PROCEEDINGS OF SCIENCE

Integrating radio detectors of cosmic-ray air showers into the open-source NuRadio framework

Sjoerd Bouma,[a,*] Mitja Desmet,[b] Christian Glaser,[c] Philipp Laub,[a] Lilly Pyras,[d] Anna Nelles,[a,e,*] Felix Schlüter[f] and Karen Terveer[a]

[a]*Erlangen Centre for Astroparticle Physics, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany*

[b]*Vrije Universiteit Brussel, Astrophysical Institute, Pleinlaan 2, 1050 Brussels, Belgium*

[c]*Dept. of Physics and Astronomy, Uppsala University, Box 516, SE-75120 Uppsala, Sweden*

[d]*Department of Physics and Astronomy, University of Utah, Salt Lake City, UT 84112, USA*

[e]*Deutsches Elektronen-Synchrotron DESY, Platanenallee 6, 15738 Zeuthen, Germany*

[f]*Université Libre de Bruxelles, Science Faculty CP230, B-1050 Brussels, Belgium*

E-mail: sjoerd.bouma@fau.de, anna.nelles@desy.de

NuRadio is an open-source, Python-based software package for the simulation, analysis and reconstruction of the radio emission from ultra-high-energy (UHE) neutrinos and cosmic rays. While NuRadio has so far mainly been used for in-ice radio neutrino detectors, such as ARIANNA, RNO-G and the future IceCube-Gen2 radio array, its modularity, provision of standard data processing steps for radio detectors, extensive documentation, and continuous integration system have allowed the LOFAR and SKA experiments to readily adopt NuRadio for the analysis of cosmic-ray air showers.

This contribution will provide a brief overview of NuRadio, covering its new features and improvements to performance and usability in the past several years. The main focus will be on the application to the reconstruction of UHE cosmic-ray air showers, including both radio emission as well as particle data. We argue that using an open, collaborative framework benefits the entire radio community by reducing the software development overhead involved in duplicating, maintaining, or refactoring code, while the open review and continuous integration processes help to ensure accuracy and reliability. We therefore invite other cosmic-ray air-shower experiments to use and contribute to NuRadio.

39th International Cosmic Ray Conference (ICRC2025)
15–24 July 2025
Geneva, Switzerland

ICRC 2025
The Astroparticle Physics Conference
Geneva July 15-24, 2025

*Speaker

PoS(ICRC2025)345

1. Introduction

NuRadio is an open-source, modular Python framework consisting of two packages: NuRadioMC [1], for the Monte-Carlo simulation of radio emission and propagation of ultra-high energy particle cascades in ice, and NuRadioReco [2], for the reconstruction and detector simulation of high-energy neutrinos and cosmic rays.

Originally, NuRadio was designed for in-ice radio detectors. However, there is significant overlap between the analysis tools required for in-ice and land-based radio detectors of ultra-high energy particle cascades. Thanks to NuRadio's modular design, it is straightforward to only use those features that are relevant for the detector or analysis at hand. This has for example allowed the Low-Frequency Array (LOFAR) experiment to port their analysis pipeline for cosmic-ray air showers to NuRadioReco, while benefiting from the established open-source NuRadio framework, such as a self-consistent unit system, coordinate system transformations, and time-dependent detector descriptions. In addition, NuRadio is an actively developed open-source framework that uses continuous integration combined with manual code reviews for new contributions. This ensures both new and existing code run and produce correct output, which help to make analyses more reliable, and reduce the time spent debugging. Finally, NuRadio features an extensive online documentation, as well as many interactive examples, that lower the barrier of entry for new users and help to allow even projects at bachelor or masters level to result in permanent contributions, rather than private software projects which become unusable once their original developer becomes unavailable.

This contribution is structured as follows: first, in section 2, we give a brief overview of the data structure and design philosophy of NuRadio. In section 3, we outline some of the features in NuRadio for the simulation and reconstruction of the radio signals from cosmic-ray air showers. Section 4 details the current status and outlook for the integration of particle detectors into NuRadioReco. Finally, some of the main recent improvements for in-ice neutrino simulations are described in section 5.

2. NuRadio structure

Data in NuRadio is structured according to hierarchical Python classes, the structure of which is illustrated in Figure 1. Each cosmic-ray (or neutrino) event corresponds to a single *Event*, which contains one or more detector *Station*s. Each *Station* in turn contains several *Channel*s, corresponding to the voltage traces at the data acquisition level (generally speaking, each *Channel* corresponds to a single antenna). Each *Event* may also contain one or more *Shower* objects that store the (reconstructed) properties of the particle shower. *Events* can be stored to and read from disk in a custom binary format.

The second important data format in NuRadio is the *Detector*. Detectors in NuRadio contain a description of the detector geometry, antennas and signal chain. Because the full, complex antenna response as a function of signal direction, and signal chain S-parameters can be large, the detector description itself usually only contains a reference to the relevant antenna or amplifier, with the full complex response stored separately; for most experiments that use NuRadio, these are stored on a public server at DESY, and downloaded on demand by NuRadio. The *Detector* object contains

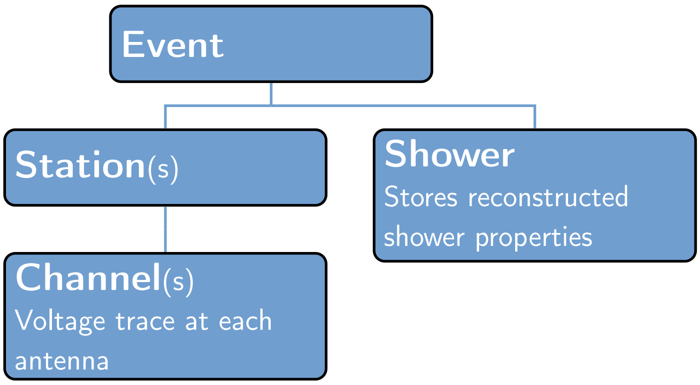

Figure 1: Sketch of the main components of the NuRadio *Event* data class. For simplicity, some additional optional components of the *Event* class that may or may not be present depending on the dataset are not shown here.

a time-dependent description of the detector, to allow for changes in the detector configuration, and can be stored either together with the *Event(s)* or independently as a json-formatted text file. The reason why the *Detector* in NuRadio is to some extent independent from the *Event* is that the knowledge of the detector geometry and signal chain response may also improve over time, for example through dedicated in-situ calibration campaigns, whereas the raw data (stored in the *Event*) does not.

One advantage of this custom data structure is that each class is aware of what type of data it contains, and has dedicated methods for the most frequently used operations; for example, one can immediately obtain either the time domain or frequency domain representation of the voltage trace of a *Channel*, without having to worry about different units or FFT normalization conventions. A second advantage is related to the modular design philosophy of NuRadio. Almost all signal processing and reconstruction modules in NuRadio have the same structure: they have a `run()` method that expects a NuRadio *Event* and a *Detector* as input. These already contain most or all of the information that is needed for the relevant module to run, and also allows for any output to be stored automatically in the *Event*. Therefore, only two steps are required in order for a new radio experiment to use NuRadio: write a reader module that converts the raw experiment data to the NuRadio *Event* structure, and provide a description of the detector. After this, one can immediately use any of the signal processing or reconstruction tools that NuRadio has to offer.

3. Radio detection of air showers

Whereas for the simulation of neutrino-induced particle cascades, NuRadioMC uses a fast, semi-analytic parameterization, for the radio emission of cosmic-ray air showers generally a full Monte-Carlo approach is used. The standard framework to simulate these is CORSIKA [3], with the CoREAS [4] extension to simulate the radio emission of an air shower. NuRadio has been able to read in the HDF5 output files from CoREAS since its inception, converting it to the NuRadio data structure, so that it is compatible with all of the signal processing and reconstruction tools that NuRadio provides.

There have been several improvements on the (reading in of) cosmic-ray air shower simulations. Although the particle-level simulations produced by CORSIKA/CoREAS are very accurate, they are also very computationally expensive. In principle, one would have to simulate an 'observer' at each antenna position, for each air shower. This becomes rapidly intractable — one would have to generate separate simulations for each shift in core position (the point where the shower impacts the surface). The computational cost for future detectors such as the Square Kilometre Array (SKAO), which will feature orders of magnitude more antennas than current radio detectors of cosmic-ray air showers, would be similarly problematic. Therefore, in practice, instead one simulates a generic 'star-shape' pattern for each air shower, computing the radio emission only for observers at regular radial and azimuthal intervals relative to the core position, and then uses some form of interpolation to obtain the emission at the actual antenna positions. NuRadio uses the *cr-pulse-interpolator* package[1], which performs a highly accurate interpolation in the Fourier domain. This method has been shown to significantly outperform previous linear methods [5], resulting in accurate interpolation not only of fluences but also of pulse shape and timing, which are critical for interferometry.

Another improvement in the simulation of the radio emission of cosmic-ray air showers is the so-called *template-synthesis* method [6], which uses detailed CoREAS simulations to generate additional air-showers with different shower parameters (Xmax, energy, direction) at a fraction of the cost of a full Monte-Carlo simulation, while maintaining an accuracy of a few percent. A Python package implementing the template synthesis method, *SMIET*[2], has been released recently, and will receive an interface with NuRadio in a future update.

In terms of data analysis and reconstruction, the largest development has been the adoption of NuRadio by the LOFAR experiment. In order to make their data analysis easier to maintain and, also looking forward to the near-future LOFAR 2.0 upgrade, easier to expand upon, it was decided to move the entire analysis pipeline to NuRadio. A read-in module that converts the raw LOFAR radio data to the NuRadio *Event* structure, and a number of noise removal, calibration and direction reconstruction modules has also been integrated into NuRadio. A description of the Dutch core LOFAR stations is also included with NuRadio, as is a model of the antenna response of the low-band antennas (LBAs). Thus, one can already read in, clean up, and reconstruct the cosmic-ray air shower from raw LOFAR data in just a couple of lines with NuRadio. Looking to the future, work is ongoing on adding more advanced reconstructions for cosmic-ray air showers, e.g. using information-field theory (IFT) [7], as well as enabling first studies for the SKA detector.

4. Particle detection of air showers

Radio detectors for cosmic-ray air showers frequently include an array of particle detectors to trigger the readout of the radio antennas. This is the case also for LOFAR, which uses the LOFAR Radboud Air shower array (LORA) array of plastic scintillator particle detectors to trigger and perform an initial reconstruction of air showers.

In an upcoming release, also the data analysis and reconstruction of particle data from LORA will be included in NuRadio. In order to facilitate this, the NuRadio data structure has been extended

[1] https://pypi.org/project/cr-pulse-interpolator/

[2] https://pypi.org/project/smiet/

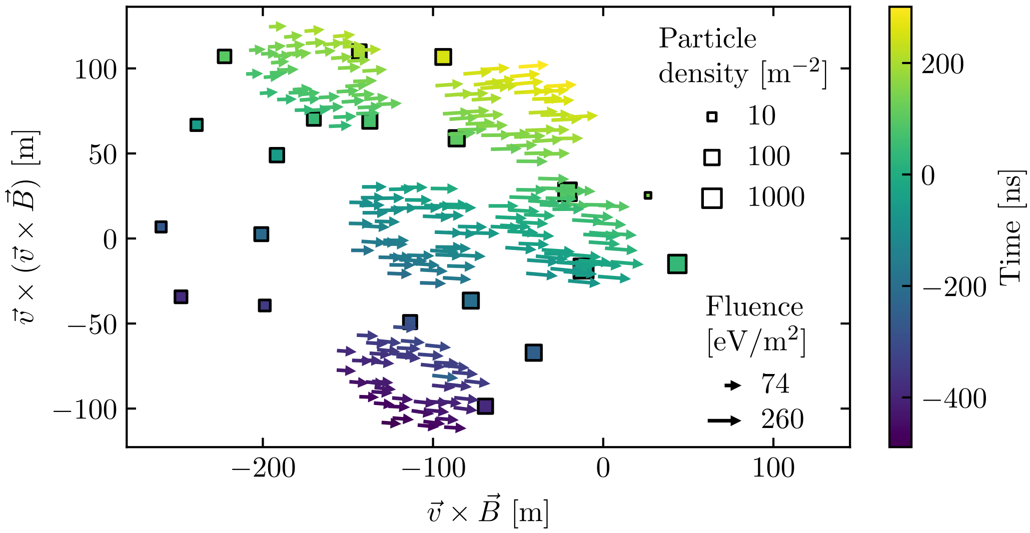

Figure 2: Example reconstruction of a single cosmic-ray air shower detected by LOFAR. Shown are both the particle detectors (squares) and the radio antennas (arrows). The direction of the arrows indicates the signal polarization, and the length is proportional to the electromagnetic fluence.

by including *ParticleStation* and *ParticleChannel* counterparts to the *Station* and *Channel* classes. Integrating the particle detector analysis in NuRadio has the advantage of improved stability and maintainability compared to the previous pipeline, which consisted of a combination of ROOT and Python code that proved non-trivial to run on newer systems. More importantly, however, having access to the radio and particle data within the same data structure will allow for analyses that exploit the information provided by both, such as e.g. combined fits, as well as providing more options for debugging. An example of such a combined analysis of a single LOFAR event, showing both the LORA particle detectors as well as the radio antennas, is shown in Figure 2.

5. Updates to in-ice neutrino use cases

In addition to the support for particle detectors, and improved features for radio detectors of cosmic-ray air showers, NuRadio has also seen several updates focussed on in-ice neutrino detectors. A more exhaustive summary of the changes since previous publications can be found in the changelog on the NuRadioMC GitHub[3]; here, we only highlight some of the most significant improvements.

As of version 3.0, it is possible to simulate birefringence in in-ice neutrino simulations. Polar ice is known to be birefringent due to a depth-dependent preferred alignment of the ice crystals caused by the ice flow and overburden pressure. This implies that the different polarization components of a radio signal propagate at different speeds, and may additionally interfere. This affects both the arrival times as well as the pulse shapes detected from the neutrino-induced radio emission by an in-ice detector. This is expected to impact especially the reconstruction of the original neutrino properties from the detected radio pulses.

[3] https://github.com/nu-radio/NuRadioMC

The implementation of the radio signal propagation in birefringent media is based on the model presented in Ref. [8], and ice models are available both for the ice at Summit Station, Greenland (the site of RNO-G) and at South Pole (the site of the proposed IceCube-Gen2 Radio array). Whereas the refractive index at Summit Station is close to isotropic, due to the limited ice flow at the top of the Greenland glacier, at South Pole a significant amount of birefringence is expected.

An additional upgrade is the improved modelling of the neutrino cross section and inelasticity, which now uses the state-of-the-art BGR18 calculations published in Ref.[9] by default. This improves the accuracy of the predicted in-ice particle cascade energies for a given simulated neutrino flux, resulting in an $O(10\%)$ change in effective volumes at the highest energies ($E_\nu \gtrsim 10^{18}$ eV).

Moreover, the core simulation routine of NuRadioMC has been rewritten in a more modular way, with benefits to simulation speed, the possibility to (re)simulate existing simulations, and exposing several functions for use in reconstruction routines. The simulation of both 'generic' (e.g. electronic) and galactic noise has also been updated, with improvements to speed, flexibility (in terms of simulating different noise spectra), and (for galactic noise) the treatment of coherence between difference antennas. Finally, there have been countless smaller additions, improvements and bug fixes, a more detailed account of which can be found on the NuRadioMC GitHub repository.

6. Summary

With a number of recent additions and improvements, NuRadio has become a suitable tool not only for in-ice neutrino detectors, but also for land-based radio and particle detectors of cosmic-ray air showers. Indeed, it has already been adopted by the LOFAR and SKA cosmic-ray groups, both for the analysis of existing data as well as simulations of the performance of future detectors. The extensive documentation, many examples, as well as the modular and Python-based nature of NuRadio help to make it easier for new users to adopt it — one can simply run `pip install nuradiomc` to immediately get started with any of the included examples. The continuous-integration testing, combined with manual code reviews, ensure that the simulation, data processing and reconstruction tools that are part of NuRadio remain accurate and reliable. Finally, the open-source, collaborative development efforts on GitHub make it possible for a large variety of users to use and contribute to NuRadio. This helps to reduce the duplication of code development efforts and to make new analyses or tools developed by both junior and senior researchers available for future users, which benefits the entire radio cosmic-ray community.

Both the in-ice and land-based radio detector features of NuRadio are continually improved by a growing number of contributors. We invite researchers working on all radio cosmic-ray air-shower experiments to join NuRadio.

Acknowledgments

We acknowledge the contribution by many members of the radio community as evident from the code repository of NuRadioMC, accessible via https://github.com/nu-radio/NuRadioMC. SB is supported by the collaborative research funding from the German Federal Ministry of Research, Technology, and Space through the LOFAR-ERIC project (05A23WEA).

References

[1] C. Glaser et al., *NuRadioMC: Simulating the radio emission of neutrinos from interaction to detector*, *Eur. Phys. J. C* **80** (2020) 77 [1906.01670].

[2] C. Glaser et al., *NuRadioReco: A reconstruction framework for radio neutrino detectors*, *Eur. Phys. J. C* **79** (2019) 464 [1903.07023].

[3] D. Heck, J. Knapp, J.N. Capdevielle, G. Schatz and T. Thouw, *CORSIKA: A Monte Carlo code to simulate extensive air showers*, *FZKA-6019* (1998) .

[4] T. Huege, M. Ludwig and C.W. James, *Simulating radio emission from air showers with CoREAS*, *AIP Conf. Proc.* **1535** (2013) 128 [1301.2132].

[5] A. Corstanje et al., *A high-precision interpolation method for pulsed radio signals from cosmic-ray air showers*, *JINST* **18** (2023) P09005 [2306.13514].

[6] M. Desmet, K. Watanabe, T. Huege and S. Buitink, *SMIET: Fast and accurate synthesis of radio pulses from extensive air shower using simulated templates*, 2505.10459.

[7] K. Terveer et al., *Reconstructing air showers using probabilistic forward modelling for LOFAR*, *PoS* **ICRC2025** (these proceedings) 413.

[8] N. Heyer and C. Glaser, *First-principle calculation of birefringence effects for in-ice radio detection of neutrinos*, *Eur. Phys. J. C* **83** (2023) 124 [2205.06169].

[9] A. Garcia, R. Gauld, A. Heijboer and J. Rojo, *Complete predictions for high-energy neutrino propagation in matter*, *JCAP* **09** (2020) 025 [2004.04756].

PoS(ICRC2025)345