p.e.}$ is given by:

$$t_{p.e} = t_{part} + t_{decay} + d \cdot u_{ph} + Gauss(1\text{ns}) \tag{4.3}$$

where t_{part} is the time spent by the particle inside the cell, t_{scint} is the scintillation decay time, d is the distance between the hit and the closest PM, $u_{ph} = 5.85$ ns/m is the inverse of the propagation velocity of the photons in the scintillator material (and thus $d \cdot u_{ph}$ is the photon propagation time) and $Gauss(1\text{ns})$ is the uncertainty, extracted from a Gauss distribution with the most probable value being 1 ns.

t_{part} is calculated from the difference between production time of the first and last photo-electron. The scintillation decay time is given by:

$$t_{decay} = t_{scin} \cdot \left(\frac{1}{r_{ph}(1)} - 1 \right) \cdot t_{scex} \text{ (ns)} \tag{4.4}$$

where $t_{scin} = 3.08ns$ and $t_{scex} = 0.588$. Each photo-electron is thus assigned a time $t_{p.e}$, producing two time arrays, one for each PM, that go in increasing order. The two TDC times t_{TDC1} and t_{TDC2}, are given by selecting the p.e. time that is positioned at 15% of the $t_{p.e.}$ list for each of the two TDC's.
At the end of the digitization process, an output file is produced, in which for each event, for every cell in which there was an energy deposition, the following information is recorded:

- The spatial position of the cell centre in the global coordinates of the ND hall;
- The module, layer and cell identification numbers and the module length;
- The two ADC values, corresponding to the number of photo-electrons produced on each photo-cathode ($N_{p.e.1}$ and $N_{p.e.2}$);
- The two TDC times (t_{TDC1} and t_{TDC2});
- Two arrays containing the photo-electron times in increasing order ($t_{p.e.1}[N_{p.e.1}]$ and $t_{p.e.2}[N_{p.e.2}]$) ;
- Two index arrays linking the p.e. times to the hits that produced them stored in the Montecarlo truth edep-sim file;

4.4.2 SAND's STT: digitization

The active component of the STT modules consists of two double layers of straw tube trackers: the first placed horizontally (XX) and the second placed vertically (YY). The first step of the digitization of these detector components, consists in dividing the hits produced by the particles propagated using edep-sim, into clusters: one for each straw contained in the detector. The hits are then reorganised from the first to the last being produced in each straw. Finally an output file is generated, where for each straw hit by at least one particle, the following information is contained:

- The straw name, in which the module and the plane are specified;

- The x, y, and z in the ND hall coordinate system and the time associated to the hit cluster. Each of the coordinates is calculated simply as the mean between the value associated to first and last hits in the cluster;
- The total energy deposition of the hit cluster;
- A flag indicating if the straw hit is horizontal or vertical;
- A index array mapping the cluster to the hits that produced it from the Montecarlo truth;

4.5 Reconstruction

The reconstruction C++ code is divided between the track reconstruction performed in the STT and the cluster reconstruction in the ECAL. The first performs a fit of the helicoidal trajectory described by the charged particles moving in the STT magnetic field. It does so using the hit coordinate information from the STT digitization output, which also allows for a reconstruction of the particle initial momentum. The second reconstructs the energy deposition in the calorimeter from the digitized cell signal.

4.5.1 SAND's ECAL: cluster reconstruction

The information from the the digitized signal produced in the calorimeter, is used to divide the cells with energy deposition into cluster, each associated to a specific particle track. In order to do so we first find for each cell, the particle that produced the most p.e., using the track id. from the Montecarlo truth. For each track that produced the largest amount of p.e. in at least one cell, we create a collection of cells, or cluster, containing each cell in which the particle had the largest energy deposition.
For each cell in the cluster we evaluate the position of where the particle hit it and when. In the ND hall coordinates the z is always taken as the centre of the cell. For the barrel ECAL modules, which are placed horizontally, the y coordinate is also estimated this way, while the x is evaluated using the information from the cell TDC's:

$$x = \frac{t_{TDC1} - t_{TDC2}}{2u_{p.e.}} + x_{cell} \tag{4.5}$$

where x_{cell} is the coordinate of the cell centre, while t_{TDC1}, t_{TDC1} and $u_{p.e.}$ are the same quantities defined in Section 4.4. For the end-cap cells, which are placed vertically the opposite is true: the x is taken as the cell centre, while the y is calculated as in Eq. 4.5. The time value for each cell is calculated by taking the mean from the two TDC value and subtracting the maximum photon propagation time:

$$t = \frac{1}{2}(t_{TDC_1} + t_{TDC2} - u_{p.e.} \times L) \tag{4.6}$$

where L is the total length of the cell.
For each cluster a weighted average of the coordinates $(x_{cl}, y_{cl}, z_{cl}, t_{cl})$ is calculated, using the total energy deposition for each cell E_{cell} as weights. These values are given in the output file together with the total energy deposition E_{cl} and the standard deviations' squares.
The average (x, y, z, t) is also calculated for each calorimeter layer in the cluster. These values are then used to perform linear fits on the zx and zy planes in order to estimate the components of the direction versor of the particle trajectory in the calorimeter. Whether the particle is entering or escaping the detector is evaluated by calculating:

$$d = \frac{(z_{out} - z_{in})(t_{out} - t_{in})}{|(z_{out} - z_{in})(t_{out} - t_{in})|} = \pm 1 \tag{4.7}$$

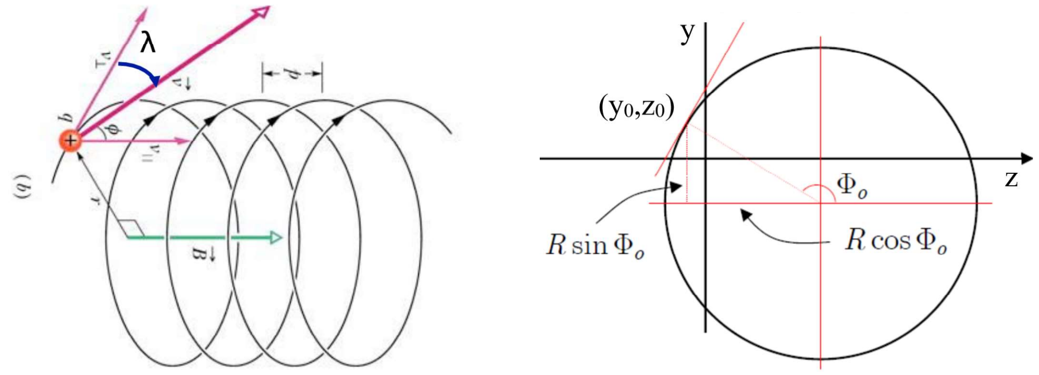
where z_{out} and z_{in} are the z coordinates for the most outer and most inner layer hit in the cluster, respectively (analogously for t_{out} and t_{in}). The particle is escaping if $d = +1$ and entering if $d = -1$.

4.5.2 SAND's STT: track reconstruction

A charged particle entering the tracking region of the SAND detector is effected by the 0.6 T magnetic field $\vec{B}$, pointing in the x direction of the ND hall coordinates. It thus describes a helical trajectory: a combination of the circular motion on ZY plane, perpendicular to the magnetic field and a linear motion parallel to B in the z direction (Figure 4.10). The helix can be described parametrically as a function of the spatial progression along the trajectory s:

$$\begin{cases} z(s) = z_0 + R\Big(\cos(\Phi_0 + \frac{hs\cos\lambda}{R}) - \cos\Phi_0\Big) \\ y(s) = y_0 + R\Big(\sin(\Phi_0 + \frac{hs\cos\lambda}{R}) - \sin\Phi_0\Big) \\ x(s) = x_0 + s\sin\lambda \end{cases} \tag{4.8}$$

where (x_0, y_0, z_0) is arbitrary and can be taken as the coordinates of the first point along the trajectory, Φ_0 is the angle between the z axis and the segment



(a) (b)

Figure 4.10: (a) Helicoidal motion of a charged particle in the magnetic field $\vec{B}$ which in the STT is parallel to the x axis in the ND hall (b) Projection of the helicoidal motion on the yz plane

connecting the centre of the circumference (y_C, z_C) and the point (y_0, z_0) in the YZ plane, R is the radius of the circumference, λ is the angle between the initial velocity vector $\vec{v}$ and the plane perpendicular to $\vec{B}$ known as the dip-angle and finally $h = \pm 1$ is the sense of rotation on the helix.
The projection of the trajectory on the YZ place thus describes a circle:

$$
\begin{aligned}
(z - z_0 + R\cos\Phi_0)^2 + (y - y_0 + R\cos\Phi_0)^2 &= \qquad (4.9)\\
(z - z_c)^2 + (y - y_c)^2 &= R^2 \qquad (4.10)
\end{aligned}
$$

Performing a circular fit on the hits recorded in the horizontal straws, on the YZ plane, we can get the most probable values for R and (y_C, z_C) for each particle trajectory.
We would now like, in order to have a complete description of the particle motion, to perform a reconstruction on the dip-angle λ. In order to do so we first approximate the circular motion in the YZ plane to a parabolic motion by applying a first order Taylor expansion in s/R. This is justified if $s/R << 1$, and thus if the portion of the circular trajectory described by the particle is small with respects to R; this is true for the energetic muons entering SAND. We also switch from Φ_0 to the angular direction of the track in (x_0, y_0, z_0) which we indicate as $\varphi_0 = \pi/2 - \Phi_0$. Finally we write (y_0, z_0) as $y_0 = d_0 \cos\varphi_0$ and $z_0 = -d_0 \sin\varphi_0$, where d_0 is simply the distance between the initial point and the centre of coordinates in the YZ plane. The equations

of motion then become:

$$\begin{cases} x(s) = -d_0 \sin\varphi_0 + s\cos\lambda\cos\varphi_0 + \frac{h}{2R}s^2\cos^2\lambda\sin\varphi_0 \\ x(s) = d_0\cos\varphi_0 + s\cos\lambda\sin\varphi_0 - \frac{h}{2R}s^2\cos^2\lambda\cos\varphi_0 \\ x(s) = x_0 + s\sin\lambda \end{cases} \tag{4.11}$$

We can now perform a coordinate rotation in the YZ plane of an angle φ_0, so that the new z axis, which we will call ρ, is directed along the track trajectory in (x_0, y_0, z_0) :

$$\rho = z\cos\varphi_0 + y\sin\varphi_0 \tag{4.12}$$

$$y' = -z\sin\varphi_0 + y\cos\varphi_0 \tag{4.13}$$

The equations of motion can then be divided between the plane $y'z$, longitudinal to the trajectory and the transverse plane ρz:

$$\begin{cases} z = z_0 + \rho\tan\lambda \\ y' = d_0 - \frac{h}{2R}\rho^2 \end{cases} \tag{4.14}$$

From the first equation we can see that performing a linear fit on the transverse plane gives us the best values for $\tan(\lambda)$ and z_0. In order to do so we use the information from the trajectory hits in the vertical straws, which give us the x and z coordinates. The y values are extrapolated using the R and (y_C, z_C) from the circular fit:

$$y = y_C + h\sqrt{R^2 - (z - z_C)^2} \tag{4.15}$$

The value of $h = \pm 1$ (+1 is clockwise, while -1 is counter-clockwise), is given accordingly by the sign of the product of the z distance between the first and second hit of the track and the y distance between y_C and y_0:

$$h = \frac{(z_1 - z_0)(y_0 - y_C)}{|(z_1 - z_0)(y_0 - y_C)|} = \pm 1 \tag{4.16}$$

In order to calculate ρ we also need the values of $\cos\varphi_0$ and $\cos\varphi_0$. We can calculate them as:

$$\cos\varphi_0 = h(y_0 - y_c)/R \tag{4.17}$$

$$\sin\varphi_0 = -h(z_0 - z_c)/R \tag{4.18}$$

If both fits were successful we now have estimates for R, (y_C, z_C) and λ. With this information we can reconstruct the initial transverse momentum of the particle on the YZ plane using the formula:

$$p_T = 0.3 \times B\cos\lambda \tag{4.19}$$

We can then get the three components of the momentum in the ND hall coordinate system as:

$$\begin{cases} p_z = p_T \cos\varphi_0 \\ p_y = p_T \sin\varphi_0 \\ p_x = p_T \tan\varphi_0 \end{cases} \tag{4.20}$$

Having the three momentum components we can then reconstruct the total momentum module, which in case of relativistic particles is about the same as the particle energy:

$$E_{reco} \simeq p_{reco} = \sqrt{p_x^2 + p_y^2 + p_z^2} \tag{4.21}$$