

A Simple Model of Muon Detection in the ANTARES Telescope and its Monte-Carlo-based Assessment

Yann Kempf, Université du Luxembourg¹
Supervisor : Jean-Pierre Ernenwein²

Internship report, Groupe de Recherche en Physique des Hautes Énergies,
Université de Haute-Alsace, July 2008

Abstract

In this paper I present the work done in July 2008, after my first year of Physics, during a four-week internship at the Groupe de Recherche en Physique des Hautes Énergies (High-Energy Physics Research Group) from the Université de Haute-Alsace, France, within the ANTARES collaboration. The ANTARES telescope is a neutrino detector based on the detection of the Čerenkov light emitted by the track of muons crossing the three-dimensional array of photomultipliers. I present the simple model I implemented, and how I then checked its validity through a Monte-Carlo analysis and a comparison with real data from the telescope.

Introduction

The ANTARES telescope (Astronomy with a Neutrino Telescope and Abyss environmental REsearch) is a photomultiplier (PM) array. Each 10-inch detector is set in a 18-inch (40 cm) glass sphere at a depth of 2.5 kilometers in the Mediterranean Sea, 40 kilometers off Toulon, Southern France. The optical modules are arranged in storeys, i.e. groups of three optical modules, along vertical strings. Each of the 12 strings, or lines, carries 25 storeys, and the lines are placed ca. 70 meters apart [Fig. 1a & 1b]. ANTARES was completed in June 2008.

High-energy μ -neutrinos very seldom do interact with matter, producing a muon. This particle travels at nearly the speed of light in vacuum, which is higher than the speed of light in the medium. This causes the matter which is on the path of the muon to emit light through what is called the Čerenkov effect. The refractive index of the medium being n , the Čerenkov light is emitted at an angle θ with respect to the muon's trajectory such

that $\cos(\theta) = \frac{1}{n}$ [1]. This gives for ANTARES an

angle of $\theta = \cos^{-1}\left(\frac{1}{1.34}\right) = 41.7^\circ$ ($n = 1.34$).

The muon can also lose energy by electron-

positron pair creation, or Bremsstrahlung, or other more complex interactions, which all provoke a shower of particles. These in the end emit a bunch of photons in all directions. Light is also detected from a natural background noise at a rate of ca. 100 kHz per PM, due to bioluminescence from organisms ranging in size from phytoplankton and zooplankton to macroscopic animals, as well as the radioactive decay of various

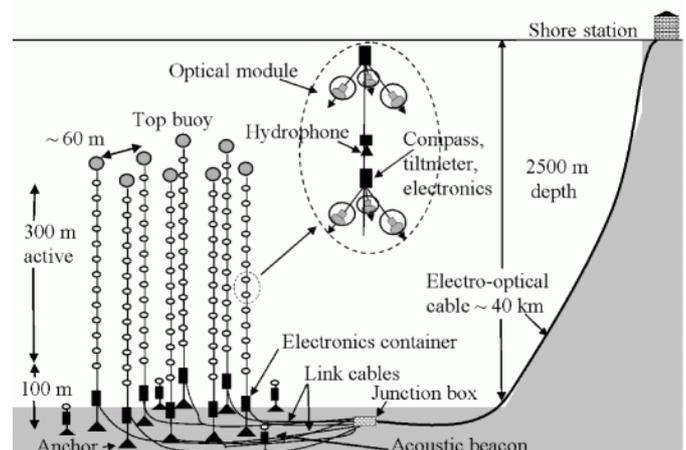


Fig. 1a : Diagram of ANTARES

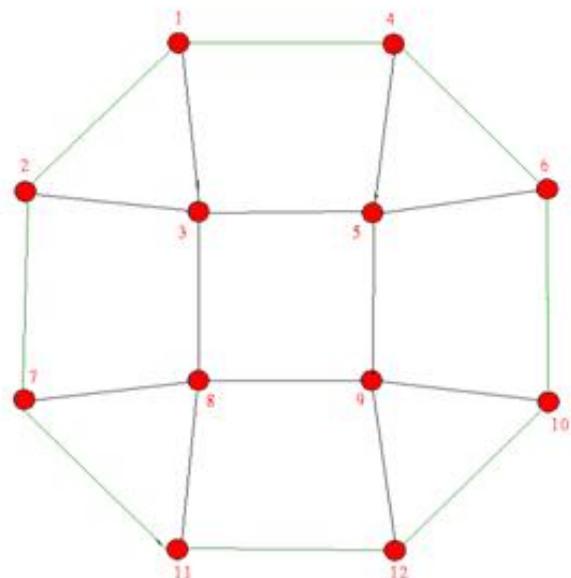


Fig. 1b : Layout of the lines (non-standard numbering)

isotopes (e.g. ^{40}K into ^{40}Ar).

Each optical module is equipped with a chip which sends out the hit data (amplitude and arrival time). These are collected by a processor on the storey, sent down to a junction box where all the lines join, and all data is sent to the shore via a standard glass fiber telecommunication link. On shore, the data stream is processed and events corresponding to a precise set of criteria trigger the recording of the stream for a brief period of time.

Many tools have been developed and are used to reconstruct the original muon's trace from the hit position and time dataset. One of them is the production of z-t plots, where the hits are plotted with their arrival time and their altitude.

To assess the quality of the modelling, I use two statistical variables, the number of hits per muon event and the number of photoelectrons per event. The comparison of histograms plotting these two variables for simulated runs and measured ones hints at possible malfunctioning of the model and/or allows to tune some of its parameters.

I. The Model and its Implementation

a. Muon Propagation

In my model, the muons propagate in straight lines through the water. I first developed a module creating the three-dimensional model of ANTARES and displaying it. I then made a module generating muons at random positions and with random unit speed vectors, either within or on a cylindrical surface containing the telescope, the 'can'. The muon is propagated by incrementing its position by one step at each iteration, where it emits light. The muon speed is nearly c , thus we take $\beta = 1$, whereas the speed of light at the location is c/n with $n = 1.34$.

Then I developed a functionality which allows to import muons from a file produced by a specific generator (Mupage), to give the muon stream a repartition which has got an astrophysical significance. But then I had to get rid of the can, which was the limit after which I stopped the propagation of a muon. Two criteria provoke the end of the propagation of a muon. Either its energy gets lower than 0.160 GeV, which corresponds to the limit below which it no longer emits Čerenkov light as it no longer travels at or faster than c/n . Or the muon no longer points towards any detector, which I discriminate based on its position and its direction with respect to the frame origin, set in the middle of the array. This way I can use muons emitted

from anywhere in space without wasting time in propagating a muon way beyond the detectors.

b. Photon Emission and Propagation

In the conditions I modelled, the track of a muon emits 350 Čerenkov photons per cm, in flight direction but with an angle θ to it as defined by formula [1]. I assumed they propagate with a uniform repartition on a cone, with its summit at the centre of the propagation step. When determining the angle between the track and the direction of a PM, I turned θ into an interval to account for the angular diameter of the PM as seen at the muon's distance.

The fraction of light getting to a detector is taken as the quotient of the detector diameter (10 in/ 25 cm) and the radius of the cone at the distance between the muon and the detector. This approximation holds for most cases, as long as the distance to the detector is large enough with respect to the radius of the sphere. Since the array is quite dispersed, I kept this model for my study anyway.

Of course, an absorption law is applied following a common exponential decay law, $N = N_0 \cdot e^{-\frac{l}{\lambda}}$ [2], where N is the number of photons, N_0 the initial number, l the propagation length and λ the absorption length. Here I assumed λ to be equal to 55 m.

Any type of photon scattering has been neglected to keep the model simple and its implementation within the possibilities of a four-week internship. Therefore the photons travel at c/n , in straight lines, and without any

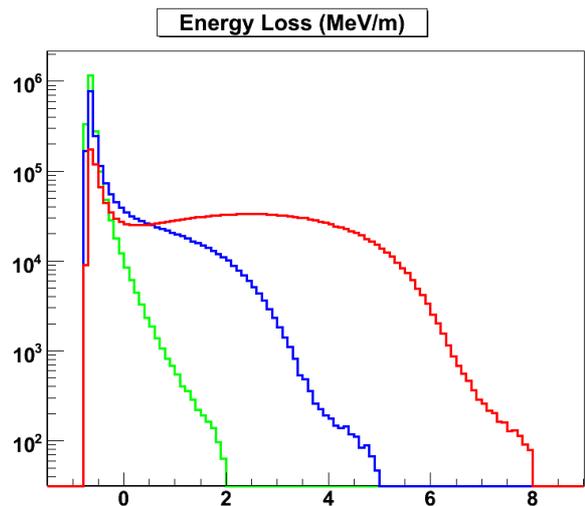


Fig. 3 : Energy loss (MeV) per meter, logarithmic scales. (green = 100 GeV, blue = 100 TeV, red = 100 PeV) These are the (unscaled) probability repartitions of the energy lost by the muon in one meter. Obviously a high-energy muon is likely to lose larger amounts of energy.

delay.

Initially I assumed a noise rate (bioluminescence and radioactivity) of 60 kHz for each module, but the large difference with the real data hints at larger values, more likely to lie around 100 kHz or more. This is modelled by allocating every detector for each event a time window ranging from the earliest hit minus 500 ns to the latest one plus 500 ns. Within this window, at random times drawn with a flat probability distribution, a number of individual photons, drawn randomly following a Poisson distribution taking the expected noise rate as an argument, hits each PM.

c. Showers

The muon loses energy by emitting photons, but

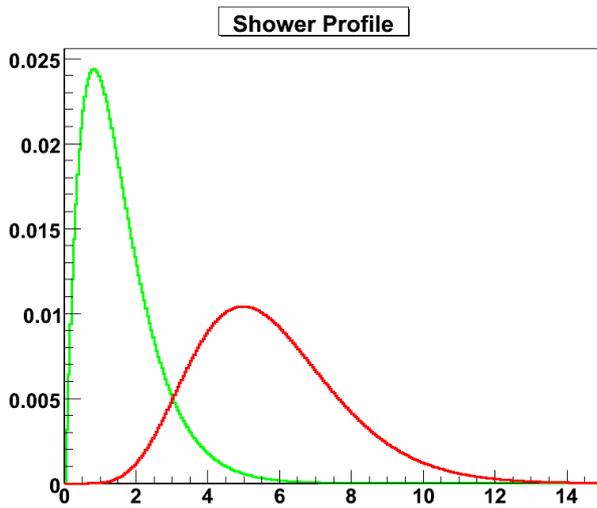


Fig. 4 : Normalised distribution of the photon emission along the shower (abscissa in m), green is a lower energy than red. A large energy loss produces a longer and more uniform shower profile.

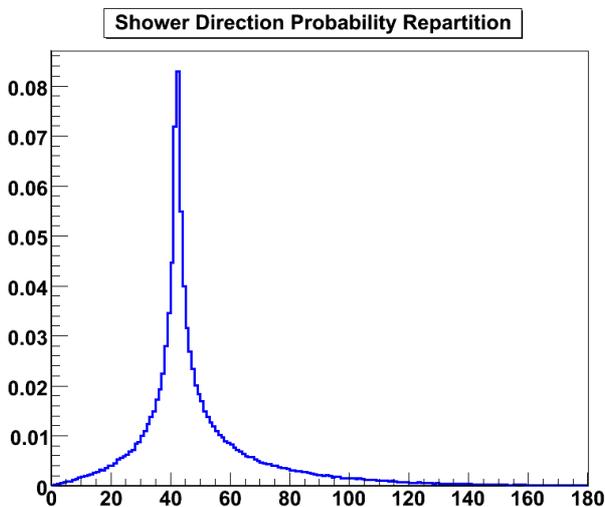


Fig. 5 : Probability repartition of the emission direction in degrees, with respect to the muon track direction.

as energy is equivalent to mass following $E = m \cdot c^2$, sometimes charged particles form along the trajectory through electron-positron pair creation or other mechanisms. Bremsstrahlung of course also occurs. All this triggers a shower of particles and it ends up in a bunch of photons being emitted along a certain length, in all directions.

The energy loss per unit length is shown on Figure 3. It depends on the initial muon energy. Depending on the amount of energy lost, the profile of the shower is different. This is easy to see on Figure 4. Finally, Figure 5 shows the angular distribution of the photons coming from the shower. The most probable direction is the same as for the Čerenkov light, however a much larger interval is possible. See the figure explanations for more details about the energy dependences.

d. Emission Implementation

The practical implementation takes place in two steps. The track is incremented by one step (typically one meter). The energy loss along this step is computed, and then the step is divided in substeps (typically 10 cm each). The Čerenkov photons are emitted in each substep (at a rate of 350 per centimeter), and a loop over the 900 PMs determines the time and amplitude of hits, after the angle, the effective detector angle (see next section) and the absorption factors have been applied. Then, from the energy loss the shower length is calculated, cut into the same substeps and the same iteration as for the original step is performed along the shower, using the aforementioned functions of profile and angular distribution [Fig. 4 & 5]. Then the next step is taken, and so forth.

e. Photon Detection

Obviously, the PM is not a perfect theoretical device, therefore some of its characteristics must be taken into account in the model to yield somewhat realistic results.

First of all, each detector is pointing in a precise direction, as the pointer segments in Figure 6 show. I included a function, obtained from experimental measurements made in Genoa, Italy, which weights the amount of light getting through to the actual PM in function of the angle of incidence.

This number must be weighted by a coefficient rendering the quantum efficiency and the absorption of the glass and gel layers in which the detector is embedded. In my model, the default value is 0.15.

However, the obtained decimal number cannot render an integer number of photons, so I apply a

Poisson distribution to the number, to make it an integer. It is then randomised again using a Gaussian, to account for the distribution of the charge the impulse from a photon gives after the amplifying dynode cascade in the PM.

The processor detecting the signals at the back of the dynode was a hard nut to simulate properly in the program. It contains two chips which work alternately. One chip has got a recording window (ca. 40 ns), during which it sums every hit occurring after being triggered by a signal higher than the equivalent of 0.33 photoelectrons. After that, it needs ca. 250 ns to send on the data (hit amplitude and start time), during which it

cannot detect, but during which the second chip can process a signal in the same way if it is triggered. If anything happens during the dead time of both detectors, it is lost, but anything in a reading window is integrated. This double system brought some trouble, but is finally working fine.

f. Trigger Logic

The “all data to shore” principle implies that everything coming from the array is sent to the computing center on shore to be processed. Nevertheless the background alone would swamp the data storage. That's why trigger algorithms have been

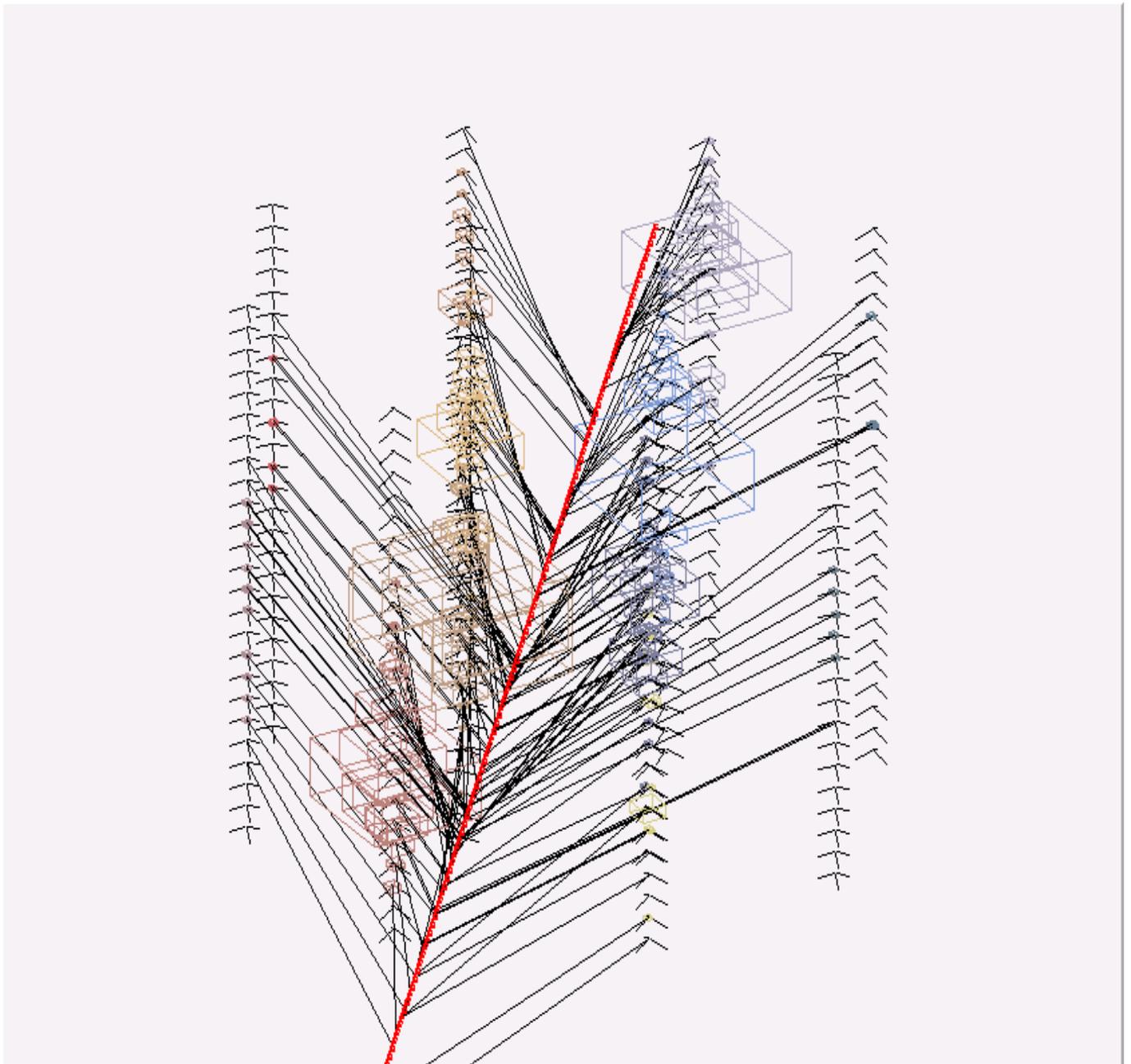


Fig. 6 : 3D graph of a simulated hit (without noise, showers or electronics). Red : muon track (originating below the bottom). Black lines : detected photon paths. Black segments : each gives the pointing direction of a PM. Boxes : size proportional to the number of photons hitting the PM.

devised to try to discriminate the data containing physically meaningful signal. The main trigger algorithm detects so-called L1 hits, and a causally linked group of at least 5 of these L1s causes the system to record a time window framing this bunch of hits. In the end, about one million muons are saved by the detector every day.

I developed a function to apply this algorithm, and in the end the output of my simulation program only includes events which have yielded those minimum 5 L1 hits. In that my trigger logic is simpler than the one usually applied in ANTARES, which adds causality.

Two criteria are decisive to flag a hit as being a Level-1 hit. Either it has got an amplitude higher than 3 photoelectrons, or (inclusively) it is coupled to at least one other hit on the same storey which is causally linked. The latter condition is expressed as two hits being linked if they are separated by less than 20 ns (the distance separating two modules on a storey at c , plus a comfortable margin). This allows to exclude most of the simulated events in which only a few dozen background hits appear and no physical photons are recorded.

Remember that the noise photons are simulated as being detected individually, and the probability for pure noise to provoke 5 causally linked L1s is very low.

However more stringent conditions are often instrumental in getting useful data. T2 hits are compound hits of L1s. Basically if two adjacent storeys on a line give L1s, it is considered a set of T2 hits. More elaborate versions can include causality, i.e. a time interval lower than 100 ns (again distance times c plus a margin). The selection criteria can rely on a certain number of T2 events instead of simple L1s. Unfortunately I came short and did not have the time to implement a T2-level trigger function.

g. Output

My work on the simulation went stepwise, my picture of the phenomena involved in simulating the muon detection in ANTARES getting more comprehensive at each step as I was introduced to them. Therefore the structure of the programs I developed was very modular. For each step, I developed first separate macros or programs. I then integrated them as functions

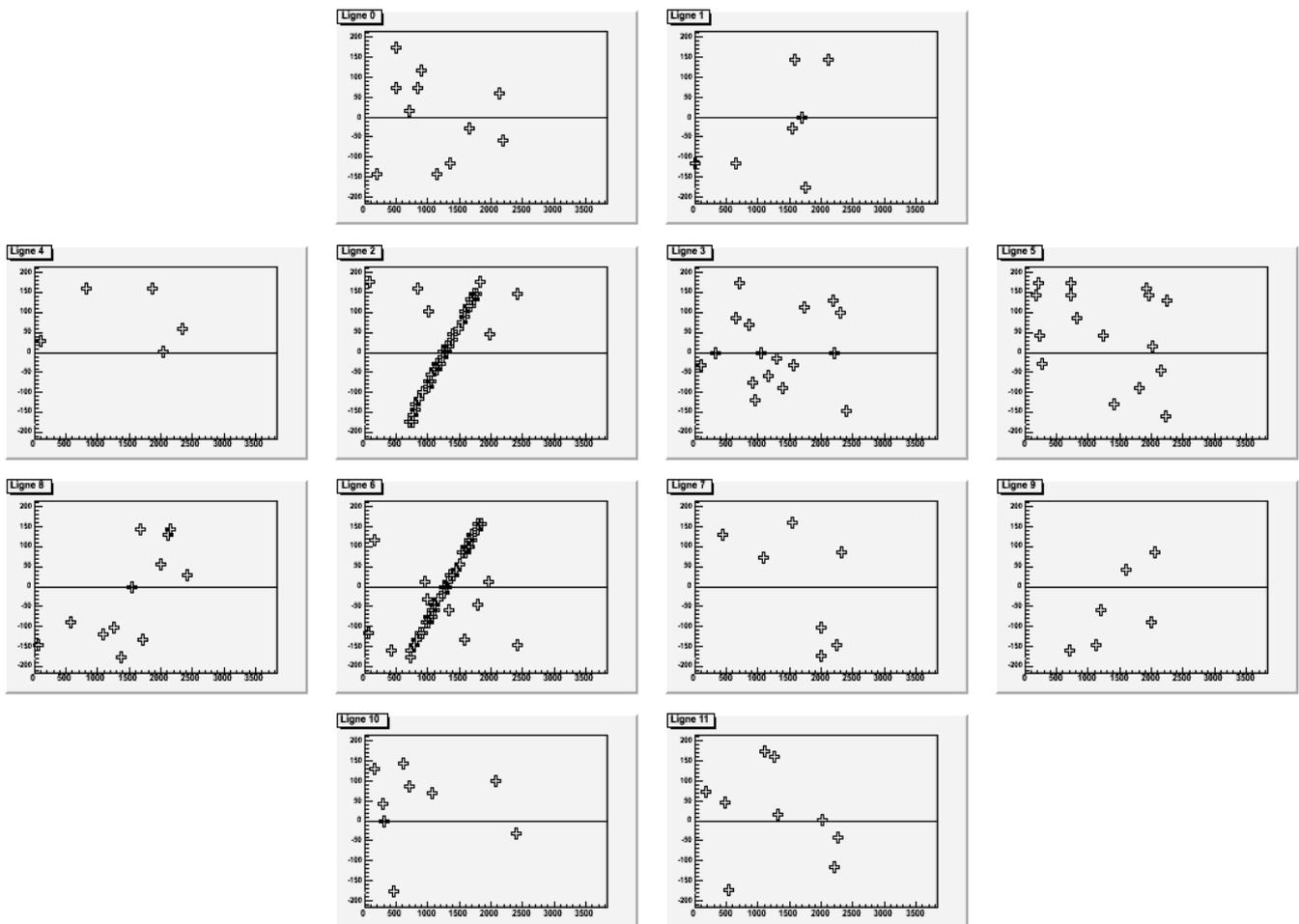


Fig. 7 : Example of z-t plots, with background noise, for a simulated vertical ascending muon in the vicinity of lines 2 and 6. The correlated hits are clearly visible.

of the larger program doing the simulations.

At the beginning, I started by creating macros which, from a coordinates input file, create and display a three-dimensional model of ANTARES, using the C++ classes provided by the ROOT software (from CERN). Along the way I learnt the basics of C++ programming, my previous experience being limited to Basic programming on a graphic calculator and the reading of a C++ tutorial. I went on with the first parts of muon and photon propagation, without the natural background and electronics, but with a 3D and colour display of the simulated tracks and hits, including real-time progress and colour-coded hit amplitude display. See Figure 6 as an example of that part. Further, I implemented the muon import (from Mupage export files), the bioluminescence function and the more elaborate stopping condition, quitting the time-consuming graphical display along the way. In the end came the electronics modelling, the important shower simulation part and the trigger logic function too, which had proved more than necessary from a 300,000-events run performed over 60 hours. Indeed the histograms without showers and L1 selection have got a shape which does not fit at all with real data. The complete version of the program, which works fine after a first aborted run due to a programming fault from me, was then used for a Monte-Carlo simulation, with several 1-million-events runs performed. This part of the project unfortunately happened only at the very end of the four weeks, once the program was complete and performing well, so we had only very limited time to tweak parameters to try to get as close as possible to the real data from ANTARES.

As I explain in the following parts, I also developed programs to build plots and histograms, with the aim of assessing the results of the Monte-Carlo simulations through graphical and statistical analysis.

II. Trace Reconstruction

One of the main goals of neutrino detection, the detection of point sources, relies on the reconstruction of the muon trace from the hits, to be able to compute the original neutrino's trajectory through space. First of all, it is necessary to calculate the local track of the muon through the detector, which is a tough task, and still is an important subject of research as none of the methods applied today give both a good accuracy and a good identification rate.

An interesting way to visualise the progression of the hits within the array is to plot, for each line, the z and t coordinates of the hits, and to arrange the display of the graphs following the detector layout [Fig. 1b]. By doing this via a program importing the hit data from the

output file of the simulation program, I was able to make helpful observations on the way towards trace reconstruction. See Figure 7 for a clear example of such a plot table.

The direction of a muon and its distance to the line considered are two factors influencing very much the shape of these plots. When a series of hits is physically correlated because it is the mark of a single muon, their z - t plot shows them characteristically grouped. The group is always wedge-shaped, with the opening toward positive t . The nearer the track is to the line, the sharper is the wedge. The further away, the more it looks like a half-circle. A purely horizontal muon gives a horizontally-oriented and symmetrical wedge. A descending one gives a wedge with a vertical upper arm, an ascending muon gives a wedge with a vertical lower arm. If the muon is purely vertical and ascending, respectively descending, the vertical part of the wedge disappears, and the corresponding plot is a single oblique segment with a positive respectively negative slope. Moreover, by arranging the z - t plots in a layout similar to Figure 1b, one can get an insight into the progression of the muon through the array, i.e. an approximate azimuthal (x - y) coordinate of the muon's speed vector (See Fig. 7).

Although being a key part of the ANTARES data interpretation, the trace reconstruction was not the main part of my work, and I did not have the time to further this research.

III. Event Generation, Simulation and Data Comparison

a. Events

The definition of an event is slightly varying, depending on the side from which one considers the problem.

During an observation run, an event is defined by a precise trigger logic, e.g. a set number of L1 hits occurring in a brief time slot. Its occurrence causes the data processing chain to store the signal during a period comprising the trigger signal. Plenty of signal is discarded when nothing considered a signature of something physical appears. If this were not done, tremendous amounts of data would have to be stored, and then sifted through for interesting things. It would be an extremely difficult task, be it only due to the sheer quantity of material there would be.

In my simulations, an event is defined as the effect one or several muons flying more or less through the array have on the PMs. In the self-generated or forced (set coordinates) modes, an event consists of the

data produced by a single muon (with the noise added, and the effect of the electronics applied, of course). In imported mode, where I take the muons from a physically meaningful flux simulated by the Mupage software, an event can have one or several muons with different origins, unit speed vectors and initial energies in a brief time slot.

The exported data however mime real data in that an event needs to comply with the trigger logic I implemented to be saved (also see above). This means any event producing less than 5 L1s might well have happened, but won't be present in the output files. This in fact is the case for the vast majority of events. In the end, when all software modules work properly and are switched on, and large numbers of input events are imported from Mupage and simulated, only about 25 to 30 % of the input events yield at least 5 L1 events, thus getting through and being stored.

b. Monte-Carlo Simulation

The low rate of saved events explains why 1-million-events runs (i.e. input events) are needed, to be able to perform proper Monte-Carlo analysis. The end-

version of the program, finally completely debugged after a pre-run on a local machine, was uploaded to the IN2P3 (Institut National de Physique Nucléaire et de Physique des Particules, National Institute for Nuclear Physics and Particle Physics) computing centre in Lyon, France, and executed there in 10-hour, twenty-fold runs to give 1 million simulated events each time.

We chose two statistical variables which are significant in the assessment of the model, and which are very easy to compute, as the basis of our Monte-Carlo. These are the number of hits and the number of photoelectrons (i.e. the sum of the amplitudes) per event. Therefore my simulation program creates two output files, one comprising for each event the individual hit data (PM, amplitude and time), useful for drawing the plots (see above and Fig. 7), the other comprising one line per event giving its number, its number of hits and its total amplitude. The latter file is the one I use to fill the histograms representing the variables (see below and Fig. 8).

I also used several scripts to process real data files, to create for a couple of real runs the pairs of files I mentioned above. This way we are truly able to compare

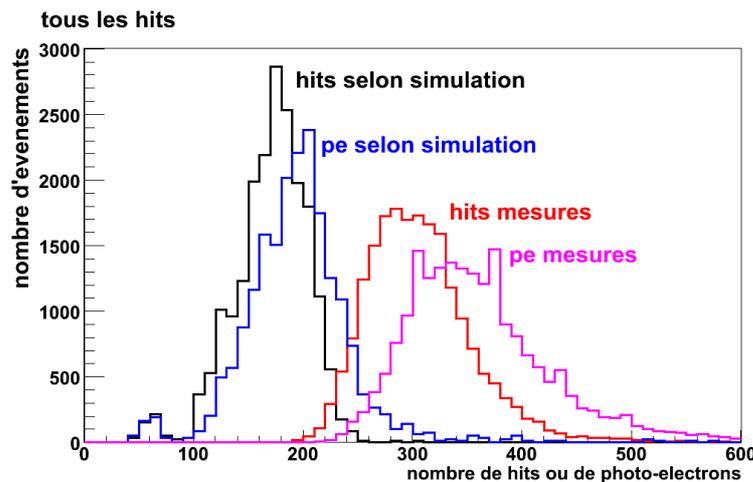


Fig. 8a : Complete datasets, black and blue simulated, red and pink measured.

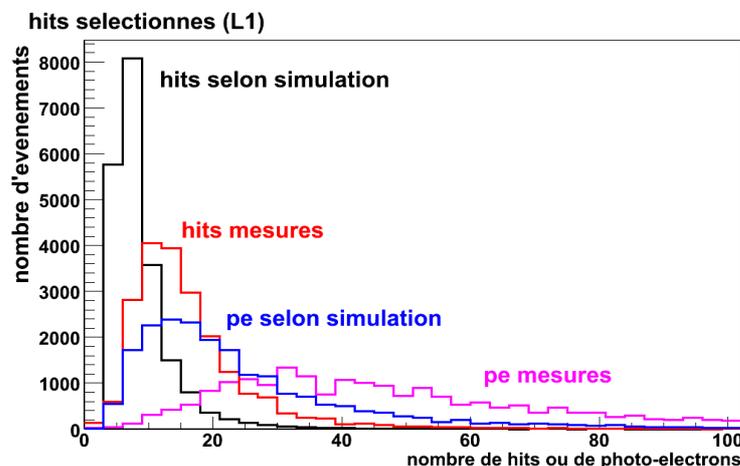


Fig. 8b : L1-sorted datasets, same colour-code as 8a.

real vs. simulated data, in order to assess the validity of the simulation model, and to adjust its parameters in such a way that the best correlation is achieved.

c. Results

Several aspects of the histograms obtained are telltale and hint at what needs adjusting. My first histograms from a 300000-event run performed over 60 hours, at a point where the showers weren't implemented yet, show a double structure. A first large maximum has got the shape of a Poisson distribution, at around 60 hits per event, and a very broad but extremely low Gaussian peaks at about 130 hits per event. This contrasts starkly with the data histograms, which do not look like that at all. They have got a much more symmetrical shape, as is conspicuous in Figure 8a, red and pink curves. The same observations hold for both variables.

Obviously many events are simulated and saved that do not appear in the data. The first "bump" corresponds to all the hit-free events, that only get a few dozen noise hits within a 1000 ns window. These can be removed largely by sorting out any event which does not comply with the trigger logic, which I then added. As the data come from the telescope and not from another simulation, they do not include pure noise events, as it is explained above.

Moreover, a large absolute quantity of hits is missing in this set. Therefore we added a fundamental feature to the model; the particle showers and the large numbers of photons associated with them.

These two new elements have a big effect on the results. First of all, the major overwhelming peak due to the noise-only events is dramatically reduced, but a first minor "bump" is still visible (see Fig. 8a, blue and black curves, on the left). This shows a few of these get through the 5-L1s condition, and it underlies the importance of a more severe selection through a T2 criterion. Second, the now major part of the histogram, which shows the distribution of the events of interest, is translated to higher values, as is expected from the showers. The histograms become more similar in overall shape. Yet the position of the main peak still shows a large discrepancy with the data (see Figure 8a).

This difference comes from the large uncertainty in and also the natural variation of the value of the background noise rate. Indeed I started with a default value of 60 kHz·PM⁻¹, whereas this would rather correspond to a minimum value on-site. Two methods can be considered to cope with this. On the one hand, many Monte-Carlos can be done, each with a slightly different background noise rate. It is a rather unpractical method, which takes a long time to perform.

And it can hide other potential factors, as at some point the result is likely to match the data despite possible faults within the program or the model itself. On the other hand, if the simulation and the data are rid of the noise hits both in the same way, the comparison gets much more effective. One more argument is the number of photoelectrons per hit on average. Figure 8a clearly shows the ratio is near 1, which is a sign that the majority of hits comes from 1-photoelectron hits, i.e. background noise hits. That's why I developed a script based on exactly the same function as the one which flags the hits as L1s in the simulation. It sorts out the L1 hits from a hit file (simulated run or real data run) and produces the same two file formats as before, but only with L1 hits. Then the same amount of noise L1s gets through and overall the noise is almost erased, hiding away the difference in noise rate between the simulation and the data. This is especially useful since the actual rate during an observing run is measured precisely, however it fluctuates in space and time and is very hard to render effectively in simulations.

That bit of processing reduces the difference between the Monte-Carlo and the observed data sets. See Figure 8b, which shows the result of the L1 sorting. The shape of the histograms clearly is of the same type of distribution, except for the remaining offset between both. The photoelectrons per hit ratio becomes much larger, indicating the higher intensity of the light observed, and underlining its physical meaningfulness. Nevertheless the model still lacks a number of photons, and that cannot be added simply by tweaking external factors.

Remark : One might object I used ANTARES data from its former configuration, i.e. with 10 operational lines. It is not influencing fundamentally the results, and does not account for the differences pointed out in the previous paragraph. Indeed most events produce light and showers in the detection zone of a few lines. Virtually never does a muon leave numerous L1 hits on every line. Therefore I did not reduce the simulated detector to the old ANTARES configuration.

Interpretation

The Monte-Carlo analysis is indeed a good tool to assess the model developed for the muon detection in the ANTARES neutrino telescope. It allows us to draw two main interpretations about the results obtained.

By the end of the internship the model reached a state in which it performs stably and with no apparent major flaw. This is the fruit of its stepwise and modular build-up, as many minor bugs occurred when testing each block individually and in various conditions. For instance the 3D graphical views or the trace

reconstruction plots helped to reveal glitches in the maths employed. Of course the picture is not yet exhaustive, and does not pretend to be, but it takes the main features of ANTARES and the muon detection process into account.

More specifically, the Monte-Carlo analysis helped to develop the model once the basic simulation was running, and then to focus on precise aspects to optimise the comparison between real and simulated data. It also made clear the model is performing quite well qualitatively, yet not quite fitting quantitatively. Further research to adjust inherent parameters is needed to try to bring the simulation closer to observations. The observed differences can be interpreted as a combination of the influence of the modelling of the PM's angular acceptance, the incomplete shower description and the remaining natural background noise hits.

Conclusions

Thanks to this challenging and exciting internship, I gained a good overview on muon detection in ANTARES, as well as muon trace reconstruction and Monte-Carlo simulation. I developed, under the guidance of Jean-Pierre Ernenwein, a relatively simple model of how ANTARES works, and how the muon signature is detected by it. I implemented it in C++ language, taking advantage of the classes and functionalities available with the ROOT software from CERN, learning to use both of them on the way. Finally, I learnt the basics of Monte-Carlo simulations. Thus I was able to assess my model through the analysis of the results of large simulations, and the comparison of these with *in situ* data from the detector.

In addition to the technical and science knowledge I gathered in Strasbourg, I also made my first research experience in a Physics lab at the GRPHE. It proved very instructive, as I was confronted to all levels of research and its organisation, and I met many researchers who willingly introduced me to the world of

research. On a personal level, I think this is the most important thing I gained there, and it definitely comforts me in my wish to go on and become myself a physicist. It was very stimulating, and I am glad I was given this opportunity as early as after my first year of bachelor's degree.

Acknowledgements

I would like to thank Jean-Pierre Ernenwein, now Professor at the CPPM (Centre de Physique des Particules de Marseille, Centre for Particle Physics of Marseille), for supervising me during my internship, and teaching me along the way all I needed to complete my tasks.

Many thanks also to everyone from the GRPHE and the IPHC (Institut Pluridisciplinaire Hubert Curien, Hubert Curien Pluridisciplinary Institute) in Cronenbourg, France, and especially the (now retired) director of the GRPHE René Blaes, for welcoming me among them for a month, providing me with a lot of advice, and above all giving me a promising taste of the world of research.

Thank you also C. Kempf, S. Neveu and A.-L. & J.-P. Kempf for successively providing me with accommodation in the vicinity of Strasbourg during the internship.

Contact

1. Yann Kempf, Student in the SaarLorLux trilateral Physics course from the French-German University, Luxembourg (L), Saarbrücken (D), Nancy (F), Semesters 1 & 2 in Luxembourg in 2007-2008

yann.kempf.001@student.uni.lu

2. Jean-Pierre Ernenwein, Physicist now professor at the Centre de Physique des Particules de Marseille (CPPM)

rnenwein@in2p3.fr