Development of a Software Framework for the ANTARES Acoustic Data and Simulations within the Framework

Diplomarbeit aus der Physik

vorgelegt von
Alexander Würstlein

Contents

1 Introduction

1.1 Motivation

Astronomy is the oldest branch of what is known today as physics, yet together with many modern advances in all fields of science, our understanding of the universe has taken a significant leap in the last centuries. One of the most important discoveries in the field of astronomy was the discovery of "Höhenstrahlung" by Victor Hess[1] in 1912. In the decades following that discovery whole new fields of astronomy were created exploring non-visible parts of the electromagnetic spectrum and particles other than photons. Astroparticle physics studies those particles and high-energy gamma rays to learn about new aspects of astronomical phenomena and to look at parts of our universe where other methods are blind.

One area of interest that will be covered in this work are neutrinos at ultra-high energies. Due to their non-interaction except through the weak nuclear force and their very small mass, neutrinos are expected to be very hard to detect. On the other hand, their small interaction cross section also enables neutrinos to pass through relatively dense regions without interaction, potentially carrying information from previously invisible domains. Therefore multiple experiments engage in the search for ultra-high energy neutrinos and in the localisation of point-sources of those neutrinos.

Neutrinos with ultra-high energy (defined as energies above $10^{18}\,\mathrm{eV}$) are very interesting because neutrinos travel directly along a virtually straight line[2], so that the source of the neutrino can easily be determined. But currently, it is unknown what the sources of such ultra-high-energetic neutrinos could be. Possibilities are for instance active galactic nuclei (short "AGNs"), supernova remnants or other phenomena capable of accelerating particles to very high energies.

An interesting question in that context is also the GZK cutoff[3, 4]. When protons reach a certain energy threshold (above $6 \cdot 10^{19}\,\mathrm{eV}$), the proton loses energy in reactions with photons from the cosmic microwave background. The proton reacts with the photon forming a Δ^+ resonance which decays into a neutron or proton and a positive or neutral pion. This puts an upper limit on the energy of protons in the cosmic radiation, if the radiation source is not closer than $50\,\mathrm{Mpc}$. While AGASA does not seem to see a

cutoff in that energy region[5], the HiRes and Auger experiments[6, 7] have published results that suggest the existence of the GZK cutoff.

The GZK effect does, amongst other particles, also produce neutrinos through the decay of charged pions, which would then reach us almost without further interaction or deflection. Therefore the detection of neutrinos from those decays would be further confirmation of the GZK cutoff. These GZK neutrinos would then also point towards the sources of ultra-high energetic particles.

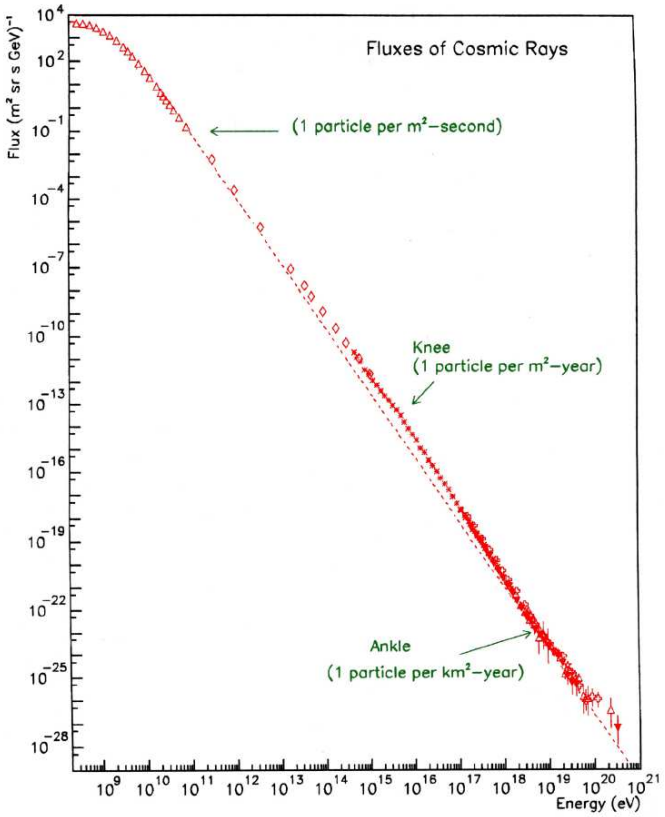

Figure 1.1: Flux of cosmic particles over energy. From [8].

Since the cross-section of a neutrino reaction is very small and the fluxes in the energy regions where the detector would be sensitive are very low (see figure 1.1), it is necessary to scale up [9] the detector mass and volume to arrive at workable expected detection rates. The detector volume of course needs to be instrumented which is where two important factors come into play, the detector medium and the detection technique.

One promising method among others is the detection of the acoustic emission of particle showers in water. Media which are easily available in very high volumes are

water, air, ice or salt, and appropriate detection techniques include the optical detection of Cherenkov photons generated by the shower, the acoustic emissions of the shower due to heating of the medium and radio emissions from the shower. Of course not all of these combinations of media and detection techniques are practical or even possible.

While for example in a water medium, with the attenuation length of light being in the range of a hundred meters, the attenuation length of sound in the relevant frequency band is an order of magnitude larger, namely in the range of a kilometer. This allows for a larger spacing of detector elements which in turn allows a larger instrumented volume with the same number of detector elements.

Also, acoustic detection requires simpler sensors and read-out electronics compared to the optical sensors. While in the optical detectors, photomultipliers requiring high-voltage supply and read-out electronics with nanosecond time stamping must be used, acoustical detectors can be built from industry standard hydrophones and signal processing electronics capable of handling signal frequencies of no more than about $100\,\mathrm{kHz}$.

Therefore an acoustical detector will be less complex and probably more reliable and easier to design and build than an optical detector. This reduces the cost for the detector planning and construction as well as the running costs of detector operations. The excessive costs make it improbable that large detectors with volumes of $10\,\mathrm{km}^3$ and more will be built using optical detection methods. However, alternative methods like acoustics i.e. using seawater as medium must prove their potential.

Within this work the ANTARES detector and its acoustical component AMADEUS will be discussed, both located in the seawater medium of the Mediterranean sea. After some general introduction of the necessary concepts of the detector and of acoustic particle detection, SeaTray will be introduced. SeaTray is a software framework for neutrino telescopes in water, which was extended in the course of this work to support the simulation of AMADEUS. The simulations and the modules that produced them will be introduced.

The demonstrated simulations will be a step towards the calculation of the detection efficiency and energy threshold for the AMADEUS detector and thereby the establishment of an upper limit on the neutrino flux.

1.2 ANTARES

The ANTARES[1] neutrino telescope [11] is located in a depth of about $2400\,\mathrm{m}$ in the Mediterranean sea near Toulon. Primarily ANTARES was constructed to employ the optical Cherenkov technique for detecting showers. To that end, ANTARES consists

[1] Acronym for **A**stronomy with a **N**eutrino **T**elescope and **A**byss environmental **RES**earch.

of 12 "lines" anchored on the sea floor straightened by a buoy. The standard lines, numbered L1 through L12, have 25 storeys each[2]. A storey consists of a so-called local control module (LCM) containing the power supply and data digitization and transmission electronics, and three photomultipliers inside glass spheres, so-called optical modules (OM). The photomultipliers detect the Cherenkov photons generated by particle showers, as well as of course other luminous phenomena that occur in the detector medium.

The "backbone" of a lines is a cable containing electrical and optical connections which power the detector elements and transmit data from and to the various components. The transmission of data is accomplished over a fibre-optic network connecting the detector with the shore station where an array of computers filters, processes and stores the transmitted data stream. Also a clock signal is generated on-shore and distributed throughout the detector to enable very precise timing of events.

In addition to the optical detection, ANTARES consists of several further components, monitoring the environment, the detector and performing various other tasks. Since the ANTARES lines are not rigid structures, the sea currents bend and twist the lines. Therefore an acoustic positioning system, consisting of "pingers" located in the anchors at the bottom of each line and of positioning hydrophones on each line regularly measures the geometry of the detector to a required precision of $20\,\mathrm{cm}$ or better[12]. A compass and tiltmeter in every storey measures the orientation of that storey. Other devices ensure calibration of the optical modules, measure temperature or sea current or provide data for biology and geophysics.

ANTARES has been recording data with the first detection line since 2007, since May 2008 the detector was completely deployed. ANTARES contains the acoustical AMADEUS detector which was completed together with ANTARES.

[2] L12 has only 24 storeys.

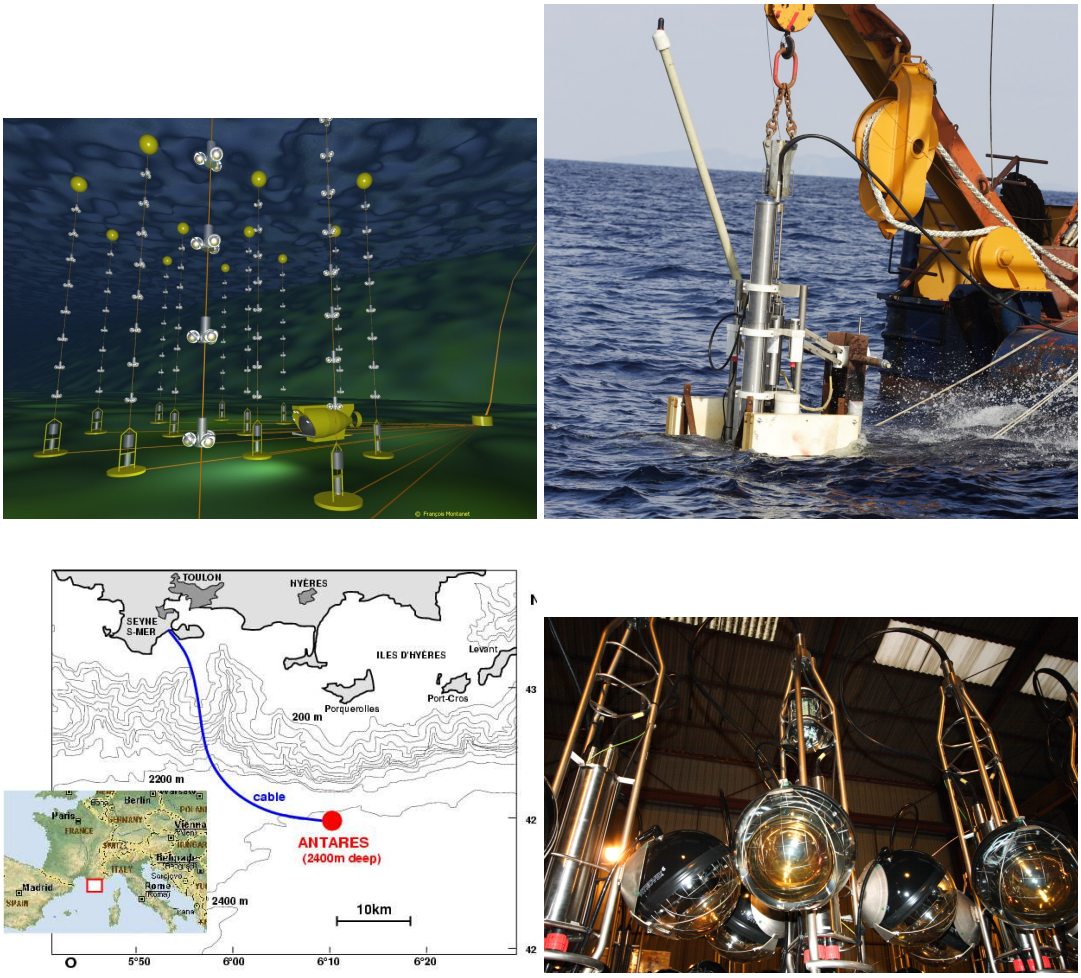

Figure 1.2: Artist's view of the ANTARES detector, an anchor which fixates a line on the sea floor during deployment, a map of the ANTARES location and a closeup of optical ANTARES storeys after the assembly of the line [10].

2 Acoustic particle detection with AMADEUS

2.1 The AMADEUS detector

AMADEUS[1] system[13] is a system inteded to study the feasibility of an acoustic neutrino detection system and the maritime acoustic environment, integrated with the ANTARES detector.

AMADEUS consists of six clusters of hydrophones on two different lines of the ANTARES detector, line 12 (L12) and the instrumentation line 07 (IL07) as shown in figure 2.1. Each cluster or storey of hydrophones comprises six hydrophones spaced approximately $1\,\mathrm{m}$ apart. The lines L12 and IL07 are located $240\,\mathrm{m}$ apart from each other, the individual storeys have the standard ANTARES spacing of $14.5\,\mathrm{m}$ except for the topmost storey of IL07, which is about $125\,\mathrm{m}$ above the closest acoustic storey on that line.

Several different types of acoustical sensors are used in AMADEUS, in addition to off-the-shelf hydrophones from a commercial manufacturer there are also some hydrophones built by the AMADEUS working group as well as so-called "acoustical modules" (AMs), which are glass-spheres used for the ANTARES photomultipliers equiped with piezo-electric glued and thus acoustically coupled to the glass. In this work the term "hydrophone" is used to denominate all these different types of acoustic sensors.

The LCM of each storey contains power supply and readout electronics for the hydrophones. The signal from each hydrophone is first amplified and band-pass filtered in an analogue signal processing part. It is then digitized by an analogue-digital-converter (ADC) into $16\,\mathrm{bit}$ values, digitally downsampled and packaged into a certain data format.

The data is usually downsampled by a factor of 2 from an ADC sampling rate of $500\,\mathrm{kHz}$ to an output sampling rate of $250\,\mathrm{kHz}$. This is still a sufficient sampling rate for the peak in the spectrum of the expected signals between 10 and $50\,\mathrm{kHz}$. If necessary, downsampling factors of 1 and 4 can also be configured. The purpose of this downsampling is the reduction of the data rate sent to shore and the amount of data which

[1] Acronym for **A**NTARES **M**odules for the **A**coustic **DE**tection **U**nder the **S**ea.

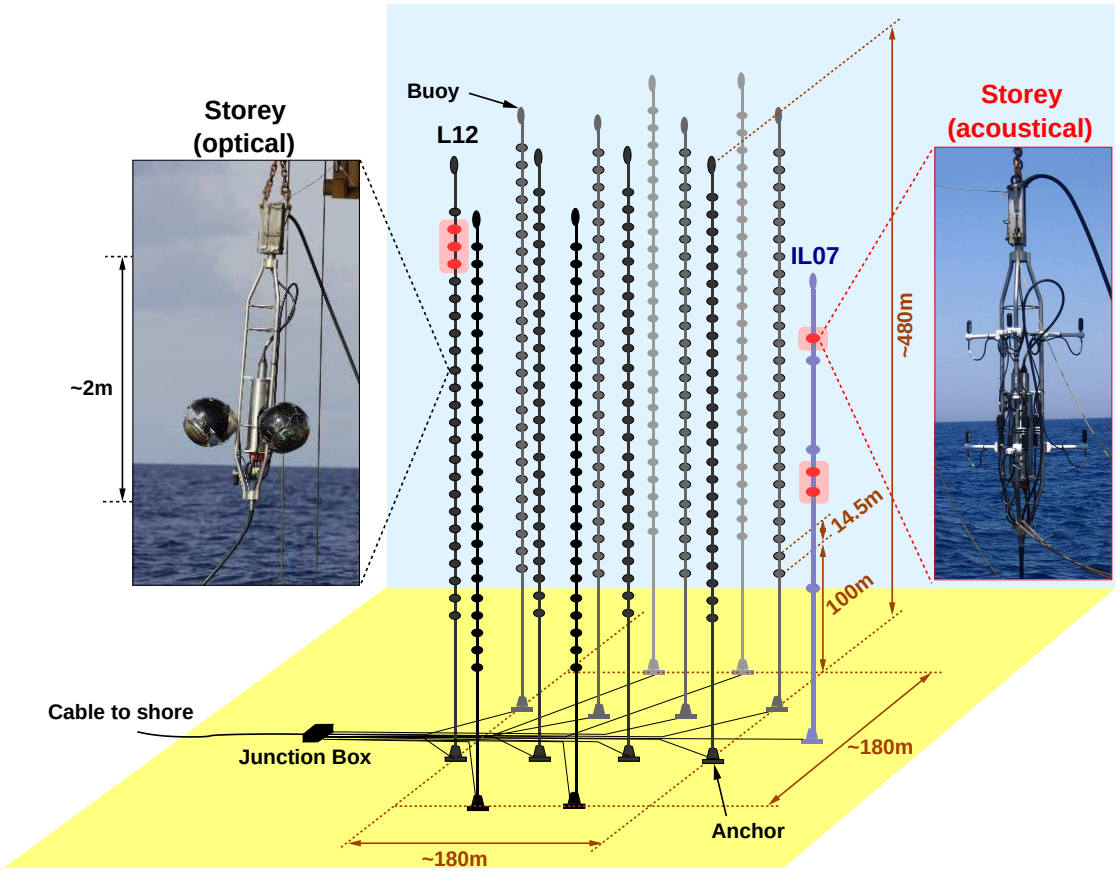

Figure 2.1: Overview of the ANTARES detector with the AMADEUS storeys marked red. In the right photo an AMADEUS storey is shown with six hydrophones at the ends of white horizontal support structures. The electronics for each storey are contained inside the central cylindrical element called LCM.

has to be filtered and stored. Aliasing effects from the downsampling are suppressed by a finite impulse response (FIR) filter.

The data format is designed to be as compatible to the optical signal formats as possible, so that many standard ANTARES parts, protocols and software components can be used. 128 digitized values are packed into such a so-called "AWFHit"[2], together with the configuration string for the LCM which sets amplification and downsampling parameters and a time stamp generated from the ANTARES clock system. Since the

[2] AWFHit[14] is an abbreviation for "**A**node **W**ave**F**orm **Hit**", which is a data format originally intended for signals from the photomultipliers which are not integrated off-shore but sent as waveforms of electrical current over time.

readout of values from the hydrophones is not triggered off-shore, a continuous stream of all data is sent to shore via standard TCP/IP[15, 16] over the ANTARES optical fibre network.

With a data rate in the order of $1.5\,\mathrm{TB/day}$, most of which is continuous background noise, storing all the data produced would be neither be practical nor sensible. Therefore measures have to be taken to further reduce the amount of data before writing it to disk. Dedicated servers on-shore receive the downsampled but unfiltered data stream and apply several triggers[17], meaning they filter out data which is to be stored permanently.

There are triggers with different criteria and purposes: First a so-called minimum bias trigger stores a $10\,\mathrm{s}$ sample of data every hour, regardless of any other conditions. This data is used whenever filtered data would not contain the necessary information or when the filtering could interfere with the later analysis, leading to wrong conclusions. One example would be the analysis of the constant background noise which the other triggers are built to ignore.

A threshold trigger accepts incoming data if the amplitude is—by a pre-defined factor—larger than the average noise level currently measured. This trigger accepts any strong enough signal, regardless of its shape. A signal shape trigger measures the cross correlation with a predefined bipolar signal and accepts if the correlation is high enough. For those last two triggers, an additional condition applies: The trigger condition must also be met on a predefined number of other hydrophones on the same storey.

After those triggers where applied, the initial data rate of $1.5\,\mathrm{TB/day}$ is reduced to about $15\,\mathrm{GB/day}$ which is then stored on disk. Since the raw data stream is sent to shore, new triggers can easily be implemented by modifying the triggering software on shore.

2.2 The acoustic environment of the detector

Since the ocean around the ANTARES detector is populated by a number of lifeforms and since the sea has a certain level of background noise, it is of course necessary to distinguish desired signals from noise and unwanted other signals.

A constant background noise is generated by the sea itself through current, turbulences and especially perturbations at the surface. Since these depend strongly on weather conditions, the noise also has a strong dependence on the weather, or more precisely, the speed of the wind. A higher speed of wind leads to a higher noise level (see figure 2.2). For higher frequencies, the thermal motion of water molecules dominates the noise spectrum.

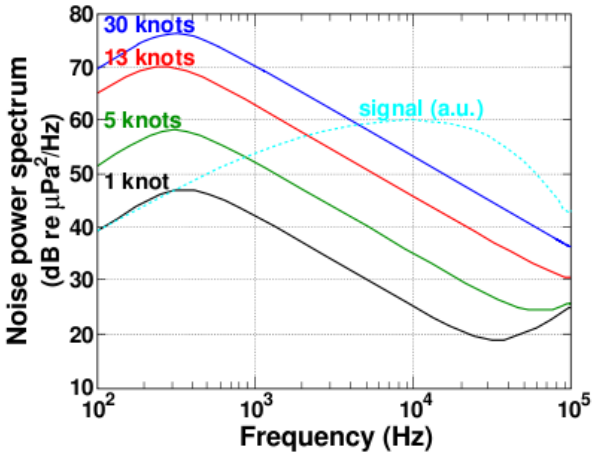

Figure 2.2: Spectra of the background noise at a depth of about $1500\,\mathrm{m}$ for different wind speeds and spectrum of the expected bipolar signals from particle showers. The wind speeds correspond to values of 0 to 7 on the Beaufort scale. The amplitude of the signal (light blue) is given in arbitrary units. The signal amplitude depends on the energy deposited in the shower and the hydrophone position relative to the shower. From [18, 19].

In addition to the ambient background noise, there are transient signals, generated by marine lifeforms such as cetaceans or anthropogenic, such as ship traffic. These signals are mostly assumed to propagate in a spherical shape from their point of origin, meaning that their angular amplitude distribution is isotropic. Together with the fact, that the temporal signal form is known, the increased rate of such signals hints at the presence of an animal or the trajectory of the object emitting a series of those signals can be reconstructed, such signals can be distinguished. Therefore it should be possible to filter such sources of noise or at least veto the respective part of the data.

2.3 Physics of particle showers

When particles with very high energies interact with the detector medium—seawater—a particle shower forms. The neutrino interacts via deep inelastic scattering with a quark in a nucleus of the detector medium. Since neutrinos only interact via the weak nuclear force, a charged $\mathrm{W}^{\pm}$ or neutral Z^0 boson will be exchanged. In both cases the products of the reaction will be accelerated to very high energies. From the neutral current reaction, $\nu_l q \xrightarrow{\mathrm{Z}^0} \nu_l' q'$, the quark will be accelerated and subsequently decay,

generating a hadronic shower. From the charged current reaction, $\nu_l q \xrightarrow{\mathrm{W}^{\pm}} l q'$ which occurs in about two thirds of all cases[20], producing again a hadronic shower and additionally a further shower from the lepton, in case of an electron a electromagnetic shower. Other leptons produce a hadronic or electronic shower depending on their decay channel. For a tauon, which can escape over some distance before it decays, a characteristic second shower in some distance from the primary, giving a "double-bang" signature[21] is produced. The details of those signatures and the probability to detect such a signature with a given detction technique depends strongly on the flavour of the primary neutrino as well as its energy.

The shower is generated by the products of the interaction of the primary particle causing secondary reactions of those products with the medium. The still highly energetic products of those secondary reactions then cause further chains of reactions, a cascade, until the energy of the particles produced is too low to cause any further interaction. Since the primary and most of the secondary particles will have a very high fraction of the speed of light, relativistic kinematics will cause the main shower direction to be the same as the direction of the primary particle. Also, the shower will only spread over a narrow angle around that direction[20].

The energy deposition of the shower into the medium causes a localized heating of the medium which is, given the very fast shower development and the, in relation to that, very slow timescale of hydro-acoustics, instantaneous. The increase in temperature causes a local increase in pressure, this overpressure will travel as a pressure wave through the medium. The details of the energy conversion from the heat deposition of the shower to the wavefront can be calculated through a thermo-acoustic[22, 23, 24] model.

Various simulations like the ones shown in [20] agree on the signal shape over time being a bipolar signal and the distribution of pressure over the emission angle measured from the shower axis to be Gaussian such that the overall emission expands in a "pancake-like" or "disk-like" shape perpendicular to the shower.

2.4 Acoustic detection of showers

These specific properties of sound generated by a shower make it possible to detect such showers, if the signal is large enough to be visible above the noise level, namely if the signal to noise ratio is larger than 1. Generally, for the acoustic method, showers which deposit a large amount of energy in a small volume are easiest to detect. [18] estimates for very calm sea that a signal to noise ratio of 1 corresponds to a neutrino energy of about $3.5\,\mathrm{EeV}$.

To recognize a particle shower and distinguish that shower from transient background events the features of the signal can be used. While background events are assumed to propagate in a spherical shape, the disk generated by a shower can be recognized if the arrangement of the hydrophones in the detector is sufficiently well chosen. As will be shown later on, this is the case for AMADEUS, at least for showers from certain directions.

The analysis of acoustic data relies on a number of algorithms to extract interesting features and parameters from the data and to reconstruct events from those features or parameters. The data from different hydrophones is analyzed in multiple stages. After the triggering stage described above, data is stored to disk for later analysis.

In the analysis, interesting signal shapes in the amplitude over time waveforms are identified, for example bipolar signals or pingers. This is usually done by calculating the cross-correlation of the measured data with a pre-defined prototype of the signal in question. After the reference signal is fitted to the data, parameters like amplitude and time are passed to the next stage.

There, the signals that arrived on all hydrophones of a storey are correlated. If the same signal shape arrived at different hydrophones of the same storey, from the differences in the time of arrival a direction can be reconstructed[25] which points toward the source of the signal. If several storeys "see" the same signal and it is possible to reconstruct a direction from each storey, together with the orientation and position of each storey an intersection point of the two or more rays can be found, thus reconstructing the point of origin of the signal.

3 SeaTray

3.1 Motivation

Other than the physically present apparatus, modern experiments in virtually every case consist of an extensive software infrastructure. Software is used during the planning phase to determine and optimize construction parameters through simulations, to eliminate sources of error and to predict the overall performance of the future experiment. During the operating phase, software is used for data taking and analyses as well as to monitor and control the state of the experiment.

Therefore it is necessary to start the development of simulation software well ahead of concrete planning of the apparatus. Long after finishing the construction and the start of operations, the data analysis will continue to be further optimized.

Because the day-to-day operations of the experiment will be automated as far as possible, the larger part of the scientists working on the experiment will be analyzing data during the operation phase. A very common mode of operations for analyzing data consists of two steps: First the expected types of events in the experiment are simulated, producing datasets which represent the expected measured data when the simulated event occurs in reality. Subsequently the measured, real data is compared to the simulated reference data to reconstruct the real event from the measurement.

In simulations from which real world data is to be predicted, the simulation needs to be calibrated to real measurements if the predictions made by the software are to be trusted. Therefore simulation packages in physics are developed in special projects which take care of this time-consuming and complex tuning. Other projects then simply make use of those software packages to avoid the necessity to develop, debug and calibrate such packages on their own. Well-known examples in the context of particle physics are AIRES[26], Geant4[27] or FLUKA[28].

From that, requirements regarding the software to be used can be specified: Simulation, data taking and analysis must be easily combined. The analysis must, for example, be able to process data from both, simulations and data taken with the real apparatus. Users who are unexperienced in programming must be able to use the software easily and be able to implement their own changes or extensions to the software. Finally, pre-existing external tools and libraries must be easy to interface with the software.

SeaTray tries to fulfill these requirements by a modular architecture where modules with inputs and outputs can be arranged in processing chains to accomplish tasks, a simple scripting language to arrange and configure those chains and the possibility to use external libraries via C++[29, 30] when writing modules.

3.2 Architecture and origin of SeaTray

SeaTray is derived from the software package "IceTray" developed by the IceCube collaboration[31] which, after a memorandum of understanding[32] with the KM3NeT consortium[1][33], was later also used and developed by ANTARES under the name "SeaTray"[34]. SeaTray and IceTray share the same basis and architecture, the differences lie primarily in the modules built on top of that. Most differences result from the different detectors in both experiments as well as the independent development of modules for data analysis. Although there are differences, most of the statements about SeaTray in this chapter also apply to IceTray.

Basic element of SeaTray is the so-called "tray", a data structure in which arbitrary[2] objects are stored referenced by a unique name. Each tray represents a certain period in time. The objects stored in the tray describe detector status, physics data, calibration information and detector geometry. The data processing schema of SeaTray is defined as follows: Trays are passed on along a chain of modules, each module adding new objects according to its role.

The objects added by a module contain results derived from objects which were already stored in the tray. A fitting metaphor would be that of a production line: Each worker (equivalent to a module) mounts certain parts on a tray until after passing all the workers in the production line the tray contains a finished product. The modules can be categorized into modules for data input and output —reading certain file formats, communicating over a network or graphical output—, production of data —by simulation for example— or analysis and processing of existing data.

Based on the modular architecture and specifications for the structure of a tray a file format[3] was defined which can be created at any place in the module chain. The data stream in form of the trays passed from module to module can be written out to a file and read again as input for a module chain. This enables easy reuse of expensive calculation results.

[1] The software was called "KM3Tray" there.

[2] The objects need to be derived from a common base class, which—because of the multiple inheritance feature of C++—is always possible.

[3] Recognizable by the file name suffix `.i3`.

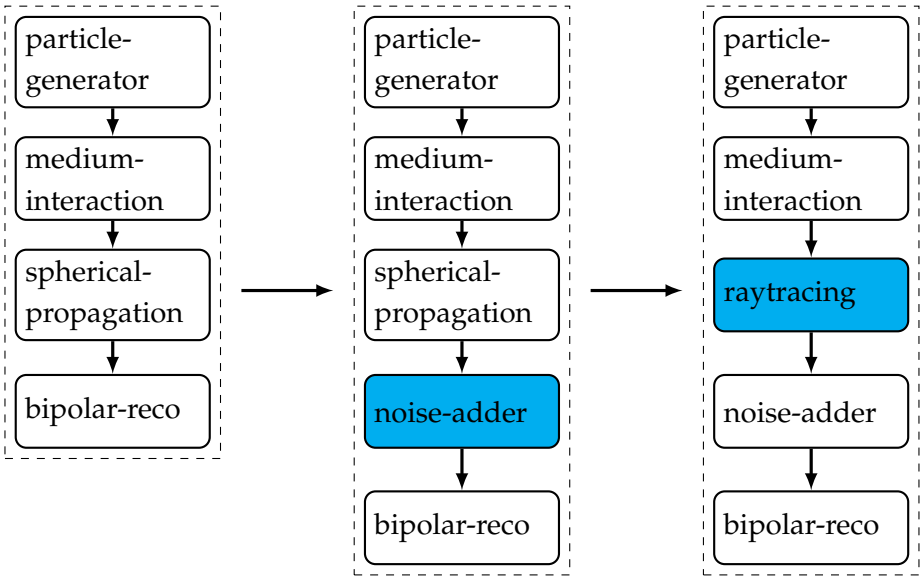

Figure 3.1: Example for the flexibility of SeaTray analysis and simulation chains: modules can be added or exchanged, often without needing to reconfigure the other modules. First, a module which generates noise is added, then the "spherical-propagation" module is replaced by a ray tracing module.

In the metaphor of a production line one can also explain the concept of "services" in SeaTray: services perform tasks which are not bound to a single step of the processing chain but are cross-cutting concerns affecting several or all modules. At a production line this would be the supply of consumables like screws, in SeaTray for example the generation of pseudo-random numbers. The service as an abstraction encapsulates a state which is relevant for all processing steps and modules, in our example the so-called "random seed" from which the next pseudo-random number will be calculated.

The configuration of the modules and services as well as their arrangement to a processing chain is done via a steering script, written in the scripting language "Python"[35, 36]. For example, for a file output module the file name where the data is to be written to is configured there. Also the in- and outputs of consecutive modules are connected in the steering script.

Ideally, a user of SeaTray only needs to create such a steering script to perform analyses or simulations by using pre-existing modules. This avoids the large barrier of entry that is presented by the need to learn C++. Also, the development cycles are shortened because changes in the steering script don't need any recompilation of the software, any changes take effect in the next execution of the script.

3.3 SeaTray in ANTARES

During the planning and design phase of KM3NeT, a number of modules for simulation and analysis of data from neutrino telescopes in a water medium were developed. Based on this foundation and because of the advantages that SeaTray offered, SeaTray was made the standard for offline-software[4] in ANTARES in the spring of 2009. In the following few paragraphs, the SeaTray environment for the optical part of the detector will be discussed.

The modules that were developed fall into three categories. First, some modules perform basic functions which are relevant to all use cases. Examples are a modules to read the ANTARES file format which is based on the software package "root"[37], to read the detector geometry or calibration from a database or from files, produce graphical representations of data, or classes which represent basic physical objects, processes or relations like particles, time or vectors.

Second, regarding simulations, there are generators for incoming particles, the showers they produce and a module interfacing to "Geant4"[27], a tool for the simulation of the interaction of particles with matter. Finally, there are also modules to reconstruct events from measured or simulated data using various algorithms.

Software development is performed using a version control system according to best practices for all larger software development efforts. Modules are grouped into packages where the packages fit the categories named above, for example the classes representing basic physical structures are compiled in the package "`dataclasses`". From time to time, when newly developed modules or features are sufficiently tested and considered stable enough for common use, a new "release" is made.

3.4 SeaTray for AMADEUS

For reasons of compatibility and to unify the software environment, the AMADEUS working group decided also to adapt SeaTray.

Yet, compared to the optical part of ANTARES previously described, there are certain differences which lead to some differing software requirements. Changes to existing components and the development of new components were necessary. For example, while the progression of the current on a photomultiplier is directly integrated by the measurement electronics and only sent to the coast as a digitized charge value, in AMADEUS the pressure curve measured by the hydrophones is digitized and continuously sent to the shore. Timescales in which optical and acoustical processes take place are different, meaning a sound wave travelling the length of the detector takes

[4] Software which is not directly involved in data taking or detector control

much longer than in the case of photons, which causes the signal runtime exceeding the length of certain data structures defined by the ANTARES hardware.

A first module to read the AMADEUS file format which is also based on "root" was needed to make existing AMADEUS data available for applications written in the SeaTray framework. To some amount it was possible to reuse existing work from the optical part because the data formats are based on the same structures which are just interpreted differently in acoustics. For the processing of the signals which are read a C++ class was also implemented in this work which represents the waveform a hydrophone has recorded.

The analysis of acoustical signals requires the recognition of characteristic signal patterns, for example bipolar or multipolar signals, maritime lifeforms or pingers from the positioning system. Therefore in the recognition process a data reduction is performed, converting from a waveform to a classified event with a type (bipolar, pinger, etc.), an amplitude, a time stamp and probably some more parameters. For those classified events another C++ class has been introduced.

To reconstruct the direction from which such a classified signal originates, the times at which the signal arrives at different hydrophones of a storey are used. If the positions of different storeys and the directions from which the signal arrives at those storeys are combined, a location where the signal was emitted can be calculated. Such an event is also represented by an appropriate C++ class.

These processing steps are already implemented in the previous software package called "AAP"[5] and are currently being ported to SeaTray. There are also plans to implement the online-software, which is the software for data taking, in SeaTray.

Another aim to be accomplished in SeaTray and the objective of this work is the realisation of simulations for AMADEUS. For this, the classes and modules which were described above are also relevant, as will be seen in the following chapters.

[5] Acronym for Amadeus Analysis Package. AAP is the previous software package used within AMADEUS.

4 Simulations

Just as measured data can be used as input data in SeaTray, the analysis and generation of simulated data is also possible. Simulated data is used to test various aspects of the software modules in SeaTray, to test and evaluate the detector design and to compare measured and simulated data to extract parameters that are otherwise hard to obtain.

4.1 Monte Carlo simulations in SeaTray

Monte Carlo simulations are a special type of simulations where random inputs are used. From a given parameter space $\mathbb{I}$ of inputs, a point $x_i \in_r \mathbb{I}$ is randomly chosen to calculate a point y_i in the result space $\mathbb{O}$ by a deterministic method (so $f : \mathbb{I} \mapsto \mathbb{O}, \quad y_i = f(x_i)$). This is repeated for a sufficiently large number of samples. The result of the simulation is obtained by aggregating all results via an appropriate aggregation function $a(x_1, ..., x_n, y_1, ..., y_n)$ such that the series a_n formed by increasing the number of samples n statistically converges[38] towards the theoretical result of the simulation.

A simple example for this would be the Monte Carlo integration of a smaller area A_s contained within a greater region $A_g \supset A_s$ of known size $s(A_g)$. One chooses random points $P_i \in_r A_g$ within the greater area and counts those $P_i \in A_s$, which also lie inside the smaller area. The ratio of points lying inside the smaller area to the whole number of points will statistically converge towards the ratio of the smaller to the greater area. If the characteristic function for each area is defined as

$$\chi_s(P_i) = \begin{cases} 1 & \text{if } P_i \in A_s \\ 0 & \text{otherwise} \end{cases}$$

the size of the smaller area $s(A_s)$ can be found via

$$s(A_g) \cdot \frac{\sum_{i=0}^{n} \chi_s(P_i)}{n} \xrightarrow[n\to\infty]{} s(A_s) \tag{4.1}$$

with n being the total number of points.

Monte Carlo simulations depend on randomness, yet in most cases, it is not necessary or desirable to go through the trouble of obtaining truly random numbers. A pseudo-random number generator (short “PRNG”) is sufficient as long as the PRNG

fulfills certain requirements: The sequence length, after which the sequence of random numbers repeats itself should be larger than the required amount of random numbers, the distribution of random numbers should be uniform over the interval of possible values and the distribution should be free of patterns which would interfere with our algorithms. PRNGs are computationally cheap and more importantly, deterministic and repeatable. When a PRNG is initialized with the same "seed" value it will output the same sequence of random numbers. That way we can repeat our Monte Carlo simulations by initializing the PRNG accordingly. From here on, when random values are mentioned, they are understood to be pseudo-random numbers.

In the context of the optical part of SeaTray for example, Monte Carlo simulations involve creating random particles with parameters like direction, energy and point of interaction with the detector medium. From that particle, a shower is simulated by randomly choosing the properties of the secondary particles while observing conservation of energy, reaction probabilities, etc. The shower is then converted into a distribution of Cherenkov photons which are propagated and detected inside the photomultipliers with a certain efficiency.

In all of those steps, the result of a physical process is computed by choosing randomly from all possible outcomes, observing basic physical laws and of course the relative probabilities of those outcomes.

In the following simulations, Monte Carlo methods will be used when generating particles with randomized properties and later summing up the outcomes by aggregating in a histogram.

4.2 Shower parametrisation

When doing simulations, a possible approach is to be as thorough as possible and include each and every known effect involved into the simulation, however small its impact might be. While this, given accurate parameters and the right theoretical background, will give a very precise simulation, its also very time-consuming, in processing time as well as in development time. Realistically, concessions need to be made to reduce the time it takes to develop a simulation as well as to run it. These concessions usually consist of simplifications and approximations. Complicated but very precise simulation algorithms are replaced by faster and simpler ones, thereby sacrificing some precision.

Also, in the context of this work, a strong focus is on defining and testing software interfaces and data formats, which is why rather simple simulation approaches to the physical problems were chosen. Now some of these simplifications and approximations relevant to this work will be shown.

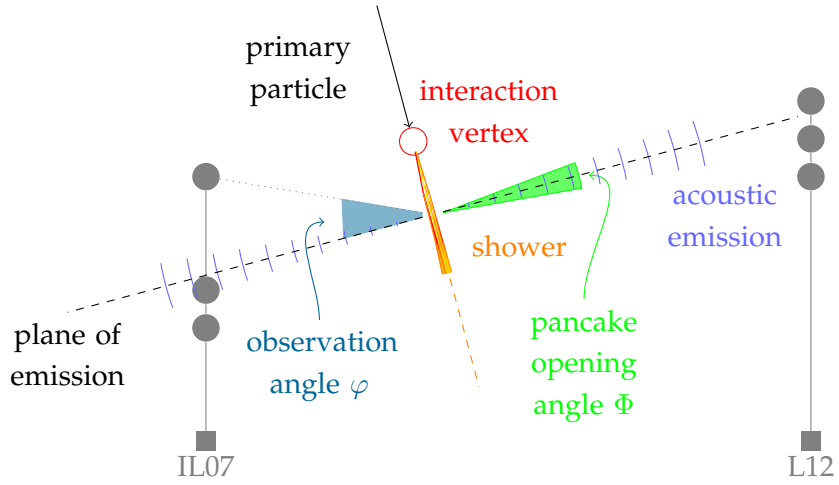

Figure 4.1: Overview of the geometry of acoustic emission from showers. Not to scale.

When a particle interacts with the detector medium, a shower is produced. This shower deposits thermal energy into the medium which expands and contracts over the affected volume, producing a characteristic sound wave. One could start by simulating a shower for each and every set of particle and its properties one wants to simulate.

But realistically, the effect of the shower itself will be governed by only a few important variables like energy distribution, size and shape of the shower. Other aspects like the temporal development of the shower formation is negligible since the shower lifetime is much shorter than the characteristic time of acoustic phenomena in water. Therefore, once the less important parameters are identified, they can be ignored without a great loss of precision.

Also, implementing and testing the simulation of the shower itself is a very complicated topic. For that reason, a simple parametrization will be used, even at the expense of some accuracy.

In [19] calculations are presented from which the following parametrizations will be derived. [19] show that for a neutrino-induced hadronic shower with an energy of the primary particle on the order of $10^{20}\,\mathrm{eV}$ an acoustic signal develops from the shower which is emitted within a plane perpendicular to the shower axis and primary particle impulse. The pressure amplitude plotted over the angle to the emission plane is roughly Gaussian in shape with a maximum at 0°[1] with respect to the plane of emission

[1]The corresponding value is about 0.6° according to [19], 0° according to [20]. This difference results from different assumptions about the placement of the point of origin on the interaction vertex or on the maximum of energy deposition in the shower from which the angles are measured as well as

if the plane of emission is defined to intersect the shower at the appropriate point. The width of the Gaussian is not explicitly given but can be interpreted from the plots to be around 0.5° to 1°.

In case of an electromagnetic shower from the same particle, the sound emission will be far less simple, since due to the LPM effect[39, 40] the shower will split up into several sub-showers each of which produces an acoustic emission as described before, but in a slightly different location. Therefore the visible sum of these acoustic emissions will consist of several superimposed Gaussian shapes extending over a larger opening angle of the pancake of about 5° and farther away from the interaction vertex, thus under a larger observation angle with respect to the emission plane. Similarly, in [41] the pancake opening angle is assumed to be on the order of a few degrees.

In the following simulations, a parametrization for the pancake-shaped acoustic emission of a shower will be used. First the parametrization assumes that the emission will have a rotational symmetry around the shower axis and particle direction, thus being independent of one of the two direction angles. Second, it will be assumed that for the remaining free angle of the emission the pressure distribution will be Gaussian over the angle, thus

$$p(\theta, \varphi) = \frac{1}{\sqrt{2\pi}\Phi} e^{-\frac{\varphi^2}{2\Phi^2}} \tag{4.2}$$

where Φ is the pancake opening angle. For some angles (which have been chosen according to [19] and [41] and will be used later on) the shape of the pancakes is shown in figure 4.2.

Similarly, the shape of pressure over time is given as bipolar in the literature. Therefore the following parametrization from [18] and [19] will be used:

$$p(t) = -\frac{t - t_0}{\tau} e^{-\frac{t-t_0}{2\tau^2}} \tag{4.3}$$

where τ is the characteristic length of the signal, which in our case is in the order of $30\,\mu s$. t_0 is the time of arrival of the signal.

These parametrizations are combined by multiplying them and applying some normalization factor.

$$p(t, \theta, \phi) = p_0 \cdot p(t) \cdot p(\theta, \phi) \tag{4.4}$$

The normalization factor p_0 needs to be chosen such that the emitted energy in the pancake equals the initially emitted energy. Since the following calculations only ratios of pressure amplitudes are relevant, the normalization was arbitrarily chosen.

Acoustic waves in water are refracted as a result of the gradient in the speed of sound caused by temperature and pressure variations. This leads to the effect that the plane in

the exact form of the emission. For this work it will be assumed that the emission is planar and the maximum of the Gaussian is at 0°.

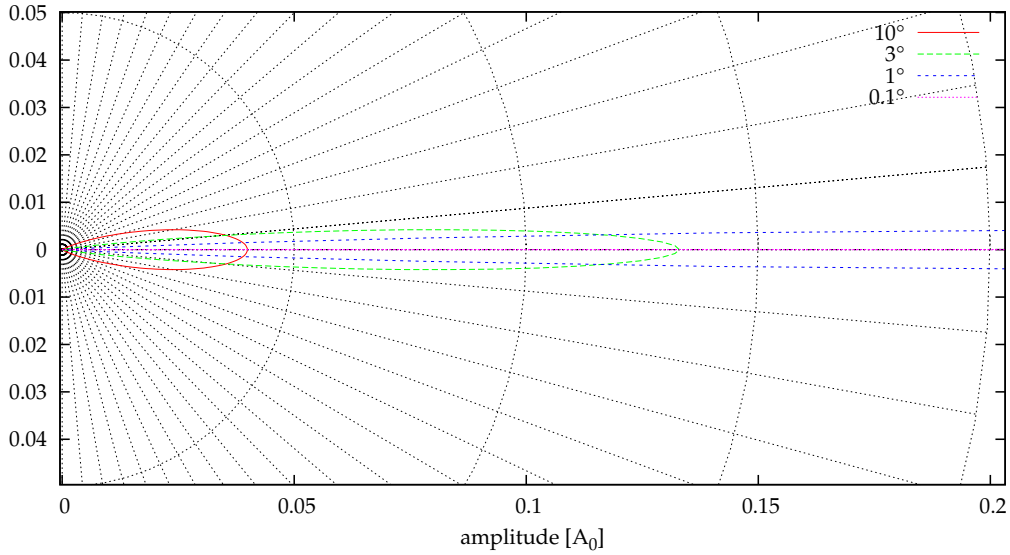

Figure 4.2: Polar plot of the amplitudes emitted over the emission angle. The shapes of the polar plots are congruent with isobaric lines of the emission. Plotted for different pancake opening angles Φ. Angular grid-lines have a distance of $5°$. The amplitudes shown are dimension-less and normalized according to equation 4.2.

which the pancake propagates is bent[42]. In some cases, thermal layers can also reflect acoustic signals.

Similarly, dispersion leads to different parts of the spectrum of the signal propagating at different velocities. More importantly, the absorption of the signal in the medium also has a rather strong frequency dependence: while signals of $1\,\mathrm{kHz}$ have an attenuation length of $70\,\mathrm{km}$, at $100\,\mathrm{kHz}$ the attenuation length is two orders of magnitude lower at $100\,\mathrm{m}$. In the relevant part of the spectrum between 10 to $50\,\mathrm{kHz}$ the attenuation length is in the order of kilometers to several hundred meters[18, 43]. These effects cause the bipolar shape of the signal to change with the distance the signal has travelled.

For the following simulations, it is assumed that the velocity of sound in seawater on the ANTARES site is constant for all depths and locations. Also, no frequency dependence is assumed for the attenuation length or speed of sound.

When differences in the speed of sound are ignored, sound from a point source propagates in a perfectly spherical shape. Pressure at a given point is then given simply by the $\frac{1}{r}$-dependence of pressure and the time the original signal takes to travel over the distance $\frac{r}{c}$. Thus the time- and direction-dependent pressure distribution given in equation 4.4 is multiplied by $\frac{1}{r}$ to give it a distance-dependence where r is the distance from the emission center to the hydrophone.

$$p(r, t, \theta, \phi) = \frac{1}{r} \cdot p(t, \theta, \phi) \tag{4.5}$$

It is assumed, that this $\frac{1}{r}$-dependence applies to all distances and forms of emission. The travelling time of the signal from its source to the hydrophone is accounted for by modifying the time of arrival t_0 in equation 4.3 by adding the travelling time, thus giving a modified version of equation 4.3:

$$p(t, r) = -\frac{t - t_0 - \frac{r}{c}}{\tau} e^{-\frac{t-t_0-\frac{r}{c}}{2\tau^2}} \tag{4.6}$$

For tests and certain analyses an even simpler parametrization for the angular dependence is used by just assuming isotropic emission over all angles, giving the trivial

$$p(\theta, \varphi) = \text{const.} \tag{4.7}$$

which is a spherical wave.

For the software to be aware of the detector configuration, especially the position of each hydrophone, a so-called "geometry" is necessary. The geometry[2] contains a list of optical modules or in the case of AMADEUS hydrophones, their positions, unique identifiers, the direction they are pointed at and their type from which their basic characteristics are known. Two different geometries are relevant to this work, the AMADEUS geometry and the ANTARES geometry. The AMADEUS geometry consists of two lines with 18 hydrophones each in slightly different arrangements: L12 has three storeys of hydrophones on top of the line. The instrumentation line IL07 has one story almost on top and two storeys around the middle of the line. Each storey consists of six hydrophones arranged in two triangles parallel to the seabed stacked about $1\,\mathrm{m}$ apart. Neighbouring storeys are spaced $14.5\,\mathrm{m}$ apart. A so-called "geometry file" was created which describes this geometry.

[2]The geometry used in simulations is usually a nominal geometry which does not include the effects of the sea current.

Within a SeaTray geometry, a unified designation scheme for the detector elements is used. For example "OM(2,5)" designates the fifth detector element from the bottom of the second line. While the "OM" in this designation originally stood for "optical module", other detector elements like hydrophones are also referred to this way, which will also be used later on.

The ANTARES geometry is the geometry of the optical components of ANTARES. The ANTARES geometry was used because it has the many more components in a much higher density which comes in handy when testing certain parts of the software. In some of the simulations presented later, the ANTARES geometry was used pretending every optical module was a hydrophone, thereby creating an array of hydrophones on 12 lines arranged in an octagonal shape, with 25 storeys on each line, each storey carrying 3 hydrophones in a triangular arrangement.

4.3 Simulation modules, simulation chains and tests

Several modules were implemented and arranged into different simulation chains during the course of this work. After a short discussion of each relevant module the simulations that were done are presented.

The simplest acoustic emission is isotropic over the emission angle as given in equation 4.7. The module implementing this form of sound propagation is called `AmadeusSphericalPropagation`. The module reads its input from an object of type `I3MCTree` which represents the particles and their interactions that make up a shower. An `I3MCTree` is usually generated from a simulation like the `SimpleGenerator` module which will be explained later. The initial pressure amplitude is then taken to be directly proportional to the energy of the most energetic primary particle of the shower. The center of the propagation sphere is placed at the interaction vertex of the primary particle. Aside from particles, this module could also be reused to simulate background noise originating from points sources like sea animals. As output, the module generates an object of type `I3AmadeusHit` for every hydrophone in the detector geometry. In that object, the calculated amplitude, time of arrival and type of the signal (for example "bipolar", "multipolar", "animal", etc.) can be stored.

The next module, called `AmadeusWaveformGenerator` generates a pressure waveform from every `I3AmadeusHit` by reading parameters out of the `I3AmadeusHit`. These values are then used to set constants in parametrization functions like equation 4.3 where for example the peak amplitude of the bipolar signal[3] is set to be the amplitude given in the `I3AmadeusHit`. The generated waveform is essentially an array of

[3] In fact, currently only bipolar signals are implemented.

floating point numbers with a given number of samples spaced equally according to a given sampling frequency. The reason for this structure is rooted in the ANTARES and AMADEUS hardware and was carried over when implementing the `AmadeusReader` module to read existing AMADEUS data files. Note that the waveform represents a pressure waveform as opposed to the ADC output waveform which the data files contain. Therefore, to have completely the same data format, one would need to implement a simulation of the hydrophone and the analogue and digital signal processing stages of AMADEUS or simply apply some conversion function.

A module called `AmadeusCSVWriter` was implemented to output the generated waveforms to a format which can be read by most software packages for plotting or further analysis. All plots of the waveforms were produced using this module and the software "gnuplot"[44, 45].

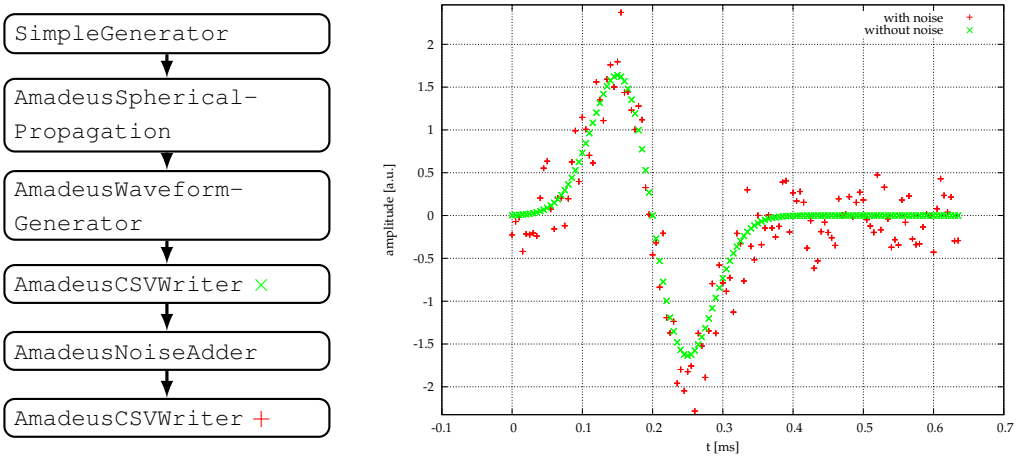

Figure 4.3: Simulation chain and an example of a generated bipolar signal, with and without added Gaussian white noise.

From figure 4.3 a bipolar signal without noise is clearly visible. Yet for example for testing reconstruction algorithms, this signal would be mostly unsuitable, because a real signal from the detector would always show a certain level of noise. To simulate that noise `AmadeusNoiseAdder` was created. This module generates Gaussian white noise by taking random numbers with a Gaussian distribution as amplitudes and mixes that noise onto its input signal, which is also shown in figure 4.3.

Using the previously described three modules, a simulation chain was implemented to test the interaction of the components. A pre-existing module named `Simple-Generator` was used to randomly create objects of the type `I3MCTree` containing only

a primary particle with random energy, direction and interaction vertex. Of course the bounds for the random parameters are definable, so that the appearance of random particles can be confined to a certain energy range or a certain area.

Starting with these random particles, for every particle a spherical wave is propagated to each hydrophone in the geometry. From the distance between the interaction vertex and the hydrophone, time of arrival and amplitude are calculated and bipolar signals are generated at each hydrophone. To those waveforms, noise is added. The result for the ANTARES geometry and the simulation chain is shown in figure 4.4.

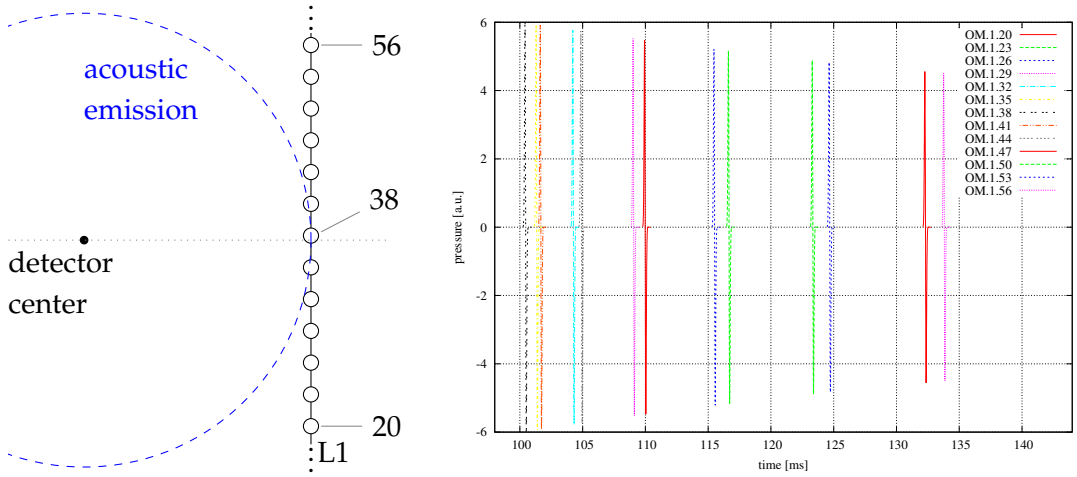

Figure 4.4: Signals from a spherical emission arriving at various fictitious hydrophone positions. The simulation chain used is the same as shown in figure 4.3, the signals plotted have no noise added. The hydrophones shown are arranged in the ANTARES geometry on line 1. One hydrophone from each storey around the middle of the line is plotted, OM(1,38) is closest to the interaction vertex. The interaction occurred at $0\,\mathrm{ms}$ in the detector center, the speed of sound was set to $1000\,\frac{\mathrm{m}}{\mathrm{s}}$ for easier distance calculations. Thus OM(1,38) is located at a distance of little more than $100\,\mathrm{m}$ from the detector center. The situation described is drawn on the left. Each circle on L1 represents one of the hydrophones on a storey. Only a part of L1 is shown in the plot and the drawing. Dashed blue is an isobaric line and a line of equal time of arrival of the emission. On the right side, the signals arriving at the selected hydrophones are shown.

The next step was to replace the spherical by a pancake-shaped signal. For that, a module named `AmadeusRotSymPropagation` was implemented. As for the spherical propagation, the time of arrival of the signal is calculated strictly by the distance

and a constant speed of sound. But other than in the spherical case, the amplitude is weighed by a factor which depends on the angle between the direction of the primary particle and the vector from the interaction vertex to the hydrophone (see figure 4.1). By only using the angle between those two vectors, the resulting signal shape is rotationally symmetric around the particle axis. The parametrization used for the pancake shape has already been discussed, see equation 4.2. Figure 4.5 uses the same simulation chain as figure 4.3 and 4.4, only with `AmadeusSphericalPropagation` replaced by `AmadeusRotSymPropagation`.

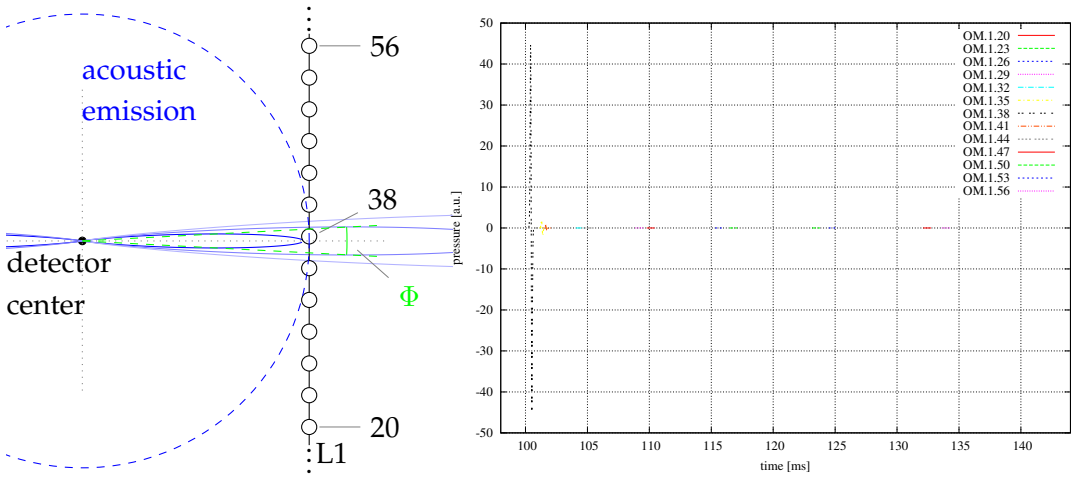

Figure 4.5: Signals from a pancake-shaped emission generated in the horizontal plane (i.e. particle coming from the zenith) in the detector center. The setup was identical to figure 4.4, only the `AmadeusSphericalPropagation` module was exchanged for `AmadeusRotSymPropagation`. A pancake opening angle of $\Phi = 3°$ was chosen, leading to a very steep decline of amplitudes when leaving the plane of emission, while OM(1,38) is hit with a peak-to-peak amplitude of 90 (in arbitrary units), the OM(1,35) on the neighboring storey only receives an amplitude of 3, the outer OMs are tens of orders of magnitude lower. The reduction of the signal amplitudes with respect to OM(1,38) for the hydrophones above and below are clearly visible. The times of arrival however are still following the same pattern as in figure 4.4. Solid blue lines are isobaric lines of the acoustic emission, the dashed blue line is a line of equal time of arrival.

The following simulations were done without the final step of creating bipolar signals. Instead, two analysis modules were created, called `AmadeusSimpleHitAnalysis` and `AmadeusPancakeSensitivityEstimator`.

Common to both modules is the mode of operation, both take objects of type `I3AmadeusHit` and analyse those. The output of each of those simulations is done via the `AmadeusCSVSimpleWriter` module, which simply outputs a line of data filled with values from an `vector<double>` for each tray.

So the whole simulation chain consists of a `SimpleGenerator` generating particles with a fixed energy and an interaction vertex (which is the same as the origin of the acoustic emission) in the detector center. The direction from which the particle enters the detector will be randomly chosen, usually from all possible directions, so an azimuth of $-180°$ to $180°$ and a zenith from $-90°$ to $90°$. The modules `AmadeusRotSymPropagation` or `AmadeusSphericalPropagation` will be used to calculate the peak amplitudes arriving at each hydrophone. From these amplitudes, `AmadeusSimpleHitAnalysis` or `AmadeusPancakeSensitivityEstimator` create an analysis by picking the maximum of all amplitudes arriving at the hydrophones or in case of `AmadeusPancakeSensitivityEstimator` the difference between the amplitudes at certain hydrophones. The values from those analyses will then be plotted in a Histogram over the direction, azimuth and zenith, of the incoming particle. This analysis will be presented in the next chapter.

With `AmadeusRotSymPropagation` it is also possible to move the maximum of the Gaussian which is usually at an angle $\alpha = 0°$ with respect to the plane of emission, in such a way that the signal travels on the envelope of a cone, similar to a Cherenkov cone with an opening angle of $90° - \alpha$. For an angle of $\alpha = \pm 90°$ off the plane of emission, a "spotlight" with a Gaussian beam profile in the same or opposite direction as the primary particle is produced. This "spotlight-mode" has been used to check the different geometries used in this work and is generally useful for debugging purposes since it produces images similar to those an observer sitting in the point of emission would "see". Two examples are shown in figure 4.6.

To distinguish a pancake-shaped signal from a spherical one, each one originating from the detector center, the following method was developed: Because of the relatively large gap between the storeys on IL07, a pancake-shaped signal will—even for relatively large opening angles—result in either the upper or lower part of the hydrophones receiving a noticeably higher amplitude, the other part will lie outside of the pancake. A spherical wave on the other hand will affect both parts in the same way, so no significant difference in amplitudes should be noticeable. This effect can also be observed when comparing figures 4.4 and 4.5.

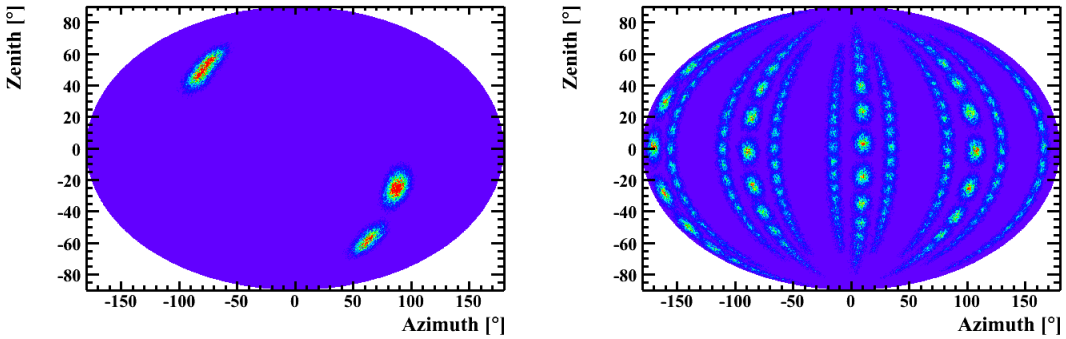

Figure 4.6: Histograms of the maximum amplitude any hydrophone receives with `AmadeusRotSymPropagation` in spotlight mode over the azimuth and zenith angles of the incoming particle and thus the direction of the spotlight. The point of emission is always the detector center. On the left side the detector geometry file used is the AMADEUS geometry file produced in the scope of this work. Clearly identifiable in the upper left part is the top of L12, the single and two storeys of IL07 are visible on the lower right. It is not possible to distinguish single storeys or hydrophones since the opening angle of the spotlight used is too large. On the right side, the ANTARES optical detector geometry was used, clearly visible as vertical structures are the 12 lines. In the zenith ranges around $0°$ storeys are separable, for larger positive or negative zenith angles, storeys on the same line overlap.

Therefore the trigger criteria will be as follows: An event is reconstructed in the detector center. The difference between the maximum amplitudes d between the upper and lower part of IL07 will assume a certain value d_{sph} in the case of spherical propagation. This value should be around 0 because for an emission in the detector center, a similar amplitude should be seen. For a non-spherical propagation on the other hand, d will deviate from d_{sph} by a relatively large factor. Depending on the absolute amplitude of the event which can be measured by the hydrophones on L12 and depending on the level of background noise, a limit for $|d|$ can be set above which the trigger accepts the event.

Two different kinds of errors can occur in this situation: spherical signals can be wrongly recognized as non-spherical, for example if noise or shadowing effects cause an overly large d. Or a non-spherical signal can wrongly be recognized as spherical, when a pancake hits all hydrophones with the same amplitude. This will occur, if the plane in which the pancake propagates is almost the same plane as the one defined by

IL07 and L12, meaning that in a certain angular range normal to the plane of IL07 and L12 this trigger will be blind.

So to test this assumption, incoming particles will be generated producing a pancake with its center point fixed at the detector center. The particle energy and thus the initial amplitude of the pancake will also be fixed to a constant value. The direction from which the particle originates which defines the normal of the plane in which the pancake propagates will be chosen randomly from a 4π solid angle. For each particle, the maximum amplitude received in any hydrophone and the amplitude difference between two hydrophones d, in this case OM(1,1) and OM(1,13), both on the IL07, namely the first hydrophone on the lowest and highest storey, will be output. From this output, histograms will be created showing the aggregated maximum amplitude or amplitude difference over azimuth and zenith of the incoming particle. The result will be presented in the following chapter.

5 Results

5.1 Results of the simulations

The previously described simulations give estimates about several aspects of the AMADEUS system. Figure 5.1 describes the sensitivity of AMADEUS to particles like neutrinos producing pancake-shaped signals from different directions and the ability to distinguish those signals from spherical signals.

Notable features in the left column of figure 5.1 are first and foremost the "sine-like" or "S-like" shapes[1]. These result from a pancake being detected by the hydrophones of a storey when the vector from the detector center towards the storey lies inside the pancake's plane. The width of those sine-like shapes is directly defined by the opening angle of the pancake we look at. Therefore, for smaller opening angles, those shapes are finer allowing smaller structures to be distinguished. Looking at the case of 10° opening angles, single storeys are not resolved, only the gap in IL07 leads to a corresponding gap and the visibility of two sine-like structures, a bolder one at smaller (positive and negative) zenith angles, and a lighter one at more extreme zenith angles. At an opening angle of 3° the bolder structure which seemed Gaussian in its cross-section is flattened, one even might see hints of a substructure. At the smallest opening angle of 0.5° that substructure is well-defined, one can see the bolder sine-like shape dissolve into five distinct shapes which are also sine-like in form. These shapes can be identified as the three storeys on L9 and the two lower storeys on IL07.

Compared to figure 4.5 the resolution of structures at an opening angle of 3° is somewhat reduced. This results from the different underlying geometries in both calculations. While in figure 4.5 the line was perpendicular to the plane of emission, in this case the hydrophones are hit under a steeper angle, leading to a smaller effective distance between storeys. Also, in figure 5.1 the signals received in all six hydrophones of a storye are taken into account, while in figure 4.5 only one hydrophone per storey is shown. This also "smears out" the signal of each storey.

[1] These curves are geodesic or great circles on the sphere that is projected onto the two-dimensional plane of the paper. The great circles themselves are generated by the intersection of the pancake with a sphere located at the point of emission of the pancake. Therefore the designation sine-like or S-like just refers to a result of the projection used, nonetheless this terminology will be used throughout this work.

For all opening angles, the described sine-like shapes intersect at two points, roughly at zenith 0° in both cases and azimuth -10° and 170°[2]. This means that the incoming particle comes exactly from the horizon such that the plane defined by IL07 and L9 is also the plane in which the pancake propagates. In that case, the sum of received amplitudes in all hydrophones is highest and the highest number of hydrophones is hit at the same time. Therefore, the area around the intersections has the highest bin contents, visible from the red coloring.

The right column of figure 5.1 shows some similar shapes, but with certain important differences. Since the values represented by the coloring are differences between two hydrophones, there will be a positive or negative difference depending on which of those hydrophones is directly hit by the pancake. This is visible as a red or blue coloring, a difference of about zero shows as green coloring. The sine-like shapes also seen in the left column are reproduced, but of course only those two shapes, that correspond to the two hydrophones that are used to compute the differences. The inner sine-like structure of course has a different sign than the outer one resulting in a different color. Most importantly at the intersection points the difference of both hydrophones amounts to zero. This means that in case of a particle from that direction, there will be no noticeable difference between a spherical and a pancake-shaped propagation in our hydrophones.

Figure 5.2 shows a "slice" of 1° at azimuth 60° of the simulations shown in figure 5.1, but evaluated for a larger number of opening angles. Comparing different opening angles one can see, that between 2° and 1° the substructure of the inner sine-like shape begins to become visible. For greater opening angles, these substructures merge into a single shape, around 10° the sine-like shapes generated by the topmost section of IL07 also begin to merge with the other ones.

From that, several conclusions can be drawn. First, the geometry of the AMADEUS system to differentiate pancake-like and spherical signals over a wide range of opening angles for the pancakes. For an opening angle around 3° it should, for pancakes where the primary particle comes from a certain range of directions, also be possible to determine the opening angle of the pancake by looking at the signal difference between neighboring storeys. Yet AMADEUS is only sensitive to pancake-like signals from a certain range of directions, such that the plane of the pancake propagation is perpendicular to the primary particle's track. The opening angle of the pancake also directly determines the solid angle within which the AMADEUS system is sensitive. Also, if the plane within which all hydrophones are located defines two directions perpendicular to that plane where AMADEUS has a blind spot for pancake-like signals, not because

[2] The 10° rotation from 0° or 180° of those angles comes from the fact that L12 and IL07 are not precisely located on the axes of the ANTARES coordinate system.

no signal reaches the hydrophones but because all hydrophones receive the same signal amplitude and therefore cannot distinguish that signal from a spherical one.

The simulations described here did not apply any kind of noise and were not used to calculate absolute values for the pressure amplitudes involved. Therefore, these results are only to be interpreted in relative terms, the different regions of higher and lower sensitivity are shown, but no fixed value for example for the lowest detectable energy is given. If the noise level would be taken into account, smaller signal amplitudes, of course according to the signal-to-noise ratio, would become undetectable, effectively setting a lower limit to the detectable amplitudes. In figure 5.2 this would for example exclude any signal lower than a constant horizontal line at a pressure corresponding to the level of noise.

All plots in figures 5.1 and 5.3 (which will be discussed below) show point symmetry around points on the horizon (zenith $0°$), for example at azimuth $-10°$ or $170°$. This is a result of the symmetry of the pancakes and the projections through which they are mapped onto the plane of the paper. In general, all the detector models show the same—only mirrored—behaviour for particles coming from above and below, that is particles passing through the atmosphere into the detector and particles travelling through the earth. At the relevant ultra-high energies, the earth is not transparent to neutrinos, therefore no particles are expected from that direction.

Also the interaction vertex which was fixed in the detector center is of course a special case, it is entirely possible to do a similar simulation for any location inside or even outside the detector volume. The central position was chosen to limit the time invested in simulations and to estimate detector performance in a best-case scenario. Yet from the simulations shown above it is also possible to draw some conclusions for larger potential acoustic detectors, because almost every position inside a larger detector which is populated with hydrophones densely enough can be reduced to a configuration like the one above. Of course if the detector consists of more hydrophones, there are far more possibilities for choosing a plane containing two clusters or more clusters of hydrophones while at the same time excluding other clusters. Therefore the sensitive angles of such a potential detector would be the sum of the distribution shown above, appropriately rotated for every hydrophone position.

This situation is shown in figure 5.3 for a hypothetical detector geometry based on the ANTARES optical geometry. What is clearly visible for all opening angles is the improved angular coverage, the blue areas representing zero sensitivity to incoming particles from that direction are far smaller. Interesting features of all plots shown in figure 5.3 are the intermittent areas of very high and no sensitivity along the horizon (zenith $0°$) and the non-sensitive spots at the poles, zenith $\pm 90°$. The pattern along the horizon is generated by the pancake plane being parallel or almost parallel to the

ANTARES lines. Therefore the pancake can either propagate within the gaps between the lines, giving a blue area of no sensitivity, or it can contain one or more lines within its plane of emission, thereby hitting a great number of hydrophones at the same time, generating a green to red area of high sensitivity. A different point of emission (i.e. by a different point of interaction of a neutrino) would move those patterns along the horizon since the gaps between the lines appear under different directions.

The non-sensitive area around the poles is generated by a particle from that direction generating a pancake parallel to the ocean floor. Since the storeys of different lines are located in parallel planes which are also parallel to the ocean floor, it is possible that a pancake will propagate in the horizontal gap between storeys, so that no hydrophones are hit, generating the non-sensitive blue area. Of course this also strongly depends on the point of emission of the pancake, if the depth of this point is also the depth at which a plane of storeys is located the poles will be a maximum of sensitivity since the pancake will then hit a very high number of hydrophones at the same time.

Looking at the plots in figure 5.3 for smaller pancake opening angles, substructures become visible. Multiple sine-like curves intersect in a grid-like pattern above and below the horizon, and in the maxima on the horizon described before. These substructures are basically the same as the superpositions of the different storey substructures from the AMADEUS geometry investigated before, only for a more densely and more regularly populated detector.

While in the case of the AMADEUS geometry, an extra simulation run with an additional set of plots generated showing the amplitude difference between two storeys of the AMADEUS system (see figure 5.1, right column) has been done, this is not necessary for the ANTARES geometry. While in the case of AMADEUS, it is possible that the plane of emission of a pancake contains all hydrophones of the detector, making it impossible to distinguish a pancake-shaped signal from a spherical one, in the ANTARES geometry there is always a number of hydrophones outside of any one plane. Therefore any pancake, that is detected can also be distinguished from a spherical signal, comparable to the difference in figures 4.4 and 4.5. For that reason, for the design of future detectors, a non-planar arrangement of hydrophones would be preferable.

The ANTARES geometry for figure 5.3 was chosen because a geometry file was readily available. But the conclusions from that simulation are significant nonetheless: A future, purely acoustical detector would have a much larger detector volume with a much larger spacing between hydrophones than in the ANTARES geometry. However, if a future optical detector like KM3NeT [46, 33] would incorporate hydrophones in every optical module for positioning purposes, these might also be usable for particle detection and in that case, the geometry would have a similar spacing to the ANTARES one. And furthermore, if one imagines a geometry which has the same arrangement of

hydrophones as the ANTARES geometry, only with all distances between the detector elements scaled up by a factor c. Then the simulation shown above for a pancake opening angle of Φ would give the same results as a simulation for the larger detector with a pancake opening angle of approximately[3] $c \cdot \Phi$. Thus for a detector with a spacing of $c = 10$ times the ANTARES spacing, so that storeys have a distance of $145\,\mathrm{m}$ instead of $14.5\,\mathrm{m}$ for example, the plot in figure 5.3 for an opening angle of $\Phi = 0.5°$ would be valid for a pancake opening angle of about $c \cdot \Phi = 5°$ in the magnified geometry.

Of course these "tricks" only work when looking at angles and relative amplitudes, not taking attenuation and other effects into account. In general, the absolute values of the arriving amplitude and its signal to noise ratio as well as the location where a pancake is generated and the distance to a hydrophone are important values for which the definition of absolute dimensions of the detector is necessary.

[3] This approximation is only valid for small angles Φ.

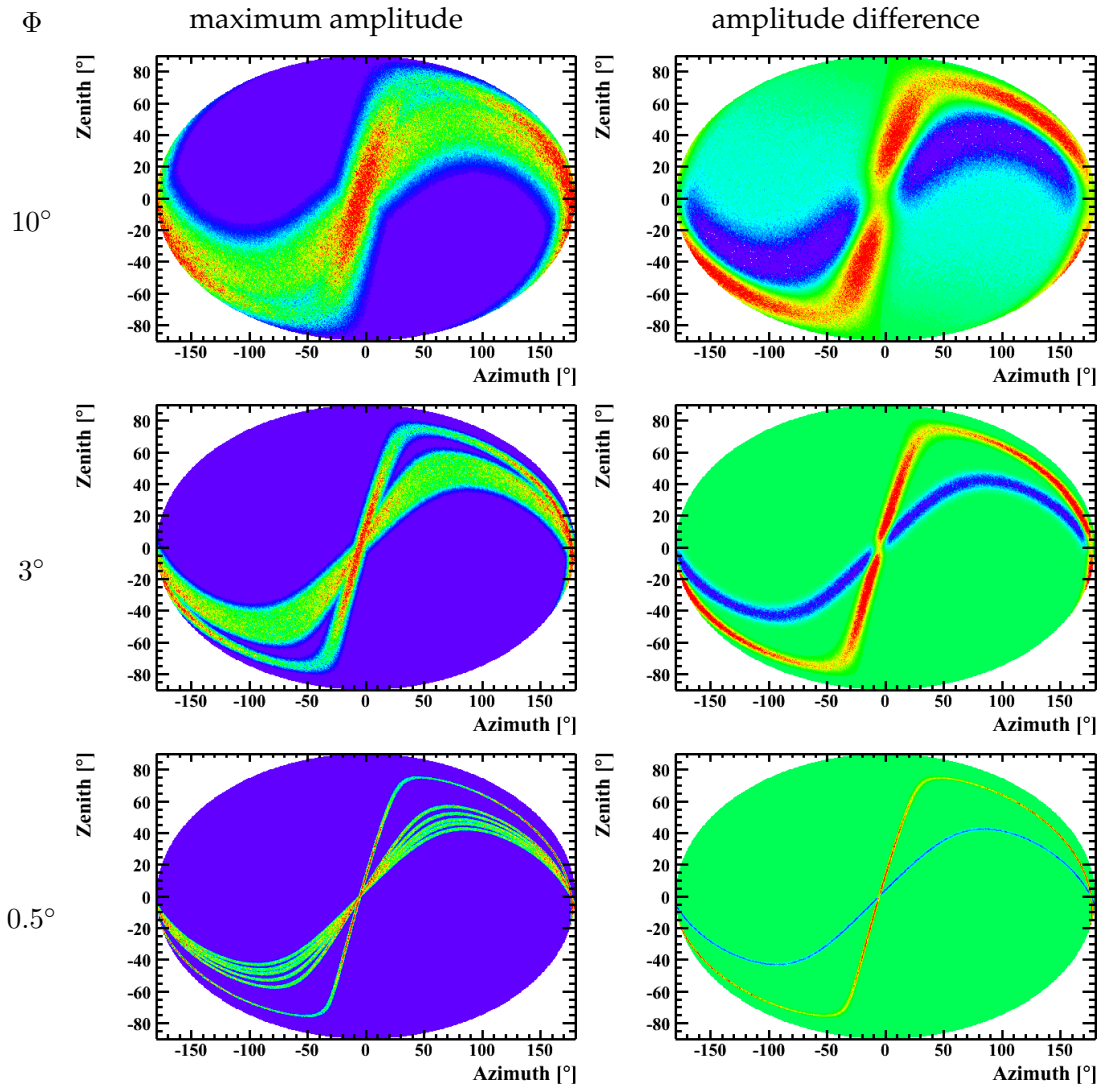

Figure 5.1: Histograms of simulations for pancake-like signals. Azimuth and Zenith values show the direction of the primary particle which caused the signal. The left column shows the maximum amplitude registered by any hydrophone. Blue shows a smaller, red a greater amplitude. Since the plots are only meant to show the spatial distribution, the scale is chosen for best visibility of the features of the plots. The right column shows the difference between two hydrophones OM(1,1) and OM(1,13) from IL07 on the lower- and topmost storey which gives a best-case estimate for the probability to recognize a pancake-like signal and distinguish it from a spherical signal. Green shows a difference of about 0, this corresponds to no ability to distinguish between spherical- and pancake-like signals. Red and blue areas show positive and negative values, both indicating the ability for distinguishing signal shapes. Both, maximum amplitude and pancake sensitivity are given for pancake opening angles Φ of $10°$, $3°$ and $0.5°$ from the top to the bottom row.

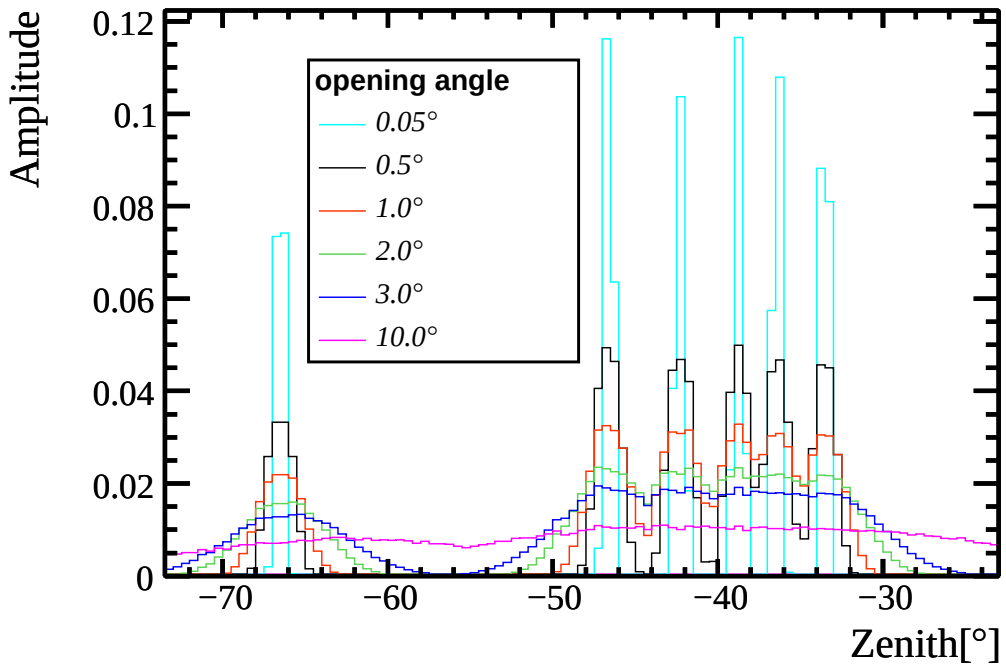

Figure 5.2: Slices of the maximum amplitudes from figure 5.1 at azimuth $(60 \pm 0.5)°$ for different pancake opening angles. Amplitudes are normalized to an arbitrary value such that the integral over each histogram is 1 to aid comparability.

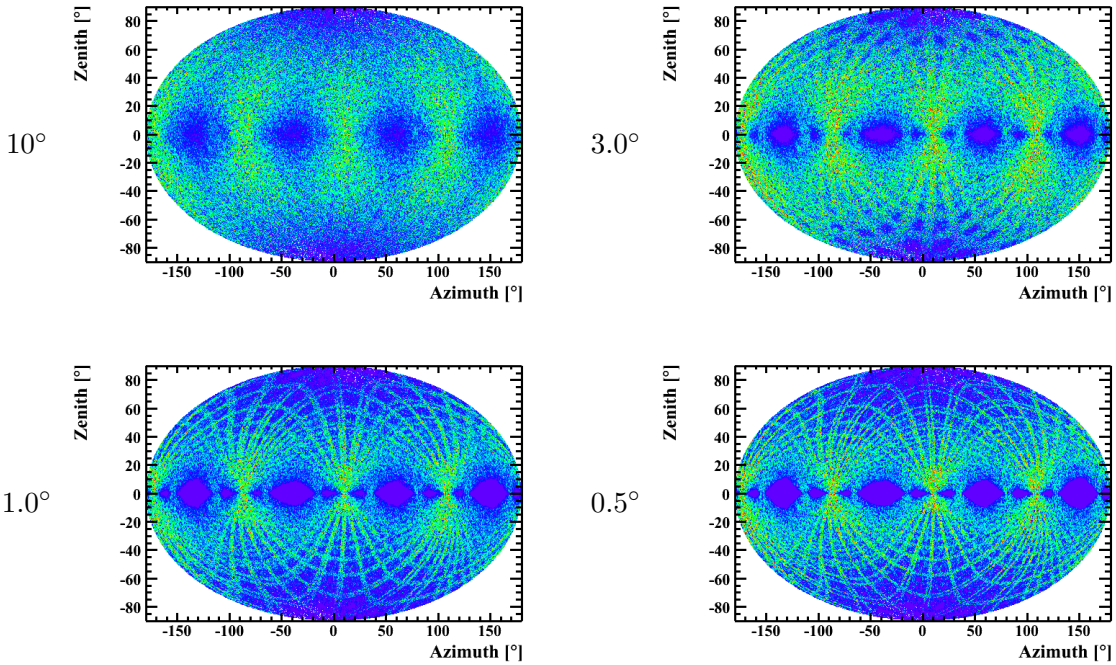

Figure 5.3: Histograms of simulations for pancake-like signals. The displayed information is the same as in the left column of figure 5.1, except for the detector geometry used, which in this case is the ANTARES geometry. The colors show the maximum amplitude received by any hydrophone, each plot for a different pancake opening angle Φ from $0.5°$ to $10°$.

5.2 Suitability of SeaTray

The simulations and analyses mentioned before were all implemented within the SeaTray framework using modules and data classes which were either preexisting or were implemented in the scope of this diploma thesis.

Pre-existing modules are mostly designed to work with optical neutrino telescopes but are often generalized enough to also fit the needs of acoustics. Modules handling the detector geometry for example could be reused without any modification just by supplying a different geometry file containing hydrophone positions. Other modules were modified to fit the special requirements of acoustics. For example the `antares-reader` was the basis for a modification named `amadeus-reader` which reads the different but similar acoustics file format. Newly written modules are easy to integrate into the framework, interaction of those new modules with preexisting ones is accomplished via well-defined and well-thought-out interfaces. Simulation and analysis itself is greatly simplified because there is no need to recompile the software for every change of a parameter. The concept of steering scripts in a suitable language like python improves productivity by shortening the cycle of adjusting a parameter, running the simulation, inspecting the results and again adjusting a parameter accordingly. Several areas still need improvement and of course additional modules have to be adapted to the needs of acoustics, but this seems to be only a matter of time and manpower, there don't seem to be any major hurdles concerning the SeaTray framework itself.

But there are also aspects which are not specific to SeaTray but speak of a generally high code quality of SeaTray and its modules. The software was built with a high degree of flexibility and without hard-coded values that would make it harder to change certain parameters. An example of this is the ease of altering the detector geometry. By simply exchanging the geometry file for another one, it is easily possible to generate simulations for any geometry desired. That way, new geometries for future detector designs can be evaluated and optimized.

6 Future prospects and possibilities

The work shown so far is of course only a foundation for future developments in and around Seatray. In the this chapter, some of those developments will be discussed.

6.1 Improvements

In the earlier chapters, several simplifications have been discussed regarding the simulation chain for acoustic detection. For accurate modeling of real events, this simulation chain will of course have to be improved. First of all, the shower model itself, which was nonexistent in this work needs to be addressed. Since this is a common concern of both optical and acoustic detection, this part of the simulation chain should be developed with both uses in mind.

From the shower model, the thermo-acoustic model derives the initial acoustic emission of the shower. In this work, only simple assumptions about the outcome of this model were used, namely a rotationally symmetric emission with a Gaussian profile leading to a pancake-shaped signal. In addition an isotropic emission for the simulation of point sources was developed. In future works, a more complete model might be used, or derived from that, some more sophisticated parametrizations.

The initial emission given by the thermo-acoustic model propagates through the detector medium, in which several effects need to be taken into account. First of all, frequency-dependent speed of sound and absorption will change the temporal shape of the signal. For example, since in water, higher frequencies undergo a stronger absorption than lower ones, the parts of the signal shape caused by higher frequencies will vanish faster. The signal will be "softened". Also, the path of the signal in space will be affected by different characteristics of the medium at different points in space.

The pressure gradient caused by different depths will be the dominant effect there, leading to a gradient in the speed of sound. This gradient will "bend" the spatial shape of a signal since a straight line will no longer be the fastest path a signal can travel between two given points. One possible approach for this problem would be the adaption of ray tracing. Ray tracing is a method originally from computer graphics, where the light distribution in a scene rendered on a plane of projection is calculated by casting rays of light from light sources into the world to be rendered.

Also, the noise spectrum of background noise is not being correctly simulated at this time. `AmadeusNoiseAdder` should be extended to generate the noise spectrum observed at different sea states according to [18]. This would later provide more realistic results when testing reconstruction algorithms and when simulating hydrophone characteristics.

6.2 Extensions

There are several more types of events, that need to be simulated. First, different types of particles need to be distinguished and from that, the different types of showers with the appropriate frequencies and relative probabilities. Even more importantly, some simulation of the transient background needs to be performed, especially when the background events like the sonar impulses of marine mammals might look similar to real particle physics events.

For example a "virtual dolphin" could be implemented, generating events along a random path in a speed and movement curve similar to that of a dolphin. These simulations can then be used in testing and development of improved triggers and vetoes improving resilience to different background events. Then for example a dolphin tracker could be implemented to veto parts of the detector where a virtual or real dolphin has been detected.

After a pressure waveform has been propagated to a hydrophone, the characteristics of the hydrophone and signal processing as well as their characteristic noise need to be simulated. Also pre-existing triggering code could be used to simulate triggers in that setup.

Using these extensions, flux and energy thresholds can be derived for the AMADEUS detector as well as other detector geometries. It would also be possible to optimize and train reconstruction algorithms for different signals from these simulations.

Different stages of the simulation and analysis could be used to check the results of either simulation or analysis. Generated bipolar signals could for example be used to test an algorithm which recognizes bipolar pulses. Or, when an algorithm that is known to recognize bipolar pulses which originate from showers is available, this algorithm can be used to create a more precise parametrization for the simulation of such bipolar signals.

Generally, most stages of the simulation chain have an equivalent stage in the analysis chain. These equivalent stages can be cross-checked against each other, and if available, against real data of known origin. That way the accurate reproduction of the real physical processes in simulation and analysis can be ensured.

6.3 Applications in other projects

In possible future large-scale acoustic neutrino telescopes, the above approach can be used to model the angular sensitivity to pancake-like events. It is possible when using an appropriate geometry for the new design to predict the angular sensitivity distribution and optimize the geometry accordingly. The same applies to other existing hydrophone arrays.

Also, all projects can benefit from the exchange of simulations and analysis algorithms and from the analysis of each others data with different algorithms. This exchange is most easily accomplished when the underlying infrastructure of both software and data formats is as similar as possible. This can be accomplished by agreeing on and using a common framework like SeaTray.

7 Summary and Conclusion

After briefly discussing the ANTARES and AMADEUS detectors and the acoustic detection of ultra-high-energetic neutrinos, SeaTray, a software framework for neutrino telescopes in water media was introduced.

In this work, SeaTray was presented as a framework for the simulation of acoustic particle detection. It has been shown, that SeaTray is suitable for the simulation of acoustic neutrino detection in a very flexible and easy way. Several modules suitable for a multitude of purposes have been introduced.

As a general impression SeaTray is well documented and usually—with very few exceptions—easy to understand and work with. The overall design principles seem sound and very well suited to the processes of simulation and analysis.

Several simulations using the software modules written in the scope of this work were performed. These simulations showed promising results regarding the sensitivity of the AMADEUS detector for particle showers of ultra-high-energetic particles.

Further simulations confirmed the assumptions about the performance of alternative detector geometries and the effects influencing this performance very well. This shows the suitability of the presented software for design and simulation of new and existing acoustic detector designs.

This work marks a step towards the calculation of flux and energy limits for the AMADEUS detector as well as for other acoustic detector designs.

A Analysis with SeaTray, an example

The workflow in SeaTray consists of several stages. First, one writes C++ modules which take input data and generate output data from these inputs. Several such modules can be plugged together to form a processing chain, where one module uses the output of the previous module(s) as input and generates new data from that. Finally, by an output module data is presented in some way to the user, by writing a file or displaying a plot for example.

As an example a part of the analysis which was used to generate figures 5.1 and 5.3 will be shown. For this analysis, a C++ module, named `AmadeusPancakeSensitivityEstimator` was developed. The following Python skript demonstrates how several analyses using this module can be performed.

First, the necessary libraries are loaded. The libraries contain the functions and objects used later on, roughly sorted by topic.

```
1  #!/usr/bin/env python
2  from I3Tray import *
3  from os.path import expandvars
4  import os
5  import sys
6  load("libdataclasses")
7  load("libphys-services")
8  load("libdataio")
9  load("libsimple-generator")
10 load("libsim-services")
11 load("libamadeus-sim")
12 load("libamadeus-writer")
13 load("libantares-reader")
```

Several variables that will be used later on are now initialized. This is the place where changes to the simulation parameters should be made.

```
15 nevents = 400003
16 geofile = expandvars("$I3_PORTS/test-data-antares/sim/
       amadeus_reference_detector.det")
```

Here the number of events to be generated is set to 4 million (plus three more events which serve a special purpose in SeaTray which will not be discussed here). Also, the geometry file is given from which the geometry for the following analysis will be read. A different geometry can be given by just passing another filename here.

```
18 xmin = 0.0 * I3Units.m
19 xmax = 0.0 * I3Units.m
20 ymin = 0.0 * I3Units.m
21 ymax = 0.0 * I3Units.m
22 zmin = 0.0 * I3Units.m
23 zmax = 0.0 * I3Units.m
24
25 azmin = 0.0 * I3Units.deg
26 azmax = 360.0 * I3Units.deg
27 zenmin = 0.0 * I3Units.deg
28 zenmax = 180.0 * I3Units.deg
```

These values set the ranges from which the `SimpleGenerator` module will choose the particle parameters like direction and point of interaction. By setting the minimum and maximum values for all coordinates, `xmin`, `xmax`, etc. to 0.0, the interaction point is set to the detector center for all particles generated. The angles in this example are set to be chosen from the whole 4π solid angle range available. But it is of course also possible to limit this range, for example to produce an analysis like in figure 5.2 where this range was set to limit the azimuth to a range of $(60 \pm 0.5)°$.

Also please note, that units are given as multiplicative factors. These factors are used to ensure consistent values in calculations. A calculation expecting to work on radians can simply divide by the desired unit to get the appropriate numerical value. So for example

```
1 angle = 63.0 * I3Units.deg
2 sine = sin(angle / I3Units.rad)
```

would provide the `sin()` function with the expected numerical angle value in radians.

The following code sets the variable for the pancake opening angle either to a default value or to the value given as a command line argument when executing the steering skript. This is very useful when doing calculations for a range of parameters.

```
30 if len(sys.argv) > 1:
31         opening_angle = float(sys.argv[1])
32 else:
33         opening_angle = 1.0
```

The following part of a shell skript could then be used for example in the job skript for a cluster job.

```
for i in 0.1 0.5 1.0 3.0 5.0 10.0 ; do
        AmadeusPancakeSensitivity.py $i
done
```

Now the module chain is built, first by initializing some basic services:

```
tray = I3Tray()

tray.AddService("I3SPRNGRandomServiceFactory", "random")(("Seed", 1), ("NStreams", 2), ("StreamNum", 1))
tray.AddService("I3EmptyStreamsFactory", "streams")(
        ("NFrames", nevents + 4),
        ("InstallGeometry", False)
        )
tray.AddService("I3AntTextFileGeometryServiceFactory", "geometry_from_file")(
        ("GeoServiceName", "geometry_from_file"),
        ("AntaresGeoFile", geofile),
        ("OMAngularParametrization", "old")
        )
tray.AddModule("I3Muxer", "muxer")(
        ("GeometryService", "geometry_from_file")
        )
```

These generate random numbers and provide the detector geometry or dummy values for streams unused in this simulation like the detector calibration.

After this basic setup the simulation chain described earlier is given.

```
tray.AddModule("I3SimpleGenerator","generator")(
        ("NMuEvent", 1),
        ("NEEvent", 0),
        ("EnergyMin", 100.*I3Units.GeV),
        ("EnergyMax", 100.*I3Units.GeV),
        ("ZenithMin", zenmin),
        ("ZenithMax", zenmax),
        ("AzimuthMin", azmin),
        ("AzimuthMax", azmax),
        ("MuStartXMin", xmin),
```

```
61          ("MuStartXMax", xmax),
62          ("MuStartYMin", ymin),
63          ("MuStartYMax", ymax),
64          ("MuStartZMin", zmin),
65          ("MuStartZMax", zmax)
66          )
67
68  tray.AddModule("AmadeusRotSymPropagation","propagation")(
69          ("pancake_opening_angle", opening_angle * I3Units.deg)
70          )
71
72  tray.AddModule("AmadeusSphericalPropagation","propagation2")(
73          ("output_name", "I3AmadeusSphericalMCHits")
74          )
75
76  tray.AddModule("AmadeusPancakeSensitivityEstimator", "analysis"
        )(
77          ("AmadeusSphericalHitSeriesName", "
                I3AmadeusSphericalMCHits"),
78          ("AmadeusPancakeHitSeriesName", "I3AmadeusMCHits"),
79          ("CompareOM1", OMKey(1,1)),
80          ("CompareOM2", OMKey(1,13))
81          )
```

First, the particle generator is initialized with the values defined above. Then, the signals are generated, one spherical and one pancake-shaped, which are then compared in the `AmadeusPancakeSensitivityEstimator`. Here also the hydrophones for which the differences are to be calculated are specified.

Now a CSV file (named using the pancake opening angle specified above) is output which can later be used in generating a plot from the data. The last few lines perform some cleanup.

```
83  tray.AddModule("AmadeusCSVSimpleWriter", "writer")(
84          ("OutputFileName", "pancake-sensitivity-" + str(
                opening_angle) + "_degree.csv")
85          )
86
87  tray.AddModule("TrashCan", "the can")
88
```

```
89 tray.Execute(nevents)
90 tray.Finish()
```

Bibliography

[1] V. F. Hess, Phys. Zeitschr. **XIII**, 1084 (1912).

[2] J. W. Cronin, (2004), arXiv:astro-ph/0402487.

[3] K. Greisen, Phys. Rev. Lett. **16**, 748 (1966).

[4] G. Zatsepin and V. Kuzmin, JETP Lett. **4**, 78 (1966).

[5] M. Takeda *et al.*, Astropart.Phys. **19**, 447 (2003), arXiv:astro-ph/0209422.

[6] HiRes Collaboration, Phys. Rev. Lett. **100:101101,2008** (2008), arXiv:astro-ph/0703099.

[7] The Pierre Auger Collaboration, Phys. Rev. Lett. **101:061101,2008** (2008), arXiv:astro-ph/08064302.

[8] C. Tsallis, J. C. Anjos, and E. P. Borges, Phys. Lett. A **310**, 372 (2003).

[9] C. Spiering, AIP Conf. Proc. **1085**, 18 (2009), arXiv:astro-ph/08114747.

[10] Antares collaboration website, `http://antares.in2p3.fr/`.

[11] ANTARES collaboration, E. Aslanides *et al.*, (1999), arXiv:astro-ph/9907432.

[12] M. Ardid, Nucl. Instr. Meth. A **602**, 174 (2009).

[13] ANTARES collaboration, J. Aguilar *et al.*, AMADEUS - The Acoustic Neutrino Detection Test System of the Deep-Sea ANTARES Neutrino Telescope (2009), to be submitted to Nucl. Instr. Meth. A, preprint obtainable from `robert.lahmann@physik.uni-erlangen.de`.

[14] D. Lachartre, ANTARES collaboration Report No. Internal Note Elec/2000-6, 2000.

[15] U. o. S. C. Information Sciences Institute, Defense Advanced Research Projects Agency Report No. RFC 791, STD 5, 1981.

[16] U. o. S. C. Information Sciences Institute, Defense Advanced Research Projects Agency Report No. RFC 793, STD 7, 1981.

[17] M. Neff, Studie zur akustischen Teilchendetektion im Rahmen des ANTARES-Experiments: Entwicklung und Integration von Datennahmesoftware, Diploma thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, FAU-PI1-DIPL-07-003, 2007.

[18] K. Graf, *Experimental Studies within ANTARES towards Acoustic Detection of Ultra-High Energy Neutrinos in the Deep-Sea*, PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2008.

[19] N. G. Lehtinen, S. Adam, G. Gratta, T. K. Berger, and M. J. Buckingham, Astropart. Phys. **17**, 279 (2002).

[20] S. Bevan *et al.*, Astropart. Phys. **28**, 366 (2007), arXiv:astro-ph/07041025.

[21] T. DeYoung, S. Razzaque, and D. F. Cowen, Journal of Physics: Conference Series **60**, 231 (2007).

[22] G. A. Askariyan, Sov. J. At. En. **3**, 921 (1957).

[23] G. A. Askariyan *et al.*, Nucl. Instr. Meth. **164**, 267 (1979).

[24] J. G. Learned, Phys. Rev. **19**, 3293 (1979).

[25] C. Richardt, Acoustic particle detection: Strategies for data reduction and source reconstruction, Diploma thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2006.

[26] Aires home page, `http://www.fisica.unlp.edu.ar/auger/aires/`.

[27] Geant4 homepage, `http://geant4.web.cern.ch/geant4/`.

[28] The fluka official site, `http://www.fluka.org/fluka.php`.

[29] B. Stroustrup, *The C++ Programming Language* (Addison Wesley Longman, 1997).

[30] B. Stroustrup, C/C++ Users J. , 43 (1999).

[31] IceCube Neutrino Observatory homepage, `http://icecube.wisc.edu/`.

[32] IceCube collaboration and KM3NeT consortium, Memorandum of understanding between the icecube collaboration and the km3net consortium on shared software development.

[33] KM3NeT consortium website, `http://www.km3net.org`.

[34] T. Eberl and C. Kopper, ANTARES collaboration Report No. ANTARES-SOFT-2009-013, 2009.

[35] G. V. Rossum, *The Python Language Reference Manual* (Network Theory Ltd., 2003).

[36] Python programming language – official website, `http://python.org`.

[37] Website of the root data analysis framework, `http://root.cern.ch`.

[38] R. Motwani and P. Raghavan, *Randomized Algorithms* (Cambridge University Press, 1995).

[39] L. Landau and I. Pomeranchuk, Dokl. Akad. Nauk SSSR **92**, 535 (1953).

[40] A. Migdal, Phys. Rev. **103**, 1811 (1956).

[41] J. Vandenbroucke, G. Gratta, and N. Lehtinen, Astrophys.J. **621**, 301 (2005), arXiv:astro-ph/0406105.

[42] R. Urick, *Principles of Underwater Sound* (Peninsula publishing, Los Altos, USA, 1983).

[43] M. A. Ainslie and J. G. McColm, J. Acoust. Soc. Am. **103**, 1671f (1998).

[44] P. K. Janert, *Gnuplot in Action – Understanding Data with Graphs* (Manning Publications Co., 2009).

[45] Gnuplot homepage, `http://www.gnuplot.info`.

[46] J. Carr *et al.*, KM3NeT: Conceptual design report for a deep-sea research infrastructure incorporating a very large volume neutrino telescope in the mediterranean sea, 2008.

Postface

Erklärung

Hiermit erkläre ich, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Erlangen, den 07. Januar 2010

Alexander Würstlein

Danksagung

Hiermit möchte ich mich bei meinen Betreuern, Uli Katz, Robert Lahmann und Kay Graf bedanken. Sie waren nicht nur immer da, wenn ich Fragen oder Anliegen hatte sondern haben mir auch durch aufmunternde Worte und ihre Hilfe und Geduld diese Arbeit überhaupt erst ermöglicht. Ich möchte behaupten, ich hätte keine besseren Betreuer als Euch finden können.

Ganz herzlicher Dank gilt auch allen Kollegen, Kommilitonen und Freunden[1], insbesondere Daniel Christiani, Daniel Danner, Kai Denker, David Eckhoff, Alexander Enzenhöfer, Jens Grabarske, Tobias Jordan, Michael Gernoth, Florian Haberl, Sebastian Kapfer, Manuel Meyer, Michael Meier, Christian Riess, Rainer Sennwitz, Rolf Weber und allen die ich zu meiner Schande hier vergesse aufzuzählen für Ratschläge, Korrekturen, moralische Unterstützung oder auch nur das Aushalten meiner schlechten Laune in stressreichen Zeiten.

Ausserdem danke ich meiner Familie und ganz besonders meinen Eltern für ihre Liebe und Unterstützung über all die Jahre meines Lebens. Nicht nur aus biologischen Gründen wäre diese Arbeit ohne Euch nicht entstanden.

[1] Diese Mengen sind nicht disjunkt.