MAGIC-TDAS 02-11
030120/AMoralejo

The Reflector simulation program v.0.6

A.Moralejo
<moralejo@pd.infn.it>

January 20, 2003

Abstract

In this document we provide a brief description of Reflector program (version 0.6) and a guide to install and run it. Some of the information contained in this document is also present in MAGIC-TDAS 02-05 dealing with the previous version of the program, but it has also been included here for clarity. Two important bugs regarding the ray-tracing routine have been corrected in this new version.

NOTE: In December 2002, a first release of Reflector 0.6 was made, together with a first version of the present TDAS note. Immediately after that (too short a time to justify a new version number), some other changes were implemented in the program to improve the atmospheric absorption routines, and the dependence of mirror reflectivity with wavelength was introduced. This document is the manual of the final 0.6 version of the reflector, including explanations of these last modifications.

Contents

1 Introduction **3**

2 Description of simulation **3**

3 Notes on version 0.6 **3**
3.1 First bug fixed in the ray tracing 3
3.2 Performance of ray-tracing in the new version 11
3.3 Second bug: photon timing 11
3.4 The new magic.def file 12
3.5 The new reflectivity.dat file 12
3.6 The *cermaker* program 16
3.7 Changes in the atmospheric absorption routines 16
3.8 Other changes in Reflector 0.6 16

4 How to Install Reflector Program **17**

5 How to Run Reflector Program **17**

6 Needed Files **18**

7 Steering Card **18**
7.1 Mandatory Commands 18
7.2 Optional Commands 19

8 Output file **20**

Appendix A : atmospheric absorption **25**
A.1 Rayleigh scattering 25
A.2 Mie scattering 27
A.3 Absorption by Ozone 29

Appendix B : files in the reflector package **30**

1 Introduction

The Reflector program was originally written by Jose Carlos González and then improved by Harald Kornmayer. The Reflector program reads in MMCS output files (cerxxxxxx files from corsika) and writes an output file with the information about all the photons which reach the telescope focal plane (taking into account atmospheric and mirror absorption) and which are within the camera radius defined in the file **magic.def**. In September 2001 D. Bastieri and C. Bigongiari released a new version (v.0.4) to adapt the program to a format change in the MMCS output. In June 2002 the version 0.5 was released, with small changes in the ouput file format and some bug fixing (see TDAS 02-05).

The aim of releasing the present version 0.6 was in principle only to add some missing information in the output. However, the ray-tracing routine was checked in detail and some inconsistencies were found. As a result of this, an important bug was detected which produced a systematic blurring of all the images. Later another bug was found in the calculation of the photon arrival time on the camera.These bugs, fixed in this version, are discussed in detail in section 3.

2 Description of simulation

The main steps of the simulation are:

1. Atmospheric absorption.
2. Checking if the photon hits the dish.
3. Aluminum absorption.
4. Determination of the mirror hitted.
5. Mirror reflection.
6. Checking if the photon is inside the camera borders.
7. Calculation of photon arrival time on camera.

The reflection of each mirror element is simulated in a realistic way by introducing a gaussian spread of the reflected photons positions on the camera plane. The sigma of this PSF is defined via the **point_spread** parameter in the telescope description file **magic.def**, and has a value of 0.5 cm, which corresponds approximately to the quality of the MAGIC mirrors produced up to date. Also the possible misalignment of mirror elements is simulated (see section 6).

3 Notes on version 0.6

3.1 First bug fixed in the ray tracing

The previous versions of the reflector were extensively checked and a strange behaviour was found in them. It is a well known fact (see for instance the discussions in the MAGIC design report) that, to focus a telescope on an object placed at a finite distance, one has to shift the camera plane *away* from the mirror dish, with respect to the position in which an object at infinity (a star) would be focused (see fig. 1). For instance, with a paraboloid of focal distance f = 1697 cm, an object placed 10 km

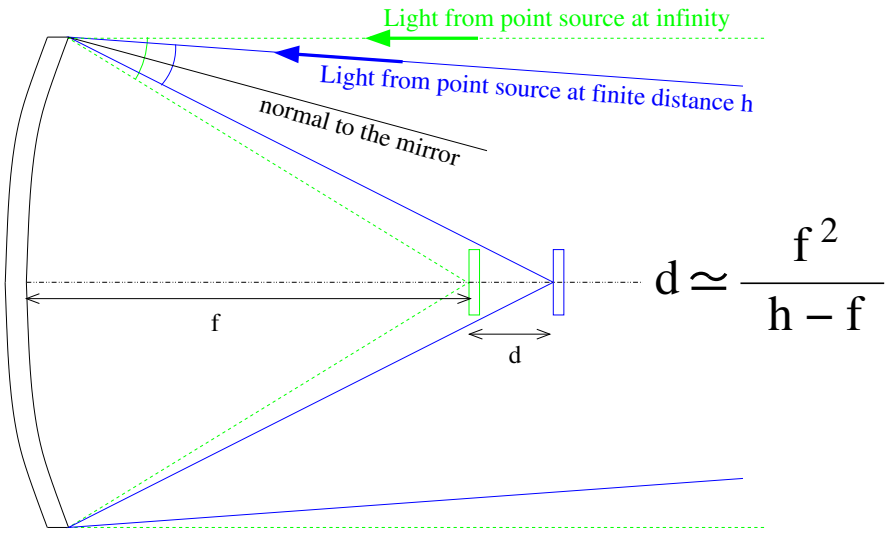

Figure 1: To focus an object at a finite distance h, the camera plane must be moved away from the mirror a distance d, given (approximately) by the formula on the right. For MAGIC, the shift is of around 3 cm for a source located 10 km above the telescope.

above the telescope would be focused on a plane at $\simeq$ 1700 cm from the dish (a distance measured along the telescope axis).

With reflector program up to version 0.5, and using the mirror parameters in the standard magic.def file, we found out just the opposite behaviour. We wrote a program to produce files with the same structure as the Corsika Cherenkov output, but containing photons from a point source. If we set the source at infinity, we found the best spot placing the camera at a distance of 1697 cm (this can be changed in the magic.def file via the *focal_distance* parameter). We can see the resulting spot in fig. 2. A completely independent ray-tracing program was used to verify that this is the spot that a f/1 16.97 m $\varnothing$ tesellated paraboloid should produce. If we now place the source 10 km above the telescope, the best spot is achieved at 1694 cm from the dish, instead of the expected 1700 cm (figs. 3 and 4). It must be noted that for these and the following checks we removed all the gaussian smearing which the program introduces to simulate the possible mirror misalignments and surface irregularities (by the way, a feature which somehow optimistically was not included in the simulations shown in the MAGIC proposal). In this way we can check just the ray tracing.

In order to rule out a possible mistake in the generation of the "false" Cherenkov files, we repeated the check using real Corsika files, and directly looking at shower images ($\Theta = 10^\circ$ incident gammas with a Crab-like spectrum). We must bear in mind that the shower maximum for gammas of 100 GeV lies around 10 km above the MAGIC level. From the result we can clearly see that the images are best focused at 1694 cm (fig. 5), therefore confirming the existence of a real problem in the reflector simulation.

Check of the mirror parameters in the magic.def file

In the standard magic.def file we have been using up to now, the spherical mirrors centers were found to be distributed on the surface of a f = 1700 cm paraboloid. Their curvature radii, though discretized in 8 "zones" as explained in the design report, corresponded in average to the local mean

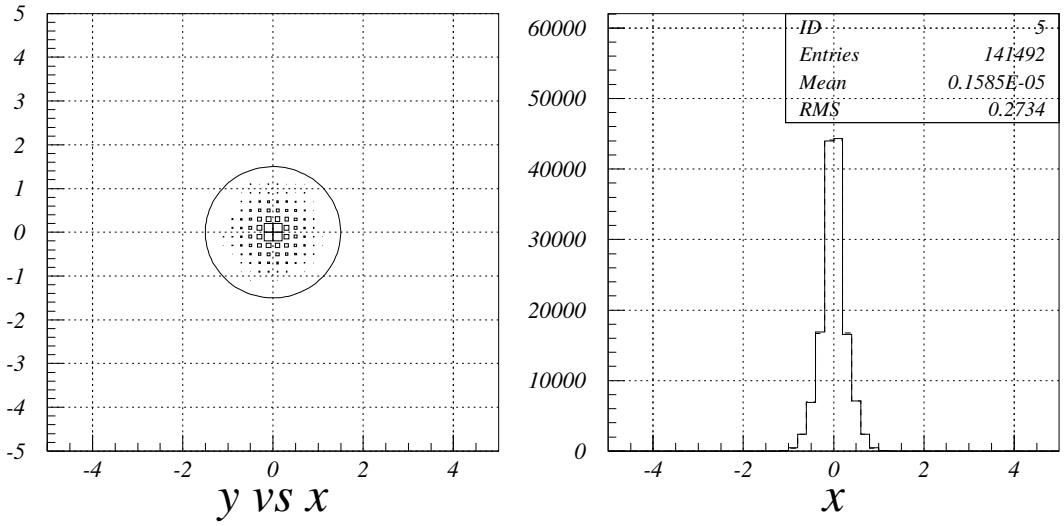

Figure 2: Image of a point like source placed at infinity with Reflector 0.5 (units of x and y are cm). Camera plane placed at 1697 cm from the mirror. The circle on the left indicates (small) pixel size. On the right side, projections on x and y of the spot.

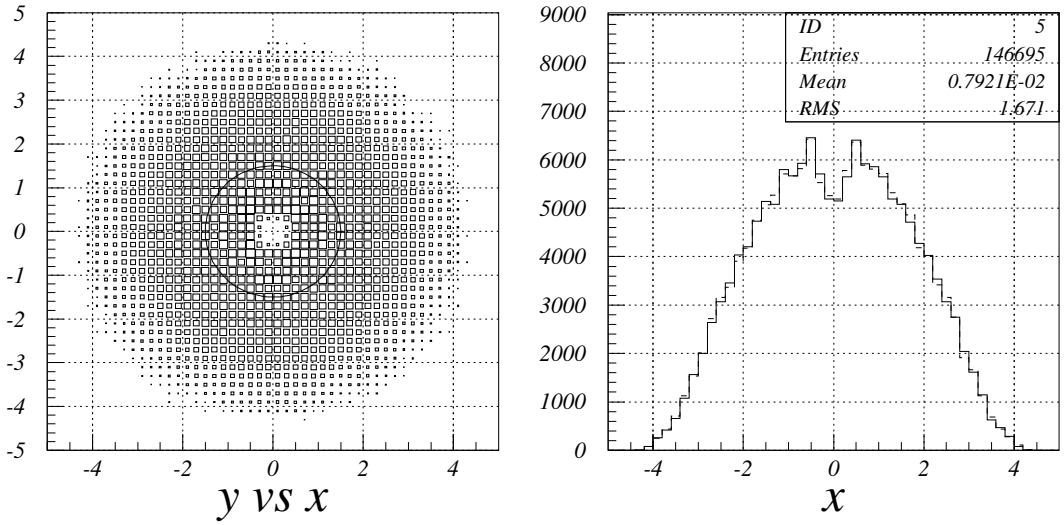

Figure 3: Image of a point like source placed 10 km above the telescope with Reflector 0.5 (units of x and y are cm). Camera plane placed at f = 1700 cm from the mirror. The fact that the spot is so large indicates a problem in the simulation, since with this camera position the telescope should be focused at 10 km (see fig. 2) and hence produce a spot similar to the one shown in fig. 2. The hole in the middle of the spot corresponds to the hole in the mirror dish, and indicates by itself a focusing problem.

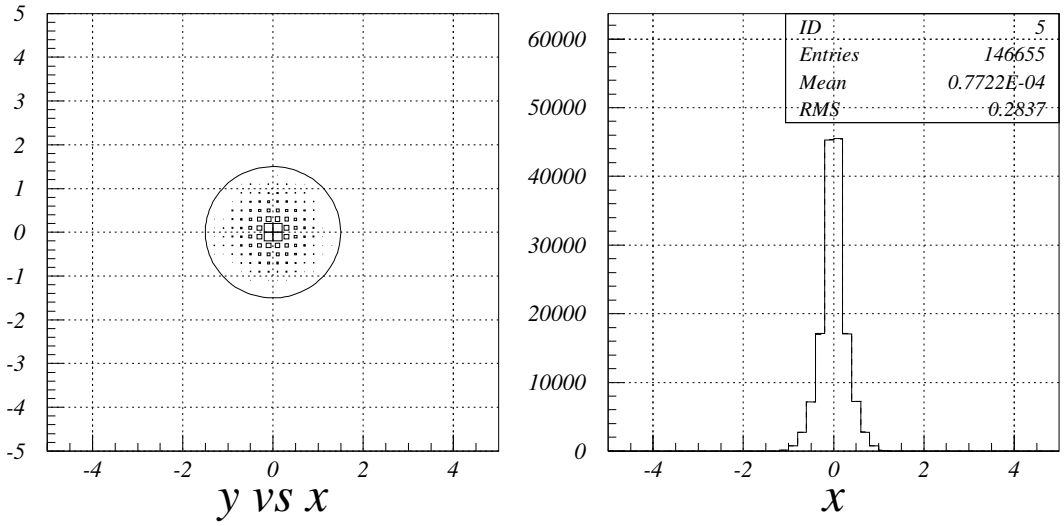

Figure 4: Image of a point like source placed 10 km above the telescope with Reflector 0.5 (units of x and y are cm). Camera plane placed at f = 1694 cm from the mirror. Now the spot is small, but the camera plane is not at the expected position (see text).

curvature radius of the same parabolic surface. However, their orientations were those corresponding to a f = 1697 cm paraboloid. Personally I do not think this is something intended, and looks more like an error, but anyway we checked that this was not the reason for the problems in the ray tracing: a test magic.def file was created with *all* the mirror parameters (positions, orientations and radii of curvature) calculated as for a f = 1697 cm paraboloid, and no significant difference could be seen. That is: i) the individual mirror orientations are the dominant factor, and the overall dish in the old reflector behaved indeed like a f = 1697 cm parabolic mirror, and ii) the bug must be somewhere else. Even though the error was not there, a new magic.def file has been produced containing the parameters for the 956 mirrors of the final MAGIC design (instead of the 920 mirrors foreseen in the beginning).

Solution of the problem

Finally, a check of the routines in *ph2cph.c* revealed that the reflector program was actually misinterpreting the information read from the Cherenkov file written by Corsika. In figure 6 we can see the definition of Corsika's coordinate system (see also the Corsika manual). In it, the x axis points north and the z axis points up. The zenith angle Θ of a particle trajectory is measured between the particle momentum and the negative z-axis, and the azimuthal angle Φ between the positive x-axis and the x-y component of the particle momentum, counterclockwise (0 - 2π). When the direction cosines of a particle's trajectory are given, they refer also to its momentum vector (a downgoing vector)[1]. The same is true for the Cherenkov photons. Unfortunately, in the output of Corsika, only the direction cosines (u, v) with respect to the x and y axis are given. These completely determine the particle's trajectory as long as one knows that they refer to a downgoing versor, in which case one gets the third direction cosine as $w = -\sqrt{1 - u^2 - v^2}$, with a minus sign. However, in *ph2cph.c* we found exactly the

[1]In Corsika the third component of the momentum is measured along the negative z-axis and this is probably the origin of the confusion which resulted in the bug

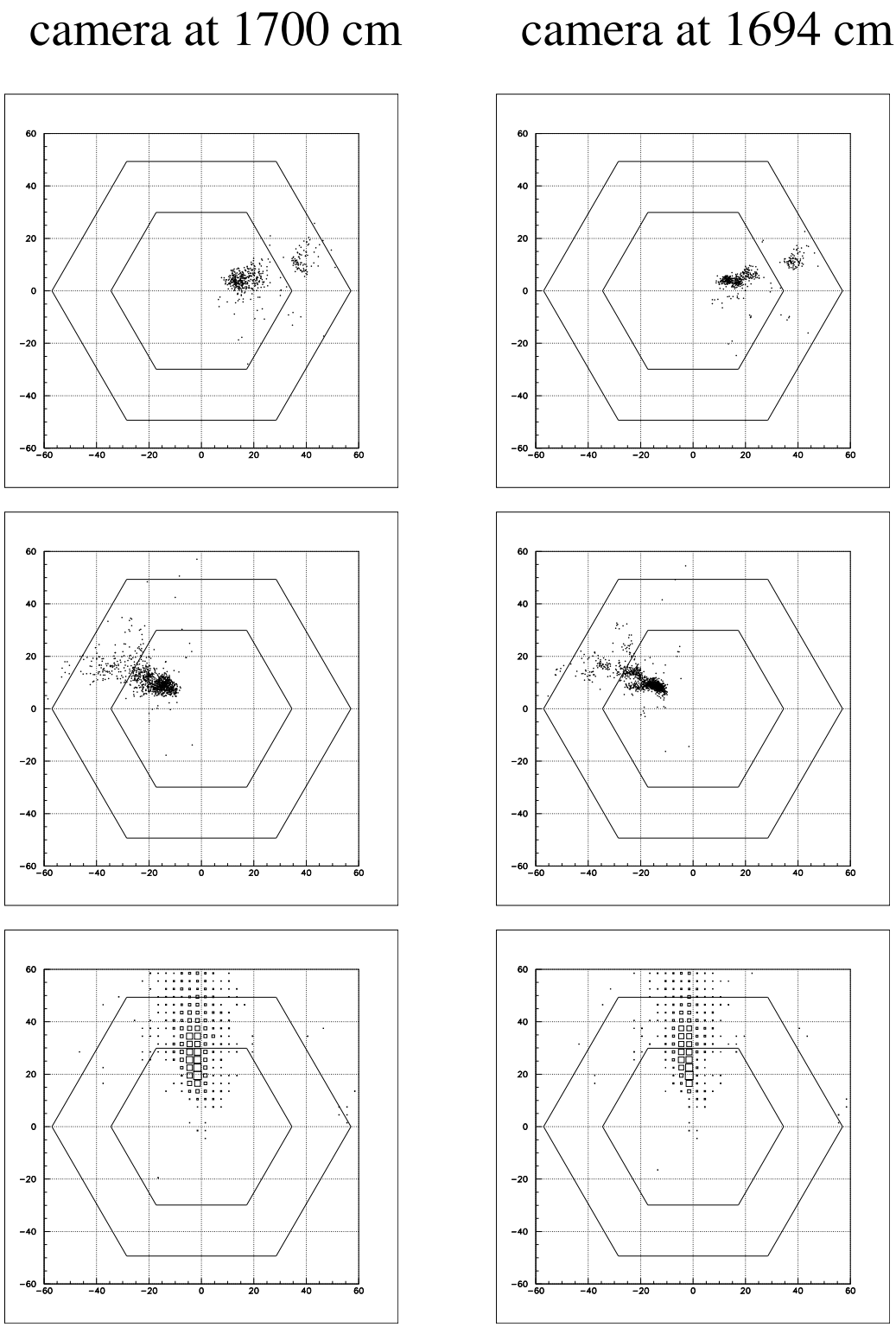

Figure 5: Gamma shower images obtained with Reflector 0.5 placing the camera at 1700 cm (left) and 1694 cm (right) from the mirror dish. Primary gammas at $\theta = 10°$, E = 16, 46 and 232 GeV.

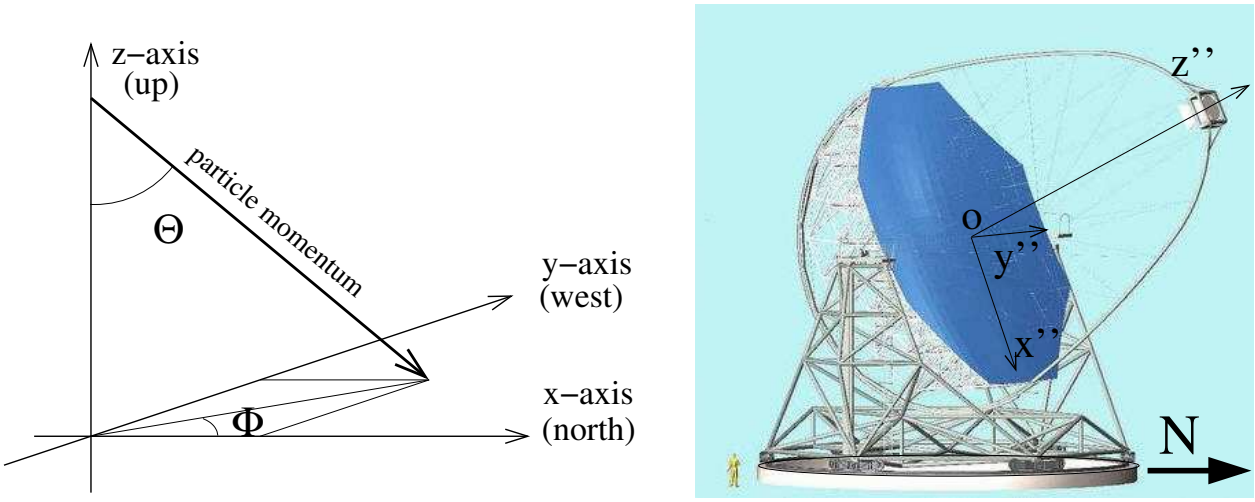

Figure 6: Coordinate systems of Corsika and MAGIC.

opposite (lines 116 to 118 in v.05):

```
r[0] = ph->u;
r[1] = ph->v;
r[2] = (float) sqrt(1.0 - r[0]*r[0] - r[1]*r[1]);
```

This means that the program was interpreting u and v as the direction cosines of the *upgoing* versor towards which one should "look" to see the incoming photon. The situation is depicted in figure 7. The two vectors at the bottom left of the plot are the two possible interpretations of the direction cosines u and v. These vectors have the same u and v, and differ only in the sign of their third components w. As the figure illustrates, the reflector simulation, by taking the wrong (positive) sign of w, was transforming the light coming from a point source 10 km away, into a convergent light beam which was then focused at a distance (1694 cm), shorter than the focal length of the paraboloid, and produced a blurred image at the theoretically optimal distance (1700 cm). Of course a paralel beam of light was focused at the right distance (1697 cm) since in that case $u = v = 0$, and both the upgoing and downgoing versors have the same direction.

The example in figure 7 shows the case in which the telescope is pointing at zenith. It is easy to see that if the telescope was pointing in an arbitrary direction a few degrees away from zenith, the fact of taking the wrong sign in the third direction cosine would spoil the reflection completely, to the point that the spot would lie outside the camera limits. However, this is not what we observed: when using the older reflector versions for different zenith angles the images were still contained in the camera. The explanation is that the transformation between Corsika's coordinates system and the telescope system was also wrong, since the angles Φ and Θ which indicate the telescope orientation (see fixed_target option in section 7.1) follow the same convention taken from Corsika: for instance, for pointing the telescope towards North we should set $\Phi = 180°$, because that would be the Φ value for a particle or photon coming from North (see again fig. 6, left). This wrong transformation of coordinates oriented the telescope in a way that the situation was always like the one shown in fig. 7, and the image was formed in the camera also for $\Theta > 0$, though also defocused.

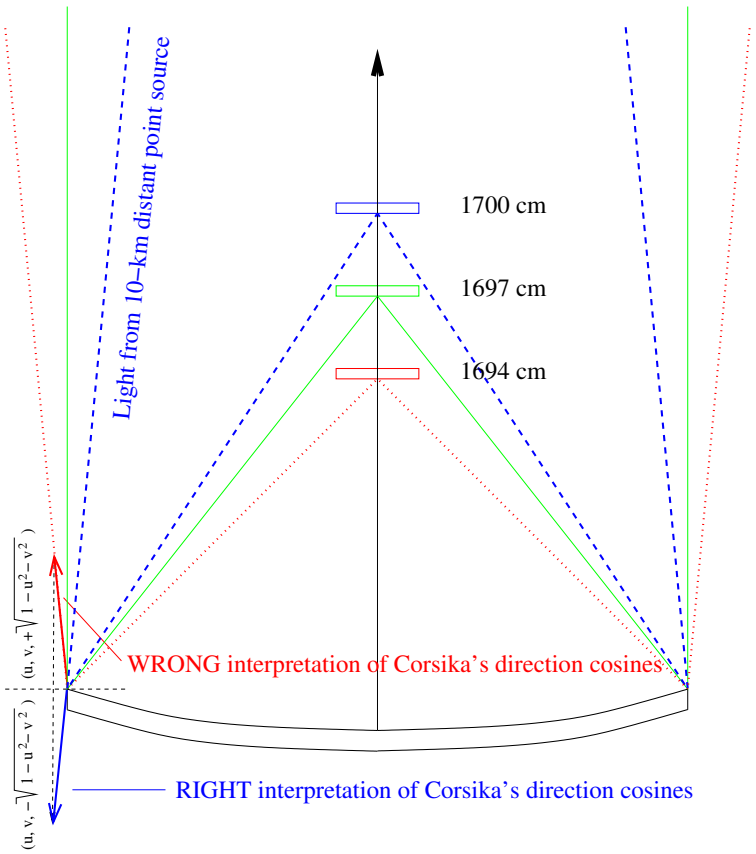

Figure 7: Misunderstanding of photon direction cosines from Corsika by Reflector v 0.5 and older versions.

The telescope coordinate system shown in figure 6 (right) has its z-axis along the telescope axis, and the origin in the center of the mirror dish. This system is used in the ray tracing routine of the reflector simulation. When the telescope points up ($\Theta = \Phi = 0$) this system matches exactly the one in Corsika. The general transformation between both is a simple rotation, since for the sake of simplicity we assume in the simulation that the origins always coincide. As we have said, in Reflector v.05 or older the rotation matrix was wrong: it had been written assuming that (Φ, Θ) indicated the direction towards which the telescope pointed. Actually, for the reasons already exposed, the telescope must point to ($\Phi + \pi$, Θ). The function in charge of building the rotation matrix is *makeOmega* (a part of *ph2cph.c*), which is called from *reflector.c*. For the present version we have simply replaced Φ by $\Phi + \pi$ in the function call. The transformation of coordinates is shown in figure 8, and can be seen as a rotation of angle $\Phi + \pi$ around the z axis of Corsika plus a rotation of angle Θ around the y" axis of the telescope (the same way in which the real MAGIC points).

New coordinate system of the camera We have introduced another change in Reflector 0.6 regarding the coordinates. In versions up to 0.5 the coordinates (x_{camera}, y_{camera}) of the photon impact point on the camera plane were given in the telescope system (x", y", z") described in fig. 6. This was a bit confusing (a rotation of the telescope in the zenith axis resulted in a displacement in x"

of the images), in particular it would have been messy when working in wobble mode. We have now adopted the camera coordinate system proposed in TDAS 01-05: when the observer is looking from the center of the reflector in the direction of the telescope axis (towards the camera) the x_{camera} axis points horizontally to the right, and the y_{camera} axis points upwards. It is trivial from figure 8 to see that the transformation needed to obtain these coordinates is: $x_{camera} = -y''$, $y_{camera} = -x''$. This has been added at the end of *ph2cph.c*.

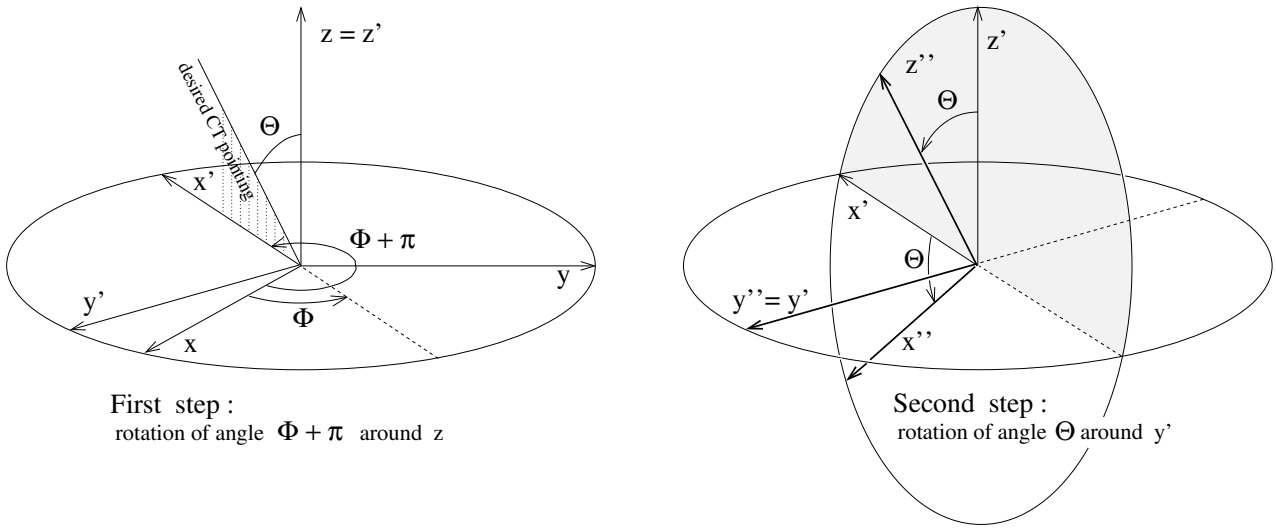

Figure 8: Transformation of coordinates between the Corsika and MAGIC coordinate systems.

How old was the ray-tracing bug?

The bug was certainly present in versions 0.4 and 0.5, but may be even older. Nevertheless, there is no doubt that the reflector program used for the simulation shown in the MAGIC design report was working fine. Extensive proof of this is provided in an appendix of the design report. A plausible explanation could be that, up to some date, the data being read in by the reflector program (the Corsika output) contained direction cosines which really referred to the upgoing versors of the photon directions, and until then the program worked well. Then may be the output of Corsika was changed to its present form, and the change went unnoticed. The Corsika version used then was 4.52, whereas all further work has been done with Corsika 5.20 or later versions. The Corsika history file shows no record of any change in this respect, but given that we have always used a slightly modified Corsika, it would not be surprising if the Cherenkov output was modified at some point in the MAGIC version of Corsika (MMCS). There is no documentation on this, so if anyone has any relevant information, please make it public.

Influence of the bug on image analysis

The fixing of the bug (resulting in sharper images) will for sure improve the results of the image analysis, in particular with regard to gamma / hadron separation. This could in part explain the differences we have been observing in the expected performance of MAGIC with respect to what was foreseen in the design report. However, we note here again that in the simulation used there, not only was the reflector working well, but also no noise was introduced in the reflecting process. This

was surely too optimistic, and it implies that it will be hard to reproduce those results using a more realistic approach which accounts for mirror imperfections. However, the introduction of the noise (fig. 12) has a less dramatic effect than the defocusing which the bug was producing.

3.2 Performance of ray-tracing in the new version

In figure 11 we show the images of a point-like source at 10 km from the telescope, produced with the Reflector version 0.6, and using the new version of the magic.def file (see see next section). No noise has been introduced in the reflection, the observed spots are just the result of optical aberrations. The light source has been put at slightly different viewing angles from the telescope. The results are comparable to those in the design report, actually these are a bit better, the difference probably being that the focal lengths of the mirror tiles in the older magic.def file were discretized in only eight values, while now they change rather continuously. Some images of a point source at infinity (a star) can be seen in fig. 13. We can see that for any incidence angle, the area within which 50% of the light is concentrated is smaller than that of a small pixel.

In figure 12 the images of three gamma events ($\theta = 10°$, E = 16, 46, 232 GeV), the same of fig. 5 are shown. They have been produced with Reflector 0.6 assuming perfect spherical mirrors (left) and realistic ones (right). The images look reasonable, much sharper than with the older versions, even when the mirror imperfections are taken into account.

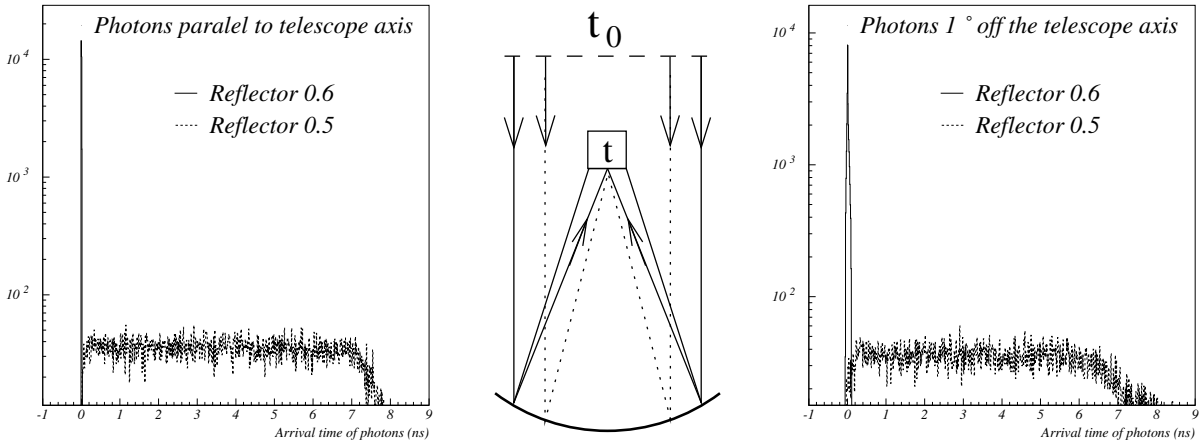

Figure 9: Test of reflector isochrony. The arrival time distributions of photons in the camera are shown for (buggy) Reflector 0.5 and for Reflector 0.6. The sketch in the center shows the test for the case in which the light beam is paralel to the telescope axis (left plot). On the right, the same test has been made with light arriving 1 degree off axis.

3.3 Second bug: photon timing

After a first release of Reflector 0.6 had been made public, another important bug was found. We tried to check whether the simulated reflecting dish was really "isochronous". A paralel beam of photons, all sharing the same arrival time on the ground, were processed by the simulation program and their arrival times on the camera plane were histogrammed. The result can be seen in fig. 9 (dashed histograms). This time the bug was quite evident: since Corsika gives us the arrival time of

photons on ground, the path from the point where the photon hits the dish to the ground has to be subtracted (or added, because since the center of the dish in the MC is at $z = 0$, the mirror reflecting the photon may have $z < 0$ when the dish is inclined). The sign in this subtraction (in *ph2cph.c*) was wrong. This bug was present in both versions 0.5 and 0.4, and might be related to the other one (a change of orientation of the z axis at some point may have produced it). Since in the simulation made for the design report the timing played little or no role (the camera simulation did not consider the arrival times of photons) it is not possible to know whether the bug was already in the code by then.

This bug means that all the studies made up to now regarding photon arrival times on the camera are completely useless (for instance, the optimization of the time parameters in the L1 trigger has to be redone from scratch!).

3.4 The new magic.def file

A new magic.def file (see sect. 6) has been created and included in the Reflector 0.6 package. Now the number of individual mirror tiles is 956, matching the number and distribution of the final MAGIC design. The mirror centers and orientations are those corresponding to a paraboloid of 1697 cm focal (hence the camera plane is placed at 1700 cm from the dish). The focal lengths have been calculated by R. Mirzoyan taking into account the so called "shortening effect" (see design report). A new axisdev.dat file (se again 6) with data for the 956 mirrors is also included.

3.5 The new reflectivity.dat file

Up to version 0.5 of the program, the reflectivity of the mirrors (which the program reads in from the file reflectivity.dat) was considered to be equal to 90% for all wavelengths. Following measurements performed in Padua of a mirror sample, the reflectivity has been found to be between 80 and 90% in the range from 280 to 650 nm, with a dependence on wavelength which is shown in figure 10.

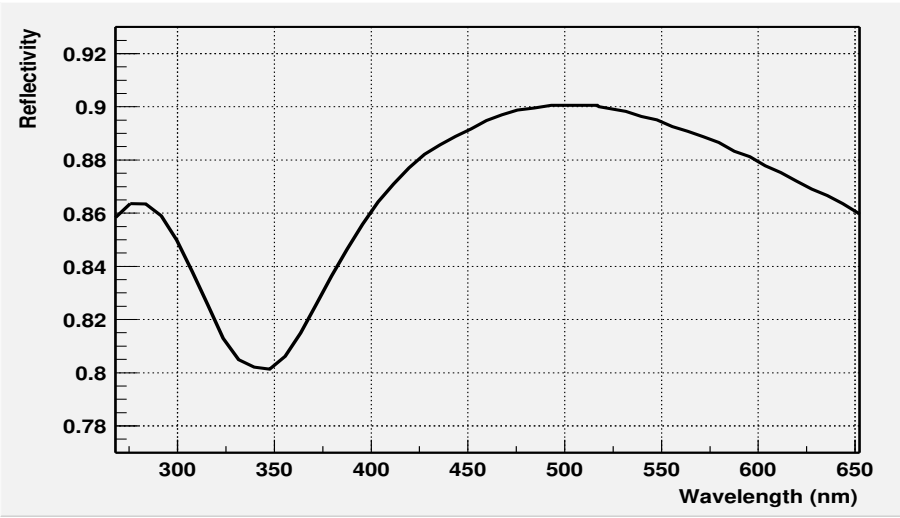

Figure 10: Reflectivity of the MAGIC telescope mirrors as a function of the wavelength of the incident light. The dip around 350 nm is due to destructive interference of the light reflected on the aluminum plate with that reflected on the protective coating of the mirror.

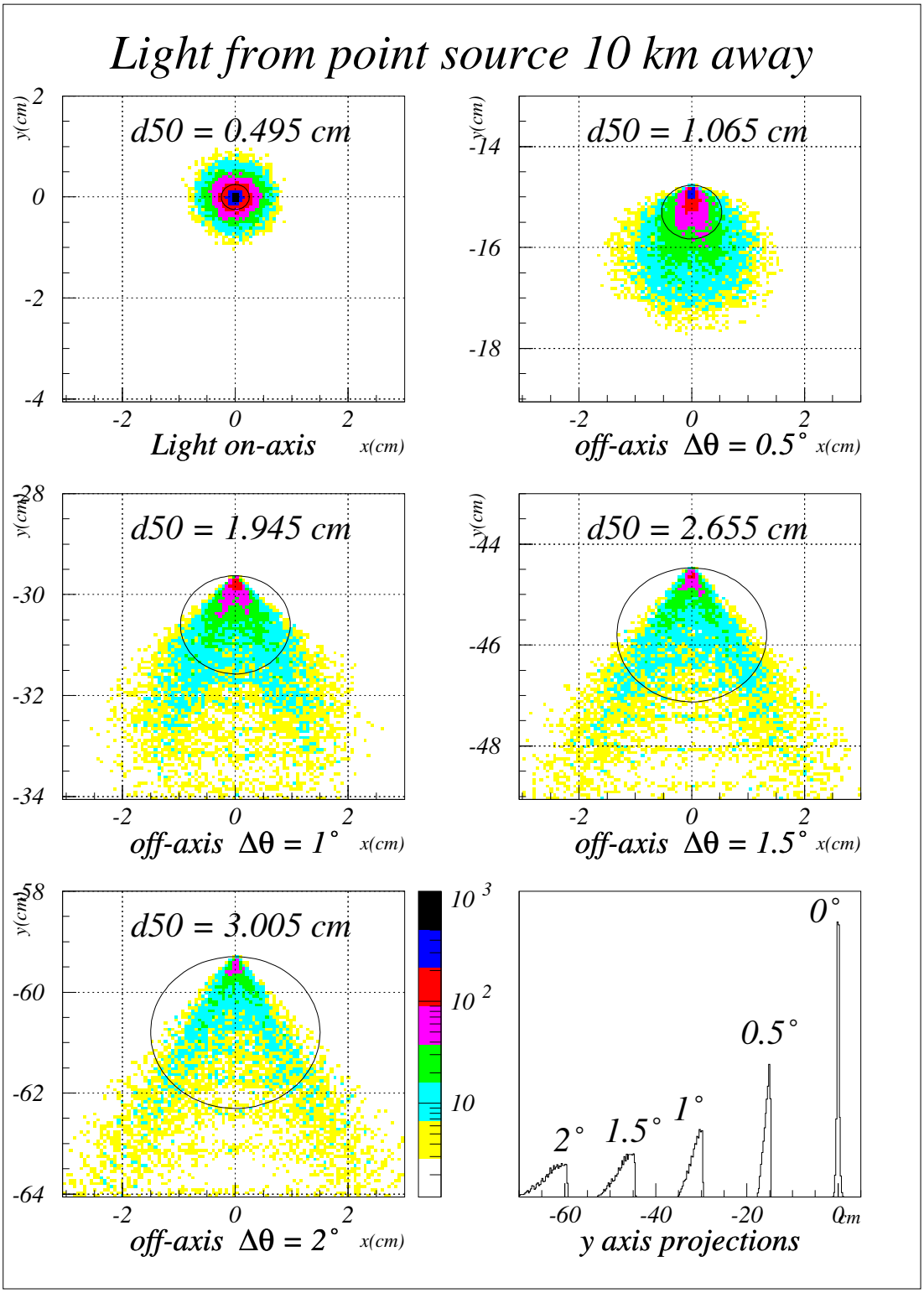

Figure 11: Reflector 0.6. Images of a point-like source at 10 km from the telescope for different incident angles (from on-axis to 2 degrees off-axis). The quantity d50 indicates the diameter of a circle (plotted) containing 50% of the reflected light. Note that the z-axis scale is logarithmic, and the same in the first five plots. The last plot shows the x-axis projections in linear scale.

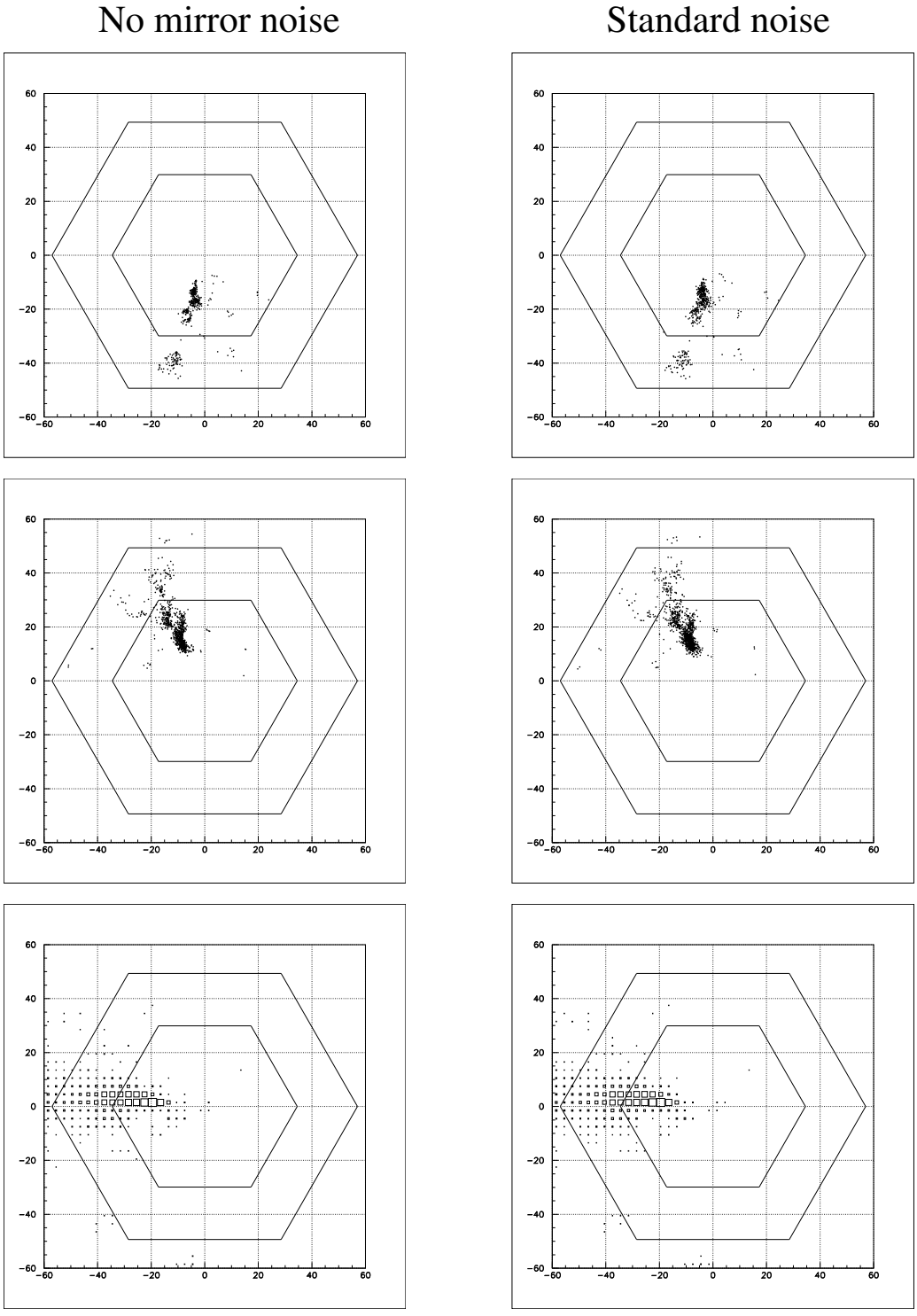

Figure 12: Reflector 0.6. Images of the three gamma showers shown in fig. 5, without noise added in the reflection (left) and with the standard noise (right) as described in section 2. Note that the orientation of the images has changed as a result of the introduction of a new camera coordinate system (see page 9).

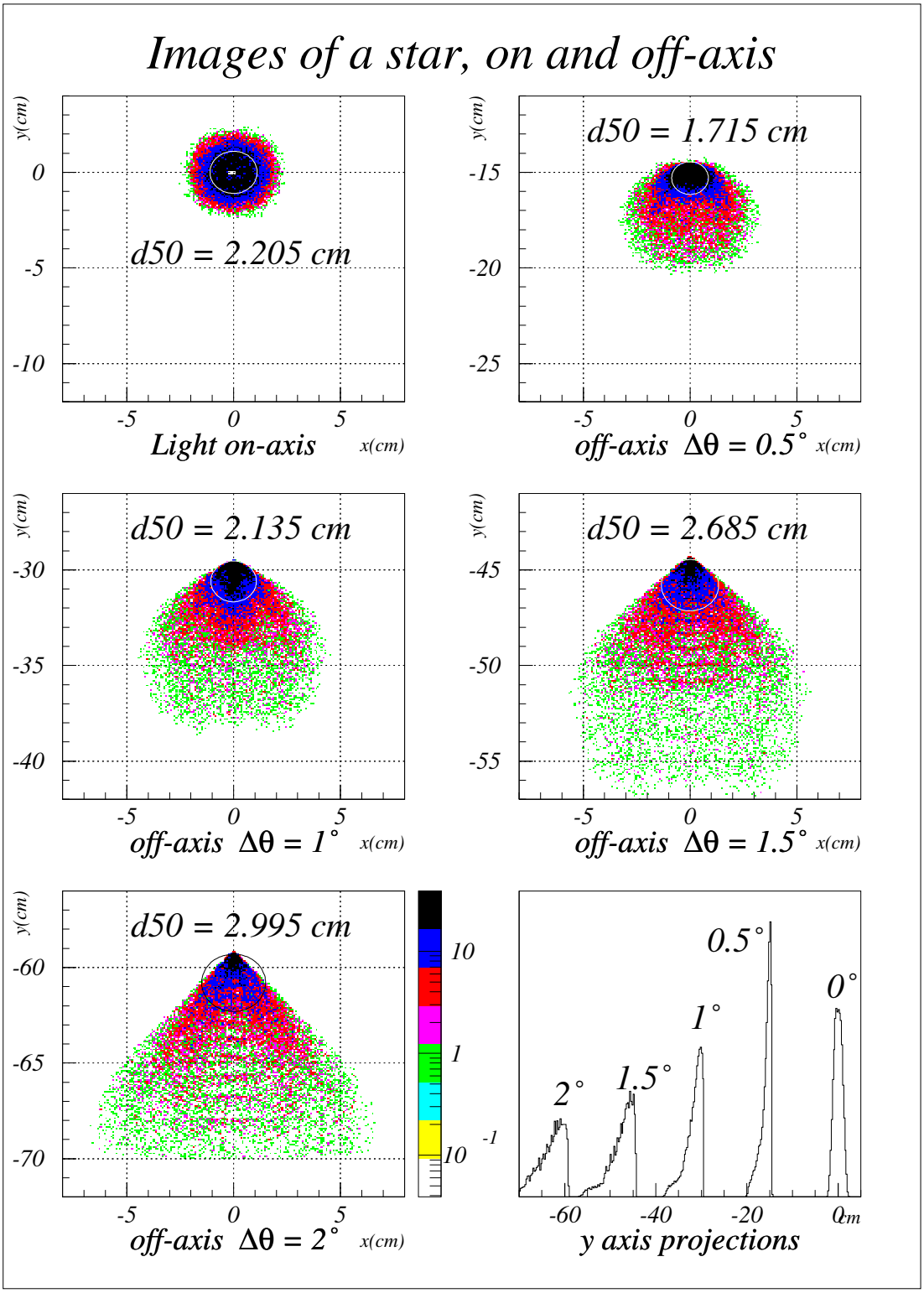

Figure 13: Reflector 0.6. Images of a star. The quantity d50 indicates the diameter of a circle (plotted) containing 50% of the reflected light.

3.6 The *cermaker* program

A test program to produce cer files (input for the reflector) containing photons from a point-light source of light placed in any position has been added to the Reflector package *(tester/cermaker.c)*. This is the same program used to produce the plots shown in this report. The usage is as follows:

```
cermaker source_x(cm) source_y(cm) source_z(cm) [events]
```

The source position is given with respect to the telescope. The output file is called *cer000001*, and can be read by the reflector program.

3.7 Changes in the atmospheric absorption routines

In the present version of the program, the simulation of atmospheric absorption has undergone major changes with respect to the original implementation by J.C. González and Aitor Ibarra. Three options are available in the simulation: ATM_NOATMOSPHERE, ATM_90PERCENT and ATM_CORSIKA (see section 7.1). In the first two, 100% and 90% of the emitted light respectively reaches the telescope mirror, regardless of the emission height. The changes in the present version affect only the third option, which is the recommended one for running reflector in the standard MAGIC MC production. In this case, a detailed estimate of the atmospheric transmission is done.

Three contributions to the atmospheric opacity are considered: Rayleigh scattering, Mie scattering by aerosols, and absorption by ozone. Details on how the effect of these three components is calculated are given in appendix A. In reflector versions older than 0.6, due to a bug, the contribution of Mie scattering and Ozone absorption was slightly overestimated because the vertical height of the emission point above sea level was interpreted as height above the telescope. The difference is small, since the density of the aerosols responsible for Mie scattering decreases very fast with height and therefore the absorption is hardly increased by going up 2.2 km in the region were most of the Cherenkov light is produced. In the case of ozone, its contribution is only important in the very low end of the Cherenkov light spectrum, and so the effect of the bug in the total amount of light reaching the telescope is negligible. Another change is that now the observation level is read in from the cer file, instead of being a fixed parameter in the routines. In this way the reflector program has become more flexible, and can now be used to process cer files produced for a detector at a different observation level.

Another problem in the old implementation of absorption was that the variation of the optical depth of the atmosphere with the zenith angle θ was assumed to be the same for Mie scattering, ozone absorption and Rayleigh scattering: the so called *air mass* (see appendix A for details) was therefore calculated only once, using the overall density profile of air molecules (which does not match that of aerosols or ozone), and used to account for the variation of all three effects with θ, while rigorously speaking it is only valid for Rayleigh scattering. Now, the simulation of Mie scattering and ozone absorption has been moved from *attenu.f* to *atm.c* and is done in a more accurate way which takes into account the vertical profile of aerosols and ozone (see appendix A for details).

3.8 Other changes in Reflector 0.6

Some other minor improvements have been introduced in Reflector 0.6:

1. reflector.c: Introduced NaN (Not a Number) check in the photon loop. If NaNs are found in a photon data block (there are some in the Corsika output from time to time, for unkown reason), it is not processed.

2. ph2cph.c: Introduced "check of positiveness" before taking a square root in the calculation of the photon trajectory intersection with the paraboloid (resulted sometimes in NaNs when the photon did not intersect the paraboloid).

3. Added an option for the wobble mode in the input card (see section 7.2).

4. New output format (see sect. 8): we have removed a null byte that was written immediately after the ascii label containing the reflector program version number at the begining of the file. This byte was there for historical reasons and had no function whatsoever. Then we have added a Run header, which is like that of Corsika, plus a couple of variables concerning the reflector parameters: the wobble mode and the atmospheric model used for the simulation. The event header has also been changed to include all the information present in the Corsika event header. We also added in the event header three new variables which tell us for each event what fraction of the Cherenkov photons on the camera plane has been produced by electrons, muons, or other particles.

   Finally, the ascii files *magic.def*, *axisdev.dat* and *reflectivity.dat* which the program has used as input are now attached at the end of the output file, so that each output file contains all the relevant information on how it was produced.

4 How to Install Reflector Program

You can get the current version of the Reflector Program from the MAGIC web page:
http://hegra1.mppmu.mpg.de/MAGICWeb/
You can find the latest public version of this program as tarred gzipped file in the Monte Carlo Download area (you need the usual password). You have to download the file reflector_0.6.tar.gz and then follow the instructions below:

Decompress the file using: gunzip reflector*.tar.gz

Unpack the tar file with: tar xvf reflector*.tar

Go to the directory where the source files are: cd MagicProgs/Simulation/Detector/Reflector_0.6/

Make symbolic links running the script: refl-install

Check if all dependencies are fulfilled: make depend

Compile the program: make

If everything goes right you should have an executable file called **reflector**.

5 How to Run Reflector Program

You need a steering card to run the Reflector program. You can find an example in the **MagicProgs/Simulation/Detector/Reflector_0.6/input.card** file. You have to modify this file according to your needs (see below for instructions about steering card) and then run the program with the following statement:

reflector < input.card

6 Needed Files

The Reflector program needs some other files to run. These files are the following:

- **magic.def**: contains the description of MAGIC telescope geometry, together with some other parameters needed by the Reflector program.
- **axisdev.dat**: contains data to simulate the possible deviation of the spot of each single mirror on the camera plane due to its non perfect alignment. The values are x, y coordinates distributed at random (according to a gaussian with $\sigma = 0.5$ cm).
- **reflectivity.dat**: contains the mirror reflectivity index as a function of the wavelength.

All these files (included in the reflector package) are usually kept in the **MagicProgs/Simulation/Detector/Dat** directory and *in principle* you should **not** make any change in them to run the program.

7 Steering Card

The steering card sets all the parameters and options to steer the reflection simulation. Each line of the steering card is a statement with its parameters, if it is the case. Lines beginning with # are considered comments. The Reflector program parses all the lines sequentially. Then if you repeat a statement with different options only the last one will be considered.

7.1 Mandatory Commands

reflector 0.6 This statement must be the first line of the steering card file. The Reflector program checks it to verify if it is reading a steering card.

output_file /disk99/reflex/Gamma_0_7_1001to1010_w0.rfl The output_file command specifies the name and the path of the output file. The path can be absolute, like in the example above, or relative. Although any name can be used, conventionally the Reflector program output file name has the .rfl extension, and starts with the primary particle name. The first number indicates the zenith angle of the incident primaries, the second one indicates the production site (7 is for Padua) and is related to the random number generator seed used by CORSIKA. Then the run number range is shown (10 runs in this case, from 1001 to 1010). Each run corresponds to 10000 showers. Finally, the label "w0" means no wobble mode was used (telescope pointing at the source). Alternatively, the "w+" or "w-" labels (only in gamma files) refer to the two pointings in the Wobble-observation mode (see TDAS 01-05 by W. Wittek).

ct_file /path/magic.def The ct_file statement defines where the program can find the telescope characteristics. Usually, the magic.def file is kept in MagicProgs/Simulation/Detector/Data

atm_model ATM_CORSIKA The atm_model statement tells the program what kind of atmospheric absorption model to use. Possible choices are: ATM_NOATMOSPHERE, ATM_90PERCENT and ATM_CORSIKA, corresponding respectively to no absorption, a 10% absorption and a model using the US Standard atmosphere (see Corsika manual, appendix C) for the Rayleigh scattering and a model by L. Elterman [1, 2] for the Mie scattering and ozone absorption (see appendix A). The third model should be chosen for the standard MC MAGIC production.

cer_files All the lines following this statement are considered files to be processed by the Reflector program, one for each line, eventually with their paths (see the example below). Therefore this command must be the last one.

cer_files
/disk99/cer001001
/disk99/cer001002
/disk99/cer001003
.....
/disk99/cer001009
/disk99/cer001010

The cer file name can be followed by two numbers, for example:
/disk99/cer001001 376 5723

In this case the program processes only the events between and including the numbers given.

7.2 Optional Commands

verbose_level 1 Sets the quantity of information printed out by Reflector when running. Possible values are 0 to 4

max_events 50000 Fixes the maximum number of events to process.

energy_cuts 100 1000 This statement forces the Reflector to process only showers with primary energy between the given values (GeV).

seeds n1 n2 Seeds for the random number generators to used by the program for the simulation of the absorption (both in the atmosphere and on the mirror). Default values are 3141592 and 2718182.

telescope_position x y Option included in version 0.5 of Reflector. Usually it is not needed, since for normal MC production for MAGIC the telescope is placed at the origin of coordinates (0,0). But, if for some reason, we produce cerxxxxxx files with the telescope in a different position, we must inform the Reflector program in the input card using this option (otherwise Reflector will fail to *find* the photons in the cer file).

reflectivity_file /path/reflectivity.dat File containing mirror reflectivity as a function of wavelength (see section 6). If this option is not supplied, the program will look for "../Data/reflectivity.dat" as previous versions of Reflector did.

axisdev_file /path/axisdev.dat File containing single mirror spot deviation in **x** and **y** on the camera in cm (see section 6) for each mirror. If this option is not supplied, the program will look for "../Data/axisdev.dat" as previous versions of Reflector did.

fixed_target Θ Φ This statement fixes the telescope axis position. The first number is the zenith angle Θ (deg) while the second is the azimuthal angle Φ (deg). This corresponds to *CORSIKA*'s definition of primary particle incident direction (see fig. 6, left). For instance, $\phi = 90°$ means that the telescope is pointing towards East. If this option is omitted the telescope will always point in the direction of the Corsika primary (whatever it is), or a slightly modified direction if

the wobble_mode option is used (see next point). When running the reflector over gamma data generated in a range of zenith angles, one should therefore ommit the fixed_target option.

wobble_mode w Indicates whether the reflection should be done in the wobble mode, that is, with shifted pointing with respect to the nominal telescope orientation (given by fixed_target or otherwise, see above). The wobble mode is described in TDAS 01-05. Possible values for w are 0 (no wobble mode), 1, -1 (image shift along x_{camera} axis) and 3 (image shift along y_{camera} axis).

8 Output file

The reflector output file begins with two ascii lines, the first of which informs us of the program version with which it has been produced (NOTE: in the following, the dollar symbol $ stands for a carriage return):

```
reflector 0.6$START---RUN$
```

Then there is run header which is basically the one from Corsika with two added variables, *wobble_mode* and *atmospheric_model*. Check the Corsika manual for the meaning and units of the rest of them. All of the variables are 4-byte real numbers except the first, which is a 4 character string containing the run header ascii label from Corsika:

	Variable	Description
1	ASCII Label	'RUNH'
2	RunNumber	
3	date	
4	Corsika_version	
5	NumObsLev	
6 to 15	HeightLev[10]	
16	SlopeSpec	
17	ELowLim	
18	EUppLim	
19	EGS4_flag	
20	NKG_flag	
21	Ecutoffh	
22	Ecutoffm	
23	Ecutoffe	
24	Ecutoffg	
25 to 74	C[50]	
75	wobble_mode	Wobble mode with which the reflector was run (TDAS 01-05)
76	atmospheric_model	Atmospheric model used for the absorption simulation 0 = no atmosphere; 1 = atm_90percent; 2 = atm_corsika.
77 to 94	dummy1[18]	not used
95 to 134	CKA[40]	
135 to 139	CETA[5]	

	Variable	Description
140 to 140	CSTRBA[11]	
151 to 254	dummy2[104]	not used
255 to 259	AATM[5]	
260 to 264	BATM[5]	
265 to 269	CATM[5]	
270 to 273	NFL[4]	

Then there comes a carriage return followed by the ascii flag which indicates the start of an event, and again a carriage return:

```
$START-EVENT$
```

and then the binary event header. Each variable is a 4-byte float number except for the first one which is the event header label from Corsika (a string of 4 characters). Some of of the variables from Corsika are not explained here (see Corsika manual instead).

	Variable	Description
1	ASCII label	'EVTH'
2	EvtNumber	Event Number
3	PrimaryID	Primary particle identification code
4	Etotal	Primary particle total energy (GeV)
5	Thick0	CORSIKA's starting altitude in g/cm2
6	FirstTarget	CORSIKA's number of first target if fixed
7	zFirstInt	Height of first interaction in cm
8 to 10	p[3]	Primary particle momentum in x,y,-z directions (GeV)
11	Theta	Primary particle zenith angle (rad)
12	Phi	Primary particle azimuth angle (rad)
13	NumRndSeq	Number of different CORSIKA random sequences (max. 10)
14 to 43	RndData[10][3]	RndData[i][0]: integer seed of sequence i RndData[i][1]: number of offset random calls (mod 10^6) of sequence i. RndData[i][2]: number of offset random calls ($/10^6$) of sequence i.
44	RunNumber	Run number
45	DateRun	Date of run yymmdd
46	Corsika_version	Version of *CORSIKA*
47	NumObsLev	Number of observation levels (should be always 1 for us)
48 to 57	HeightLev[10]	Height of observation levels in cm
58	SlopeSpec	Energy spectrum slope
59	ELowLim	Energy lower limit (GeV)
60	EUppLim	Energy upper limit (GeV)
61	Ecutoffh	
62	Ecutoffm	
63	Ecutoffe	

	Variable	Description
64	Ecutoffg	
65	NFLAIN	
66	NFLDIF	
67	NFLPI0	
68	NFLPIF	
69	NFLCHE	
70	NFRAGM	
71	Bx	
72	By	
73	EGS4yn	
74	NKGyn	
75	GHEISHAyn	
76	VENUSyn	
77	CERENKOVyn	
78	NEUTRINOyn	
79	HORIZONTyn	
80	COMPUTER	
81	ThetaMin	Minimum Theta of primaries (deg)
82	ThetaMax	Maximum Theta of primaries (deg)
83	PhiMin	Minimum Phi of primaries (deg)
84	PhiMax	Maximum Phi of primaries (deg)
85	CBunchSize	
86	CDetInX	
87	CDetInY	
88	CSpacInX	
89	CSpacInY	
90	CLenInX	
91	CLenInY	
92	COutput	
93	AngleNorthX	
94	MuonInfo	
95	StepLength	
96	CWaveLower	Wavelength lower limit (nm)
97	CWaveUpper	Wavelength upper limit (nm)
98	Multipl	
99 to 138	CorePos[2][20]	Core positions of randomized shower
139 to 140	SIBYLL[2]	
141 to 142	QGSJET[2]	
143 to 144	DPMJET[2]	
145	VENUS_cross	
146	mu_mult_scat	
147	NKG_range	
148 to 149	EFRCTHN[2]	
150 to 151	WMAX[2]	

	Variable	Description
152	rthin_rmax	
153 to 154	viewcone_angles[2]	Inner and outer angles of Corsika's VIEWCONE option.
155	telescopePhi	Telescope azimuth (rad). Measured from South, counter-clockwise
156	telescopeTheta	Telescope zenith angle (rad)
157	TimeFirst	Arrival time on camera of first photon (ns)
158	TimeLast	Arrival time on camera of last photon (ns)
		6 next variables: CORSIKA longitudinal particle fit parameters (see CORSIKA manual for precise meaning and units)
159	longi_Nmax	Numer of charged particles at maximum
160	longi_t0	Atmospheric depth of shower starting point (N=0)
161	longi_tmax	Atmospheric depth of shower maximum (g/cm^2)
162	longi_a	
163	longi_b	For **longi_a**, **longi_b**, **longi_c**, see CORSIKA manual
164	longi_c	
165	longi_chi2	χ^2/dof of the fit
166	CORSIKAPhs	Original photons written by *CORSIKA*
167	AtmAbsPhs	Photons absorbed by the atmosphere
168	MirrAbsPhs	Photons absorbed by the mirror
169	OutOfMirrPhs	Photons outside the mirror
170	BlackSpotPhs	Photons lost in the "black spot"
171	OutOfChamPhs	Photons outside the camera
172	CPhotons	Photons reaching the camera
173	elec_cph_fraction	Fraction of C-photons produced by electrons
174	muon_cph_fraction	Fraction of C-photons produced by muons
175	other_cph_fraction	Fraction of C-photons produced by electrons
176 to 182	dummy[7]	not used

The event header is followed by 8-word blocks, one for each Cherenkov photon that reaches the camera. A photon block contains the following variables (as 4-byte float numbers):

Variable	Description
w	100000 × Particle_type + 1000 × n + Wavelength (nm) Particle_type indicates what particle produced the C-photon. Its value is 2 for electrons and positrons, and the standard codes of Corsika for the rest. n is the index from 1 to ICERML (see Corsika manual) for the case in which each Corsika shower is used more than once (normally, in MMCS will be just 1).
x, y	Impact point in camera coordinates (cm)
u, v	Director cosines of down-going versor indicating the photon direction
t	Arrival time on camera (ns), measured from the time of first interaction of the primary
h	Production height (cm), measured above sea level on the vertical of the telescope location (it is not the *true* height which would be measured on the vertical of the emitting particle!)
phi	Incidence angle with respect to camera plane (rad)

After the last photon block of an event there is a carriage return followed by the ascii flag indicating the event end, and then two more carriage returns before the ascii flag of the beginning of the next event, and so on:

```
$END---EVENT$$START-EVENT$Event_header....$END---EVENT$$END-----RUN$$START---RUN$...
```

The flag "END-----RUN" appears after the last event in a run (that is, the last event processed of each of the input cer files). After the last processed run, an end of file flag is written:

```
...$END---EVENT$$END-----RUN$$END----FILE$
```

Finally, after this flag an "ascii tail" has been attached to the file: it consists of the ascii files *magic.def*, *axisdev.dat* and *reflectivity.dat* one after the other, separated by carriage returns:

```
$magic.def$axisdev.dat$reflectivity.dat
```

In this way all the relevant parameters used to produce the output are kept together with the reflected events.

Appendix A : atmospheric absorption

The simulation of the absorption of Cherenkov light in the atmosphere has been included in the *Reflector* program because this feature was not yet available in the first versions of CORSIKA used within the MAGIC collaboration. In the latest CORSIKA versions, the atmospheric absorption has been included as an option, but it is not compatible with the simulation of a curved atmosphere [3], and hence we have kept this step as a part of our reflector simulation. This appendix describes how the atmospheric absorption is implemented in the program when the option ATM_CORSIKA (see section 7.1) is chosen.

The geometry of the problem is sketched in figure 14. A Cherenkov photon is emitted in point A and travels towards the telescope placed at B. At any moment, the height h of the photon above sea level is related to the distance L between the photon and the telescope through

$$(R+h)^2 = (R+h_1)^2 + L^2 + 2L\ (R+h_1)\ \cos\theta \tag{1}$$

, where R is the Earth radius, h_1 the height (a.s.l.) of the observation level and θ is the zenith angle of the photon trajectory measured at the telescope site. The Cherenkov output of CORSIKA contains for each photon the height h_C of the emission point A, measured along the vertical of the observer. The *true vertical height* h_2 of the emission point can be obtained by replacing L by $(h_C - h_1)/\cos\theta$ in equation (1).

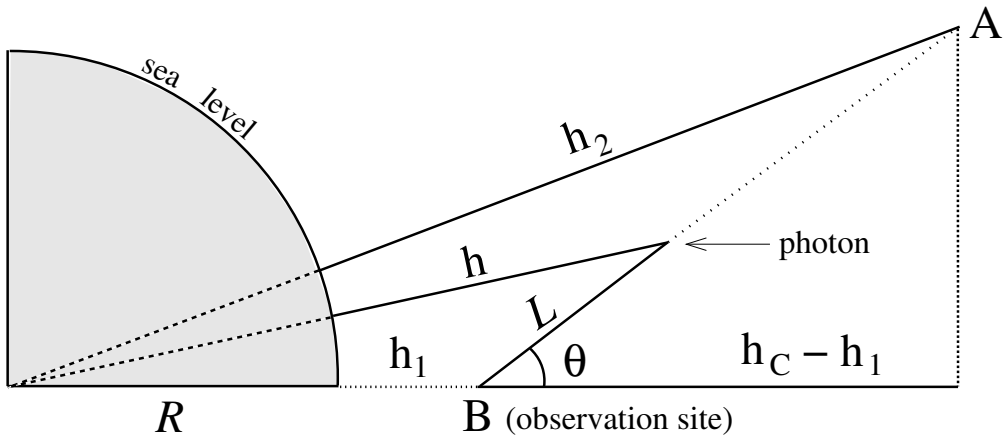

Figure 14: Calculation of the true vertical height h_2 of the emission point of a Cherenkov photon (point A), and the optical path traversed down to the telescope (point B).

The optical path $I(\theta, h_1, h_2)$ traversed by the photon can be calculated integrating the air density along the trajectory $\overline{\mathrm{AB}}$. For the case $h/R \ll 1$, we can drop in (1) the terms in $(h/R)^2$ and smaller, and then solve for L. Deriving now with respect to h, we have:

$$\frac{dL}{dh} \simeq \sqrt{\frac{R}{2\ (h-h_1) + R\ \cos^2\theta}} \qquad \text{for} \qquad \frac{h}{R} \ll 1 \tag{2}$$

A.1 Rayleigh scattering

Rayleigh scattering is the scattering of light by particles smaller than its wavelength. These are in our case the air molecules. The transmission coefficient due to Rayleigh scattering is a strong function of

the wavelength λ:

$$T_{\mathrm{Rayl}}(\lambda) = \exp\left[-\frac{I(\theta, h_1, h_2)}{x_R}\left(\frac{400\ \mathrm{nm}}{\lambda}\right)^4\right] \tag{3}$$

Here $I(\theta, h_1, h_2)$ is the optical path (in g/cm^2) traversed between points A and B, and $x_R = 2970$ g/cm^2 is the mean free path of the Rayleigh scattering at $\lambda = 400$ nm.

A convenient way of expressing the optical path is the following:

$$I\ (\theta, h_1, h_2) = (x_1 - x_2) \cdot \mathcal{AM}\ (\theta, h_1, h_2) \tag{4}$$

Here $x_{i=1,2}$ is the mass overburden of the atmosphere above a height h_i (in the vertical direction) and $\mathcal{AM}$ is the so called *air mass*[2], defined as

$$\mathcal{AM} \equiv \frac{I\ (\theta, h_1, h_2)}{I\ (0^\circ, h_1, h_2)} \tag{5}$$

, which is a function mainly of the zenith angle θ (see fig. 15). In our simulation, for the calculation of the mass overburden x_i we have used the U.S. standard atmosphere parametrized by J. Linsley [3], the same we used in Corsika for the shower development simulation. It consists of five layers: in the lower four the density decreases exponentially with height, and in the upper one the mass overburden cecreases linearly until it vanishes at $h = 112.8$ km.

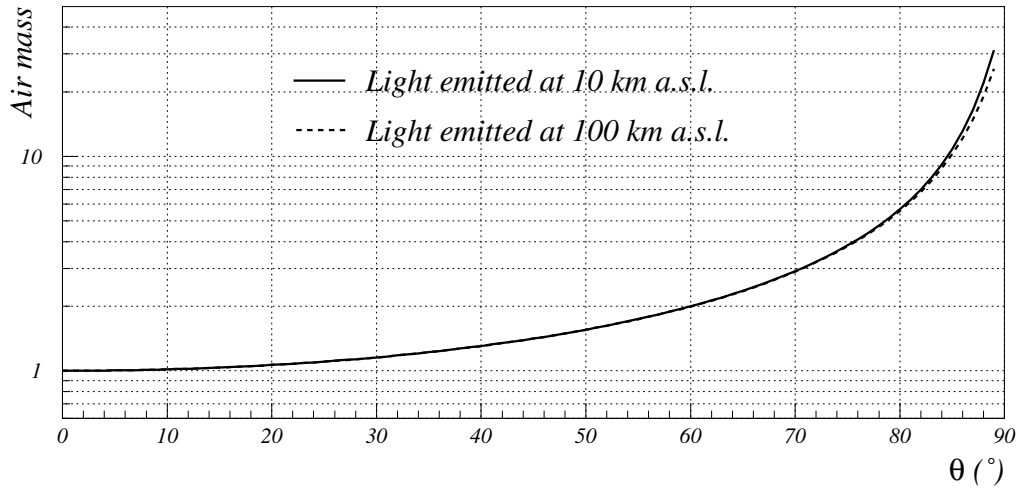

Figure 15: Dependence with zenith angle of the air mass as defined in the text. The air mass has been calculated for an exponential atmosphere with scale height $H = 7.4$ km, for the observation level of MAGIC (2.2 km a.s.l.), and for light coming from $h_2 = 10$ and at 100 km a.s.l. As we can see the dependence with the emission height h_2 is very small.

For the estimate of $\mathcal{AM}$, a simpler atmospheric model has been used, in which the vertical density profile is described by a single exponential: $\rho = \rho_0\, e^{-h/H}$ with scale height $H = 7.4$ km. This simplifies

[2] If we set $h_2 = \infty$, we have the usual definition of *air mass* in optical astronomy.

the calculations, and is accurate enough for our purposes. Using (2) the optical path $I(\theta, h_1, h_2)$ can then be obtained approximately as:

$$I(\theta, h_1, h_2) = \int_A^B \rho\,(h)\,\frac{dL}{dh}\,dh \simeq \sqrt{\frac{R}{2}}\int_{h_1}^{h_2} \frac{\rho_0\, e^{-h/H}}{\sqrt{h - h_1 + \frac{1}{2}\,R\,\cos^2\theta}}\,dh \tag{6}$$

and finally:

$$\mathcal{AM} \simeq e^{-\frac{R\sin^2\theta}{2H}} \cdot \frac{\mathrm{erfc}\,(\sqrt{\frac{R\cos^2\theta}{2H}}) \;-\; \mathrm{erfc}\,(\sqrt{\frac{2(h_2-h_1)+R\cos^2\theta}{2H}})}{\mathrm{erfc}\,(\sqrt{\frac{R}{2H}}) \;-\; \mathrm{erfc}\,(\sqrt{\frac{2(h_2-h_1)+R}{2H}})} \tag{7}$$

where we have used the complementary error function $\mathrm{erfc}\,(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt$. From (7), $\mathcal{AM}$ can be readily evaluated for any value of θ, h_1 and h_2. In fig. 16 the resulting Rayleigh transmission coefficient T_{Rayl} finally obtained from (3) is plotted versus the zenith angle for three wavelengths.

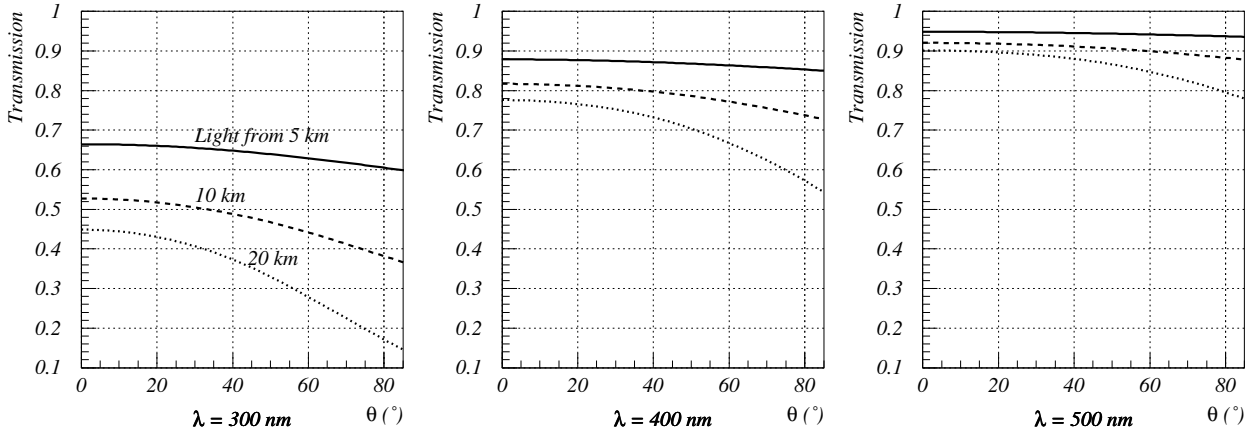

Figure 16: Rayleigh transmission coefficient as a function of zenith angle for three different wavelengths. The solid, dashed and dotted lines correspond respectively to light coming from 5, 10 and 20 km distance from the telescope.

A.2 Mie scattering

Cherenkov light also suffers Mie scattering through interaction with small dust particles suspended in the air (aerosols), whose size is comparable to the wavelength of the light. In our simulation of the attenuation due to aerosols we have used the model proposed by Elterman [1, 2], which considers an aerosol number density N_p which (roughly) decreases exponentially up to 10 km a.s.l. with scale height $H \simeq 1.2$ km, followed by a more tenuous layer between 10 and 30 km (see fig. 17a). In this model, the aerosol size distribution is considered to be unchanged with altitude.

Measured values of the aerosol attenuation coefficients at sea level $\beta_p(0)$ for different wavelengths [2] are shown in figure 17b. To obtain the attenuation coefficient at a given height h, we simply do $\beta_p(h,\lambda) = \beta_p(0,\lambda)\cdot N_p(h)/N_p(0)$. In the Elterman model the aerosol transmission coefficient for the trajectory from A to B depicted in figure 14 would then be:

$$T_{\mathrm{Mie}}(\lambda) = e^{-\tau_{mie}} \quad \text{, with} \quad \tau_{mie}(h_1, h_2, \theta, \lambda) = \frac{\beta_p(0,\lambda)}{N_p(0)}\int_{h_1}^{h_2} N_p(h)\;\frac{dL}{dh}\,dh \tag{8}$$

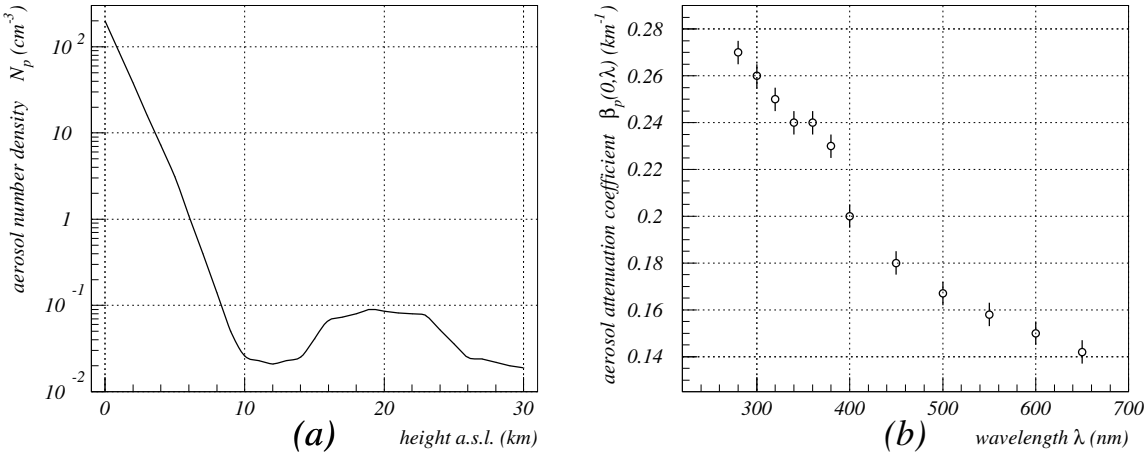

Figure 17: Aerosol model by Elterman: in (a), number density of aerosols as a function of height above sea level; in (b), aerosol attenuation coefficient at sea level as a function of wavelength.

Here τ_{mie} is the aerosol optical depth of the path from A to B. Given that the aerosol density distribution is not a simple exponential, we have to do the integral in (8) numerically. The integral depends on h_2 and on θ, through dL/dh (it also depends on h_1, but this is the observation level and therefore fixed). In this case we use the exact expression for dL/dh which can be obtained from (1). At the beginning of each simulation run we calculate and store the results of the integral for values of θ between 0 and 90° (in steps of 1°), and of h_2 from h_1 up to 30 km (in steps of 100 m). To do the integral a linear interpolation has been used to obtain the value of N_p for any height. During the simulation of each Cherenkov photon, we get the corresponding precalculated value of the integral and deduce $T_{\rm Mie}$ from expression (8).

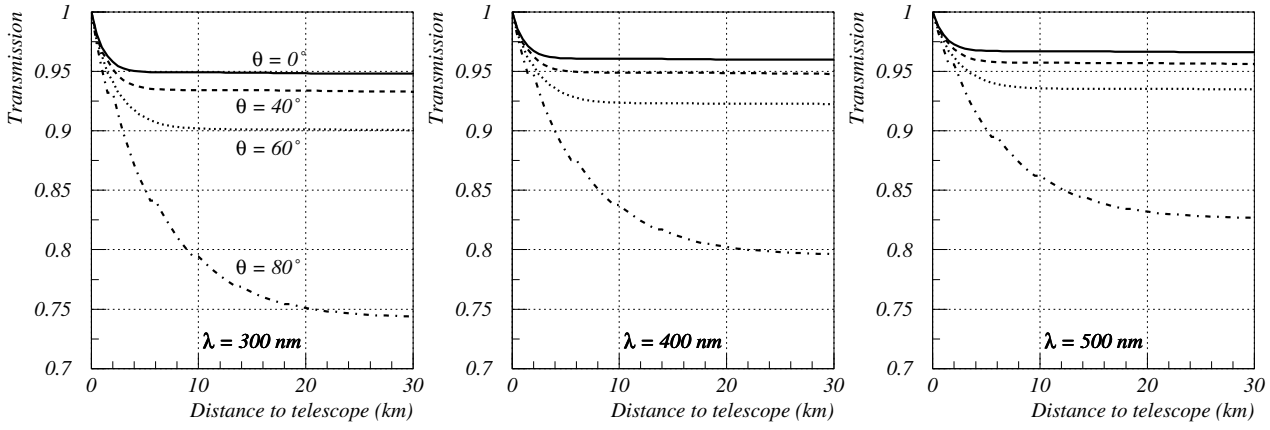

Figure 18: Aerosol transmission coefficient for three different wavelengths as a function of the distance between the photon emission point and the telescope. Plots for four different zenith angles between 0 and 80 degrees are shown.

Since in this model the aerosols are concentrated mainly at very low altitude, the transmission coefficient is more or less constant above a certain height (which depends on θ), as can be seen in fig. 18. For instance, for vertically incident 300 nm light emitted higher than 4 km above the telescope, the Mie transmission is about 0.95.

A.3 Absorption by Ozone

The absorption of Cherenkov light by Ozone has been implemented following also the Elterman standard atmosphere [2]. The coefficient for ozone absorption is given by

$$\beta_3(h,\lambda) = A_v(\lambda) \cdot D_3(h) \tag{9}$$

, where $A_v(\lambda)$ is the Vigroux [4] ozone absorption coefficient (cm^{-1}) and $D_3(h)$ is the ozone concentration ($\mathrm{cm\ km}^{-1}$) according to Elterman. The transmission coefficient of Ozone in the path $\overline{\mathrm{AB}}$ is then:

$$T_{\mathrm{Ozone}}(\lambda) = e^{-\tau_{oz}} \quad \text{, with} \quad \tau_{oz}(h_1,h_2,\theta,\lambda) = A_v(\lambda) \int_{h_1}^{h_2} D_3(h)\ \frac{dL}{dh}\, dh \tag{10}$$

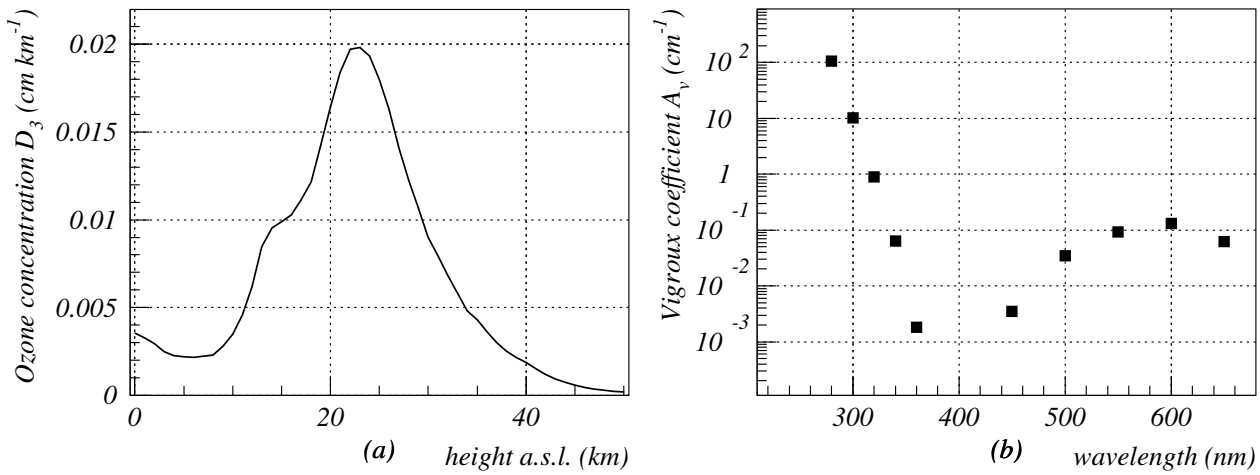

Figure 19: Ozone concentration vertical profile (a) and Vigroux coefficients for Ozone absorption (b). The Vigroux coefficients for $\lambda = 380$ and 400 nm are zero.

Once again, like in the case of Mie scattering, the optical depth τ_{oz} is the product of a factor which depends on λ and an integral which depends on h_1, h_2 and θ. We proceed in the same way as before, precalculating the values of the integral in steps of $\Delta\theta = 1°$ and $\Delta h = 100$ m, up to a height of 50 km a.s.l. (where the ozone concentration becomes negligible), and then reading the appropriate values for every simulated photon.

Finally, the overall atmospheric transmission coefficient is calculated as

$$T_{total} = T_{Ray} \cdot T_{Mie} \cdot T_{Ozone} \tag{11}$$

In figure 20 the atmospheric transmission as a function of the distance to the telescope for $\theta = 0°$ is shown. Ozone absorption turns out to be dominant below 320 nm, while Rayleigh scattering is the main cause of the loss of photons at longer wavelengths.

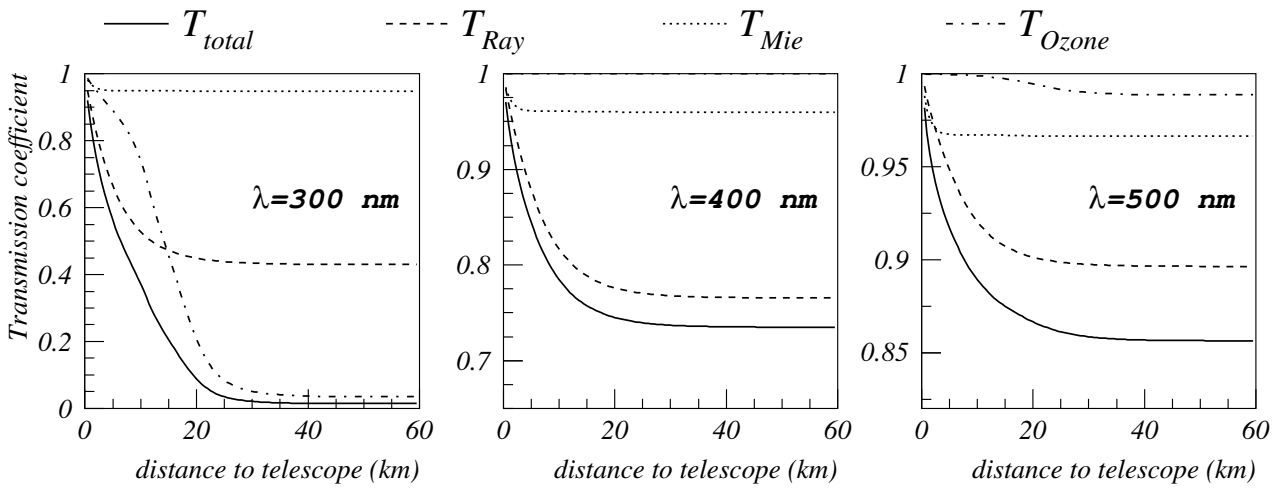

Figure 20: Total transmission coefficient for vertically incident light as a function of the distance between the emission point and the telescope. The contributions of absorption by ozone and of Rayleigh and Mie scattering are also shown for comparison.

Appendix B : files in the reflector package

The list of all Reflector files follows.

```
MagicProgs/Simulation/Detector/Reflector_0.6/Changelog
MagicProgs/Simulation/Detector/Reflector_0.6/Makefile
MagicProgs/Simulation/Detector/Reflector_0.6/atm.c
MagicProgs/Simulation/Detector/Reflector_0.6/atm.h
MagicProgs/Simulation/Detector/Reflector_0.6/attach.c
MagicProgs/Simulation/Detector/Reflector_0.6/attenu.f
MagicProgs/Simulation/Detector/Reflector_0.6/config.mk.linux
MagicProgs/Simulation/Detector/Reflector_0.6/config.mk.linux-gnu
MagicProgs/Simulation/Detector/Reflector_0.6/config.mk.osf1
MagicProgs/Simulation/Detector/Reflector_0.6/diag.c
MagicProgs/Simulation/Detector/Reflector_0.6/diag.h
MagicProgs/Simulation/Detector/Reflector_0.6/geometry.c
MagicProgs/Simulation/Detector/Reflector_0.6/geometry.h
MagicProgs/Simulation/Detector/Reflector_0.6/header.c
MagicProgs/Simulation/Detector/Reflector_0.6/header.h
MagicProgs/Simulation/Detector/Reflector_0.6/init.c
MagicProgs/Simulation/Detector/Reflector_0.6/init.h
MagicProgs/Simulation/Detector/Reflector_0.6/input.card
MagicProgs/Simulation/Detector/Reflector_0.6/lagrange.h
MagicProgs/Simulation/Detector/Reflector_0.6/parms.c
MagicProgs/Simulation/Detector/Reflector_0.6/parms.h
MagicProgs/Simulation/Detector/Reflector_0.6/ph2cph.c
MagicProgs/Simulation/Detector/Reflector_0.6/refl-install
```

```
MagicProgs/Simulation/Detector/Reflector_0.6/reflector.c
MagicProgs/Simulation/Detector/Reflector_0.6/version.h

MagicProgs/Simulation/Detector/Reflector_0.6/doc/Tdas0211.ps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/Tdas0211.tex
MagicProgs/Simulation/Detector/Reflector_0.6/doc/magic-tdas.sty
MagicProgs/Simulation/Detector/Reflector_0.6/doc/magiclogo.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/absorplot.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/aerosol.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/airmass.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/atmoscheme.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/colimation.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/coma.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/coma_star.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/coorsystems.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/evcompare.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/mie.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/ozone.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/parabola.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/rayleigh.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/refl06images.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/reflec.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/spot10kmf1694.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/spot10kmf1700.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/spot_inf_f1697.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/telecoor.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/timing.eps

MagicProgs/Simulation/Detector/Reflector_0.6/tester/Makefile
MagicProgs/Simulation/Detector/Reflector_0.6/tester/cermaker.c

MagicProgs/Simulation/Detector/Data/axisdev.dat
MagicProgs/Simulation/Detector/Data/magic.def
MagicProgs/Simulation/Detector/Data/reflectivity.dat

MagicProgs/Simulation/Detector/lib/libranlib.a.osf1
MagicProgs/Simulation/Detector/lib/libranlib.a.linux
```

References

[1] L. Elterman, Applied Optics Vol. 3, No. 6 (1964) 745.

[2] L. Elterman, R.B. Toolin, S.L. Valley (editor), Handbook of geophysics and space environments, McGraw-Hill, N.Y. (1965).

[3] D. Heck and J. Knapp, EAS Simulation with CORSIKA: A User's Manual, 2002.

[4] E. Vigroux, Contributions à l'tude exprimentale de l'absorption de l'ozone, Annales de Physique, v. 8 (1953) 709.