



NuRadioMC: simulating the radio emission of neutrinos from interaction to detector

C. Glaser^{1,a}, D. García-Fernández^{2,3,b}, A. Nelles^{2,3,c}, J. Alvarez-Muñiz⁴, S. W. Barwick¹, D. Z. Besson⁵, B. A. Clark⁶, A. Connolly⁶, C. Deaconu⁷, K. D. de Vries⁸, J. C. Hanson⁹, B. Hokanson-Fasig¹⁰, R. Lahmann^{1,3}, U. Latif⁵, S. A. Kleinfelder¹¹, C. Persichilli¹, Y. Pan¹², C. Pfendner¹³, I. Plaisier^{2,3}, D. Seckel¹², J. Torres⁶, S. Toscano¹⁴, N. van Eijndhoven⁸, A. Vieregg⁷, C. Welling^{2,3}, T. Winchen^{8,15}, S. A. Wissel¹⁶

¹ Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA

² DESY, Platanenallee 6, 15738 Zeuthen, Germany

³ Erlangen Centre for Astroparticle Physics, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany

⁴ IGFAE, Depto. de Física de Partículas, Universidade de Santiago de Compostela, Santiago de Compostela, Spain

⁵ Department of Physics and Astronomy, University of Kansas, Lawrence, USA

⁶ Department of Physics and Center for Cosmology and Astroparticle Physics, The Ohio State University, Columbus, USA

⁷ Kavli Institute for Cosmological Physics, University of Chicago, Chicago, USA

⁸ Vrije Universiteit Brussels, Brussels, Belgium

⁹ Department of Physics, Whittier College, Whittier, CA, USA

¹⁰ Department of Physics and Wisconsin IceCube Particle Astrophysics Center, University of Wisconsin, Madison, USA

¹¹ Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697, USA

¹² Department of Physics and Astronomy, Bartol Research Institute, University of Delaware, Newark, USA

¹³ Otterbein University, Westerville, OH, USA

¹⁴ Université Libre, Brussels, Belgium

¹⁵ Max-Planck Institute for Radio Astronomy, Bonn, Germany

¹⁶ Physics Department, California Polytechnic State University, San Luis Obispo, CA 93407, USA

Received: 3 June 2019 / Accepted: 5 January 2020 / Published online: 31 January 2020
© The Author(s) 2020

Abstract NuRadioMC is a Monte Carlo framework designed to simulate ultra-high energy neutrino detectors that rely on the radio detection method. This method exploits the radio emission generated in the electromagnetic component of a particle shower following a neutrino interaction. NuRadioMC simulates everything from the neutrino interaction in a medium, the subsequent Askaryan radio emission, the propagation of the radio signal to the detector and finally the detector response. NuRadioMC is designed as a modern, modular Python-based framework, combining flexibility in detector design with user-friendliness. It includes a state-of-the-art event generator, an improved modelling of the radio emission, a revisited approach to signal propagation and increased flexibility and precision in the detector simulation. This paper focuses on the implemented physics processes and their implications for detector design. A variety of models and parameterizations for the radio emission of neutrino-induced showers are compared and reviewed. Comprehensive examples are used to discuss the capabilities of

the code and different aspects of instrumental design decisions.

1 Introduction

High-energy neutrino astronomy is a most promising approach to address the still unanswered question of the origin of high-energy cosmic rays [1]. Neutrinos are the perfect messenger. Because they have negligible mass, are electrically neutral and have an extremely low interaction probability, they traverse the universe essentially unimpeded and point directly back to their sources. However, measuring neutrinos requires the instrumentation of large volumes to observe sufficient target material in which a rare interaction of these particles may occur. Currently the largest detector having observed neutrinos is IceCube, which uses the Antarctic ice as a target medium and instruments it with optical sensors [2].

Neutrino astronomy recently took a significant leap forward when the IceCube detector at the South Pole was used to measure a yet unexplained excess of events that provides

^a e-mail: christian.glaser@uci.edu

^b e-mail: daniel.garcia@desy.de

^c e-mail: anna.nelles@desy.de

the first strong evidence for astrophysical neutrino sources [3]. The sources have not yet been identified, though compelling evidence for a first source was delivered with the observation of a spatial and temporal coincidence between a flaring blazar, observed with gamma-ray telescopes, and a high-energy neutrino [4]. However, detection of astrophysical neutrinos above a few tens of PeV has not been achieved yet, possibly due to the neutrino flux expected to steeply fall with energy, which calls for instrumented volumes larger than those currently existing. A two orders of magnitude increase in the volume instrumented by IceCube is considered cost-prohibitive due to the attenuation and scattering of optical light in ice [5]. Such a detector may measure the continuation of the neutrino flux, as well as the expected fluxes in the ultra-high energy regime [1].

1.1 Experimental and physical context of radio detection

High-energy neutrinos ($E_\nu > 10^{16}$ eV) can be most efficiently observed with the radio technique. Radio signals are produced via the Askaryan effect [6] from particle cascades generated in the ice following interactions of the neutrinos. The Askaryan effect arises from the development of a charge excess in the shower front as it accumulates electrons from the surrounding medium. The resulting changing current leads to measurable radio emission in the MHz–GHz frequency range. The Antarctic ice is transparent to these radio signals which allows for a cost-effective instrumentation of large volumes with sparse arrays. The attenuation length is about 1 km, depending on the frequency and ice temperature [7]. This results in an effective volume in the order of 1km^3 per single detector station, similar to the size of the entire IceCube detector.

The radio technique has already been successfully piloted with detectors at the South Pole and at Moore’s Bay on the Ross ice-shelf. The ARIANNA project [8,9] uses an array of autonomous detector stations with antennas located close to the ice surface, whereas the ARA project [10] uses antennas at a depth of up to 200 m below the firn layer. The experimental techniques matured substantially over the last years [11,12] and the community is well prepared for the construction of a large scale Askaryan detector with enough exposure to measure the continuation of the astrophysical neutrino flux to higher energies [1], to potentially discover cosmogenic neutrinos [13–15], and measure particle physics properties at yet unachieved energies [16].

With the developments on the experimental side, improved Monte Carlo simulations became imperative, leading to the development of NuRadioMC, which is presented in this article. A versatile and validated simulation of the radio signal in an Askaryan detector is crucial in many areas: for the determination of the sensitivity of a specific detector, for the optimization of the detector layout, to establish the require-

ments of the hardware to record the relevant parts of the signal, for the computation of a realistic signal expectation that is used to search for neutrino induced signals out of a large background of thermal and anthropogenic triggers, and finally, for the development of reconstruction techniques to determine the neutrino properties from the short radio flashes. In particular, the usage of modern deep-learning techniques requires a large and precise training data set.

The diversity of possible station layouts (e.g. compare the ARA and ARIANNA approach) requires a flexible software which is one of the main limitations of existing codes that were each targeted at a very specific experimental layout [17–19]. NuRadioMC is not tailored to a specific experimental design, and a detector station can have any number of antennas at arbitrary positions. In addition, the Askaryan radio technique is not limited to in-ice detectors. For example the lunar regolith has similar radio properties as ice and provides a immense neutrino target that can be observed from Earth with radio telescopes [20,21], providing the opportunity for synergies in simulations. Hence, from the beginning NuRadioMC was designed for maximum flexibility while maintaining user-friendliness.

1.2 Structure of NuRadioMC

The Monte-Carlo simulation of Askaryan signals from neutrino induced in-ice¹ particle showers is logically split up into four independent steps, the four pillars of NuRadioMC:

- 1. Event generation** The simulation of a neutrino flux. This includes the simulation of different neutrino properties (energy, direction, flavor, etc.), lepton propagation, the position of the interaction vertices, and the properties of the induced particle shower, i.e., how much neutrino energy is transferred into the shower, whether it is an electromagnetic or hadronic shower, etc.
- 2. Signal generation** The calculation of the Askaryan radio pulse generated from the particle shower.
- 3. Signal propagation** The propagation of the radio signal through the medium, from its origin to each antenna. Naturally occurring media typically have a density gradient resulting in bent rather than straight trajectories of the radio signal. Also, multiple distinct paths from the interaction vertex to the antenna may exist for typical geometries and ice typically shows a frequency-dependent attenuation length.
- 4. Detector simulation** The simulation of all components of the detector hardware. This step includes the conver-

¹ We will continue to refer to the standard case of a neutrino interaction in ice, when describing NuRadioMC. However, the code is designed in such a way that it can also support media other than ice, and exotic particles such as for instance dark photons [22].

sion from the electric-field pulses at the antenna positions to the measured voltages of each antenna channel, as well as the simulation of the trigger. It accounts for frequency dependent gain and group-delay, sampling-speed, record-length, etc.

The separation of the four steps follows the temporal structure of the physical processes. In a MC simulation this sequence will be different and not linear, e.g., we determine the signal path before generating it, so that we only need to calculate the Askaryan signal at the particular emission angle leading to that path. Moreover, after having calculated the signal, we need to use the propagation module again to determine the signal attenuation along the path.

We note that the separation of signal generation and propagation is a valid approximation when the difference in travel time from different points of the emission region to an observer in a homogeneous medium and one in a medium with a density gradient (bent trajectories) is small with respect to the observation frequency. We find that this assumption holds for all but rare and extreme geometries of an in-ice detector at frequencies up to 1 GHz.

The four pillars are complemented by a set of utility classes that are accessible at all times throughout the simulation such as a model of the medium, or a model of the signal attenuation. To ensure maximum flexibility and ease of use of different codes and programming languages the four pillars are separated as much as possible. The modules can be written in any language but Python wrappers of the relevant functions are required (this can be achieved e.g. with Cython [23]), so that the simulation can be steered from Python. This design was chosen to maximize user-friendliness and allow for the interfacing with other existing frameworks.

1.3 Improvements on the simulated physics in NuRadioMC

NuRadioMC does not only improve in flexibility and ease of use over existing codes, but also includes more physics processes in the simulation than previous codes and improves on precision. In the event generation, the subsequent decay of taus following a tau-neutrino interaction is modelled and the interface to simulate any *multi-bang* model is provided. Hence, models predicting several spatially-separated interactions can be implemented and simulated.

In the signal generation pillar, various Askaryan signal generation models are implemented. Previous MC codes relied on parameterizations of the frequency spectrum of radio emission [24] or on time-domain calculations mostly restricted to electromagnetic shower profiles [25]. NuRadioMC improves this approach by providing a time-domain calculation from an extensive library of electromagnetic, hadronic and tau-initiated showers. In particular, this

allows for a realistic treatment of the Landau–Pomeranchuk–Migdal effect (LPM effect) [26,27].

In the signal propagation pillar, new ray-tracing techniques based on an analytic solution of possible signal paths are implemented. This implementation results in unprecedented combination of speed and accuracy. Furthermore, we provide the interface to a more detailed numerical calculation that can simulate the signal paths in arbitrary 3D density profiles.

In the detector simulation pillar, we use the *NuRadioReco* code [28] that allows for the simulation of any detector geometry. In particular, it includes a detailed antenna response for a variety of antenna types and arbitrary orientations, treating the full set of complex gains as well as complex triggers such as phased-arrays.

In this article, we first describe each of the four pillars in detail and discuss different approaches. Then, we present three examples of how to use NuRadioMC and discuss the implications for the design of a high-energy neutrino radio detector.

2 Event generation

The event generation is logically separated from the simulation and provides general event parameters as input to the simulation. The results of the event generation are stored in an HDF5 file [29], which ensures that the event generator is easy to change in order to cover a variety of physics cases, as well as practical cases such as the simulation of calibration pulser data. This section describes the standard case implemented in NuRadioMC and provides an outlook for future implementation and special cases.

Having the event generation separated from the other simulation steps is beneficial because it allows the user to test the influence of different parameters on the same events. For example, the influence of different signal generation models, ice properties that influence the signal propagation or attenuation, and trigger schemes and thresholds, while using the same set of events.

2.1 Considerations concerning the coordinate system

All coordinates are specified in a local Cartesian coordinate system with its origin centered at the surface of the ice (see Fig. 1). The implementation of a global coordinate system that takes into account the curvature of the Earth is not required at this stage of precision: Due to attenuation of radio signals in the ice, the maximum propagation distance of radio signals is $\mathcal{O}(1 - 5)$ km where the impact of Earth attenuation is less than 2 m. Thus, effects of Earth curvature can be ignored from the signal propagation step onwards. The maximum propagation distance also defines the neces-

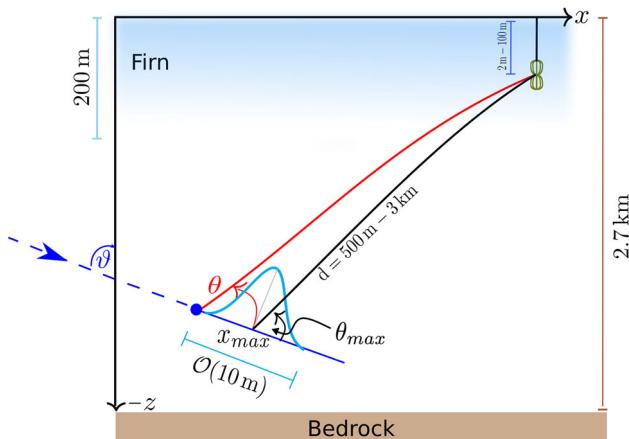


Fig. 1 Sketch of the coordinate system used by NuRadioMC and typical dimensions in the radio detection of neutrino interactions. The coordinate origin is at the ice surface. A quantity of particular interest is the viewing angle θ , i.e., the angle at which the in-ice shower is observed. Due to the longitudinal extent of the shower, the viewing angle is not uniquely defined. By default, we measure the angle with respect to the neutrino interaction vertex, but sometimes it is appropriate to measure the angle with respect to the maximum of the charge-excess profile, which we denote with $\theta_{X\max}$. It should be noted that this is just one typical set-up, other choices of geometry are supported

sary volume where neutrino interactions are simulated in. Thus, also for the standard event generation, a flat Cartesian coordinate system is sufficient.

Earth curvature starts to matter in the tracking of tau leptons and simulation of their subsequent decay as the tau decay length can reach values above 10 km. At 10 km distance, the difference between a flat and curved surface is 8 m which is still small compared to the thickness of the ice sheet at the South Pole of 2.7 km. Hence, the difference in target volume is also small. Another effect is that the probability of a neutrino reaching the simulation volume (referred to as *neutrino event weight*, see Sect. 6.2) is calculated based on the angle between the incident neutrino direction and the (flat) surface. Consequently, the neutrinos originating close to the horizon will have a systematic uncertainty in their assigned weights. However, at 10 km distance, this effect is again small with a displacement of only 0.1°. In the future, effects of Earth curvature can be considered by correcting this angle in the neutrino event weight calculation. The additional complexity of implementing a global coordinate system does not seem required at this point.

2.2 Default event generator and file format details

The default event generator creates a list of neutrino interaction vertices, specifies all relevant neutrino properties, and stores everything in an HDF5 file (see structure in Appendix A).

The event generator specifies the following parameters:

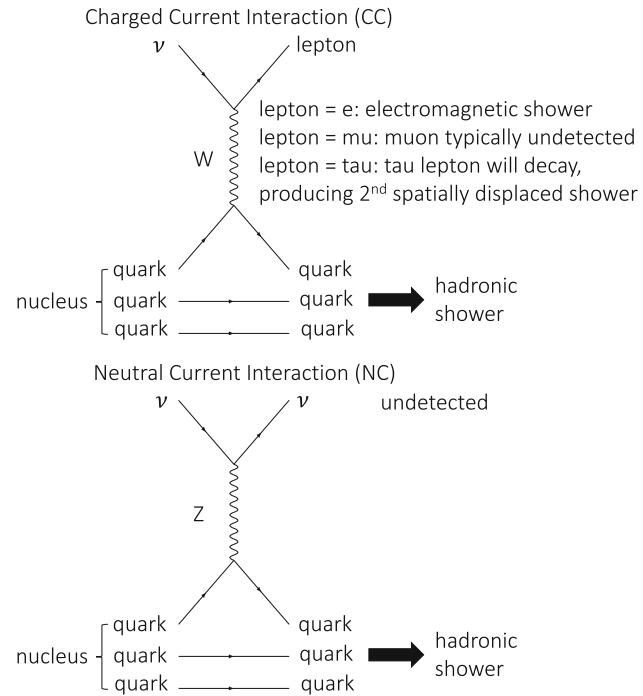


Fig. 2 Feynman diagrams of a charged current and neutral current neutrino interaction

- the position of the neutrino interaction, randomly placed in a cylindrical volume surrounding the detector. The user can control the minimum and maximum radius and the vertical extent.
- the neutrino energy, drawn from a user definable energy spectrum between a minimal and maximal energy. We also allow to specify the *deposited* energy instead, i.e., the amount of neutrino energy that ends up in a particle shower producing an Askaryan signal.
- the neutrino flavor. By default all flavors and particle/anti-particle nature have equal probability. Internally, this is specified using the Particle Data Group ID (PDGID) [30], which allows for cross-referencing with other Monte-Carlo codes.
- the neutrino direction. By default the full sky is uniformly covered but the user can restrict neutrino directions to specific ranges in zenith and azimuth angles.
- whether the neutrino undergoes a neutral current (NC) or charged current (CC) interaction (see Fig. 2 for an illustration of the two interaction types). We use a constant ratio CC:NC 0.7064:0.2936 according to the CTEQ4-DIS cross sections for the neutrino energy between 10^{16} eV and 10^{21} eV [31].
- the inelasticity, i.e., the fraction of the neutrino energy going into the hadronic part of the interaction. The inelasticity distributions from [32–34] have been implemented.

We note that we place neutrino vertices with equal probability per volume. The probability of a neutrino reaching the detection volume is taken into account later by assigning a *weight* to each event (see Sect. 6.2 for how the neutrino absorption is calculated). Similarly, it is currently ignored, if the density of the simulation volume is not uniform which changes the neutrino interaction cross section and thereby the interaction probability. As the density of the typical use-case of ice, only changes in the upper ~ 100 m this effect is ignored at this stage of precision. It can be taken into account in the future by an additional weighting factor or by an event-by-event calculation of the neutrino cross section.

All these parameters are saved in a HDF5 table. This has several advantages. The data is saved efficiently, the format is platform and programming-language independent, stand-alone viewers exist to quickly inspect the files, and apart from storing the actual data tables, it allows saving meta attributes such as the parameters the event set was generated for.

Typical data sets consist of millions of events which would take too long to simulate in a single process. Therefore, the event generator allows to automatically split up the data set into smaller chunks, i.e., into separate HDF5 files with typically 10,000 to 100,000 events per file. Then, the NuRadio-MC simulation can be performed for each file separately, and we provide the tools to merge the individual output files back together.

2.3 Multiple showers

Previous radio simulations only considered particle showers created by the initial neutrino interaction. However, in case of charged current interactions of muon and tau neutrinos, the produced muons and taus might interact or decay producing a second spatially displaced particle shower that generates Askaryan radiation.

The typical decay length of a tau lepton range from 50 m at tau energies of 1 PeV to 50 km at tau energies of 1 EeV. This increases the sensitivity of an Askaryan detector because tau neutrinos can interact far away from the detector but still produce a visible signal if the tau happens to decay close enough to the detector.

Muons in turn are unlikely to decay but they can undergo a catastrophic dE/dX energy loss, depositing a substantial fraction of their energy into the ice and initializing a hadronic shower [35, 36]. In general, more exotic models can also be considered that predict multiple spatially displaced showers per neutrino. Hence, NuRadioMC offers the flexibility to specify an arbitrary number of interaction vertices per event. This is incorporated into the file format by inserting additional events into the event list with the same event ID.

We consider several levels of detail. While a simple treatment of tau decays exists in NuRadioMC itself, we also foresee the inclusion of more complete particle decay codes,

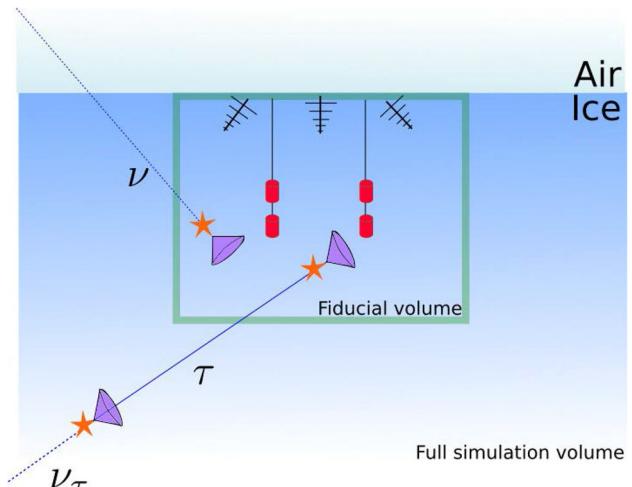


Fig. 3 Sketch of the geometry and the concept of a fiducial volume of the event generator. Neutrino tracks are generated in a full simulation volume, but only the radio emission of primary or secondary interactions are considered, when they take place in a fiducial volume encompassing the detector

such as PROPOSAL [35, 36] that tracks secondary losses of all types of lepton.

2.4 Tau neutrinos

In NuRadioMC, for the first time in an in-ice simulation, we provide the inclusion of secondary sub-showers from tau-decays that add additional detection channels, flavor sensitivity and contribute to the effective volume.

Due to the large decay length of tau leptons, a large volume needs to be simulated to catch the few cases in which there is a secondary interaction close enough to the detector. This increases the computation time enormously as it scales proportionally to the simulated volume, and makes this brute-force approach unfeasible. Therefore, we developed the following technique: we generate neutrino interactions in an arbitrarily large volume including all secondary interaction vertices (e.g. from tau decays) but save only those primary and secondary interactions that take place in a much smaller fiducial volume surrounding the detector while keeping track of the total number of simulated events (see Fig. 3 for an illustration). The user needs to make sure that the fiducial volume is chosen large enough such that the probability to trigger the detector is negligible for interaction vertices outside of this volume. This allows for a computationally efficient simulation of complex physics models.

Once a tau is created after the interaction of a tau neutrino in the volume, we calculate its decay time t_{decay} and energy at decay. We first randomly sample a decay time τ_{decay} in the tau particle rest frame from an exponential distribution using a mean tau decay lifetime 2.903×10^{-13} s [37]. If the tau energy is less than $E_\tau = 1$ PeV, we do not account for tau

energy losses along the path, and the decay time is simply given by the product of the Lorentz factor γ and the sampled decay time τ_{decay} in the tau rest frame

$$t_{\text{decay}} = \gamma(E_\tau)\tau_{\text{decay}}. \quad (1)$$

The decay length l_τ is calculated multiplying t_{decay} by the particle speed, while the energy of the τ at decay is equal to the initial tau energy.

In the case the tau has an energy greater than 1 PeV, we include photonuclear tau energy losses in our calculation. These are not very well constrained and we use a simple model inspired by the results in [38]. We take the mean energy loss per amount of traversed matter in ice to be,

$$\left\langle \frac{dE_\tau}{dX} \right\rangle \approx f(E_\tau) = b_1 E_\tau + b_2 E_\tau \log_{10}(E_\tau/E_0), \quad (2)$$

with $b_1 = 1 \times 10^{-7} \text{ cm}^2/\text{g}$, $b_2 = 1.8 \times 10^{-7} \text{ cm}^2/\text{g}$, and $E_0 = 1 \text{ PeV}$. Above $E_\tau = E_0$, it is a good approximation to assume that the tau speed is equal to the speed of light in vacuum c . This allows us to write the time t that it takes a tau with initial energy $E_{\tau,i}$ to reach a lower energy E_τ as,

$$t(E_\tau) = \frac{1}{c\rho_{\text{ice}}} \int_{E_{\tau,i}}^{E_\tau} \frac{dE'}{f(E')}. \quad (3)$$

Once $t(E_\tau)$ is known, we numerically obtain the inverse function $E_\tau(t)$ for equally-spaced times by interpolation. The decay time is obtained by solving the following integral equation for t_{decay} :

$$\int_0^{t_{\text{decay}}} \frac{m_\tau}{E_\tau(t)} dt = \tau_{\text{decay}}, \quad (4)$$

from which the tau decay length above 1 PeV is obtained as:

$$l_\tau \approx ct_{\text{decay}}. \quad (5)$$

In Fig. 4, left, we show the decay length l_τ as a function of tau energy. The straight dashed line represents the mean decay length without tau energy losses, which increases linearly with energy. The solid line indicates the decay length assuming that the decay time in the rest frame is equal to the mean decay time τ_{decay} and accounting for deterministic tau-energy losses during propagation given in Eq. (2). The shaded band represents an 80% confidence interval for the decay length, where the decay time has been drawn from an exponential distribution. Stochastic energy losses have not been accounted for. In Fig. 4, right, we show the tau energy at decay obtained with the same assumptions used for obtaining the tau decay length shown in the left panel. Tau energy losses become important around 100 PeV.

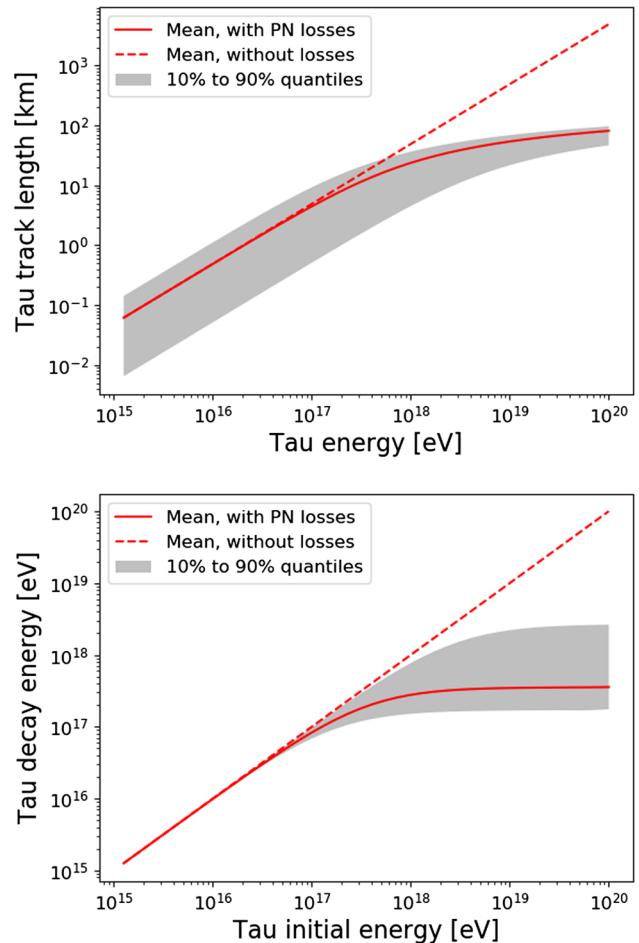


Fig. 4 Top: Tau decay length as a function of the tau energy. Bottom: Tau decay energy as a function of the initial tau energy. Due to the one-tailed nature of the exponential decay function, we show the decay length for the mean proper decay time with photonuclear losses (solid line) and without any losses (dashed line). The shaded band represents the area spanning from the 10% proper decay time quantile to the 90% quantile (80% of total probability). This implementation matches what has been shown previously in [39]

2.5 Options for additional physics processes or calibration purposes

The event generation described above is the default event generator in NuRadioMC. However, emission from a standard-model neutrino-induced shower is only one possible scenario that can be covered. The users have the freedom to implement their own event generators according to other physics assumptions, e.g., new physics or for simulating calibration signal generators. We provide an example to simulate a calibration measurement online [40]. As long as the events are saved according to the well-defined file structure, NuRadioMC can process any input files. A skeleton event generator is provided as an example [41].

3 Signal generation

NuRadioMC provides several modules for the generation of the radio signal from showers. The user may choose from a selection ranging from well-known frequency-domain parameterizations of the Askaryan signal to a state-of-the art semi-analytic calculation.

A uniform interface in the form of a simple function is provided for all models (see [42] and List. 3 in Appendix D.3). In this way the NuRadioMC code also serves as a reference implementation for all models. Furthermore, the well-defined interface allows for an easy extension of NuRadioMC with additional models. Even calibration emitters can be (and are) implemented to simulate a calibration measurement with NuRadioMC.

In the following, we first present the different signal generation models available in NuRadioMC before discussing their differences and giving recommendations for use in different cases. We discuss a variety of models, some for more pedagogical reasons, others because they are fast, and others because they are accurate. We hope that this section also serves as reference discussion of several widely used emission models, however, it is not an attempt at completeness.

3.1 Frequency-domain parametrizations

NuRadioMC currently provides two frequency-domain parameterizations of the Askaryan signal. One, referred to as *Alvarez2000*, is also used in the simulation code for the ANITA detector (IceMC) [19] and for the ARIANNA array (ShelfMC) [17,43], and is an implementation of the parameterization of [24], which was validated against a full simulation of Askaryan radiation performed with the ZHS Monte Carlo [44]. This is a microscopic simulation of the shower and its radio emission, that does not contain signal propagation and detector simulation.

The other parameterization (*Alvarez2009*) is an updated version of the first one. It is based on the so-called “box model” of shower development [45] and separate parameterizations for electromagnetic [46] and hadronic [47] showers are provided. Both parameterizations are the product of three functions. The first is a scaling function A that grows linearly with the primary energy E_0 , frequency f , and the sine of the observing angle θ . The second and third functions are two continuous cutoff frequency factors d_L and d_R that account for deviations from linearity due to incoherence effects associated to the longitudinal and lateral extensions of the shower. For electromagnetic showers, the LPM effect is modelled including random fluctuations of the size of the effect.

Although we encourage the use of the *Alvarez2009* parameterization, we have also included the older parameterization *Alvarez2000* for comparison with previous work and other

codes. The latter can be understood as a simplified version of the former, with constant factors, a simple continuous cut-off factor instead of two, and a Gaussian function for the dependence of emission on viewing angle. Because of its simplicity, it provides qualitative and easily understandable, however, not necessarily precise insights into the main dependencies of the Askaryan signal. For pedagogical reasons, we explicitly provide the parameterization of this model here and give an example of the resulting Askaryan signals.

If the shower is observed on the Cherenkov angle, the electric field (scaled to a distance of 1 m) according to *Alvarez2000* is given by

$$\frac{\varepsilon_c^{lm}}{V/m/MHz}(E_{sh}, f) = 2.53 \times 10^{-7} \cdot \frac{E_{sh}}{\text{TeV}} \cdot \frac{f}{f_0} \cdot \frac{1}{1 + (\frac{f}{f_0})^{1.44}}, \quad (6)$$

with the shower energy E_{sh} , frequency f and $f_0 = 1.15$ GHz. Signal amplitudes off the Cherenkov cone, ε^{lm} , are modeled as a Gaussian profile according to

$$\varepsilon^{lm}(E_{sh}, f, \theta_v) = \varepsilon_c^{lm}(E_{sh}, f) \cdot \frac{\sin \theta_v}{\sin \theta_c} \cdot \exp \left[-\ln 2 \cdot \left(\frac{\theta_v - \theta_c}{\sigma_\theta} \right)^2 \right] \quad (7)$$

with ε_c^{lm} given in Eq. (6), and where θ_v is the viewing angle relative to the shower axis. The angular width of the cone around the Cherenkov angle σ_θ is a function of both frequency and energy. For hadronic showers σ_θ is given in Eq. (6) of [48], for which a factor to account for the so-called missing energy, energy going mainly into muons and neutrinos that does not contribute to the Askaryan signal, is included in Eq. (6).

For electromagnetic showers above 2 PeV, the shower profile becomes elongated due to the Landau–Pomeranchuk–Migdal (LPM) effect. In the simple model of *Alvarez2000* such an elongation corresponds to a reduced σ_θ which is modeled according to the prescription in [49]. This in turn manifests itself as a rapid decrease in the high frequency content of the Askaryan signal off the Cherenkov cone for EM showers, as seen in Fig. 5.

For NuRadioMC, the time-domain signal based on *Alvarez2000* and *Alvarez2009* is generated by taking the simple approximation of a phase that is constant with frequency and equal to 90°, yielding a bipolar pulse in the time domain.

3.2 Fully analytic treatment including the LPM effect and Cascade Form Factor

NuRadioMC provides an implementation of the analytic model of Askaryan radiation (*HCRB2017*) [50] that builds on previous work by [51]. This fully analytic model accounts

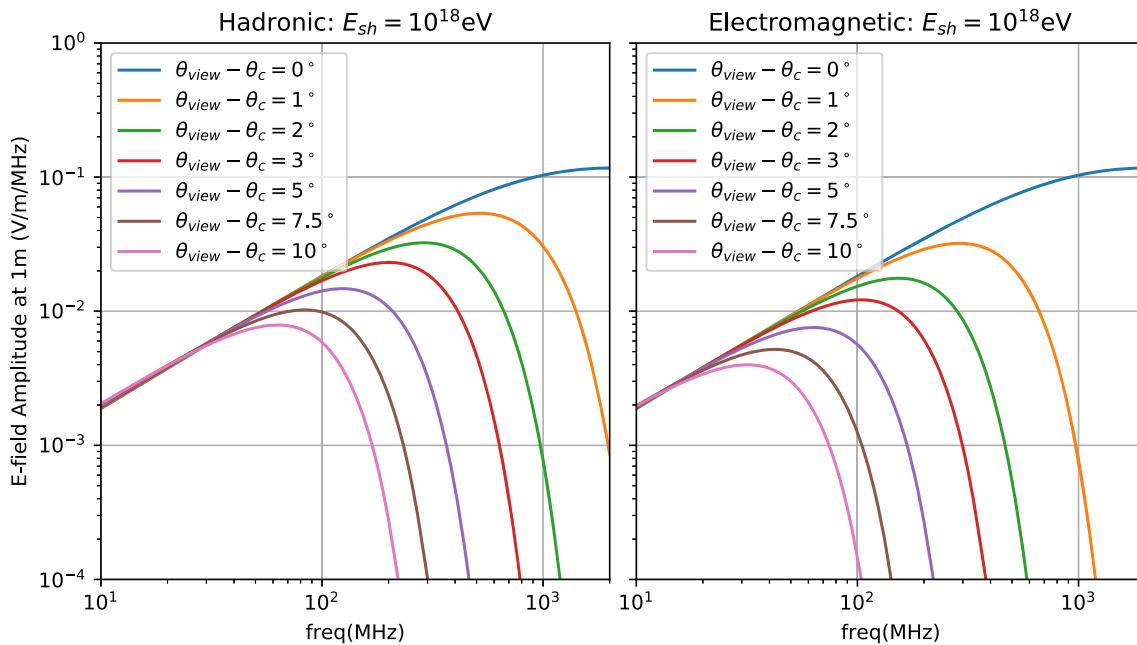


Fig. 5 Electric field amplitude ε^{1m} , 1 m from the neutrino interaction vertex (Eq. (7)) for hadronic (left) and electromagnetic (right) showers with $E_{sh} = 10^{18}$ eV using the parameterization Alvarez2000. Note that

as the viewing angle shifts away from the Cherenkov cone angle, high frequency components fall off. For the EM showers, the cone width σ_θ is reduced due to the LPM effect

simultaneously for the three-dimensional *form factor* of the cascade, and the cascade elongation. The form factor is the spatial Fourier transform of the instantaneous charge distribution of the cascade. The form factor affects the Askaryan signal properties in the same way a multi-pole filter affects any time-domain signal. Although some authors have provided partial solutions for the three-dimensional form-factor in the past [52], in [50] a complete solution is presented that includes dependence on the viewing angle θ . This allows for the analytic exploration of the relevant parameter space affecting σ_θ and σ_v , the width of the Cherenkov cone and the Fourier spectrum, respectively.

This module builds upon the work of [51] where the authors provide analytic functions for Askaryan radiation correct in both the near and far-field regimes. When a cascade is elongated due to the LPM effect, both regimes become important given the three-dimensional nature of the form-factor. *HCRB2017* treats the LPM effect as a smooth stretching of the shower profile using the results of [53].

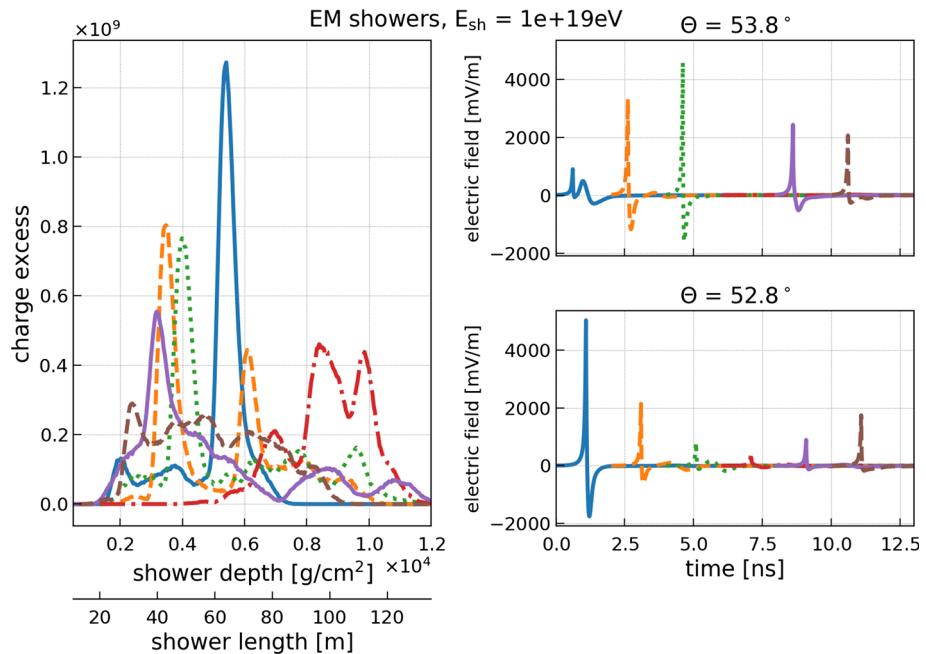
The fully analytic nature of this model has the advantage that it gives direct insights into the physical dependencies of the Askaryan signal. However, as shown in the radio emission of air showers [54] a purely analytic model comes at the cost of a poorer accuracy.

3.3 Semi-analytic model in the time domain

A third option for the signal generation is to calculate the Askaryan radiation individually from detailed charge-excess profiles in the time domain, following the approach in [55]. The implementation in NuRadioMC referred to as *ARZ*, is based on a realistic shower library. This allows to precisely model the effects of LPM elongation [26,27] and the resulting large shower-to-shower fluctuations on the Askaryan signal on a single event basis, rather than describing an average behaviour. The model also captures subtle features of the cascades like sub-showers and accounts for stochastic fluctuations in the shower development which can alter the Askaryan signal amplitudes significantly (see e.g. discussion in [47] or Fig. 6). This model is the most accurate treatment of Askaryan radiation implemented in NuRadioMC, but it comes at the expense of larger computation times as it involves computationally expensive convolutions of the Askaryan vector-potential with Monte-Carlo generated cascade profiles.

The main idea behind the *ARZ* method is that the electromagnetic vector potential \mathbf{A} in Coulomb gauge can be expressed as an integral in shower depth containing the shower profile, a factor that accounts for polarization, another factor that accounts for distance to the emitting point of the shower, and a form factor F_p :

Fig. 6 Charge-excess profiles and resulting Askaryan signal (unfiltered). (left) Charge-excess, i.e., number of electrons minus number of positrons, as a function of shower depth and length of six electromagnetic shower with an initial energy of 10^{19} eV. The variation in the charge-excess profile is due to the stochastic nature of the shower development effected by the LPM elongation. (right) The resulting Askaryan signal for the charge-excess profiles according to the ARZ model for two different viewing angles at 1 km distance. The pulse start time is shifted for a better visibility of all pulses



$$\mathbf{A}(r, z, t) = \frac{\mu}{4\pi} \int_{-\infty}^{\infty} dz' \frac{Q(z')}{\sqrt{r^2 + (z - z')^2}} \mathbf{p}(z') F_p \left(t - \frac{z'}{v} - \frac{n\sqrt{r^2 + (z - z')^2}}{c} \right), \quad (8)$$

where r is the radial distance of the observer to the shower, z is the vertical coordinate of the observer, z' is the shower depth, $Q(z')$ the excess charge profile, \mathbf{p} is the polarization vector and F_p is the form factor (see [55] for more details). This form factor F_p has approximately the same shape for every particle shower in ice, which allows us to treat it as a constant function. It only depends on the type of the shower, i.e., hadronic or electromagnetic, and a parameterization of the form factor for both shower types is provided.

The charge profile $Q(z')$ depends on the nature of the shower (hadronic or electromagnetic), the shower energy, and is also subject to random fluctuations. The LPM effect, for instance, modifies the charge profile, which in turns modifies \mathbf{A} through Eq. (8). All the physical processes that are relevant for the electric-field calculation contribute to $Q(z')$, so as long as a correct description of the charge profile is available in the shower library, an accurate electromagnetic potential \mathbf{A} can be calculated with Eq. (8).

Once \mathbf{A} is known, the radiation electric field can be calculated with a derivative, since in Coulomb gauge $\mathbf{E}_{\text{rad}} = -\frac{\partial \mathbf{A}}{\partial t}$. The agreement between the electric field predicted by the ZHS Monte Carlo and the one obtained with the ARZ model is quite satisfactory, yielding a few percent of error up to 2 GHz (see Fig. 3 in [25]). The ARZ model considers that the shower has a volume and therefore is adequate for computing the fields of observers near the shower as long as the con-

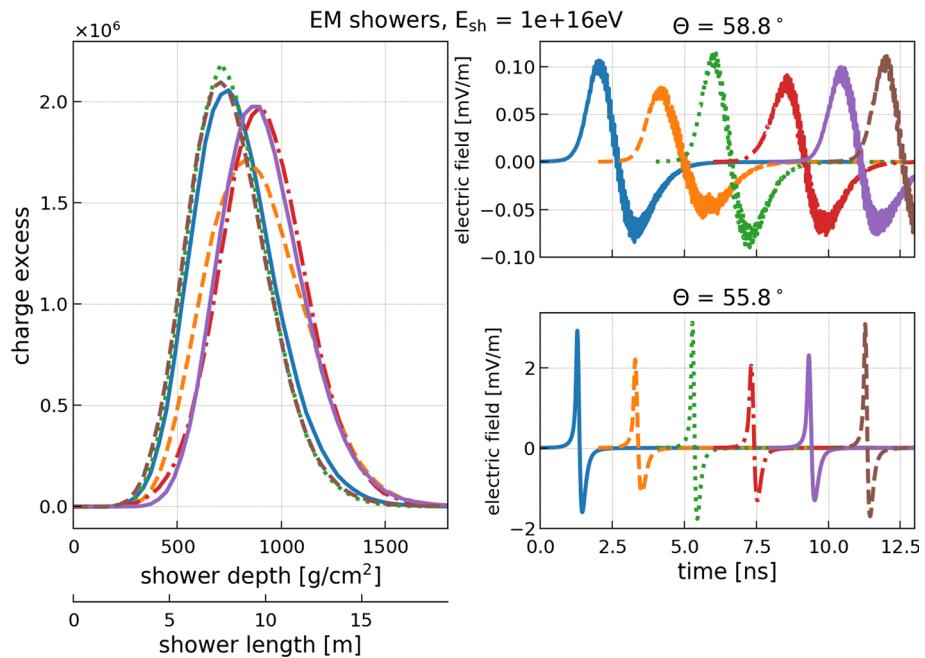
sidered wavelengths are small with respect to the distance to the shower.

NuRadioMC provides a modern Python-based implementation of the code used in [55] and optimized routines for numerical integration. The code includes a shower library of charge-excess profiles for different shower types:

1. electromagnetic: purely electromagnetic showers from ν_e charge current interactions.
2. hadronic (neutrino): showers started by the fragmentation of the nucleon struck by the neutrino, i.e., the result of neutrino neutral current interactions and the hadronic part of an electron neutrino charged current interaction.
3. hadronic (tau): showers initiated by a hadronic decay of a tau lepton. A tau decay into muons will not produce any significant shower, and tau decays into electrons correspond to purely electromagnetic showers.

The last category is not simulated explicitly. Instead, the branching ratios of a tau decay and the fraction of energy ending up in the particle cascades is parameterized using the results of [35, 36]. Then, the shower library of electromagnetic (category 1) or hadronic (category 2) showers is used with the appropriate shower energy. We note that the initial hadronic particles that start the hadronic shower are different between a fragmenting nucleon and a hadronic tau decay. This might lead to small differences in the hadronic shower developments. However, for now we ignore this subtle difference and use category 2 also for hadronic tau decays. In the future, we will provide a separate shower library for category 3. Currently, NuRadioMC comes with version 1.2

Fig. 7 Charge-excess profiles and resulting Askaryan signal (unfiltered). Same as Fig. 6 but for electromagnetic showers with an initial energy of 10^{16} eV. At this energy the LPM effect only has a small influence on the shower development and stochastic shower-to-shower fluctuations are small



of the shower library that will be described in the following.

The showers were simulated using HERWIG [56] for the simulation of the first neutrino nucleon interaction, and ZHAireS [57] for the subsequent simulation of the particle shower in ice. The charge-excess profiles are binned in bins of 37 g/cm^2 for electromagnetic showers and 18 g/cm^2 for hadronic showers. To optimize the computation speed, we integrate Eq. (8) numerically using the trapezoid rule given the binning of the charge-excess profile. The form factor is a strongly peaked function which requires a more precise integration around the peak. This is achieved by dynamically interpolating the charge-excess profile at the positions corresponding to the peak of the form factor.

The shower library (version 1.2) contains 10 showers for every shower energy ranging from 10^{15} eV to $10^{20.5}$ eV in steps of $\Delta \log_{10}(E) = 0.1$ for both electromagnetic and hadronic showers. To obtain charge-excess profiles for shower energies that were not explicitly simulated we do the following: At first order, the charge-excess amplitude scales with shower energy. Hence, in a simulation, we pick one shower realization randomly from the nearest energy bin and re-scale the charge-excess amplitude by $E_{\text{event}}/E_{\text{library}}$.

To discuss and illustrate the improvement in accuracy when using the ARZ approach as opposed to a parameterization, we consider the influence of the LPM effect on the radio signal. The main consequence of the LPM effect is that the interaction probability of high-energy electrons, positrons and photons is suppressed leading to an elongation of the shower profile. The strength of the effect is proportional to the energy of the particle. Therefore, it mostly affects highly-

energetic electromagnetic showers above a few PeV in ice, in which a large amount of energy is carried by individual particles. Previously in the literature (e.g. [50, 57]), the effect was often modelled via stretching of a smooth shower profile. However, this does not take into account the stochastic nature of the process and the fact that the first few particles of an electromagnetic shower are impacted differently by the LPM effect as the energy is not equally distributed. As a consequence, one gets multiple spatially displaced EM showers as shown in Fig. 6. In this figure, also the resulting Askaryan signals are shown for two different viewing angles θ which are significantly different for different realizations of the shower (see Fig. 1 for a sketch of the coordinate system). Low energy EM showers are less influenced by the LPM effect and the resulting Askaryan signals are similar for all shower realizations (cf. Fig. 7). Hadronic showers exhibit little shower-to-shower fluctuations except for the rare cases where a high-energy electromagnetic shower is initiated in one of the first interactions that then gets LPM elongated (see Fig. 8).

3.4 Comparison of models

Each signal generation module in NuRadioMC has its own strengths and shortcomings. We first compare the signal models with respect to their resulting signal properties and then discuss practical considerations. We provide a quick overview of the discussion in Table 1. In Fig. 9, a comparison of the predicted peak-to-peak amplitudes in a typical detector bandwidth of 100 MHz–500 MHz is presented that will be discussed below.

Fig. 8 Charge-excess profiles and resulting Askaryan signal (unfiltered). Same as Fig. 6 but for hadronic showers with an initial energy of 10^{17} eV. Most hadronic showers are not influenced by the LPM effect and show only very little shower-to-shower fluctuations. Different energies mostly scale the charge-excess and electric-field amplitudes approximately linear with energy but have a small effect on the shower length. However, sometimes a high-energy π^0 that is created in one of the first interactions decays instead of interacting leading to an electromagnetic sub-shower that experiences significant LPM elongation (green dotted curve in this figure)

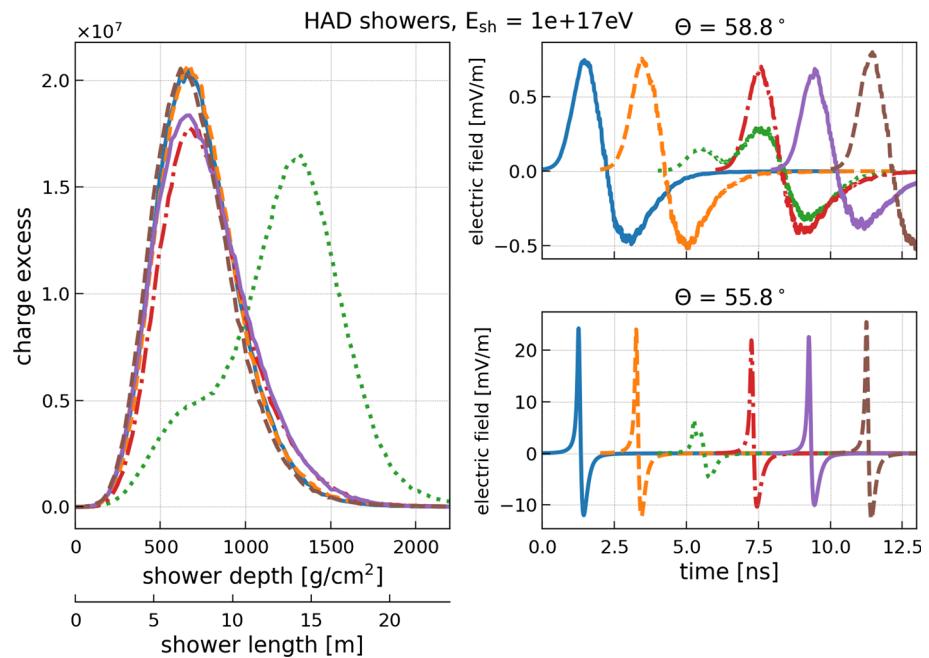


Table 1 Overview of alternative methods implemented in NuRadioMC to calculate the signal following a neutrino interaction

Model	Advantages	Shortcomings
Parameterization (<i>Alvarez2009</i>)	Fast, accurate representation of the signal amplitudes, includes statistical fluctuations from LPM	No phase information, only valid in far-field
Fully analytic (<i>HCRB2017</i>)	Fast, phase information provided, valid in near and far-field, LPM is treated as elongated shower	No statistical fluctuations from LPM, generalization, absolute amplitudes less accurate
Semi analytic (<i>ARZ</i>)	Phase information provided, near and far-field, realistic LPM treatment based on simulated shower library	Computationally expensive
Full MC	Precise modelling of all details of shower development	Slow, no implementation in NuRadioMC yet

The frequency-domain parameterizations are based on a detailed full Monte Carlo simulation of the particle shower and a calculation of the resulting radio signal using the ZHAireS code [46]. Thus, their predictions of the signal amplitudes are accurate, the narrowing of the Cherenkov cone due to the LPM effect is modelled and even statistical fluctuations in the shower development are parameterized (only *Alvarez2009*). The models are fast to evaluate and the computing time is negligible compared to the other parts of the simulation. We also provide an older version, *Alvarez2000*, that was most commonly used in previous simulation frameworks and is therefore important for comparison. However, we strongly recommend the usage of the newer model *Alvarez2009* as the older model typically overestimates the Askaryan amplitudes by roughly 20–30%. The

Alvarez2009 model is in good agreement with the more precise *ARZ* time-domain calculation (cf. Fig. 9).

The main shortcomings of such parametrizations are that no phase information is provided which leads to inaccuracies in the time domain. Typically, the phases are approximated as constant 90° as function of frequency, which results in a perfectly symmetric bipolar pulse. While this may be a reasonable approximation for many cases, it does not capture the details of the shape of the pulses and does not account for physical time delays. Thus, these models are suitable for general sensitivity calculations given the correct prediction of amplitudes. However, more detailed models are recommended to study trigger efficiencies and event reconstruction that are based on pulse shape and timing.

Another option is the fully analytic model *HCRB2017* that also calculates the phases and is thus suitable for the time-

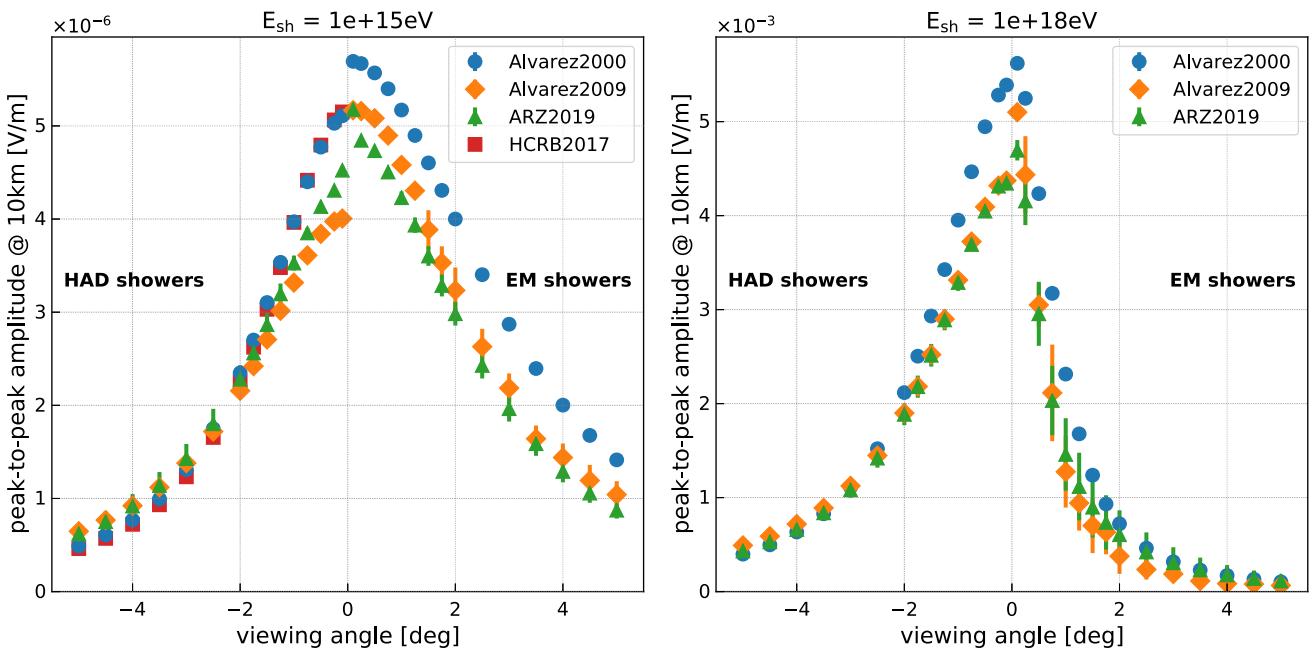


Fig. 9 Comparison of Askaryan models. Shown the peak-to-peak amplitude (the difference between the maximum and the minimum of the Askaryan pulse) as a function of viewing angle. The pulses are filtered in a typical experimental bandwidth of 100 MHz to 500 MHz. The

left part of the plot (negative angles) shows the prediction for hadronic showers and the right part of the plot (positive angles) the prediction for electromagnetic showers of the same shower energy. (left) 10^{15} eV shower energy. (right) 10^{18} eV shower energy

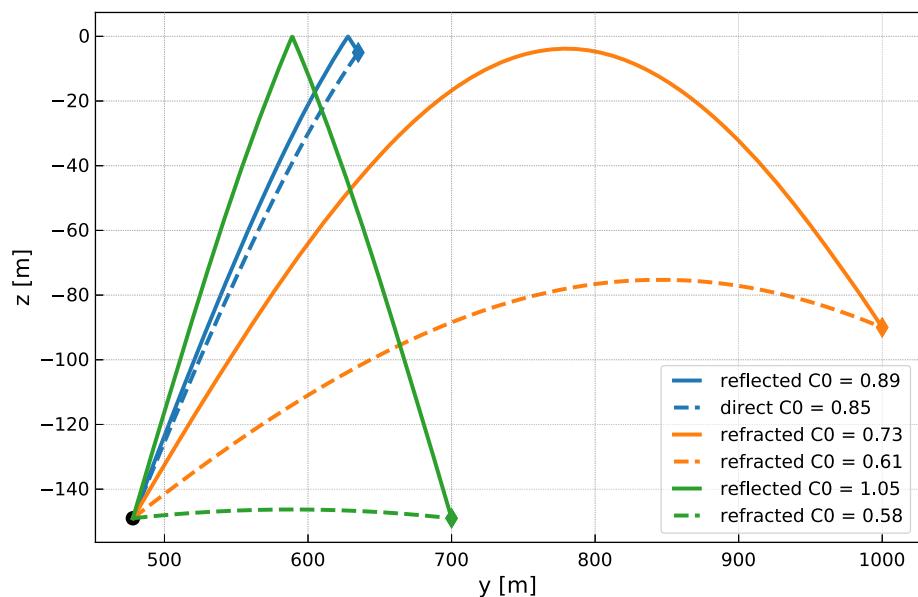
domain. It provides helpful insights into the dependencies of the Askaryan signals on shower elongation and shower width. As being analytically it does not model the statistical fluctuations occurring in showers that can be substantial as shown in Fig. 6. The signal strength prediction depends strongly on the longitudinal cascade width a , which has to be approximated with a Gaussian function for different cases (electromagnetic, hadronic and LPM showers). The approximations lead to a mis-match between the predictions of this model and the ones of the other models that are based on a microscopic Monte Carlo simulation where the calculation of the radio signal is based on first principles resulting in a few percent accuracy as shown in the radio emission of air showers [58]. In particular, the HCRB2017 model overpredicts the amplitudes at higher shower energies and the reduction of the cone width due to the LPM effect. Therefore, we only show the HCRB2017 model for low-energy hadronic showers in Fig. 9. Furthermore, the treatment of pulse arrival times is complex in an analytic model, complicating the integration with the different signal propagation modules (see Sect. 4). Naturally, the model is computationally very fast given its analytic approach.

The semi-analytic model ARZ builds on a shower library of charge-excess profiles and thus models all details like sub-showers including statistical fluctuations in the shower development. The calculation is performed in the time domain. It therefore includes all phase information and gives an accurate

prediction of the pulse shape and timing. The model provides valid results even when the distance from observer to shower is comparable to or smaller than the shower dimensions, as long as the distance is large compared the considered wavelengths. Above 100 MHz, and at distances greater than 10 m, the use of the ZHS formula, on which the ARZ model is based, is justified [59]. It is the most precise model available and recommended for the development of neutrino identification and reconstruction algorithms. Its disadvantage is that it is computationally more expensive. In a full end-to-end simulation it takes up roughly 90% of the computing time. When using this model, the computing time increases roughly by a factor of 10.

The next level of precision can be achieved with full Monte Carlo simulations where each shower particle is tracked and the radio emission is calculated from the acceleration and creation of each charged particle. This is done for air showers in codes like CoREAS [60] and ZHAireS [57], which are required to achieve the necessary accuracy for modern air shower experiments that are pushing the reconstruction uncertainties (e.g. [58, 61–63]). Currently, there is no urgency to require this level of accuracy for neutrino predictions, given the experimental uncertainties and the computational costs of a full Monte Carlo. However, future developments like a next generation of CORSIKA [64] are followed closely to allow for synergies and compatibility with NuRadioMC.

Fig. 10 Example of typical ray-tracing solutions for receiver locations differing in depth and horizontal distance to a given emitter. The emitter is indicated by the black circle at the bottom left. Lines of the same color belong to the same receiver location. Shown are the combinations of direct and reflected ray (blue), refracted and reflected ray (green), and two refracted rays (orange). The numbers in the legend show the C_0 parameter of Eq. (10) that defines the shape of the curve



One could also consider another future improvement in the combination of signal generation and propagation. As discussed earlier, the decoupling of signal generation and propagation leads to noticeable inaccuracies in an inhomogeneous medium (where the signal trajectories are bent, cf. next section) if the extent of the emission region becomes large with respect to the distance to the receiver and if the trajectory is substantially refracted in the firn. Then, the time delay of the propagation time from different emission points to the receiver vary between a homogeneous and inhomogeneous medium, so that signal generation and propagation cannot be separated without loss of accuracy. This effect can be taken into account naturally in a microscopic Monte Carlo simulation by calculating the (curved) path from each emission point to the observer. In an intermediate step, one could use the ARZ2019 model, where the Askaryan signal is calculated from the charge-excess profile to address the issue: Instead of calculating the emission from the full charge-excess profile at once, a shower can be subdivided into small chunks. The Askaryan radiation can then be calculated per chunk and propagated individually to the receiver.

4 Signal propagation

The signal propagation pillar of NuRadioMC handles the propagation of the Askaryan signal through the medium to the observer positions. Like the other pillars, this part of the code is clearly separated so that different signal propagation modules can be implemented and exchanged by the user. This is achieved by defining an interface in form of a Python class (see general example in [65]).

The signal propagation problem is typically approximated via ray tracing but more general techniques such as a finite difference time-domain (FDTD) method that evolves Maxwell's equation can be foreseen in the future [66, 67]. In the ray-tracing approximation, the different ray paths connecting an emitter and receiver can be classified as *direct*, if the depth is monotonously decreasing or increasing along the path between emitter and receiver, as *refracted*, if the path shows a turning point, and as *reflected*, if the ray is reflected off the ice-air interface at the surface which acts as a perfect mirror for most geometries. A few typical ray-tracing solutions are presented in Fig. 10.

4.1 Analytic ray tracing

The default signal propagation module in NuRadioMC is an analytic ray-tracing technique that provides an unprecedented combination of speed and precision relative to traditional ray-tracing techniques. Traditional ray-tracing techniques locate the path connecting an emitter and receiver by time intensive trial-and-error methods, where numerous rays are “thrown” until a ray which connects the emitter and receiver is found. This is necessary because the index-of-refraction (n) of glacial ice is known to vary with depth, and so a light ray is bent and follows a curved path as it travels from an emitter to a receiver. Because the index-of-refraction does not need to be a well-behaved function it is impossible to predict the path traversed by the ray with full generality.

However, ice density measurements and the resulting index-of-refraction profiles from the South Pole and Moore’s Bay site exhibit a simple, depth-dependent index-of-refraction $n(z)$. The data can be described to within a few percent

[68] by an exponential function of the following form:

$$n(z) = n_{\text{ice}} - \Delta_n e^{z/z_0}, \quad (9)$$

where z is the depth and n_{ice} , Δ_n , z_0 are the parameters of the model. For this specific exponential $n(z)$ profile, an *analytic* solution of the ray path as a function of depth ($y(z)$) exists and is given by

$$y(z) = \pm z_0 \sqrt{n_{\text{ice}}^2 C_0^2 - 1} \cdot \ln \left(\gamma / \left[2\sqrt{c(\gamma^2 - b\gamma + c)} - b\gamma + 2c \right] \right) + C_1, \quad (10)$$

with $\gamma = \Delta_n e^{z/z_0}$, $b = 2n_{\text{ice}}$, and $c = n_{\text{ice}}^2 - C_0^{-2}$. We provide a derivation of this equation in Appendix C.1. The parameters C_0 and C_1 uniquely describe the ray path and need to be determined from two initial conditions which are given by the two points the ray goes through, e.g., the neutrino interaction vertex (the point of emission) and the observer position.

The parameter C_1 corresponds to a vertical translation in the coordinate system and can be calculated analytically from the initial conditions. The parameter C_0 must be determined numerically, and is found through a least-squares minimization. For each receiver-emitter coordinate pair, we can either have no, one or two solutions, corresponding to no connecting ray, one connecting ray, or two connecting rays. To quickly and stably find all possible solutions, we leverage numerical algorithms as documented in Appendix C.2.

4.1.1 Derived quantities

Once a ray path is found, several derived quantities are needed in the simulation. The *launch vector* of the ray is needed to calculate the *viewing angle* (the angle between shower axis and launch vector) which is required to calculate the Askaryan emission. The *receive vector* is needed to evaluate the antenna response for the arrival direction of the incident radiation. As discussed in Appendix C.2, the ray-tracing problem can be reduced to the y-z plane with a simple coordinate rotation. Hence, only the launch and receive angles are required, which can be calculated analytically from the derivative $dy(z)/dz$ which we specify in appendix Appendix C.4.

The path length can be calculated numerically via the following line integral

$$d = \int_{z_1}^{z'_2} \left| \frac{d\mathbf{x}}{dz} \right| dz = \int_{z_1}^{z'_2} \sqrt{1 + \left(\frac{dy(z)}{dz} \right)^2} dz, \quad (11)$$

where $\mathbf{x} = (y(z), z)^T$, and $z_1/2$ refer to the z position of the emitter/receiver. In case of a direct ray we have $z'_2 = z_2$. In case of a refracted or reflected ray, we first need to integrate from z_1 to the turning point and then the same path backwards to z_2 .

Similarly, the travel time t and the signal attenuation $\exp(-A)$ can be calculated as

$$t = \int_{z_1}^{z'_2} n(z)/c \left| \frac{d\mathbf{x}}{dz} \right| dz = \int_{z_1}^{z'_2} n(z)/c \sqrt{1 + \left(\frac{dy(z)}{dz} \right)^2} dz, \quad (12)$$

and

$$A = \int_{z_1}^{z'_2} \left| \frac{d\mathbf{x}}{dz} \right| / L(z, f) dz \quad (13)$$

$$= \int_{z_1}^{z'_2} \sqrt{1 + \left(\frac{dy(z)}{dz} \right)^2} / L(z, f) dz \quad (14)$$

where $L(z, f)$ is the attenuation length as a function of depth and frequency which is discussed in Sect. 6.5.

If the index of refraction profile is described with an exponential function as in Eq. 9, an analytic expression for the path length and travel time can be derived. This analytic function is used by default due to its improved computing time. The derivation can be found in Appendix C.5. For the attenuation factor no analytic solution has been found and a numerical integration is required.

4.1.2 Computational speed

We provide a Python implementation of the analytic ray-tracing technique described above which leverages the NumPy [69] and SciPy [70] computational packages. In addition, we implemented the time critical operations of finding the ray-tracing solution and determining the signal attenuation in a standalone C++ module. This C++ module leads to a substantial speed improvement of a factor of 20, so that the calculation of the ray-tracing solutions and the calculation of travel time and distance as well as the signal attenuation takes less than 4 ms in ice. The C++ module utilizes the highly optimized and broadly supported GNU Scientific Library (GSL) [71] for numerical integration and root-finding.

We provide a Cython wrapper to the C++ implementation so that it can be called as a sub-routine. Selection of routine (C++ or Python) is done in a transparent fashion. If the user compiled the C++ extension, NuRadioMC will automatically pick the faster C++ implementation, and otherwise utilize the Python implementation. In this way, the

NuRadioMC code works out-of-the-box without additional dependencies. The Python implementation is still sufficiently fast to be used for many problems.

4.2 Focusing effect due to ray bending

Applying the ray approximation to signals from neutrinos in case of ray bending, requires an additional correction factor on the signal amplitude. In general, when considering many rays which are bent there can either be a convergence or divergence of rays. If there is a convergence the ray density and thereby the amplitude of the signal will increase, and conversely so for a divergence. For the ice geometry, refraction contains the signal within the ice, and an amplification is expected if the receiver is above the point of emission and the ray is not reflected from the surface.

We calculate a correction factor from an energy conservation argument: The intensity along the ray is given by

$$I = \sqrt{\frac{\epsilon \varepsilon^2}{\mu c}} = \frac{n \varepsilon^2}{c}, \quad (15)$$

for $\mu = \mu_0$ where ε is the electric-field amplitude, c the speed of light and n the index of refraction. The total power contained in a ray bundle is $P = I A$ with A being an area perpendicular to the propagation direction, so the electric field strength propagates as

$$\varepsilon' = \varepsilon \sqrt{\frac{n A}{n' A'}}. \quad (16)$$

The power radiated into a given solid angle is fixed by the source. For a spherical geometry we have

$$dA = R^2 d\Omega = R d\theta \times R \sin \theta d\phi. \quad (17)$$

For refracted rays the relation $\frac{dA}{d\Omega}$ changes during propagation. Assuming a planar index of refraction model, i.e., it only depends on the depth z , only the $R d\theta$ part changes and is given by

$$dA' = \frac{dz}{d\theta} \sin \theta' d\theta \times R \sin \theta' d\phi. \quad (18)$$

See Appendix C.6 for a derivation of this relation. Then, the ratio of electric field amplitudes is given by

$$\frac{\varepsilon'^2}{\varepsilon^2} = \frac{n}{n'} \frac{dA}{dA'} = \frac{n}{n'} \frac{R}{\sin \theta' \frac{dz}{d\theta}} \quad (19)$$

in the limit of $\theta \approx \theta'$, which is applied as a correction factor to the calculated electric field amplitude from the signal generation module. The factor $\frac{dz}{d\theta}$ is calculated numerically using the ray tracing code by calculating a new ray to the

receiver position which is vertically displaced by a small amount $\Delta z \approx 1$ cm.

Emitter positions very close to the shadow zone boundary require special attention as the correction diverges because $\frac{dz}{d\theta}$ approaches zero. This is not physical but an artifact from treating both emitter and receiver as a point. However, in reality the emission region is extended over several meters due to the extent of the particle shower (cf. Fig. 7) and also the antenna is an extended object. Thus, we studied the stability of the correction factor under small changes of the emitter position by ± 5 m corresponding to typical dimensions of the emission region. We find that correction factors below about a factor of $2\times$ in amplitude vary by less than 10% when the emitter position is varied. Larger amplification factors in-turn are not stable. Hence, limiting the amplification to a maximum of $2\times$ removes unphysical correction factors. Furthermore, we studied the effect of the limit value. Limiting the focusing correction to a factor of $1.5\times$, $2\times$ and $3\times$ results in essentially the same effective volume (i.e. sensitivity of the detector) over a broad range of neutrino energies. Thus, the exact choice limit value is not that important as long as very large amplification factors are removed. As default we limit the focusing correction to a factor of $2\times$ but allow the user to configure this value via the config file.

The effect of focusing is strongest when the rays pass near the surface and experience significant refraction. For a receiver close to the surface we find an increase in the effective volume of the order of 10% due to this correction.

4.3 Numerical ray tracing for arbitrary density fields

In the future, it may become necessary to describe the ice in more detail than an exponential profile that only depends on the depth. This will require a more detailed ray tracing that takes into account an arbitrary 3D index of refraction profile $n(x, y, z)$. We have already foreseen this case and ensured that necessary hooks are available in the code.

Interestingly, the computational problem of the propagation of ultra-high energy cosmic rays through the universe is similar to propagating a ray through the ice. Instead of magnetic fields bending the trajectories of charged cosmic-ray particles, the ray is bend according to the spatial distribution of the index of refraction. Where the cosmic ray can spallate into secondaries, a ray can be partly transmitted and reflected. Consequently, we considered the cosmic-ray propagation code CRPropa [72] as one option and have started to modify it for our needs.

The resulting code RadioPropa [73] solves the Eikonal equation in a local paraxial approximation thus enabling casting of rays through materials with arbitrary varying refractive index as may be required here. In addition, RadioPropa handles effects from boundary traversals such as reflection or partial reflection and allows for the implementation of prop-

agating components of the electric field differently, such as needed for birefringence. It automatically tracks several parts of the original ray, making it also suitable for other less well understood phenomena in the ice. In the same way as NuRadioMC, RadioPropa is modular and flexible, leaving room for future developments. It is currently under heavy development and therefore not yet fully included in NuRadioMC.

4.4 Signal propagation beyond ray tracing

Ray tracing describes the path taken by light in the limit where the wavelength is much smaller than any relevant feature sizes. While this is appropriate in most practical cases, i.e., when the ice is uniform or has a slowly-varying index of refraction, ray tracing does not offer a full description of light propagation near dielectric interfaces, where additional solutions to Maxwell's equations exist, (see e.g. [74] for a pedagogical tutorial on some of the solutions, or [75] for a complete solution for the field of a particle track). In addition to the ice-air interface at the surface, variations in ice density are present below the surface, producing a set of dielectric interfaces. These may result in signals being observed at locations, where simple models assuming a smooth gradient predict no radio signals [68]. While adaptations to the analytic ray-tracing requiring a smooth gradient of the index of refraction, deliver solutions for special cases, the finite-difference time-domain (FDTD) method may be used to model propagation in ice even in the presence of inhomogeneities in all its aspects [66, 67].

Interesting phenomena that arise include the existence of potentially detectable (though generally small) signals coming from regions where there is no ray-tracing solution, diffraction and interference of the radio waves, and the presence of caustics, where the small electric field may be significantly amplified in some geometries [67].

While these effects will slightly modify the effective volume of a detector and provide additional opportunities for event reconstruction, direct integration of an FDTD solver into NuRadioMC is challenging for the purpose of providing a simulation framework. FDTD methods are very computationally and memory intensive, requiring discretization on the scale of a tenth of the smallest relevant wavelength in all spatial dimensions as well as time. Directly simulating the entire volume seen by a typical in-ice station is extremely computationally challenging in three dimensions with our present capabilities – we estimate a single simulation of a cubic kilometer volume valid up to 500 MHz would take $\mathcal{O}(10^7)$ CPU-hours. One can envision the usage for a single event (in case of re-simulation of a detected shower for example), the integration for all events is, however, impractical.

By considering only azimuthally-symmetric antennas and density variations dependent only on depth, it is possible to simulate a transmitting in-ice antenna in just two dimensions,

greatly reducing the computational burden. We are investigating techniques exploiting reciprocity in order to tabulate the propagation properties of the equivalent time-reversed geometry, corresponding to a receiving antenna. Such tabulated properties could then be incorporated into NuRadioMC in an efficient manner.

5 Detector simulation

The fourth pillar of NuRadioMC is the detector simulation, i.e., the calculation of the detector response to an electric field at the antenna and subsequent trigger simulation. We use the software *NuRadioReco* for this task [28]. NuRadioReco is a software for the detector simulation and event reconstruction of radio neutrino and cosmic-ray detectors. It is written in Python and also follows a modern modular design so that it nicely integrates into NuRadioMC.

5.1 Antenna simulation

The most important part in the simulation of the detector response is the impact of the antenna. NuRadioReco provides antenna response pattern of typically used antennas such as LPDAs, dipoles or bicone antennas that were simulated with dedicated codes such as WIPL-D [76] and XFDTD [77]. NuRadioReco also provides an interface to the output of these codes such that new antenna models can be added if necessary.

In earlier software, the response of the antennas was typically treated in a simplified way, only assuming real gain factors and a simple polarization response, i.e. ignoring contributions polarized orthogonal to the main antenna sensitivity. According to methods already standard in the treatment of radio signal from cosmic rays (e.g. [63, 78]), the antenna response is modelled fully frequency-dependent in NuRadioReco, also taking into account the group delay induced by the antenna and its sensitivity to two orthogonal polarization components.

5.2 Trigger simulation

Especially when looking for small signals, as expected from neutrinos, the simulation of the trigger mechanism is essential. The trigger simulation is set up as such that any instrumental trigger can be rebuilt in software. NuRadioReco offers modules to simulate different trigger conditions, e.g., a simple threshold trigger, a high and low trigger as implemented on the SST electronic [79] used by ARIANNA [8] that also allows to specify temporal coincidences between different channels, or more complex triggers such as the phased array concept used by ARA [12] have been included to model the instrument response as implemented in the fields.

5.3 Usage in complex detectors

NuRadioReco was built to reconstruct data from an existing detector. In order to facilitate complex detectors without creating too much overhead, the detector description is stored in a database allowing for a description of every single detector component. While this functionality will be helpful to simulate specific events for an existing detector, it is much too complex for design studies. Therefore, NuRadioReco also allows the user to define the detector description in a human readable JSON format, with reduced complexity. This means both that the detector description only needs to be as complex as minimally required and it significantly speeds up simulations. The information ranges from basic parameters such as the positions of the antennas, their type and orientation to more detailed properties such as the sampling rate of the digitizing electronics, the cable lengths or details about the amplifier and ADC. The detector simulation modules have access to these properties and will simulate the detector response accordingly. An example of a typical detector simulation is provided in Appendix E.

6 Utilities

The four pillars of NuRadioMC are complemented by a set of utility classes that are available to all modules, such as units and medium properties.

6.1 Cross-sections and inelasticities

The cross-section of neutrinos at energies relevant for radio detection are still subject to study, given that these energies have never been probed. Different current extrapolations [31, 33, 34] have been implemented in NuRadioMC in the central utilities, so that the cross-sections can easily be exchanged throughout the code, if so desired.

6.2 Earth models for neutrino absorption

To simulate the sensitivity of a neutrino detector, we need to calculate the probability of a neutrino reaching the detection volume. The Earth atmosphere has negligible absorption for high energy neutrinos but the Earth becomes opaque at high neutrino energies. Hence, NuRadioMC comes with multiple models to calculate the Earth absorption so that we can assign each simulated neutrino a weight, i.e., a probability of reaching the detection volume.

Right now, NuRadioMC provides two Earth models: a simple Earth model with a constant density and a core-mantle-crust Earth model with three layers of different densities. Due to the modularity, it is straight forward to add more detailed models if deemed necessary.

Currently, we do not model *tau regeneration*: A tau lepton that is created following a tau neutrino interaction can propagate significantly through the Earth and potentially decay with a relatively large energy and producing another tau neutrino that can interact close to the detector. We plan to include this effect in a future version of NuRadioMC using e.g. the code of [80, 81].

6.2.1 Simple Earth model

This model uses a constant density of 2900 kg/m^3 and by default uses the cross section (σ) based on [31]. It then calculates the distance the neutrino goes through the Earth as

$$d = 2R_e \cos(\pi - \vartheta), \quad (20)$$

where R_e is the radius of the Earth and ϑ is the zenith angle of the neutrino direction. The weight of an event is then

$$\text{weight} = e^{-d\sigma\rho/\text{AMU}}, \quad (21)$$

where ρ is the constant density of the Earth and AMU is the atomic mass unit in kg.

6.2.2 Core-mantle-crust Earth model

NuRadioMC provides a more realistic Earth model with three layers of different densities which is the default model. In this model, the cross section is per default calculated based on [33] and the propagation distance is calculated through three different layers. The weight is calculated as

$$\text{weight} = e^{-(d_1\rho_1 + d_2\rho_2 + d_3\rho_3)\sigma/\text{AMU}}, \quad (22)$$

where d_1, d_2, d_3 are the distances through three layers and ρ_1, ρ_2, ρ_3 are the three densities.

6.3 Handling of Fourier transforms

NuRadioMC provides a consistent internal handling of Fourier transforms. A common source of errors when using time- and frequency-domain calculations simultaneously is the normalization of the Fourier transforms. There are several reasons for different normalizations depending on the purpose and context. All NuRadioMC Fourier transforms adhere to Parseval's theorem and previously existing Askaryan signal parameterizations have been adjusted to match the FFT definition used in NuRadioMC. Details are discussed in Appendix D.

6.4 Handling of units

In simulations, typical errors occur during the handling of units. To prevent that, NuRadioMC (just like NuRadioReco)

employs a default system of units, a concept borrowed from the Pierre Auger Observatory offline analysis framework [82]: every time a physical variable is defined, it is multiplied by its unit, and every time a variable is plotted or printed out in a certain unit, it is divided by the unit of choice. All other calculations within the code can then be done without considering units.

```
from NuRadioMC.utilities import units
time = 132. * units.ms # define 132 milliseconds
distance = 5. * units.mm # define 5 mm
speed = distance/time
print("the speed is {:.2f} km/h"
      .format(speed/units.km*units.hour))
# the speed is 0.14 km/h
```

The units utilities are available to modules written in both Python and C. In order to facilitate this, no standard Python package was used.

6.5 Attenuation length and other medium characteristics

As discussed in Sect. 4 the signal propagation is a significant part of the neutrino simulation and an area where lots of development is still to be expected. Consequently characteristics of the interaction medium are stored centrally in the utilities to avoid contradicting definitions in modules. We describe the index-of-refraction profile and signal attenuation properties separately to allow for simulation with different combinations of the two. Which model is being used in a NuRadioMC simulation is controlled via the central config file (see Sect. 7.2).

Currently, a signal attenuation model for South Pole ice is provided that is based on a custom model used by the ARA experiment [83]. For the index-of-refraction profile we provide exponential parameterizations to data from for the South Pole and Moore's Bay [68], as well as from Greenland [84, 85].

6.6 Flux calculations and sensitivity limits

In order to compare the performance of different experimental designs, typically quantities like the effective area, volume or expected limits are compared. Since also here, many definitions are common (e.g. 90% confidence upper limits vs. 5σ discovery fluxes), utility functions are provided centrally.

7 Example 1: calculation of the sensitivity of an Askaryan neutrino detector

In this section we present a full example of the capabilities of NuRadioMC to simulate the sensitivity of an Askaryan

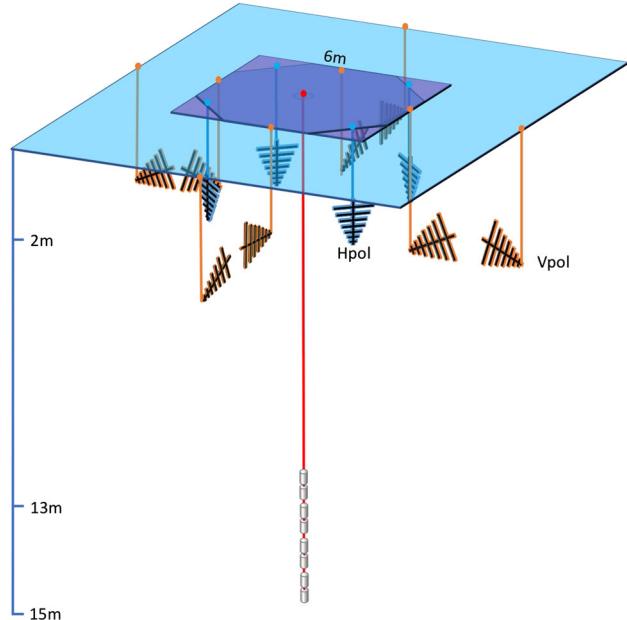


Fig. 11 Sketch of the station layout simulated in Example 1

detector. We choose a station layout that combines log-periodic dipole antennas (LPDA) near the surface with slim dipoles deployed in a borehole deeper in the ice. The specific layout is depicted in Fig. 11. This station layout does not necessarily reflect the authors' opinion on the optimal detector layout but was chosen because it highlights NuRadioMC's capabilities: Antennas of different type, orientation and depth are simulated, the location close to the surface makes a detailed propagation of the signal through the firn necessary, and multiple trigger conditions need to be calculated for different sets of antennas. In the following, only the relevant code snippets are shown. A comprehensive tutorial can be found online [86].

7.1 Event generation

The first step in the simulation is the event generation. The event generation is done stand-alone and produces a list of neutrino interactions in the ice with all necessary properties saved in a simple HDF5 format (see Sect. 2 for details and advantages of separating this step). We choose to generate several input lists, each for a fixed neutrino energy to study the energy dependence. We only consider the initial neutrino interaction. A discussion of the impact of additional Askaryan signals from decaying taus or interacting muons goes beyond the scope of this publication.

A list of one million neutrino interactions with an energy of $E_\nu = 10^{18}$ eV in a cylindrical volume saved in chunks of 10,000 events can be generated with

```
generate_eventlist_cylinder('1e18_n1e6.hdf5',
    n_events=1e6, n_events_per_file=1e4,
    Emin=1e18 * units.eV, Emax=1e18 * units.eV,
    fiducial_rmin=0, fiducial_rmax=5 * units.km,
    fiducial_zmin=-2.7 * units.km, fiducial_zmax=0)
```

The radius needs to be set large enough to include all events that can trigger the detector and is set to 4 km here. For larger neutrino energies, the radius needs to be extended and for lower energies the simulation volume can be decreased to save computing time. The vertical extent of the volume ranges from the surface to the bottom of the ice layer at a depth of 2.7 km at the South Pole.

7.2 Configuration of simulation parameters

The settings of the simulation are controlled with a config file in the human-readable yaml format. The user only needs to specify a parameter if it should be different from its default value. An example configuration with typical settings is shown in listing 1. Typical parameters are the choice of signal generation model (*Alvarez2009* in this example), the ice model, or if noise should be generated and added to the signal in the simulation.

```
noise: False # specify if simulation should be
    run with or without noise
sampling_rate: 5. # sampling rate in GHz used
    #internally in the simulation.
speedup:
    minimum_weight_cut: 1.e-5
    delta_C_cut: 0.698 # 40 degree
propagation:
    ice_model: southpole_2015
signal:
    model: Alvarez2009
trigger:
    noise_temperature: 300 # in Kelvin
weights:
    weight_mode: core_mantle_crust # core_mantle_crust:
        #use the three layer earth model,
        #which considers the different densities of the
        #core, mantle and crust.
        #Simple: use the simple earth model,
        #which applies a constant earth density
```

Listing 1: Example of NuRadioMC’s config file. All parameters are specified in a default system of units. See text for details.

7.3 Detector description

The detector description consists of two parts. First, we need to define the layout of the detector (position, type, and orientation of the antennas), and the sampling rate. Additional parameters such as cable delays and amplifiers can be specified if needed (cf. Sect. 5.3 and NuRadioReco [28]). However, in this example we will perform a simplified detec-

tor simulation sufficient to estimate the sensitivity of an Askaryan detector. The detector description is specified in a JSON file presented in List. 2.

```
{
    "channels": {
        "1": {
            "station_id": 101,
            "channel_id": 0,
            "ant_type": "createLPDA_100MHz",
            "ant_position_x": 3,
            "ant_position_y": 0,
            "ant_position_z": -2.0,
            "ant_rotation_phi": 180,
            "ant_rotation_theta": 90,
            "ant_orientation_phi": 0,
            "ant_orientation_theta": 180,
        },
        ...
    },
    "stations": {
        "1": {
            "pos_altitude": 0,
            "pos_easting": 0,
            "pos_northing": 0,
            "pos_site": "southpole",
            "station_id": 101
        }
    }
}
```

Listing 2: Example of detector description. Only the first channel is shown which defines a downward facing LPDA at a depth of 2 m with its tines oriented along the Northing direction.

Second, we need to specify basic details of the signal chain, i.e., what filter is being used and which triggers are calculated. These tasks are done by dedicated NuRadioReco modules [28] (see Sect. 5.3) that interface directly with NuRadioMC. Instead of simulating just a single trigger condition as shown in the example, a separate trigger can be simulated for each parallel pair of LPDA antennas and for the dipole antennas. This is achieved by calling the same trigger module several times with different arguments. The full example can be found in the online tutorial [86].

7.4 Running the simulation, results, and visualization tools

The NuRadioMC simulation is run by executing the steering script from the command line. The flexibility to split up the input data set into smaller chunks is part of the event generator, so multi-processing computing resources can be used right away. A detailed example on how to run NuRadioMC on a cluster is available in the online tutorial [87].

The sensitivity of the detector is quantified in terms of effective volume to an isotropic neutrino flux. It is given by the weighted sum of all triggered events divided by the total

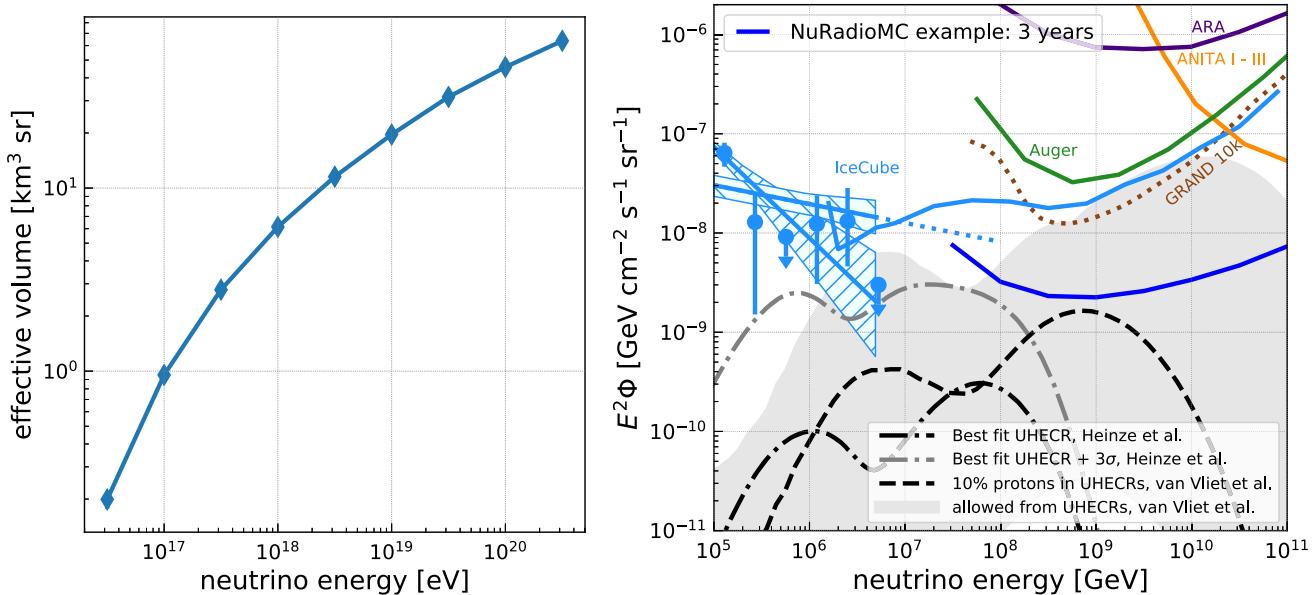


Fig. 12 (left) Effective volume of one example detector station (right) corresponding expected limit for a diffuse neutrino flux for a detector comprising 100 stations and an uptime of 3 years. Shown are for comparison neutrino flux measurements from IceCube [88–90], the Pierre

Auger Observatory [91], ANITA [92], and ARA [10], as well as neutrino flux prediction models from [93,94] calculated using the restrictions from ultra-high energy cosmic rays. We also compare to other proposed arrays [95]

number of events multiplied by the simulation volume and the simulated solid angle (typically 4π). The weighting factor is the probability of a neutrino reaching the simulation volume (and not being absorbed by the Earth). The effective volume of our example detector station is presented in Fig. 12 (left). This effective volume can be converted into an expected limit on the diffuse neutrino flux which is shown in the right panel of Fig. 12. The required tools to make these standard post-processing plots are also part of NuRadioMC.

Furthermore, a standard set of debug plots can be automatically generated from the output files. The distribution of the neutrino interaction vertices of events that triggered the detector is shown in Fig. 13 (left). The upper right (triangular) part of the volume correspond to positions in the *shadow zone* where signals cannot reach the detector according to the ray tracing. The lower left region has little events because the Askaryan signal is only emitted towards the antennas if the neutrino is up-going, i.e., it travelled through the Earth and its probability of reaching the detector is small. The right panel shows the ratio of neutrino flavors and interaction types that triggered the detector. In this case, most triggered events were electron neutrino charged-current (CC) interactions where the full neutrino energy is deposited in particle showers producing an Askaryan signal.

8 Example 2: calculation of the efficiency to detect a signal from both the direct and reflected path

In this example, we calculate the efficiency of an in-ice antenna to observe both the direct Askaryan signal and the signal reflected at the ice surface. For most shower geometries there is total internal reflection of the Askaryan signal at the ice surface, i.e., the ice-air interface acts as a mirror. Consequently, an antenna installed within the ice has the chance to see two pulses: one pulse that propagated straight to the antenna and a second pulse that was reflected off the surface. Detecting this *D'n'R* (direct and reflected) signature is advantageous and an Askaryan neutrino detector will benefit strongly from detecting both pulses: First, it provides a unique method to identify a neutrino interaction in the ice as origin of the detected radio signal, and second, the time difference between the two pulses allows for an improvement in the reconstruction of the distance to the neutrino interaction vertex which is a crucial ingredient for the reconstruction of the neutrino energy. See [9] and [96] for first experimental results concerning this effect using pulsars deployed in the Antarctic ice at South Pole.

There are several effects that influence the efficiency of detecting both pulses that are all taken into account in the NuRadioMC simulation:

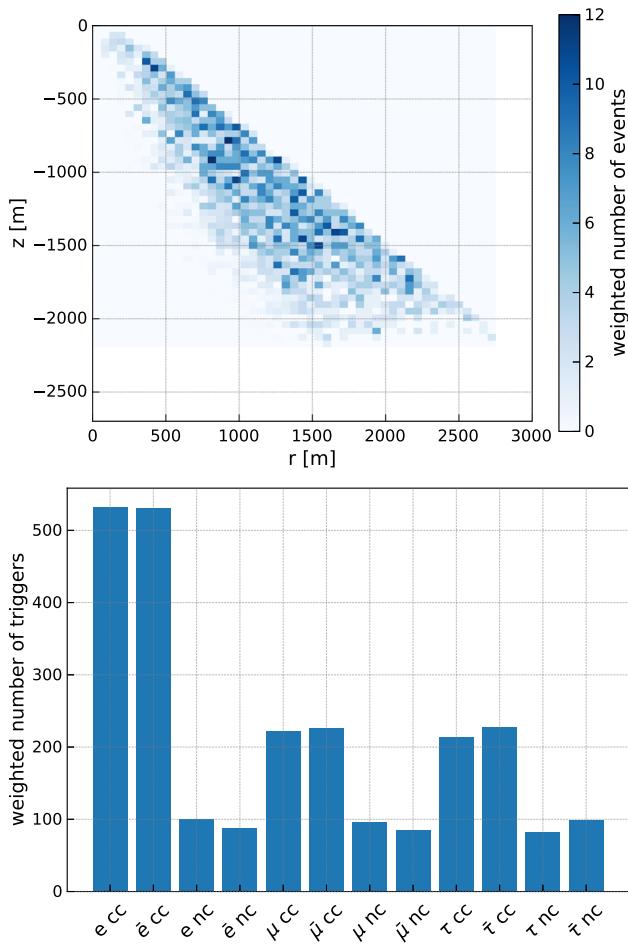


Fig. 13 Visualization for the energy bin of 10^{18} eV neutrino energy. (top) Distribution of neutrino interaction vertices of all triggered events. (bottom) Flavor and interaction type (charged or neutral current) distribution of triggered events

- The reflection coefficient depends on the incident angle of the radio pulse at the ice surface and can range from 1 (total internal reflection) to 0 (no reflection) at the Brewster angle.
- The reflection results in a phase shift of the Askaryan pulse which can alter the amplitude of the pulse. This is modelled using the complex Fresnel coefficients.
- Due to the changing index of refraction in the upper ice layers the signal propagates on curved paths. We find all possible paths to each antenna via ray-tracing. We note that not only a ‘direct’ and ‘reflected’ path will provide a useful signature but any two distinct paths through the ice to the antenna. In case only one solution exists, the efficiency to detect two pulses is of course zero.
- The different ray paths correspond to different launch angles of the signal. This results in a potentially large difference of the amplitude of the Askaryan signal as the launch angles correspond to different viewing angles.

- Antennas have a different sensitivity to different incoming signal directions.
- The two ray paths have different propagation distances and potentially propagate through ice with different attenuation lengths.

In the following we describe an example of how to simulate the D’n’R detection efficiency with NuRadioMC and explain the relevant parts of the code. The full code of this example can be found online at [97].

The D’n’R efficiency depends on the depth of an antenna, hence, we want to define a detector with several antennas of the same kind at different depths. As antenna type we choose a bicone antenna as used by the ARA experiment as such an antenna is sensitive to the dominant vertical polarization, fits into narrow boreholes, and has very little signal dispersion which helps to measure the time difference between the two pulses. Hence, we set up a detector with vertically oriented bicone antennas every 10 m down to a depth of 100 m.

It does make sense to study the D’n’R efficiency as a function of neutrino energy. Therefore, we can use the same script to generate the input event list as in the previous example.

8.1 Set-up of detector simulation

In the previous example we have discussed how to simulate the detector response and the trigger. In the detector simulation so far, all signals that reach the antenna from the different ray path solutions, are combined into a single voltage trace on which the trigger condition is determined. However, for the D’n’R study, we not only need to determine if the detector could observe/trigger a certain event, but also if both pulses are visible. Hence, a dedicated *NuRadioReco* module called *calculateAmplitudePerRaySolution* was written, which simulates the antenna response to each pulse separately and calculates and saves the resulting maximum amplitude. Following this we can calculate if a triggered event has two visible pulses.

As trigger condition we choose a simple threshold trigger of $2 V_{rms}$ that runs on all channels (i.e. antennas) independently. The NuRadioMC simulation is then executed as described in Example 1.

8.2 Results

We now assume a more stringent cut in which all events that produce at least a 3σ ($3 V_{rms}$) signal can be recorded. For the second pulse the requirement for identification is assumed smaller at 2σ . Furthermore, we require that the time difference between the two pulses is smaller than 430 ns

which is assumed as typical record length. We then calculate if an event has triggered via

$$B_i = A_1^i \geq 3 V_{\text{rms}} \quad \text{or} \quad A_2^i \geq 3 V_{\text{rms}} \quad (23)$$

and if both pulses are visible via

$$C_i = ((A_1^i \geq 3 V_{\text{rms}}) \text{ or } (A_2^i \geq 3 V_{\text{rms}})) \quad (24)$$

$$\text{and } ((A_1^i \geq 2 V_{\text{rms}}) \text{ and } (A_2^i \geq 2 V_{\text{rms}})) \quad (25)$$

$$\text{and } (\Delta T < 430 \text{ ns}), \quad (26)$$

where A_1^i and A_2^i are the amplitudes of the two pulses of event i .

Then the D'n'R efficiency is then given by

$$\epsilon = \sum_i C_i / \sum_i B_i \quad (27)$$

where the summation runs over all simulated events i . This calculation is performed for each simulated antenna depths, and for each set of simulated neutrino energy separately.

We simulated 10 million events per neutrino energy and obtain the result presented in Fig. 14. The D'n'R efficiency depends strongly on depth and energy and is best at shallow depth and high energies.

It should be noted that D'n'R efficiency is not the only parameter that one should optimize an array for. For example, a shallower station generally has a smaller effective volume than a deep station, and the fraction of sky coverage also depends of depth. Together with a diverse choice of antennas influencing reconstruction capabilities, data volume restrictions, and instrument costing, optimizing a detector layout is a complex problem, for which NuRadioMC provides guidance.

9 Example 3: Optimization of station spacing for an Askaryan neutrino detector

In this example we calculate the probability to detect a signal from the same neutrino in multiple stations of an array. For a discovery detector, one objective is a large sensitivity which means that it is beneficial to separate stations far enough to minimize station coincidences. However, one may want to optimize differently in the future to have a large fraction of coincidences to improve reconstruction quality. Here, we show how the coincidence fraction can be studied as a function of station separation distance, neutrino energy, and antenna depth. The full code of this example can be found online at [98].

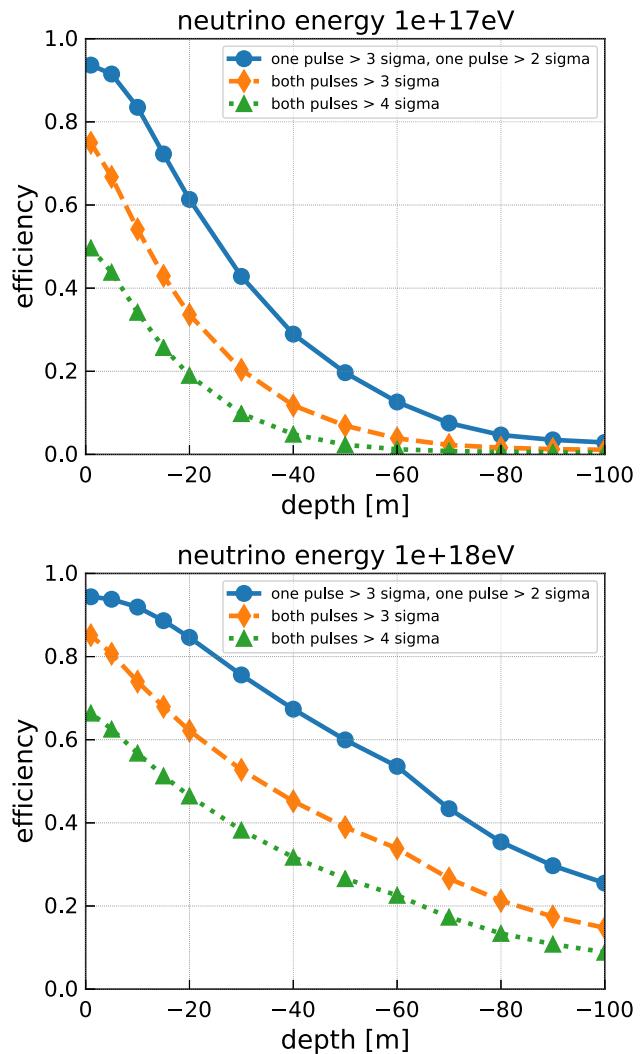


Fig. 14 Efficiency to detect both the direct and reflected Askaryan pulse as a function of depth of the receiver. (top) For a neutrino energy of 10^{17} eV. (bottom) For a neutrino energy of 10^{18} eV. Different markers and colors correspond to different trigger thresholds. All events with a signal of at least a 3σ in any of the pulses were considered which explains the smaller efficiency at the surface for the 'both pulses $> 4\sigma$ ' criterion

9.1 Simulation strategy

We consider a simplified detector with two components. The first one is a surface oriented component consisting of LPDAs and dipoles. To save computing time, we only simulate two orthogonally-oriented horizontal LPDAs at 2 m depth and one dipole at 5 m depth to be sensitive to all signal polarizations. The second component is a deep one, approximated with a single dipole antenna at 50 m depth. We combine the four antennas into a single station so that only one simulation needs to be run, but we can still evaluate the coincidence fraction independently.

In principle, one would need to simulate a full 2D grid for every station separation distance that one wanted to test, because there might be cases where not the nearest station triggered but the next-to nearest neighboring station or stations even further out. However, as this will drastically increase computing time (which scales linearly with the number of stations) this small second order effect is ignored in this example. Our analysis will show that the coincidence rate is dominated by the nearest neighbors, i.e., the coincidence rate quickly drops if the separation between stations is doubled, justifying this approximation.

For every station separation distance we consider the eight nearest stations around the central station as illustrated in Fig. 15 on the left. We consider distances ranging from 100 m to 3 km.

We run the NuRadioMC simulation for event lists of different neutrino energies. The Askaryan signal is filtered from 80 MHz - 500 MHz and all events are saved that exceed a signal threshold of $1V_{\text{RMS}}$ for a noise temperature of 300 K.

9.2 Accessing the results and coincidence fraction

Part of the HDF5 output file is the maximum amplitude of each channel of each event stored in a two dimensional array. This allows for a quick calculation of the coincidence requirements. We first check if the central station fulfilled the trigger condition which we assume to be a signal above $3V_{\text{RMS}}$ in any channel. Then, for each simulated distance, we select the channels corresponding to this distance and check if any channel fulfills the trigger condition. The coincidence rate is then given by the ratio of events where both the central station and any of its nearest neighbors triggered, divided by the number of triggers of the central station alone. The result is presented in Fig. 15 (right). It shows that the coincidence fraction increases with energy. At a station distance of 1 km more than 20% of the events at 10^{18} eV for a surface station (and more than 40% for a 50 m deep station) are detected in at least two stations. This suggests that for a design optimizing on effective volume, stations should be separated further than 1 km from each other, or even further when optimizing for the highest energies. An array of surface stations shows in general a smaller coincidence fraction.

10 Summary and outlook

We have presented NuRadioMC as a versatile framework to simulate different aspects of radio neutrino detectors. NuRadioMC provides a state-of-the-art implementation of the four pillars of a radio neutrino simulation: *event generation*, *signal generation*, *signal propagation*, and *detector simulation*. All properties of the simulation chain can be adapted and compared to each other. Following the design goals of flex-

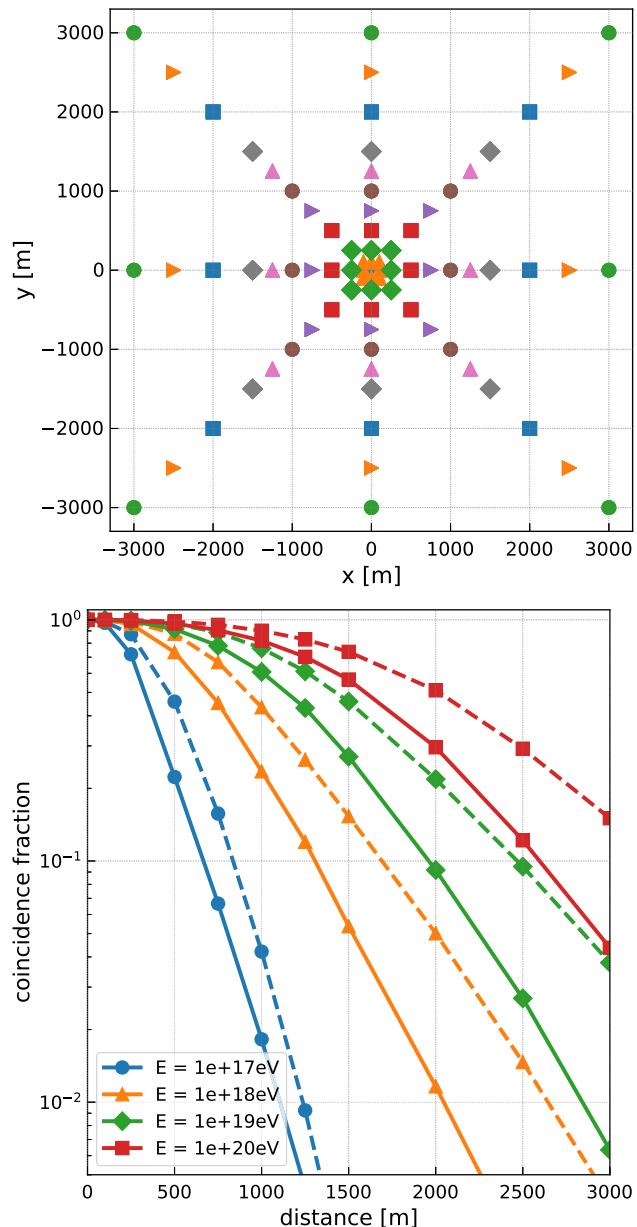


Fig. 15 (top) Station layout of example 3 to determine the station coincidence rate. Each color and symbol combination corresponds to the nearest neighbors of one station separation distance. (bottom) The probability to detect the same neutrino in multiple stations as a function of separation between the stations. The different colors/symbols correspond to different neutrino energies. The solid line is the result for a surface detector, the dashed line is the result for a 50 m deep detector

ibility and usability, NuRadioMC combines the knowledge and experience from all previous radio detectors for neutrino and cosmic-rays. We have presented a detailed discussion of many radio emission models and documented an improved time-domain approach using a shower library which provides a realistic treatment of the LPM effect and its random fluctuations. In three comprehensive examples, we have shown how to calculate effective volumes and sensitivities,

the efficiency to detect multiple pulses from the same shower (multi-path events), and the coincidence fraction between stations in a large array, depending on the distance between stations. This provides valuable tools for design decisions, depending on the goals one wants to optimize for. Proposed radio neutrino experiments such as RNO, ARIANNA, GRAND, ANITA/PUEO or BEACON [9, 95, 99, 100] may soon or already have profited from the capabilities of NuRadioMC.

NuRadioMC provides a solid foundation for reliable simulations, but also leaves room for future developments from the radio neutrino community. NuRadioMC is publicly available on github [101] and is open to low-threshold further code development from interested parties. As experiments progress and as soon as neutrinos are detected through their radio emission, the areas of prioritized need for development will be indicated by the data.

Acknowledgements This article and NuRadioMC itself would not exist without the constructive spirit of the InIceMC working group of the ARA and ARIANNA collaborations. We acknowledge funding from the German research foundation (DFG) under grant GL 914/1-1 (CG) and grant NE 2031/2-1 (DGF, AN, IP, and CW). JAM is supported by Ministerio de Economía, Industria y Competitividad (FPA 2017-85114-P), Xunta de Galicia (ED431C 2017/07), Feder Funds, RENATA Red Nacional Temática de Astropartículas (FPA2015-68783-REDT) and María de Maeztu Unit of Excellence (MDM-2016-0692). We are grateful to the U.S. National Science Foundation-Office of Polar Programs, the U.S. National Science Foundation-Physics Division (grant NSF-1607719) and the U.S. Department of Energy (SWB and CP). BAC thanks the National Science Foundation for support through the Graduate Research Fellowship Program Award DGE-1343012. AC acknowledges funding from the NSF CAREER award 28820 and NSF award 49285. We acknowledge Belgian Funds for Scientific Research (FRS-FNRS and FWO) (ST and NvE), the FWO programme for International Research Infrastructure, and funds from the ERC-StG (No. 805486) of the European Research Council (KDdV).

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The source code and example data files are available open source on github: <https://github.com/nuradio/NuRadioMC>.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funded by SCOAP³.

Appendix A: HDF5 event files structure

The HDF5 files created by the event generator consist of a collection of arrays containing the properties of the neutrinos and other secondary particles (taus, for instance). The array keys and contents are the following:

- *azimuths*, the arrival azimuth angles in radians.
- *zeniths*, the arrival zenith angles in radians.
- *xx*, *yy*, and *zz*, the *x*, *y* and *z* coordinates in meters for the point where the particles interact or decay.
- *event_ids*, the event identification numbers
- *n_interaction*, the interaction number. 1 indicates a neutrino interaction, 2 and greater indicates decay or interaction of a lepton created after the neutrino interaction.
- *flavors*, neutrino flavors. 12 for electron neutrino, 14 for muon neutrino, and 16 for tau neutrino. Antineutrinos are represented by –12, –14, and –16. A value of 15 indicates a tau lepton. The numbers are following the standard of [30].
- *energies*, the particle energies in electronvolts
- *interaction_type*, the interaction type. 'cc' for charged current, and 'nc' for neutral current. 'tau_had', 'tau_em', 'tau_mu' indicate the tau decays into the hadronic, electromagnetic and muonic channels respectively.
- *inelasticities*, the inelasticities for the neutrino interactions and the tau decays, that is, the energy fractions taken by the product cascades.

In these HDF5 files we also save as attributes the number of events and the characteristics of the fiducial and total simulated volumes, along with maximum and minimum energies and angles for the neutrinos.

Appendix B: NuRadioMC HDF5 output files structure

NuRadioMC creates as output an HDF5 file with information on the events and on the simulation outcome. The user can choose between saving all the information for all events or only for those that have triggered. The NuRadioMC HDF5 output files contain all the values that can be found in the event files (Appendix A), along with the following additional arrays:

- *triggered*, with ones indicating a triggering event and zeroes a non-triggering event.
- *weights*, the weights given to each event as a consequence of propagation through the Earth.
- *multiple_triggers*, indicates if the triggering condition has been met individually for each simulated trigger. The first axis of this array gives the event number, and the second the type of trigger.

The rest of the output arrays are stored in several HDF5 groups, each group corresponding to a simulated station. The following arrays (except for the SNRs array) contained within the station group are multidimensional. Their first axis is the event number, and the second one the antenna. Each group for a given station contains:

- *SNRs*, the signal to noise ratios for each event defined as the maximum signal amplitude divided by the RMS noise.
- *triggered*, with ones indicating a triggering station and zeroes a non-triggering station.
- *multiple_triggers*, indicates if the triggering condition has been met individually for each simulated trigger. The first axis of this array gives the event number, and the second the type of trigger.
- *maximum_amplitudes*, the maximum amplitudes for the voltages of each antenna.
- *maximum_amplitudes_envelope*, the maximum amplitudes of the voltage envelope of each antenna.
- *travel_distances*, the distances traveled by the rays. There can be up to two, one for each ray-tracing solution. The third axis of the array indicates the ray-tracing solution. The same principle applies to all arrays containing ray-tracing information.
- *travel_times*, the times taken by the rays from emitter to observer.
- *ray_tracing_C0*, C_0 parameters for the ray tracing solutions.
- *ray_tracing_C1*, C_1 parameters for the ray tracing solutions.
- *ray_tracing_solution_type*, strings containing the type of ray tracing solutions: direct, reflected, or refracted.

The following arrays of the HDF5 group contain three-dimensional vectors, and therefore they have a fourth axis that allows us to find the x , y , and z components of said vectors.

- *launch_vectors*, the launch vectors for the ray tracing solutions.
- *receive_vectors*, the receive vectors for the ray tracing solutions.
- *polarization*, the polarization of the electric field.

In the attributes of the output files the names of the simulated triggers (using the string *trigger_names*) can be found.

Appendix C: Analytic ray tracing

The analytic ray tracing in NuRadioMC provides a novel and fast solution of the ray-tracing problem. For completeness we

provide the full derivation of the analytic solution, the path, the path length and the travel time.

Appendix C.1: Derivation of analytic solution

In this section, we will derive the analytic solution to the ray tracing problem. Fermat's principle states that the optical path of a ray of light travelling between two points is stationary. Suppose the index of refraction depends on one coordinate in a three-dimensional Cartesian coordinate system:

$$n(x, y, z) = n(z) \quad (C.1)$$

Further, let $dx/dz = \dot{x}$ and $dy/dz = \dot{y}$, so that the metric may be expressed as:

$$ds = \sqrt{dx^2 + dy^2 + dz^2} = dz\sqrt{\dot{x}^2 + \dot{y}^2 + 1} \quad (C.2)$$

The symmetry of $n(z)$ implies that the coordinate system may be rotated such that $\dot{x} = 0$. Thus the metric becomes

$$ds = dz\sqrt{\dot{y}^2 + 1} \quad (C.3)$$

Inserting this metric into Fermat's Principle gives

$$S = \int_A^B n ds \quad (C.4)$$

$$\delta S = 0 \quad (C.5)$$

$$\delta \int_A^B n(z)\sqrt{1 + \dot{y}^2} dz = 0 \quad (C.6)$$

Defining $u = \dot{y}$ and applying the Euler-Lagrange equations yields

$$\dot{u} = -\frac{\dot{n}}{n}(u^3 + u) \quad (C.7)$$

Letting $v = -\ln n$, Eq. C.7 simplifies to

$$\dot{u} = \dot{v}(u^3 + u) \quad (C.8)$$

Noting that $\dot{v} = dv/dz$, and applying the chain rule gives

$$\frac{du}{dz} \frac{dz}{dv} = \frac{du}{dv} = u^3 + u \quad (C.9)$$

Rearranging and then integrating gives

$$\int \frac{du}{u^3 + u} = \int dv \quad (C.10)$$

$$\ln u - \frac{1}{2} \ln(u^2 + 1) = v + C_0 \quad (C.11)$$

Equation C.11 may be solved for dz/dy after re-scaling C_0 :

$$\frac{dz}{dy} = \pm \sqrt{C_0^2 n^2 - 1} \quad (\text{C.12})$$

In the case of South Pole and Moore's Bay glacial ice, it is found that $n(z)$ is described to within a few percent by an exponential function [68] which allows us to proceed further in solving for the ray-path.

$$n(z) = n_{ice} - \Delta_n \exp(z/z_0) \quad (\text{C.13})$$

Let $\gamma = \Delta_n \exp(z/z_0)$, which implies

$$n(z) = n_{ice} - \gamma \quad (\text{C.14})$$

$$dz = \gamma^{-1} z_0 d\gamma \quad (\text{C.15})$$

Inserting Eq. C.13 into Eq. C.12 and integrating, with $b = 2n_{ice}$ and $c = n_{ice}^2 - C_0^{-2}$:

$$\int \frac{d\gamma}{\gamma(\gamma^2 - b\gamma + c)^{1/2}} = \pm C_0 \left(\frac{\gamma}{z_0} + C_1 \right) \quad (\text{C.16})$$

The second integration constant is C_1 . Intriguingly, for depths much greater than the scale height ($|z_i| \gg z_0, z_i < 0$), the integral in Eq. C.16 has a singularity in the denominator when the ray is initially horizontal. This is discussed further below. The solution to Eq. C.16 is available in standard tables. The solution with y as a function of z via γ is:

$$y(z) = \pm C_0^{-1} c^{-1/2} z_0 \ln \left(\frac{\gamma}{2c^{1/2}(\gamma^2 - b\gamma + c)^{1/2} - b\gamma + 2c} \right) \mp z_0 C_1 \quad (\text{C.17})$$

Let the function within the logarithm in Eq. C.17 be $F(\gamma)$:

$$F(\gamma) = \frac{\gamma}{2c^{1/2}(\gamma^2 - b\gamma + c)^{1/2} - b\gamma + 2c} \quad (\text{C.18})$$

Inserting Eq. C.18 into Eq. C.17, we recover a function which returns the ray path as a function of depth:

$$y(z) = \pm C_0^{-1} c^{-1/2} z_0 \ln(F(\gamma)) \mp z_0 C_1 \quad (\text{C.19})$$

Because the ice model is horizontally symmetric, the constant C_1 is set by the choice of origin. All that remains is to understand the physical meaning of C_0 . Let the initial angle

with respect to the horizontal be θ_i , which should obey

$$\frac{dy}{dz} = \cot(\theta_i) \quad (\text{C.20})$$

$$\frac{dy}{d\gamma} = z_0 \gamma^{-1} \cot(\theta_i) \quad (\text{C.21})$$

Given Eq. C.19, Eq. C.21 may be solved in terms of $F(\gamma)$. The result is

$$\tan \theta_i = \pm C_0 c^{1/2} \frac{F(\gamma)}{\gamma F'(\gamma)} \quad (\text{C.22})$$

Inserting the definition of c and solving for C_0 :

$$C_0(\gamma, \theta_i) = \pm n_{ice}^{-1} \left(\frac{\gamma^2 F'^2(\gamma)}{F^2(\gamma)} \tan^2 \theta_i + 1 \right)^{1/2} \quad (\text{C.23})$$

The right-hand side of Eq. C.23 resembles a secant function. Restricting to initial depths much greater than the scale depth ($|z_i| \gg z_0, z_i < 0$) causes

$$\frac{\gamma^2 F'^2(\gamma)}{F^2(\gamma)} \rightarrow 1 \quad (\text{C.24})$$

If this limit is taken, then Eq. C.23 simplifies:

$$C_0(\gamma, \theta_i) = \pm n_{ice}^{-1} \left(\tan^2 \theta_i + 1 \right)^{1/2} = \pm n_{ice}^{-1} \sec \theta_i \quad (\text{C.25})$$

C_0 is a constant that depends on the boundary conditions, so Eq. C.25 may be inverted:

$$n_{ice} \cos \theta_i = \pm C_0^{-1} \quad (\text{C.26})$$

Equation C.26 is Snell's Law, because C_0 is constant and θ_i is defined with respect to the horizontal. Thus, in the limit ($|z_i| \gg z_0, z_i < 0$) the singularity in Eq. C.16 is for $\cos \theta_i = \pm 1$, i.e. horizontal propagation. Further, in the limit ($|z_i| \gg z_0, z_i < 0$) the factor in front of Eq. C.19, $C_0^{-1} c^{-1/2}$, simplifies:

$$c = n_{ice}^2 - C_0^{-2} \quad (\text{C.27})$$

$$c^{-1/2} = \left(n_{ice}^2 - C_0^{-2} \right)^{-1/2} \quad (\text{C.28})$$

$$C_0^{-1} c^{-1/2} = \left(C_0^2 n_{ice}^2 - 1 \right)^{-1/2} \quad (\text{C.29})$$

$$C_0^{-1} c^{-1/2} = \cot(\theta_i) \quad (\text{C.30})$$

In the last step, Eq. C.12 has been used. Thus, the closed form of $y(z)$ is

$$y(z) = \pm z_0 \cot \theta_i \ln(F(\gamma)) \quad (\text{C.31})$$

If the depth z does not satisfy the limit ($|z_i| \gg z_0, z_i < 0$), C_0 must first be obtained from Eq. C.23, and then inserted into Eq. C.19 to obtain the ray-tracing path.

Appendix C.2: Putting the analytic solution into practical usability

In this section, we demonstrate how to efficiently solve the analytic equations for the ray path derived in Appendix C.1. Without loss of generality, we can use only the positive solution which corresponds to rays propagating into the positive y direction. Equally, we can only consider rays in the $y - z$ plane. This is because such a start configuration can always be achieved with a simple coordinate transformation.

In addition, it is sufficient to only compute solution from a deeper to a shallower position without loss of generality by flipping the initial condition. Hence we can always reduce the problem to finding all possible path's between two points

$$\mathbf{x}_1 = (y_1, z_1)^T \quad \text{and} \quad \mathbf{x}_2 = (y_2, z_2)^T \\ \text{with } y_1 < y_2 \quad \text{and} \quad z_1 < z_2. \quad (\text{C.32})$$

The analytic solution only describes the “first part” of the solution until the *turning point*. This is the position where the ray either hits the surface and is reflected down, or it reaches the point where the propagation direction of the ray becomes horizontal (i.e. into the y direction) due to continuous refraction. This is of course a consequence of the solution being $y(z)$ and not $z(y)$ which is needed to describe the ray path in a single analytic function (because $z(y)$ is not bijective).

The turning point is the position where the second root of Eq. (10) becomes undefined, i.e., for

$$\gamma^2 - b\gamma + c = 0 \Rightarrow \gamma_{\text{turn}} = \frac{1}{2}b - \sqrt{\frac{b^2}{4} - c}. \quad (\text{C.33})$$

The z_{turn} position can be calculated from γ_{turn} . If z_{turn} is positive, the turning points is above the surface. Hence, the ray is reflected off the surface and z_{turn} is set to zero. Then, y_{turn} can be calculated by inserting z_{turn} into Eq. (10).

Hence, from an implementation perspective, we have two distinct cases: either we have a direct ray ($y_2 < y_{\text{turn}}$) or we have a reflected or refracted ray ($y_2 > y_{\text{turn}}$).

Appendix C.3: Determination of free parameters

Now, we present how to determine the two free parameters C_0 and C_1 in a fast and robust way from the initial condition that the ray path goes through the points \mathbf{x}_1 and \mathbf{x}_2 . The parameter C_1 is given by

$$C_1 = y_1 - y(z_1, C_0 = C'_0, C_1 = 0) \quad (\text{C.34})$$

with $y()$ being Eq. (10) evaluated for $C_0 = C'_0$ and $C_1 = 0$.

The parameter C_0 needs to be determined numerically by minimizing the following objective function:

$$\chi^2 = (y_2 - y'(z_2, C_0, C_1))^2. \quad (\text{C.35})$$

As Eq. (10) describes only half of the solution, we first check if \mathbf{x}_2 is before or after the turning point. It is after the turning point if $y_{\text{turn}} < y_2$. Then the following coordinate transformation is performed.

$$y'(z_2, C_0, C_1) = 2y_{\text{turn}} - y(z_2, C_0, C_1). \quad (\text{C.36})$$

To increase the numerical stability of the minimizer it is useful to perform the following coordinate transformation

$$D = \ln(C_0 - 1/n_{\text{ice}}). \quad (\text{C.37})$$

Then Eq. (10) is defined for all values of D .

For typical geometries not just one but two solutions are present. Once one solution is found, the second solution can be determined fast and efficiently using the Brent root finding algorithm [102], and using the displacement in y at position \mathbf{x}_2 as objective function (cf. Fig. 16 right). Utilization of Brent’s algorithm is possible because for a second solution to exists, Δy needs to change sign in one of the open intervals $(-\infty, C_0^1)$ and (C_0^1, ∞) , where C_0^1 is the first solution.

Appendix C.4: Derivative of analytic ray tracing path

The derivative of the analytic ray tracing solution is given by

$$\begin{aligned} & \frac{dy(z)}{dz} \\ &= \left(-\sqrt{c} e^{\frac{z}{z_0}} b \Delta_n + 2 \sqrt{-b \Delta_n e^{\frac{z}{z_0}} + \Delta_n^2 e^{2 \frac{z}{z_0}} + c} c + 2 c^{3/2} \right) \\ & \times \left(2 \sqrt{c} \sqrt{-b \Delta_n e^{\frac{z}{z_0}} + \Delta_n^2 e^{2 \frac{z}{z_0}} + c} - b \Delta_n e^{\frac{z}{z_0}} + 2 c \right)^{-1} \\ & \times \frac{1}{\sqrt{-b \Delta_n e^{\frac{z}{z_0}} + \Delta_n^2 e^{2 \frac{z}{z_0}} + c}} \frac{1}{\sqrt{C_0^2 n_{\text{ice}}^2 - 1}}. \end{aligned} \quad (\text{C.38})$$

Appendix C.5: Analytic solution of path length and travel time

In this section, the analytic solution of the path length and travel time for an exponential index-of-refraction profile is derived.

To find the path(s) between two given points in the ice, (r_0, z_0) and (r_1, z_1) , we need to find the launch angle(s) θ_0 of the ray(s). The radial coordinate r is equivalent to the y coordinate used in the previous sections, since we are restricted to the vertical plane where the wave propagates. Given the

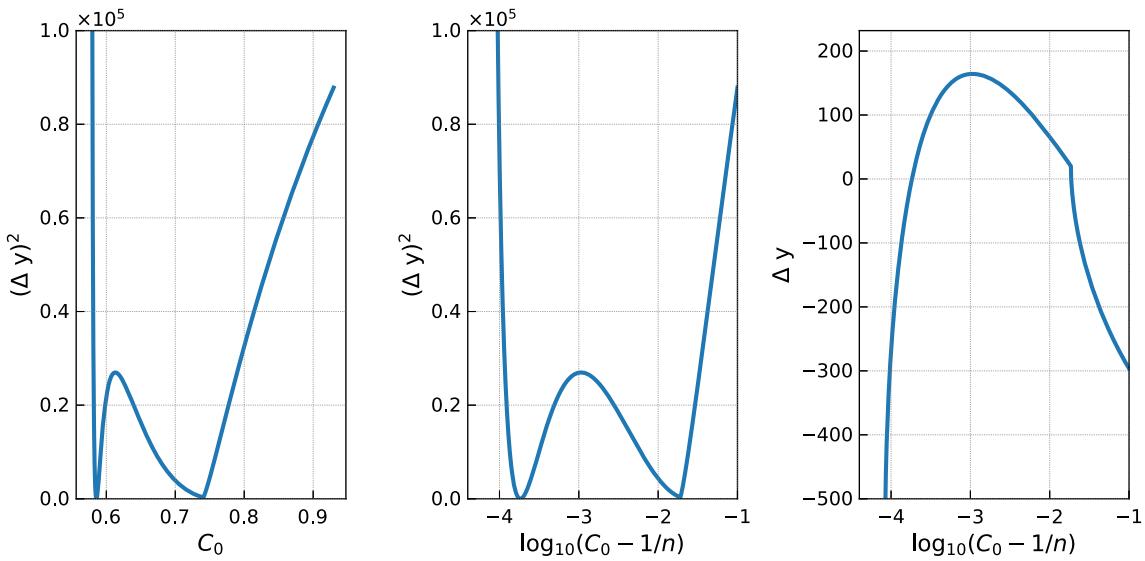


Fig. 16 Example of a typical objective function as a function of C_0 (left) and $\log_{10}(C_0 - 1/n)$ (center). Displacement in y as used for the determination of the second solution via the root finding algorithm (right)

launch angle θ_0 then we can find θ as a function of z using Snell's Law:

$$n(z) \sin(\theta(z)) = n(z_0) \sin(\theta_0) \quad (\text{C.39})$$

$$\theta(z) = \arcsin\left(\frac{n(z_0) \sin(\theta_0)}{n(z)}\right) \quad (\text{C.40})$$

Since we know the radial distance between our starting and ending points, we can calculate the launch angle by first working out the radial distance integral as a function of launch angle, and then inverting it.

$$\begin{aligned} \frac{dr}{dz} &= \frac{dr}{ds} \frac{ds}{dz} = \tan(\theta) \\ \int_{r_0}^{r_1} dr &= \int_{z_0}^{z_1} \tan(\theta) dz \end{aligned}$$

And then using Eq. C.40, this becomes

$$r_1 - r_0 = \int_{z_0}^{z_1} \tan\left(\arcsin\left(\frac{n(z_0) \sin(\theta_0)}{n(z)}\right)\right) dz \quad (\text{C.41})$$

To calculate the launch angle(s) for ray(s) between our two points, solve this equation for θ_0 . While we will continue solving this problem in generality for any $n(z)$ now, in a following section we will simplify the answer for a specific ice model.

Once we know the launch angle of our path we have all we need to calculate its properties. The total path length can be calculated by integrating $\frac{dz}{ds}$:

$$s = \int_{z_0}^{z_1} \frac{1}{\cos(\theta)} dz \quad (\text{C.42})$$

$$= \int_{z_0}^{z_1} \sec\left(\arcsin\left(\frac{n(z_0) \sin(\theta_0)}{n(z)}\right)\right) dz \quad (\text{C.43})$$

The time of flight t along the path can be calculated by combining $\frac{dz}{ds}$ with the following differential equation for the time of flight (where c is the speed of light):

$$\frac{dt}{ds} = \frac{n(z)}{c} \quad (\text{C.44})$$

Which then gives

$$\begin{aligned} \frac{dt}{dz} &= \frac{dt}{ds} \frac{ds}{dz} = \frac{n(z)}{c} \frac{1}{\cos(\theta)} \\ t &= \int_{z_0}^{z_1} \frac{n(z)}{c} \frac{1}{\cos(\theta)} dz \end{aligned} \quad (\text{C.45})$$

$$= \frac{1}{c} \int_{z_0}^{z_1} n(z) \sec\left(\arcsin\left(\frac{n(z_0) \sin(\theta_0)}{n(z)}\right)\right) dz \quad (\text{C.46})$$

For an exponential index-of-refraction profile of the form

$$n(z) = n_{\text{ice}} - \Delta_n e^{z/z_0} \quad (\text{C.47})$$

we can finish the calculations. We will use a few substitutions to make our equations clearer. The substitutions are as follows, where $n(z)$ is as above, z_0 is the starting depth, and θ_0 is the launch angle:

$$\begin{aligned}\beta &= n(z_0) \sin(\theta_0) \\ \alpha &= n_{\text{ice}}^2 - \beta^2 \\ \gamma &= n(z)^2 - \beta^2 \\ \ell_1 &= n_{\text{ice}} n(z) - \beta^2 - \sqrt{\alpha \gamma} \\ \ell_2 &= n(z) + \sqrt{\gamma}\end{aligned}\quad (\text{C.48})$$

Plugging in our ice model, the radial distance integral in equation C.41 becomes

$$r_1 - r_0 = \frac{\beta}{\sqrt{\alpha}} (-z + z_0 \log(\ell_1)) \Big|_{z_0}^{z_1} \quad (\text{C.49})$$

after equation C.48's substitutions. Solving this equation for the launch angle is an alternative approach to find the ray tracing path. Unfortunately, since the launch angle appears in so many places (α , β , and ℓ_1), this equation is not invertible and so cannot be directly solved for θ_0 . As a result, root-finding algorithms will need to be used to calculate the launch angle(s) for the ray(s) between (r_0, z_0) and (r_1, z_1) . In the NuRadioMC code, we calculate the ray paths using the approach of Sect. Appendix C.2 and just calculate the launch angle from the parameter C_0 of the analytic ray-tracing path.

Plugging in our ice model and substituting according to equation C.48, the path length (equation C.42) becomes

$$s = \frac{n_{\text{ice}}}{\sqrt{\alpha}} (-z + z_0 \log(\ell_1)) + z_0 \log(\ell_2) \Big|_{z_0}^{z_1} \quad (\text{C.50})$$

By the same process, the time of flight (equation C.45) becomes

$$t = \frac{1}{c} \left(z_0 \left(\sqrt{\gamma} + n_{\text{ice}} \log(\ell_2) + \log(\ell_1) \frac{n_{\text{ice}}^2}{\sqrt{\alpha}} \right) - z \frac{n_{\text{ice}}^2}{\sqrt{\alpha}} \right) \Big|_{z_0}^{z_1} \quad (\text{C.51})$$

Note that these integrals are specifically for a direct path. For an indirect path, the bounds must be changed to reflect the fact that the path goes up to z_{turn} before coming back down to z_1 .

Appendix C.6: Derivation of focusing correction

Here, we derive how ray density per unit area changes. The geometry in case of straight line propagation is depicted in Fig. 17. We read off that $a = R \sin \Delta\theta$. In the limit of $\Delta\theta \ll 1$ we get $a = R \Delta\theta$. The relation between the length a and vertical displacement Δz is given by $a = \sin \theta \Delta z$. Thus, we get

$$R = \frac{\Delta z}{\Delta\theta} \sin \theta \quad (\text{C.52})$$

and in the limit $\Delta z \Rightarrow 0$

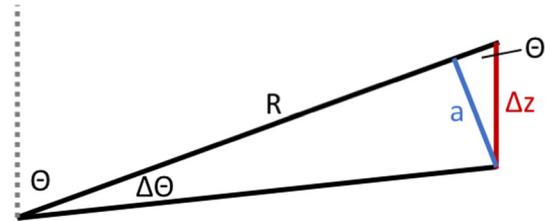


Fig. 17 Sketch of geometry for focusing correction

$$R = \frac{dz}{d\theta} \sin \theta . \quad (\text{C.53})$$

The area dA perpendicular to a ray is given by

$$dA = R d\theta \times R \sin \theta d\phi , \quad (\text{C.54})$$

and will change due to ray bending to

$$dA = \frac{dz}{d\theta} \sin \theta d\theta \times R \sin \theta d\phi . \quad (\text{C.55})$$

Appendix D: FFT normalization in NuRadioMC

In NuRadioMC we use a real fast Fourier transform (rFFT) as it only deals with real valued signals in the time-domain. Furthermore, we assume that the number of samples in the time domain is even. Then, n_t bins (with real values) in the time domain correspond to $n_f = n_t/2 + 1$ bins (with complex values) in the frequency domain where the first bin is the zero frequency component. This is because we exploit the symmetry between negative and positive frequencies for real valued input and only compute the positive frequency components.

The rFFT is normalized such that Parseval's theorem holds without any additional normalization factor, i.e.,

$$\sum_{m=0}^{n_t-1} x_m^2 = \sum_{k=0}^{n_t/2} \tilde{X}_k^2 . \quad (\text{D.56})$$

where x_m are the time domain samples of the signal, and \tilde{X}_k are the frequency domain samples. In the case of electric fields, the dimensions of both x_m and \tilde{X}_k are voltage/length.

This means that the energy fluence, i.e., the time integral over the pulse amplitudes, calculated in the frequency domain and in the time domain give the same results which is a useful physical property. Then, the rFFT and inverse rFFT is defined as

$$\tilde{X}_k = \frac{\sqrt{2}}{\sqrt{N}} \times \sum_{m=0}^{n_t-1} x_m \exp \left(-2\pi i \frac{mk}{n_t} \right) , k = 0, \dots, n_t/2 \quad (\text{D.57})$$

and

$$x_m = \frac{1}{\sqrt{2}\sqrt{N}} \times 2 \sum_{k=0}^{n_t/2} \tilde{X}_k \exp\left(2\pi i \frac{mk}{n_t}\right), m = 0, \dots, n_t - 1 \quad (\text{D.58})$$

We added an additional factor of $\sqrt{2}$ with respect to the standard orthogonal normalization to compensate for the negative frequencies that we did not compute so that the Eq. (D.56) holds.

Appendix D.1: Relation to a continuous Fourier transform

In literature, one also finds the continuous Fourier transform with different conventions for the normalization. One typical choice is to define the Fourier transform as

$$\tilde{E}(v) = \int_{-\infty}^{\infty} dt \exp(i2\pi vt) E(t) \quad (\text{D.59})$$

and

$$E(t) = \int_{-\infty}^{\infty} dv \exp(-i2\pi vt) \tilde{E}(v). \quad (\text{D.60})$$

If the signal in the time domain has units V/m the units in the frequency domain become V/m/Hz. A common task is to transform a frequency-domain parameterization of the Askaryan signal into the time domain via a discrete Fourier transform. For the definition of Eq. (D.59), the corresponding discrete inverse transform is

$$x_m = \frac{1}{n_t} \times 2 \sum_{k=0}^{n_t/2} \tilde{X}_k / \Delta t \exp\left(2\pi i \frac{mk}{n_t}\right) \quad (\text{D.61})$$

$$= 2 \sum_{k=0}^{n_t/2} \tilde{X}_k \Delta f \exp\left(2\pi i \frac{mk}{n_t}\right) \quad (\text{D.62})$$

where we exploit the relation $\Delta t = 1/(n_t \Delta f)$ of a discrete Fourier transform. The additional factor of 2 was added because we only sum over the positive frequencies here. This factor of 2 is already part of real FFT packages such as `numpy.fft` and does not need to be taken into account by the user (see Sect. Appendix D.3 for details).

Appendix D.2: Adjustments to different normalizations

All publications of a frequency-domain parameterization of the Askaryan signal that is based on the *ZHS* model use an unusual normalization of the continuous Fourier transform

where an additional factor of 2 is added to the forward transform (Eq. D.59), and correspondingly a factor of 1/2 in the backward transform (Eq. D.60) (see e.g. [47]). Therefore, Eq. (D.62) needs an additional factor of 1/2 if a ZHS parameterization is used.

Appendix D.3: Implementation details

Most parts of the code use the `numpy` real fft routines. The default normalization has the direct transforms unscaled and the inverse transforms are scaled by $1/n_t$. Hence, a analytic parametrization of the amplitudes in the frequency domain $A(v)$ with units V/m/Hz can be transformed into the time domain via

```
import numpy as np
n = 2**12 # number of bins in time domain
dt = 0.5 * units.ns # bin width in time domain
ff = np.fft.rfftfreq(n, dt)
# get array of frequencies
trace = np.fft.irfft(A(ff) / dt)
```

If $A(v)$ is a parametrization from a ZHS paper, we get the correct time domain representation via

```
trace = 0.5 * np.fft.irfft(A(ff) / dt)
# additional factor of 2 due to
# ZHS Fourier transform normalization
```

All other Fourier transforms are normalized such that Eq. (D.56) is satisfied which is achieved with `numpy` via:

```
def time2freq(trace):
    """
    performs forward FFT with correct
    normalization that conserves the power
    """
    return np.fft.rfft(trace,
axis=-1, norm="ortho") * 2 ** 0.5
# an additional sqrt(2) is added because
# negative frequencies are omitted.
```

```
def freq2time(spectrum):
    """
    performs backward FFT with correct
    normalization that conserves the power
    """
    return np.fft.irfft(spectrum, axis=-1,
norm="ortho") / 2 ** 0.5
```

Appendix E: Detector simulation

The code snippet in List. 4 shows a typical detector simulation. With just a few lines of code, we can calculate the antenna response, downsample the time trace to the detec-

tor sampling rate, bandpass filter the signal and simulate a high/low trigger with a 2 out of 4 antennas coincidence logic.

```
def get_time_trace(energy, theta, N, dt, shower_type, n_index, R, model,
                   interp_factor=None, interp_factor2=None,
                   same_shower=False, **kwargs):
    """
    returns the Askaryan pulse in the time domain of the eTheta component

    We implement only the time-domain solution and obtain the frequency spectrum
    via FFT (with the standard normalization of NuRadioMC). This approach assures
    that the units are interpreted correctly. In the time domain, the amplitudes
    are well defined and not details about fourier transform normalizations needs
    to be known by the user.

    Parameters
    -----
    energy : float
        energy of the shower
    theta: float
        viewangle: angle between shower axis (neutrino direction) and the line
        of sight between interaction and detector
    N : int
        number of samples in the time domain
    dt: float
        time bin width, i.e. the inverse of the sampling rate
    shower_type: string (default "HAD")
        type of shower, either "HAD" (hadronic), "EM" (electromagnetic) or
        "TAU" (tau lepton induced), note that TAU showers
        are currently only implemented in the ARZ2019 model
    n_index: float
        index of refraction
    R: float
        distance from vertex to observer
    model: string
        specifies the signal model
        * ZHS1992: the original ZHS parametrization from E. Zas, ...
        * Alvarez2000: parameterization based on ZHS mainly based on J. Alvarez-...
        * Alvarez2009: parameterization based on ZHS from J. Alvarez-...
        * HCRB2017: analytic model from J. Hanson, A. Connolly ...
        * ARZ2019 semi MC time domain model
    interp_factor: float or None
        controls the interpolation of the charge-excess profiles in the ARZ model
    interp_Factor2: float or None
        controls the second interpolation of the charge-excess profiles in the ARZ model
    same_shower: bool (default False)
        controls the random behaviour of picking a shower from the library in the ARZ model,
        see description there for more details

    Returns
    -----
    time trace: array
        the amplitudes for each time bin
    """


```

Listing 3: Signature of the signal generation interface. NuRadioMC provides a uniform interface in form of simple function to all implemented Askaryan modules. This allows to use the Askaryan modules outside of a NuRadioMC simulation and is a well tested resource/reference implementation for the radio community.

```

class mySimulation(simulation.simulation):
    def _detector_simulation(self):
        # 1st convolve efield with antenna pattern
        efieldToVoltageConverterPerChannel.run(self._evt, self._station, self._det)
        # downsample trace back to detector sampling rate
        channelResampler.run(self._evt, self._station, self._det, sampling_rate=1. / self._dt)
        # bandpass filter the signal
        channelBandPassFilter.run(self._evt, self._station, self._det,
                                  passband=[80 * units.MHz, 500 * units.GHz],
                                  filter_type='butter', order=2)
        # run a high/low trigger on the 4 downward pointing LPDAs
        triggerSimulatorHighLow.run(self._evt, self._station, self._det,
                                     threshold_high=4 * self._Vrms,
                                     threshold_low=-4 * self._Vrms,
                                     coinc_window=40 * units.ns,
                                     triggered_channels=[0, 1, 2, 3], # select the LPDA channels
                                     number_cedicences=2, # 2/4 majority logic
                                     trigger_name='LPDA_2of4_4sigma')

```

Listing 4: Example of performing a detector simulation using NuRadioReco.

References

1. M. Ackermann et al., Astrophysics uniquely enabled by observations of high-energy cosmic neutrinos, astro 2020 decadal whitepaper (2019). [arXiv:1903.04334](https://arxiv.org/abs/1903.04334)
2. IceCube collaboration, The IceCube Neutrino Observatory: instrumentation and online systems. *J. Instrum.* **12**, P03012 (2017). [arXiv:1612.05093](https://arxiv.org/abs/1612.05093). <https://doi.org/10.1088/1748-0221/12/03/P03012>
3. IceCube collaboration, A combined maximum-likelihood analysis of the high-energy astrophysical neutrino flux measured with IceCube. *Astrophys. J.* **809**, 98 (2015). [arXiv:1507.03991](https://arxiv.org/abs/1507.03991). <https://doi.org/10.1088/0004-637X/809/1/98>
4. IceCube collaboration, Neutrino emission from the direction of blazar TXS 0506+056 prior to the IceCube-170922A alert. *Science* **361**, 147 (2018)
5. IceCube collaboration, Measurement of South Pole ice transparency with the IceCube LED calibration system. *Nucl. Instrum. Methods* **A711**, 73 (2013). [arXiv:1301.5361](https://arxiv.org/abs/1301.5361). <https://doi.org/10.1016/j.nima.2013.01.054>
6. G.A. Askar'yan, Coherent radio emission from cosmic showers in air and in dense media. *Sov. J. Exp. Theor. Phys.* **21**, 658 (1965)
7. S. Barwick, D. Besson, P. Gorham, D. Saltzberg, South polar in situ radio-frequency ice attenuation. *J. Glaciol.* **51**, 231–238 (2005). <https://doi.org/10.3189/172756505781829467>
8. ARIANNA collaboration, Design and performance of the ARIANNA HRA-3 neutrino detector systems. *IEEE Trans. Nucl. Sci.* **62**, 2202 (2015). [arXiv:1410.7369](https://arxiv.org/abs/1410.7369). <https://doi.org/10.1109/TNS.2015.2468182>
9. ARIANNA collaboration, Targeting ultra-high energy neutrinos with the ARIANNA experiment. *Adv. Space Res.* **64**(12), 2595–2609 (2019). [arXiv:1903.01609](https://arxiv.org/abs/1903.01609) <https://doi.org/10.1016/j.asr.2019.06.016>
10. ARA collaboration, Performance of two Askaryan Radio Array stations and first results in the search for ultrahigh energy neutrinos. *Phys. Rev. D* **93**, 082003 (2016). [arXiv:1507.08991](https://arxiv.org/abs/1507.08991). <https://doi.org/10.1103/PhysRevD.93.082003>
11. ARIANNA collaboration, A first search for cosmogenic neutrinos with the ARIANNA Hexagonal Radio Array. *Astropart. Phys.* **70**, 12 (2015). [arXiv:1410.7352](https://arxiv.org/abs/1410.7352). <https://doi.org/10.1016/j.astropartphys.2015.04.002>
12. P. Allison et al., Design and performance of an interferometric trigger array for radio detection of high-energy neutrinos. *Nucl. Instrum. Methods Phys. Res. A* **930**, 112 (2019). <https://doi.org/10.1016/j.nima.2019.01.067>. [arXiv:1809.04573](https://arxiv.org/abs/1809.04573)
13. K. Greisen, End to the cosmic-ray spectrum? *Phys. Rev. Lett.* **16**, 748 (1966). <https://doi.org/10.1103/PhysRevLett.16.748>
14. G.T. Zatsepin, V.A. Kuz'min, Upper limit of the spectrum of cosmic rays. *Sov. J. Exp. Theor. Phys. Lett.* **4**, 78 (1966)
15. V.S. Beresinsky, G.T. Zatsepin, Cosmic rays at ultra high energies (neutrino?). *Phys. Lett. B* **28**, 423 (1969). [https://doi.org/10.1016/0370-2693\(69\)90341-4](https://doi.org/10.1016/0370-2693(69)90341-4)
16. M. Ackermann et al., Fundamental physics with high-energy cosmic neutrinos, Astro 2020 Decadal Whitepaper (2019). [arXiv:1903.04333](https://arxiv.org/abs/1903.04333)
17. K. Dookayka, *Characterizing the Search for Ultra-High Energy Neutrinos with the ARIANNA Detector*; Ph.D. thesis (University of California, Irvine, 2011)
18. ARA collaboration, First constraints on the ultra-high energy neutrino flux from a prototype station of the Askaryan Radio Array. *Astropart. Phys.* **70**, 62 (2015). [arXiv:1404.5285](https://arxiv.org/abs/1404.5285). <https://doi.org/10.1016/j.astropartphys.2015.04.006>
19. ANITA collaboration, ICEMC: a Monte Carlo Simulation for Cosmogenic Neutrinos interacting in the Antarctic ice as viewed by the Antarctic Impulsive Transient Antenna (ANITA). [arXiv:1903.11043](https://arxiv.org/abs/1903.11043)
20. T. Winchen et al., Realtime processing of LOFAR data for the detection of nano-second pulses from the Moon, in *Journal of Physics Conference Series*, vol. 898, p. 032004 (2017). [arXiv:1612.06592](https://arxiv.org/abs/1612.06592). <https://doi.org/10.1088/1742-6596/898/3/032004>
21. C.W. James et al., Overview of lunar detection of ultra-high energy particles and new plans for the SKA, in *European Physical Journal Web of Conferences*, vol. 135, p. 04001 (2017). [arXiv:1704.05336](https://arxiv.org/abs/1704.05336). <https://doi.org/10.1051/epjconf/201713504001>
22. R. Essig et al., Dark sectors and new, light, weakly-coupled particles, FERMILAB-CONF-13-653 (2013). [arXiv:1311.0029](https://arxiv.org/abs/1311.0029)
23. “Cython.” <http://cython.org>
24. J. Alvarez-Muñiz, R.A. Vázquez, E. Zas, Calculation methods for radio pulses from high energy showers. *Phys. Rev. D* **62** (2000). <https://doi.org/10.1103/physrevd.62.063001>

25. J. Alvarez-Muñiz, A. Romero-Wolf, E. Zas, Practical and accurate calculations of Askaryan radiation. *Phys. Rev. D* **84**, 103003 (2011). [arXiv:1106.6283](https://arxiv.org/abs/1106.6283). <https://doi.org/10.1103/PhysRevD.84.103003>
26. L.D. Landau, I. Pomeranchuk, Limits of applicability of the theory of bremsstrahlung electrons and pair production at high-energies. *Dokl. Akad. Nauk Ser. Fiz.* **92**, 535 (1953)
27. A.B. Migdal, Bremsstrahlung and pair production in condensed media at high-energies. *Phys. Rev.* **103**, 1811 (1956). <https://doi.org/10.1103/PhysRev.103.1811>
28. C. Glaser, A. Nelles, I. Plaisier, C. Welling, S.W. Barwick, D. Garca-Fernndez, et al., NuRadioReco: A reconstruction framework for radio neutrino detectors. *Eur. Phys. J. C* **79** (2019). [arXiv:1903.07023](https://arxiv.org/abs/1903.07023). <https://doi.org/10.1140/epjc/s10052-019-6971-5>
29. The HDF Group, Hierarchical Data Format, version 5, 1997-NNNN
30. C. Amsler et al., (Particle Data Group), Monte Carlo particle numbering scheme. *Phys. Lett. B* **667** (2018)
31. R. Gandhi, C. Quigg, M.H. Reno, I. Sarcevic, Neutrino interactions at ultrahigh energies. *Phys. Rev. D* **58**, 58 (1998). <https://doi.org/10.1103/PhysRevD.58.093009>
32. R. Gandhi, C. Quigg, M.H. Reno, I. Sarcevic, Ultrahigh-energy neutrino interactions. *Astropart. Phys.* **5**, 81–110 (1996). [https://doi.org/10.1016/0927-6505\(96\)00008-4](https://doi.org/10.1016/0927-6505(96)00008-4)
33. A. Connolly, R.S. Thorne, D. Waters, Calculation of high energy neutrino-nucleon cross sections and uncertainties using the MSTW parton distribution functions and implications for future experiments. *Phys. Rev. D* **83**, 113009 (2011). [arXiv:1102.0691](https://arxiv.org/abs/1102.0691). <https://doi.org/10.1103/PhysRevD.83.113009>
34. A. Cooper-Sarkar, P. Mertsch, S. Sarkar, The high energy neutrino cross-section in the Standard Model and its uncertainty. *JHEP* **08**, 042 (2011). [arXiv:1106.3723](https://arxiv.org/abs/1106.3723). [https://doi.org/10.1007/JHEP08\(2011\)042](https://doi.org/10.1007/JHEP08(2011)042)
35. J.H. Koehne, K. Frantzen, M. Schmitz, T. Fuchs, W. Rhode, D. Chirkov et al., PROPOSAL: a tool for propagation of charged leptons. *Comput. Phys. Commun.* **184**, 2070 (2013). <https://doi.org/10.1016/j.cpc.2013.04.001>
36. M. Dunsch, J. Soedingrekso, A. Sandrock, M. Meier, T. Menne, W. Rhode, Recent improvements for the lepton propagator PROPOSAL. [arXiv:1809.07740](https://arxiv.org/abs/1809.07740)
37. Particle Data Group collaboration, Review of particle physics. *Phys. Rev. D* **98**, 030001 (2018). <https://doi.org/10.1103/PhysRevD.98.030001>
38. T. De Young, S. Razzaque, D.F. Cowen, Astrophysical tau neutrino detection in kilometer-scale Cherenkov detectors via muonic tau decay. *Astropart. Phys.* **27**, 238 (2007). [arXiv:astro-ph/0608486](https://arxiv.org/abs/astro-ph/0608486). <https://doi.org/10.1016/j.astropartphys.2006.11.003>
39. S. Iyer Dutta, M.H. Reno, I. Sarcevic, D. Seckel, Propagation of muons and taus at high energies. *Phys. Rev. D* **63**, 094020 (2001). <https://doi.org/10.1103/PhysRevD.63.094020>
40. NuRadioMC, “Example: Pulser calibration measurement.” <https://github.com/nu-radio/NuRadioMC/wiki/Example:-Pulser-calibration-measurement>
41. NuRadioMC, “Event generation skeleton.” https://github.com/nu-radio/NuRadioMC/blob/master/NuRadioMC/EvtGen/generator_skeleton.py
42. NuRadioMC, “Online documentation.” <https://github.com/nu-radio/NuRadioMC/wiki/Interface-to-Askaryan-modules>
43. C.R. Persichilli, *Performance and Simulation of the ARIANNA Pilot Array, with Implications for Future Ultra-high Energy Neutrino Astronomy*, Ph.D. thesis (University of California, Irvine, 2018)
44. E. Zas, F. Halzen, T. Stanev, Electromagnetic pulses from high-energy showers: implications for neutrino detection. *Phys. Rev. D* **45**, 362 (1992). <https://doi.org/10.1103/PhysRevD.45.362>
45. J. Alvarez-Muñiz, E. Marqués, R.A. Vázquez, E. Zas, Coherent radio pulses from showers in different media: a unified parameterization. *Phys. Rev. D* **74**, 023007 (2006). [arXiv:astro-ph/0512337](https://arxiv.org/abs/astro-ph/0512337). <https://doi.org/10.1103/PhysRevD.74.023007>
46. J. Alvarez-Muñiz, W.R. Carvalho, M. Tueros, E. Zas, Coherent Cherenkov radio pulses from hadronic showers up to EeV energies. *Astropart. Phys.* **35**, 287 (2012). [arXiv:1005.0552](https://arxiv.org/abs/1005.0552). <https://doi.org/10.1016/j.astropartphys.2011.10.002>
47. J. Alvarez-Muñiz, C. James, R. Protheroe, E. Zas, Thinned simulations of extremely energetic showers in dense media for radio applications. *Astropart. Phys.* **32**, 100 (2009). <https://doi.org/10.1016/j.astropartphys.2009.06.005>
48. J. Alvarez-Muñiz, E. Zas, The LPM effect for EeV hadronic showers in ice: implications for radio detection of neutrinos. *Phys. Lett. B* **434**, 396 (1998). [https://doi.org/10.1016/s0370-2693\(98\)00905-8](https://doi.org/10.1016/s0370-2693(98)00905-8)
49. J. Alvarez-Muñiz, E. Zas, Cherenkov radio pulses from EeV neutrino interactions: the LPM effect. *Phys. Lett. B* **411**, 218 (1997). [https://doi.org/10.1016/s0370-2693\(97\)01009-5](https://doi.org/10.1016/s0370-2693(97)01009-5)
50. J.C. Hanson, A.L. Connolly, Complex analysis of askaryan radiation: a fully analytic treatment including the LPM effect and cascade form factor. *Astropart. Phys.* **91**, 75 (2017). [arXiv:1605.04975](https://arxiv.org/abs/1605.04975). <https://doi.org/10.1016/j.astropartphys.2017.03.008>
51. R.V. Buniy, J.P. Ralston, Radio detection of high energy particles: coherence versus multiple scales. *Phys. Rev. D* **65**, 016003 (2001). <https://doi.org/10.1103/PhysRevD.65.016003>
52. C.-Y. Hu, C.-C. Chen, P. Chen, Near-field effects of Cherenkov radiation induced by ultra high energy cosmic neutrinos. *Astropart. Phys.* **35**, 421 (2012). [arXiv:1012.5155](https://arxiv.org/abs/1012.5155). <https://doi.org/10.1016/j.astropartphys.2011.11.008>
53. L. Gerhardt, S.R. Klein, Electron and photon interactions in the regime of strong LPM suppression. *Phys. Rev. D* **82**, 074017 (2010). [arXiv:1007.0039](https://arxiv.org/abs/1007.0039). <https://doi.org/10.1103/PhysRevD.82.074017>
54. K. Werner, K.D. de Vries, O. Scholten, A realistic treatment of geomagnetic Cherenkov radiation from cosmic ray air showers. *Astropart. Phys.* **37**, 5 (2012). [arXiv:1201.4471](https://arxiv.org/abs/1201.4471). <https://doi.org/10.1016/j.astropartphys.2012.07.007>
55. J. Alvarez-Muñiz, A. Romero-Wolf, E. Zas, Čerenkov radio pulses from electromagnetic showers in the time domain. *Phys. Rev. D* **81**, 123009 (2010). [arXiv:1002.3873](https://arxiv.org/abs/1002.3873). <https://doi.org/10.1103/PhysRevD.81.123009>
56. J. Bellm et al., Herwig 7.1 Release Note. [arXiv:1705.06919](https://arxiv.org/abs/1705.06919)
57. J. Alvarez-Muñiz, W.R. Carvalho, Jr., E. Zas, Monte Carlo simulations of radio pulses in atmospheric showers using ZHAireS. *Astropart. Phys.* **35**, 325 (2012). [arXiv:1107.1189](https://arxiv.org/abs/1107.1189). <https://doi.org/10.1016/j.astropartphys.2011.10.005>
58. M. Gottowik, C. Glaser, T. Huege, J. Rautenberg, Determination of the absolute energy scale of extensive air showers via radio emission: systematic uncertainty of underlying first-principle calculations. *Astropart. Phys.* **103**, 87 (2018). [arXiv:1712.07442](https://arxiv.org/abs/1712.07442). <https://doi.org/10.1016/j.astropartphys.2018.07.004>
59. D. García-Fernández, J. Alvarez-Muñiz, W.R. Carvalho, A. Romero-Wolf, E. Zas, Calculations of electric fields for radio detection of ultrahigh energy particles. *Phys. Rev. D* **87**, 023003 (2013). <https://doi.org/10.1103/PhysRevD.87.023003>
60. T. Huege, M. Ludwig, C.W. James, Simulating radio emission from air showers with CoREAS, in American Institute of Physics Conference Series, R. Lahmann, T. Eberl, K. Graf, C. James, T. Huege, T. Karg et al. (eds.) American Institute of Physics Conference Series, vol. 1535, pp. 128–132 (2013). [arXiv:1301.2132](https://arxiv.org/abs/1301.2132). <https://doi.org/10.1063/1.4807534>
61. Pierre Auger collaboration, Measurement of the radiation energy in the radio signal of extensive air showers as a universal estimator of cosmic-ray energy. *Phys. Rev. Lett.* **116**, 241101

- (2016). arXiv:1605.02564. <https://doi.org/10.1103/PhysRevLett.116.241101>
62. C. Glaser, M. Erdmann, J.R. Hörandel, T. Huege, J. Schulz, Simulation of radiation energy release in air showers. *JCAP* **1609**, 024 (2016). arXiv:1606.01641. <https://doi.org/10.1088/1475-7516/2016/09/024>
 63. LOFAR collaboration, Detecting cosmic rays with the LOFAR radio telescope. *Astron. Astrophys.* **560**, A98 (2013). arXiv:1311.1399. <https://doi.org/10.1051/0004-6361/201322683>
 64. R. Engel, D. Heck, T. Huege, T. Pierog, M. Reininghaus, F. Riehn et al., Towards a next generation of CORSIKA: a framework for the simulation of particle cascades in astroparticle physics. *Comput. Softw. Big Sci.* **3**, 2 (2019). arXiv:1808.08226. <https://doi.org/10.1007/s41781-018-0013-0>
 65. NuRadioMC, “Signal propagation base class.” https://github.com/nu-radio/NuRadioMC/blob/master/NuRadioMC/SignalProp/propagation_base_class.py
 66. S. Buitink, Using FDTD simulations to study radio propagation effects, Proceedings of the ARENA 2018, Catania Sicily (2018)
 67. C. Deaconu, A.G. Vieregg, S.A. Wissel, J. Bowen, S. Chipman, A. Gupta et al., Measurements and modeling of near-surface radio propagation in glacial ice and implications for neutrino experiments. *Phys. Rev. D* **98**, 043010 (2018). arXiv:1805.12576. <https://doi.org/10.1103/PhysRevD.98.043010>
 68. S. Barwick et al., Observation of classically ‘forbidden’ electromagnetic wave propagation and implications for neutrino detection. *J. Cosmol. Astropart. Phys.* **07**, 055 (2018). arXiv:1804.10430. <https://doi.org/10.1088/1475-7516/2018/07/055>
 69. S. van der Walt, S.C. Colbert, G. Varoquaux, The numpy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* **13**, 22 (2011). <https://doi.org/10.1109/MCSE.2011.37>
 70. E. Jones, T. Oliphant, P. Peterson et al., “SciPy: Open source scientific tools for Python.” <http://www.scipy.org/> (2001)
 71. M. Galassi et al., “GNU Scientific Library Reference Manual.” <http://www.gnu.org/software/gsl/%22>
 72. R.A. Batista, A. Dundovic, M. Erdmann, K.-H. Kampert, D. Kuempel, G. Müller et al., CRPropa 3—a public astrophysical simulation framework for propagating extraterrestrial ultrahigh energy particles. *JCAP* **1605**, 038 (2016). arXiv:1603.07142. <https://doi.org/10.1088/1475-7516/2016/05/038>
 73. T. Winchen, RadioPropa—a modular raytracer for in-matter radio propagation, in *Proceedings of the ARENA 2018*, Catania Sicily (2018). arXiv:1810.01780
 74. F. Frezza, N. Tedeschi, Electromagnetic inhomogeneous waves at planar boundaries: tutorial. *J. Opt. Soc. Am. A* **32**, 1485 (2015). <https://doi.org/10.1364/JOSAA.32.001485>
 75. D. García-Fernández, B. Revenu, A. Escudie, L. Martin, Influence of a planar boundary on the electric field emitted by a particle shower. *Phys. Rev. D* **99**, 063009 (2019). arXiv:1811.11003. <https://doi.org/10.1103/PhysRevD.99.063009>
 76. B. Kolundzija, WIPL-D: from university software to company product, in *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*, pp. 2844–2846 (2011)
 77. K.S. Kunz, R.J. Luebers, *The Finite Difference Time Domain Method for Electromagnetics* (CRC Press, Boca Raton, 1993)
 78. Pierre Auger collaboration, Advanced functionality for radio analysis in the Offline software framework of the Pierre Auger Observatory. *Nucl. Instrum. Methods Phys. Res. A* **635**, 92 (2011). arXiv:1101.4473. <https://doi.org/10.1016/j.nima.2011.01.049>
 79. S.A. Kleinfelder, E. Chiem, T. Prakash, The SST Multi-G-Sample/s Switched Capacitor Array Waveform Recorder with Flexible Trigger and Picosecond-Level Timing Accuracy. arXiv:1508.02460
 80. J. Alvarez-Muñiz, W. R. Carvalho, K. Payet, A. Romero-Wolf, H. Schoorlemmer, E. Zas, Comprehensive approach to tau-lepton production by high-energy tau neutrinos propagating through the earth. *Phys. Rev. D* **97** (2018). <https://doi.org/10.1103/physrevd.97.023021>
 81. J. Alvarez-Muñiz, W.R. Carvalho, A.L. Cummings, K. Payet, A. Romero-Wolf, H. Schoorlemmer et al., Erratum: Comprehensive approach to tau-lepton production by high-energy tau neutrinos propagating through the earth [Phys. Rev. D 97, 023021 (2018)]. *Phys. Rev. D* **99** (2019). <https://doi.org/10.1103/physrevd.99.069902>
 82. S. Argirò et al., The offline software framework of the Pierre Auger Observatory. *Nucl. Instrum. Methods Phys. Res. A* **580**, 1485 (2007). <https://doi.org/10.1016/j.nima.2007.07.010>. arXiv:0707.1652
 83. ARA Project, “Attenuation model.” <https://icecube.wisc.edu/araproject/radio/#icetabsorption>
 84. R.L. Hawley, E.M. Morris, J.R. McConnell, Rapid techniques for determining annual accumulation applied at Summit. Greenland. *J. Glaciol.* **54**, 839 (2008). <https://doi.org/10.3189/00221430878779951>
 85. R. Alley, B. Koci, Ice-Core analysis at site A, Greenland: preliminary results. *Ann. Glaciol.* **10**, 1–4 (1988). <https://doi.org/10.3189/S026030550004067>
 86. NuRadioMC, “Tutorial.” <https://github.com/nu-radio/NuRadioMC/wiki/Tutorial>
 87. NuRadioMC, “Cluster documentation.” <https://github.com/nu-radio/NuRadioMC/wiki/Running-on-a-cluster>
 88. IceCube collaboration, A measurement of the diffuse astrophysical muon neutrino flux using eight years of IceCube data. *PoS ICRC2017*, 1005 (2018). arXiv:1710.01191. <https://doi.org/10.22323/1.301.1005>
 89. IceCube collaboration, Observation of astrophysical neutrinos in six years of IceCube data, *PoS ICRC2017*, 981 (2018). arXiv:1710.01191. <https://doi.org/10.22323/1.301.0981>
 90. IceCube collaboration, Differential limit on the extremely-high-energy cosmic neutrino flux in the presence of astrophysical background from nine years of IceCube data. *Phys. Rev. D* **98**, 062003 (2018). arXiv:1807.01820. <https://doi.org/10.1103/PhysRevD.98.062003>
 91. Pierre Auger collaboration, Improved limit to the diffuse flux of ultrahigh energy neutrinos from the Pierre Auger Observatory. *Phys. Rev. D* **91**, 092008 (2015). arXiv:1504.05397. <https://doi.org/10.1103/PhysRevD.91.092008>
 92. ANITA collaboration, Constraints on the diffuse high-energy neutrino flux from the third flight of ANITA. *Phys. Rev. D* **98**, 022001 (2018). arXiv:1803.02719. <https://doi.org/10.1103/PhysRevD.98.022001>
 93. J. Heinze, A. Fedynitch, D. Boncioli, W. Winter, A new view on Auger data and cosmogenic neutrinos in light of different nuclear disintegration and air-shower models. arXiv:1901.03338
 94. A. van Vliet, R. Alves Batista, J.R. Hörandel, Determining the fraction of cosmic-ray protons at ultra-high energies with cosmogenic neutrinos. arXiv:1901.01899
 95. GRAND collaboration, The Giant Radio Array for Neutrino Detection (GRAND): science and design. arXiv:1810.09994
 96. ARA collaboration, Measurement of the real dielectric permittivity ϵ_r of glacial ice. *Astropart. Phys.* **108**, 63 (2019). <https://doi.org/10.1016/j.astropartphys.2019.01.004>
 97. NuRadioMC, “Example: D’n’r analysis.” <https://github.com/nu-radio/NuRadioMC/wiki/Example:-D'n'R-analysis>
 98. NuRadioMC, “Example: Multi station coincidences.” <https://github.com/nu-radio/NuRadioMC/wiki/Example:-Multi-station-coincidences>

99. A. Karle, A. Nelles, The Radio Neutrino Observatory, in *Proceedings of the XVIII International Workshop on Neutrino Telescopes*, Venice (2019)
100. S. Wissel, *A new concept for high-elevation radio detection of tau neutrinos*, in *Proceedings of the ARENA 2018* (Catania, Sicily, 2018)
101. “NuRadioMC.” <https://github.com/nu-radio/NuRadioMC>
102. R.P. Brent, *An algorithm with guaranteed convergence for finding a zero of a function*, in *Algorithms for Minimization without Derivatives* (Prentice-Hall, Englewood Cliffs, 1973)