33RD INTERNATIONAL COSMIC RAY CONFERENCE, RIO DE JANEIRO 2013
THE ASTROPARTICLE PHYSICS CONFERENCE

Simulations of the MAGIC telescopes with matelsim

MARCOS LOPEZ[1] FOR THE MAGIC COLLABORATION.
[1] *Universidad Complutense de Madrid, E28040 Madrid, Spain*
marcos@gae.ucm.es

Abstract: Monte Carlo simulations are essential in the design of ground-based γ-ray observatories, as well as in the analysis of the data recorded by these facilities. With the main aim of performing simulations for the new MAGIC stereoscopic system, which consists of two 17 m diameter Cherenkov telescopes, we have developed a new simulation code named matelsim. This package is distributed together with the standard MAGIC analysis software, MARS [1]. matelsim is composed of independent modules, which simulate each of the different elements of the telescopes, such as the reflectors, the photosensor planes, the trigger logic or the signal digitization units. It also includes ray-tracing algorithms to simulate the reflection of photons by the individual telescope mirrors, and the attenuation of light due to Rayleigh and Mie scattering and Ozone absorption. The program has been designed to allow an easy customisation for testing different telescope configurations.

Keywords: simulation software, IACT, MAGIC

1 Introduction

The detection technique employed by Imaging Atmospheric Air Cherenkov Telescopes (IACTs) is based on the detection of the faint flashes of Cherenkov light produced when γ-rays (or cosmic-rays) plunge into the earth atmosphere and initiate extensive air showers of secondary particles (EAS). The Cherenkov light emitted by the charged secondary particles is reflected by the mirrors of the telescopes and an image of the shower is obtained in each telescope camera. An offline analysis of the shower images allows the rejection of the hadronic cosmic ray background, the measurement of the direction of the incoming γ-rays, and the estimation of their energy. This analysis is based on the comparison of image parameters with Monte Carlo simulations. Hence, IACTs rely entirely on detailed simulations for their calibration and data analysis.

Any simulation of an IACT involves two major stages. First, one has to simulate the development in the atmosphere of the showers of particles initiated by cosmic-rays and the corresponding emission of Cherenkov light. Then, one simulates the response of the telescopes to this input light.

In MAGIC, the simulation of EAS is done with a modified version [3] of the CORSIKA program [4, 5].

In this contribution we describe a new simulation program, termed matelsim, with deals with the simulation of the different elements of the Cherenkov telescopes: mirrors, photosensors, trigger and readout systems, and their response to the incoming Cherenkov light.

2 Matelsim

matelsim (MAgic TELescopes SIMulation) has been developed to simulate the recently upgraded MAGIC stereoscopic system. The code is object oriented and written in C++. It is distributed within MARS [1] and made use of the ROOT libraries [2] as its underlaying platform for I/O operations and graphics (e.g. histograms).

2.1 The need for a new simulation package

Simulations of the MAGIC telescopes has been done by two separate programs, namely reflector and camera [6, 7]. They were first developed to simulate the first MAGIC telescope, which began operating in 2004. At that time, the extension of MAGIC into a stereoscopic array was not foreseen, and hence the code was not designed for multi telescope simulations. In addition, the MAGIC telescopes have undergone major hardware upgrades in the past years. The old code lacked the flexibility and modularity to implement in an easy way all the changes required to simulate these hardware changes. Moreover, these programs were kept separate from the rest of the analysis code used in MAGIC, namely MARS [1].

This motivated the development of a new code, written from the the beginning with the following goals:

- To allow an easy maintenance and a full integration within the standard MAGIC analysis software.
- To allow simulations of arrays of any number of telescopes.
- To help in the design phase of new telescopes subsystems, as the new MAGIC analogue-sumtrigger.
- To allow the implementation of new simulation algorithms in a fast and robust way. Most of the code can be run directly from the the ROOT-Cint interface. This provides direct access to the implemented methods allowing an easy debugging of the different simulation steps.

This was achieved by a modular and multilayer design in which every element of the telescope (e.g. mirrors, photosensors, trigger cells, digitalisation chips...) is described by a base class containing all the required configuration parameters as well as the algorithms implementing its physical working. Specific components of the telescopes, as a given photomultiplier model, just inherit from the base class overriding the default settings. In this way, custom telescopes can be simulated by choosing its components from the class library and new hardware components can be added without changing the existing code.

2.2 Input/Output files

2.2.1 Configuration file

A configuration file is passed to the program to steer all the steps of the simulation. This file is implemented in such a way that it follows the modular structure of the program, aiming an easy use. There is only one mandatory keyword to set, which refers to the kind of observatory to simulate, e.g. magicMono, magicStereo, etc. Nevertheless, a high degree of customisation is possible. The user may specify the different 'building-blocks' with which they want to build up every telescope, and set every single detail of the simulation, like e.g. the pulse shape of a given channel of a given telescope.

2.2.2 Input photon files

matelsim can starts by reading two kind of input files: CORSIKA output files or output files produced by the old MAGIC reflector program [7]. In addition, matelsim can be run without input files if only calibration or pedestal files are to be simulated (see §2.3.7). In any case, both kind of input files are binary files containing, for each shower, the Cherenkov photons hitting a given area around the telescopes (see figure 1). Each photon consists of an array of 7 real numbers: coordinates of the emission point, director cosines, wavelength and arrival time at the observation level. In the case of reflector files, there is one additional parameter, storing the incident angle of the photon in the camera plane. The input file names can be passed from the input card or directly from the command line, for easy batch processing.

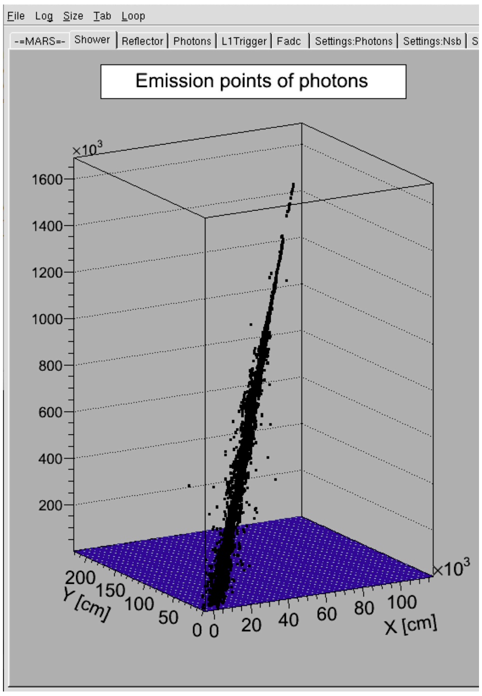

Figure 1: matelsim event display showing the emission points of the Chrenkov photons to be processed.

2.2.3 Output files

The goal of the Monte Carlo simulation is to produce files similar to the ones generated during normal data taking. In this way, they can be used with the same analysis tools to e.g. estimate the performance of the telescopes. For that, the simulated events produced by matelsim are saved in the output files using the same data structures, so-called *raw data*, as the one produced by the MAGIC telescopes. The only difference is that the output files of matelsim are already in ROOT format while the ones produced by the telescopes are binary files that are later translated into ROOT format at the first step of the analysis chain. Along with these *raw data*, additional information is written to the output files such as the energy and arrival direction of the simulated events, which are later used e.g. for training the γ/hadron separation methods.

2.3 Simulation steps

In what follows we describe the main steps or simulation modules implemented in matelsim.

2.3.1 Atmosphere simulation

A crucial part is the simulation of the atmosphere, as it is part of the detector for IACTs. Though CORSIKA includes atmospheric models needed for the simulation of EASs, it does not take into account the different attenuation processes that photons undergo as they travel in the atmosphere. This means that the output CORSIKA files contains all the Cherenkov photons produced by the shower particles. Hence, the first step of the simulation, when starting from CORSIKA files, is to simulate the ozone and aerosol absorption. The mean free path for a given photon is calculated by interpolation from pre-calculated tables, and it will depend on the photon wavelength and production height and on the model used for the atmosphere.

2.3.2 Reflector simulation and ray-tracking

Cherenkov telescopes have big tessellated reflectors made typically of spherical mirror tiles of different shapes and materials. The mirrors used in the MAGIC telescopes are square mirrors of two different kind (aluminium and glass) and of two different sizes (see figure 2).

The reflector is defined by its overall paraboloid focal length, and by a table containing the position and optical axis of each individual mirror. Imperfections in the focussing of the mirrors as well as the sagging due to the gravity are simulated by randomising the optical axis of each mirror.

For every Cherenkov photon, the ray-tracing algorithm starts by calculating the intersection of the photon trajectory with the overall reflector. Then, the mirror (if any) laying at that intersection point is found and the reflected trajectory is calculated taking into account the specific mirror tile parameters (as curvature radius and optical axis). The empty space between mirrors, as well as the whole at the reflector centre (used for placing calibration instruments) is taking into account. Finally, the coordinates of the impact point of the reflected photon's trajectory in the camera focal plane are obtained as well as the incident angle of the photon.

The optical ray-tracing module of matelsim can be used for determining the optical point-spread-function (PSF) of the reflector for different observation conditions, as well as for comparing the performance of different mirrors or optical systems (see figure 2).

2.3.3 Simulation of the photosensor's response and generation of the electronic signals

Once the photons are ray-traced to the camera plane, the program proceeds by finding which pixel is hit by the photon. Then the detection probability is calculated from the

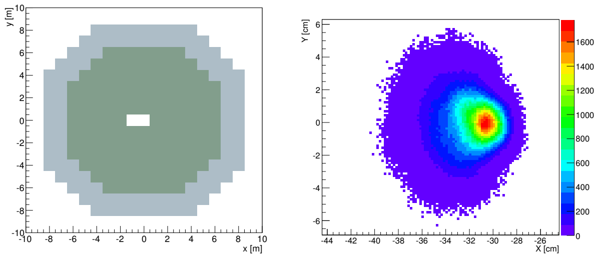

Figure 2: **Left**: Example of the implementation of a tessellated reflector with two mirror types, as used in the MAGIC-II telescope. **Right**: Simulation of the reflection of star at 1deg off-axis

pixel's specific collection efficiency (*CE*) which accounts for the camera light losses in the plexiglass protective front-plane and the reflexion in the light concentrators on top of the pixels (the so-called Winston cones). If the photon is not absorbed before hitting the photocathode, the probability of releasing a photo-electron (*ph.e.*) is evaluated from the quantum efficiency (*QE*) curves. The probability of detecting the photon will be then the product of $CE\, x\, QE$ which depends on the photons incident angle and wavelength (see figure 3). The simulation also includes fluctuations in the transit times of the *ph.e.* signals from pixel to pixel and from event to event, the so-called time jitter.

For every *ph.e.* released, detailed simulations of the electronic signals distributed to the trigger and digitalisation branches are carried out. For a given pixel, the intrinsic statistical behaviour of a photomultplier tube (PMT) is such that the output signals after the release of one single *ph.e.* varies from *ph.e.* to *ph.e.*. In the simulation, every time that a *ph.e.* is produced, a random number is drawn from the single *ph.e.* amplitude distribution (which is obtained from lab measurements) to get the amplitude of the output electronic signal. From this amplitude, and using template pulse shapes for the each electronic channel (trigger or digitalisation) the corresponding electronic signals are generated. On top of these signals, the electronic analogue noise of each channel is added.

matelsim allows to 'wire' every pixel to more that one output channel of a given type, allowing a flexibility beyond to what it is possible in real hardware. For instance, one can send the signals in parallel to different trigger systems to performance trigger studies, or to different digitalisation units to see e.g. the impact of different digitalisation speeds.

2.3.4 Simulation of the diffuse light from the night-sky background

The camera's photosensors are continuously exposed to the light of the night-sky background (NSB) and to the light of stars in the field-of-view. The level of the NSB depends on the telescope's location and on the region of the sky under observation (galactic or extragalactic). The user can set this level directly in the input card in units of *ph.e.* per ns per pixel (typical values for the MAGIC cameras are of $0.13\, ph.e./ns$ for extragalactic observations) or more generally by passing the spectrum of the NSB at the observatory place. If this later option is used, the number of *ph.e.* induced by the NSB is calculated from the convolution of the NSB spectrum with the mirror reflectivity and the photosensor *CE* and *QE* curves. This module can be also used to estimate the expected *ph.e.* rate from a given star,

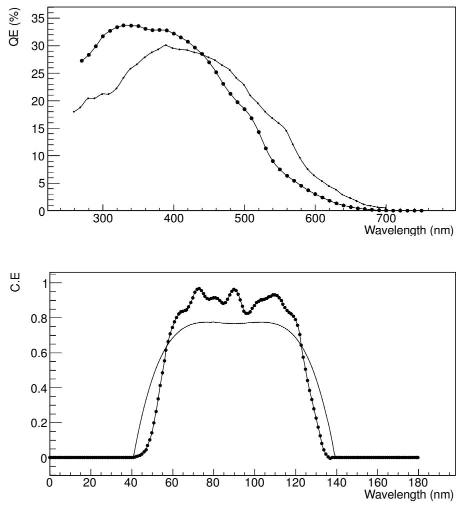

Figure 3: Collection efficiency (top panel) and Quantum efficiency curves (bottom panel) of two types of photomultipliers tubes used in the MAGIC cameras.

providing its optical spectrum. The signals produced by these background photons has to be added to the ones from Cherenkov photons. For speeding up the simulations, at the beginning of the program execution a database containing random arrival times of NSB photons (according to a poissonian distribution) is filled, and the corresponding electronic signals are computed and stored in the database. In this way, once all the Chrenkov photons for a given event have been read and processed, the contribution of the NSB is added on top of the pixel signals.

When simulating the response of PMTs to the NSB, one has to take into account the so-called afterpulses, typically generated by the ionization of residual gases in the PMT. Afterpulses appear hundreds of nanosecond after the main pulse, and can have amplitudes much larger than that of normal signals. NSB photons can then produce signals due to afterpulses affecting Cherenkov events registered hundreds of nanosecond later. These NSB-induced afterpulse signals are properly simulated and included in the NSB database.

2.3.5 Trigger logic

The trigger implementation of matelsim is very flexible, aiming to be used for different trigger logics and specially, for helping in the design phase of new trigger schemes and in the optimisation of the trigger threshold. Currently, it implements the standard digital logic of IACTs and the new one introduced by MAGIC, namely the analogue-sumtrigger.

When the digital trigger is used, the electronic analogue signals sent though the trigger branches are compared against a discriminator threshold which can be set independently for every pixel. The algorithm checks for every simulation time step the value of the analogue signal, and it is above the discriminator thresholod for a given amount of time (of de order of few hundreds of ps), it opens a digital output signal of a given length. The simulated discriminator signals are then sent to the subsequent trigger level, in which for every trigger cell (typically containing 36 pixels) coincidences among fired pixels within a given time window are searched for. Then for every trigger cell, a digital output signal is generated, being these trigger signal combined in an OR-logic in the later stage, for producing the final trigger signal.

In the case of the analogue-sumtrigger, the analogue signals are first clipped at a given value to reduce the fraction of triggers induced by afterpulses. Then, the clipped signal within a trigger cell (of typically 18 pixels) are sum up and the resulting signal compared against a threshold value. As in the digital logic, the digital trigger signals of each cell are combined to produce the final camera trigger signal which is then distributed to central trigger unit when the trigger signal of all the telescope's cameras are combined to produced the final trigger. matelsim allows to simulate both types of trigger logic, at the same time, for every event.

2.3.6 Digitalisation and generation of the output signals

If a trigger occurs, the pixel electronic analogue signals are digitalised. Different kinds of digitalisation units are implemented, as multiplexing units and units based on the DRS2 and DRS4 ring-sampling chips. First, a pedestal signal is added to the signals going through the digitalisation branch and then, the signal are digitalised according to the resolution of the digitalisation units. The digitalisation process includes also the simulation of saturation effects. To make sure that the readout window will contain a realistic signal, the whole history of each channel is simulated in a larger time window and stored in a buffer to account for earlier photons which can produce signals (such as NSB-induced afterpulses) affecting later the region of interest. Finally, the digitalised samples, selected based on the trigger time, are read from the inner buffer and saved to the output file.

2.3.7 Simulation of special runs

In addition to normal data runs, containing cosmic-ray showers, Cherenkov telescopes takes periodically special runs for the calibration of the electronic chain. These special files has to be also simulated.

- Calibration runs. During these runs, the telescope's cameras are illuminated by short flashes of light, mimicking the spectral distribution of Cherenkov light and its duration. matelsim can simulate calibration runs in an easy way, just launching the program with a specific option. The characteristics of the calibration pulses, such as the pulse intensity in *ph.e.*, the colour of the pulses, of the pulse width in ns., can be set from the input card by specific keywords.

- Pedestal runs. One can generate files containing only electronic noise, corresponding to pedestal files taken with the camera closed, or files containing both electronic and NSB noise, corresponding to pedestal files taken with the camera open.

2.3.8 Interactive mode and built-in event display

matelsim includes the possibility to run the simulation in an interactive mode. In this mode, an event display pops up and the simulation proceeds event by event. The user can see what the program is doing in the different steps of the simulation: reading of the photons produced in the atmospheric shower, pixelation of the photons, conversion of photons to *ph.e.*, simulation of the electronic signals which are distributed to the trigger and digitalisation channels up to their final digitalisation. The different displays are also interactive, so by clicking on a given pixel on the camera displays (see figure 4), the user can see signal of that particular channel. Apart from illustrative purposes, this interactive mode allows an easy way to debug new algorithms and to check visually that the setting passed to the programs are the right one for the desired simulation.

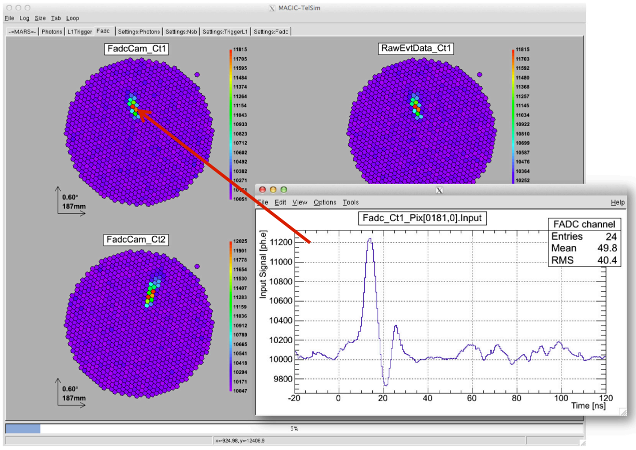

Figure 4: Event display showing the digitalised signals on each telescope camera. On top of it, canvas with the electronic signal simulated in a given pixel, opened by clicking on the desired pixel on the camera.

3 Conclusions

In this work, a new modular simulation framework for the MAGIC telescopes has been presented. The program has the following main parts: first reading of data file containing Cherenkov photons and simulation of the absorption that they undergo in the atmosphere; implementation of ray-tracing algorithms for simulating the telescope's optics; simulation of the photosensors installed in the camera plane and of the trigger and digitalisation systems; to finally produce event files in the same raw format that the one produced by the telescopes during normal operations. The program has been designed to be easy to maintain and highly expandable to new hardware elements or even to be used for future IACTs.

Acknowledgments

The author would also like to thank the support of the *Ramón y Cajal* program from the Spanish MICINN.

References

[1] R. Zanin et al., (2013) These proceedings.
[2] http://root.cern.ch/drupal/
[3] D. Sobczynska, MAGIC-TDAS 02-10, MAGIC internal notes (2002)
[4] D. Heck at al., Forschungszentrum Karlsruhe Report FZKA 6019 (1998)
[5] http://www-ik.fzk.de/ corsika/
[6] O. Blanch, MAGIC-TDAS 01-04, MAGIC internal notes (2001)
[7] A. Moralejo, MAGIC-TDAS 02-11, MAGIC internal notes (2003)