

A Geant4 framework for generic simulations of atmospheric muon detection experiments

Nicola Mori^{1,4,*}, Fabio Ambrosino^{2,3}, Lorenzo Bonechi¹, Luigi Cimmino^{2,3},
Raffaello D'Alessandro^{1,4}, Pasquale Noli³, Giulio Saracino^{2,3},
Paolo Strolin^{2,3}, Lorenzo Viliani^{1,4}

¹ Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Firenze, Sesto Fiorentino, Florence, Italy

² Università di Napoli Federico II, Dipartimento di Fisica, Naples, Italy

³ Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Napoli, Naples, Italy

⁴ Università di Firenze, Dipartimento di Fisica e Astronomia, Sesto Fiorentino, Florence, Italy

Article history

Received July 5, 2016; accepted October 5, 2016.

Subject classification:

Atmospheric muons, Simulations, Muon radiography.

ABSTRACT

A flexible and adaptive simulation framework based on Geant4 is presented. The framework has been developed in order to speed up the deployment of full-featured Monte Carlo simulation codes for small/medium sized particle detection experiments with generic geometries. Specific components related to atmospheric muon experiments have been developed and are presented as well.

1. Introduction

The detection of atmospheric muons for investigating the density structure of target bodies is a growing field with many real-world applications. Modern fields of investigation relying on this technique ranges from geology [Tanaka et al. 2005, Ambrosi et al. 2011] to archaeology [Menichelli et al. 2007]. Novel possibilities opened up recently also for monitoring legacy nuclear waste storage sites [Clarkson et al. 2014], nuclear power plants [Morris et al. 2014] and nuclear contraband [Gnanvo et al. 2008].

The different sizes of the target structure in the various scenarios, as well as the conditions of the environment where the measurement is performed (e.g. difficult access to the site), influence the design of the detectors, resulting in a wide variety of target-detector cases.

Monte Carlo simulations are of the utmost importance for atmospheric muon detection experiments, in order to optimize the design of the detector, to estimate the level of the backgrounds and of the expected signals, and so on.

Currently, any simulation program for a specific experiment or measurement scenario must include the

details of the geometry of the target and of the detector, the detector response, an accurate description of the muon flux etc. These details must be implemented for the specific target structure and detector involved, so the software development effort has to be repeated for each measurement scenario.

However, a careful planning of the software architecture could allow to reuse those elements which are common to every atmospheric muon detection simulation, i.e. the generation of a realistic spectrum of muons. Also, some elements can be implemented in a way that is sufficiently general that they can be reused in all scenarios where the required accuracy is not very high: for example, a generic detector may be useful in an early stage where a coarse estimation of the expected performance with a given target structure has to be assessed.

In this paper a Monte Carlo simulation framework based on the Geant4 [Agostinelli et al. 2003] simulation toolkit is presented. The framework has been developed to enhance the speed of deployment of simulation software for atmospheric cosmic ray detection experiments by implementing all the common features (generation of muons, creation of the output files and so on) and providing a reference implementation of a detector to be used in early stages. A particular focus has been put on making the framework fully extensible to accommodate any experiment and/or scenario specific feature (e.g. the target structure, a more detailed description of the detector, a different muon generator), either as a side-by-side implementation or by actually replacing the reference ones. In the follow-

ing sections, the software architecture of the framework is described.

2. The Monte Carlo framework

Starting from the Geant4 toolkit, a full Monte Carlo simulation stack has been developed. The implementation is layered with different levels, from the most generic up to the most specific, in order to maximize code re-usage and development efficiency.

The foundation software layer

All the software stack is implemented on top of the Geant4 simulation toolkit. Geant4 is a generic (i.e. experiment-independent) package providing tools for simulating the propagation of particles inside matter. It requires the end user to write some mandatory C++ classes describing the detector geometry, handling particle generation and managing the output, in order to produce a full simulation program. While giving the opportunity to finely tune the simulation details, this also increases the time required to deploy the simulation program. In order to mitigate this issue, an implementation of the mandatory Geant4 classes, called GGS (generic Geant4 simulation), has been developed. GGS implements all the experiment-independent bits of the mandatory Geant4 classes in a generic and reusable way.

In particular, it automatically detects the sensitive volumes (i.e. those volumes where the energy released by the traversing particles has to be recorded on the output files) in the geometry and adapts the format of the output files to the sensitive volumes layout in the current geometry. The implementation leverages the I/O engine of the ROOT framework [Brun and Rademakers 1997] and the output is produced in the form of ROOT files.

While being experiment-independent, GGS has been designed to be fully customizable *via* plugin libraries. A plugin API allows the user to define the detector geometry and/or any particular particle generator, to add information to the output files and so on. All these eventual customizations can be implemented as C++ classes, and allow the introduction of experiment-specific features in the simulation while still taking advantage of the generic code paths implemented in GGS.

The atmospheric muon software layer

All the simulation details which are relevant for atmospheric muon detection have been implemented as GGS plugins. While the geometry of the target structure is a specific feature of the simulated scenario, and thus cannot be implemented in a generic way, a generic, reference muon detector can be defined. The chosen

implementation is a tracking device made of N pairs of X-Y tracking layers made of plastic scintillator, either segmented (in fibers or triangular bars; see Ambrosi et al. [2011]) or monolithic (in this case the muon impact point can be determined only by means of the Monte Carlo truth). The user can set any parameter of the detector (number and size of planes, plane spacing, number of fibers etc.) at runtime. An arbitrary number of these detectors can be placed with any orientation in the geometry describing the target volume, allowing for a quick geometry development. The plugin infrastructure allows also to replace the reference detector with a more specific one if needed.

Two different muon generators based on a theoretical model [Tanaka et al. 2001] and on ground measurements with a magnetic spectrometer [Bonechi et al. 2005] have been implemented. These generators cover different ranges in kinetic energy and zenith angle and generate realistic spectra for vertical and nearly-horizontal muons up to some TeV. Besides particle generation inside the simulation, the generators are also used by the flux normalization tool, a program which converts the number of generated events to the corresponding data acquisition time. The software architecture allows for an easy definition and integration of additional models of the muon flux.

The simulation runs can be very time consuming for large targets, since only a few muons hitting the target hit also the detector. Those events which do not hit the target might not be relevant, while those not hitting the detector would be actually undetected. Simulating and saving on disk these events increase the run time, the analysis time and the size of output files with no benefit whatsoever. Thus to increase efficiency while retaining maximum accuracy, an optional two-step acceptance check has been implemented. The underlying assumption is that the space surrounding the target is filled with a light material in which the scattering of the primary muon is negligible, so that a straight line constitutes a good approximation of the real trajectory. This is generally true since often the target structure is surrounded by air. When the initial kinematics of a muon is generated the framework checks if the extrapolated trajectory hits the target volume: if not the event is discarded and another kinematics is generated, otherwise the muon is propagated until it exits from the target. At this point the framework checks if the extrapolated trajectory hits the detector: if not the event is discarded, otherwise the simulation is carried on. The user can enlarge or restrict the acceptance criteria in order to define enlarged or fiducial acceptances. This mechanism minimizes the time spent to simulate useless (i.e. undetected) events

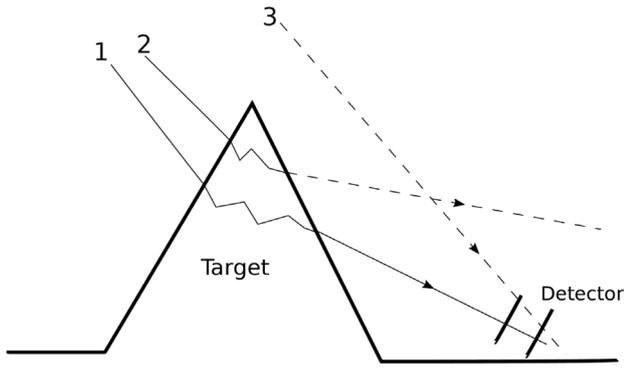


Figure 1. The acceptance checking algorithm: track 1 is propagated down to the detector, track 2 is killed at the exit of the target since its extrapolation does not hit the detector, track 3 is killed just after the generation of initial kinematics since its extrapolation does not hit the target.

without artificially reducing the background coming from those events whose extrapolated track at generation lies out of the acceptance of the detector but are deviated inside the acceptance by multiple scattering inside the target. A sketch illustrating the mechanism is shown in Figure 1.

The scenario-specific simulations

If no specific tuning is required (e.g. a particle generator implementing a different flux parameterization or a detailed detector) the simulation of a specific atmospheric muon detection experiment can be implemented simply by creating a C++ GGS geometry describing the target and then placing the reference muon detector(s) in the desired position(s) with the desired orientation(s). Every kind of target can be simulated quickly with a minimum effort in terms of code writing. In fact the entity of the effort may vary according to the complexity

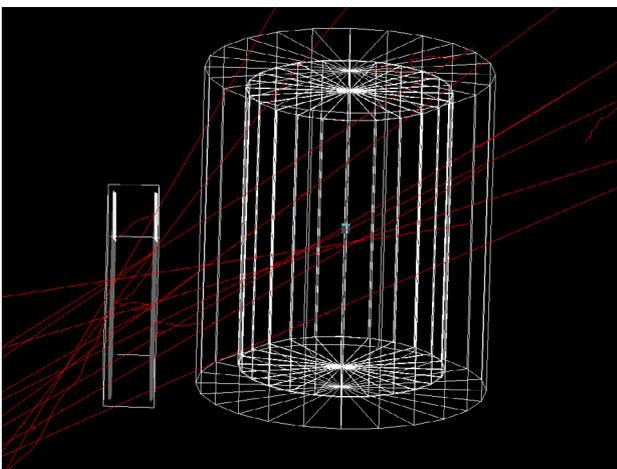


Figure 2. A simulation scenario implemented with the framework. The target is a cylindrical storage silo for nuclear waste. The detector is constituted by two X-Y pairs of layers with area $2 \times 2 \text{ m}^2$. The muons (red tracks) are generated from a horizontal surface placed above the silo.

of the target geometry: modeling the surface and the internals of a large geological structure like a volcano would require a considerable effort anyway. Yet this is the only thing that the user has to take care of, reducing the amount of the needed scenario-specific software development to a bare minimum.

The benefits of this approach are even more evident when simulating many different scenarios: besides the further enhanced reduction in development time, the user will also take advantage of having a common output format for all the different scenarios which will speed up also the production of data analysis codes. An example of a simulation scenario implemented with the framework is shown in Figure 2.

3. Discussion

The architecture of a software framework for Monte Carlo simulations aimed at atmospheric cosmic ray detection has been presented. Its design guidelines are geared towards code reuse and customizability, to speed up the simulation deployment leveraging standard implementations of common features while retaining the ability to finely tune the simulation by introducing experiment-specific bits. The framework is currently being used with success for simulating different muon detection scenarios (i.e. nuclear waste storage sites, archeological sites, geological structures), meeting all the design requirements. This framework is currently being used by the MURAVES collaboration for their Mount Vesuvius muography project, and the INFN-University of Glasgow collaboration for their evaluation of muography applied to the Sellfield nuclear stockage site.

References

- Agostinelli, S., et al. (2003). Geant4 - a simulation toolkit, Nucl. Instrum. Meth. A, 506, 250-303.
- Ambrosi, G., et al. (2011). The MU-RAY project: Volcano radiography with cosmic-ray muons, Nucl. Instrum. Meth. A, 628, 120-123.
- Bonechi, L., M. Bongi, D. Fedele, M. Grandi, S. Ricciarini and E. Vannuccini (2005). Development of the ADAMO detector: test with cosmic rays at different zenith angles, Proceedings of the 29th International Cosmic Ray Conference, 9, 283-286.
- Brun, R., and F. Rademakers (1997). ROOT - An Object Oriented Data Analysis Framework, Nucl. Instrum. Meth. A, 389, 81-86.
- Clarkson, A., et al. (2014). GEANT4 simulation of a scintillating-fibre tracker for the cosmic-ray muon tomography of legacy nuclear waste containers, Nucl. Instrum. Meth. A, 746, 64-73.
- Gnanvo, K., P. Ford, J. Helsby, R. Hoch, D. Mitra and

- M. Hohlmann (2008). Performance expectations for a Tomography System using cosmic ray muons and micro pattern gas detectors for the detection of nuclear contraband, *IEEE Nucl. Sci. Symp. Conf. Record*, 1278-1284.
- Menichelli, M., S. Ansoldi, M. Bari, M. Basset, R. Battiston, S. Blasko, F. Coren, E. Fiori, G. Giannini, D. Iugovaz, A. Papi, S. Reia and G. Scian (2007). A scintillating fibres tracker detector for archaeological applications, *Nucl. Instrum. Meth. A*, 572, 262-265.
- Morris, C.L., et al. (2014). Analysis of muon radiography of the Toshiba nuclear critical assembly reactor, *Appl. Phys. Lett.*, 104, 024110.
- Tanaka, H., K. Nagamine, N. Kawamura, S.N. Nakamura, K. Ishida and K. Shimomura (2001). Development of the Cosmic-Ray Muon Detection System for Probing Internal-Structure of a Volcano, *Hyperf. Int.*, 138 (1), 521-526.
- Tanaka, H.K.M., K. Nagamine, S.N. Nakamura and K. Ishida (2005). Radiographic measurements of the internal structure of Mt. West Iwate with near-horizontal cosmic-ray muons and future developments, *Nucl. Instrum. Meth. A*, 555, 164-172.

*Corresponding author: Nicola Mori,
Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Firenze,
Sesto Fiorentino, Florence, Italy; email: mori@fi.infn.it.