MAGIC-TDAS 02-11
021009/AMoralejo

The Reflector simulation program v.0.6

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

November 12, 2002

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. An important bug regarding the ray-tracing routine has been corrected in this new version.

Contents

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. This bug, fixed in this version, is 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 The 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 above the telescope would be focused on a plane at $\simeq$ 1700 cm from the dish (a distance measured along the telescope axis).

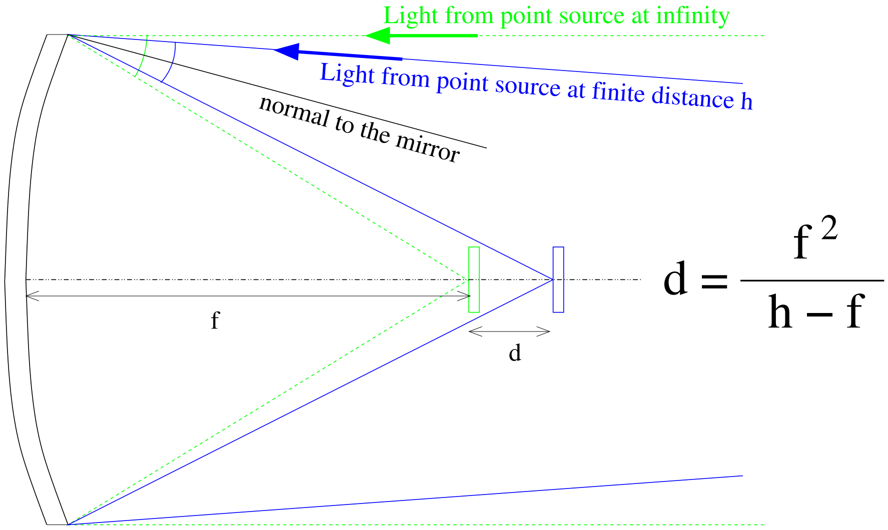

Figure 1: To focus an object at a finite distance h, the camera plane must be moved away from the mirror a distance d, given by the formula on the right. For MAGIC, the shift is of around 3 cm for a source located 10 km above the telescope.

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 our 17 m 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 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 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

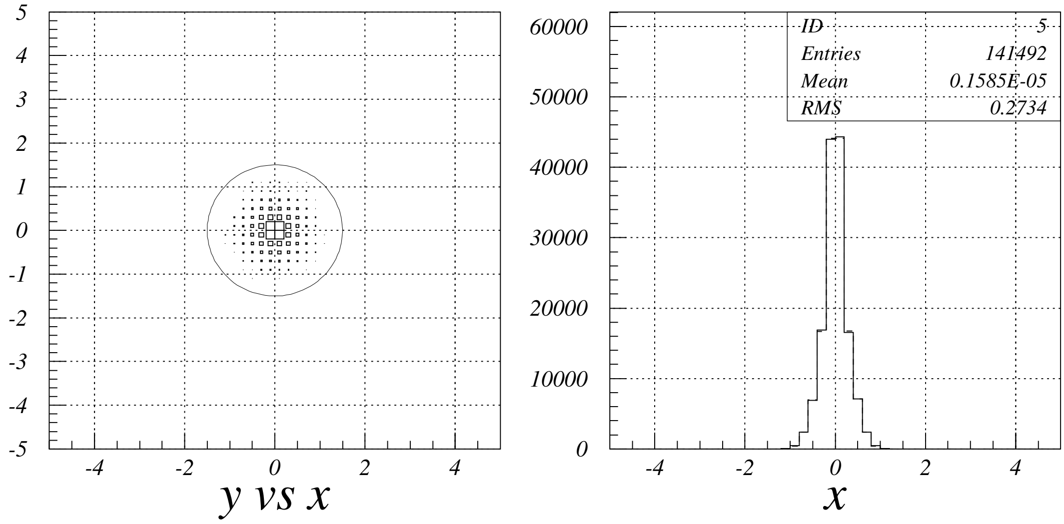

Figure 2: Image of a point like source placed at infinity with Reflector 0.5 (units of x and y are cm). Camera plane placed at 1697 cm from the mirror. The circle on the left indicates (small) pixel size. On the right side, projections on x and y of the spot.

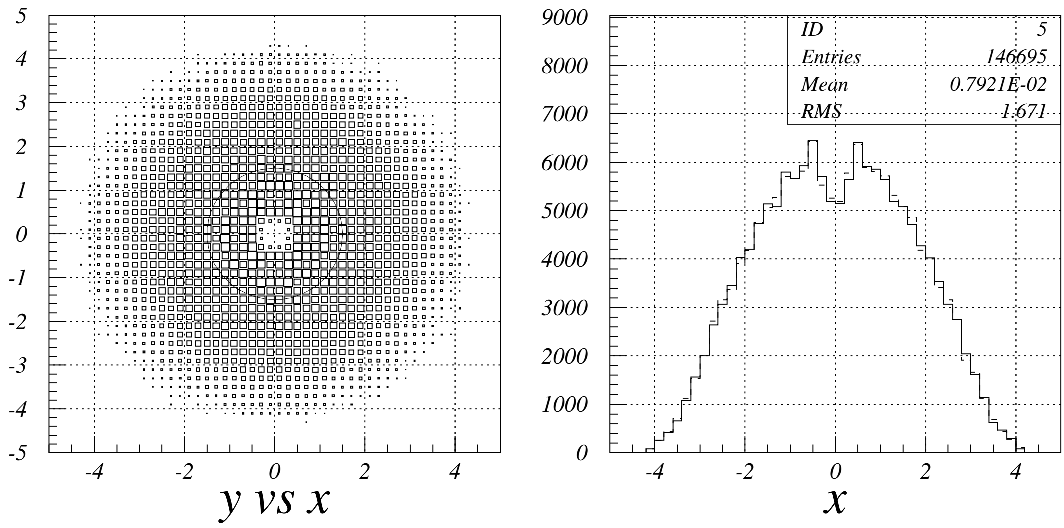

Figure 3: Image of a point like source placed 10 km above the telescope with Reflector 0.5 (units of x and y are cm). Camera plane placed at f = 1700 cm from the mirror. The fact that the spot is so large indicates a problem in the simulation, since with this camera position the telescope should be focused 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.

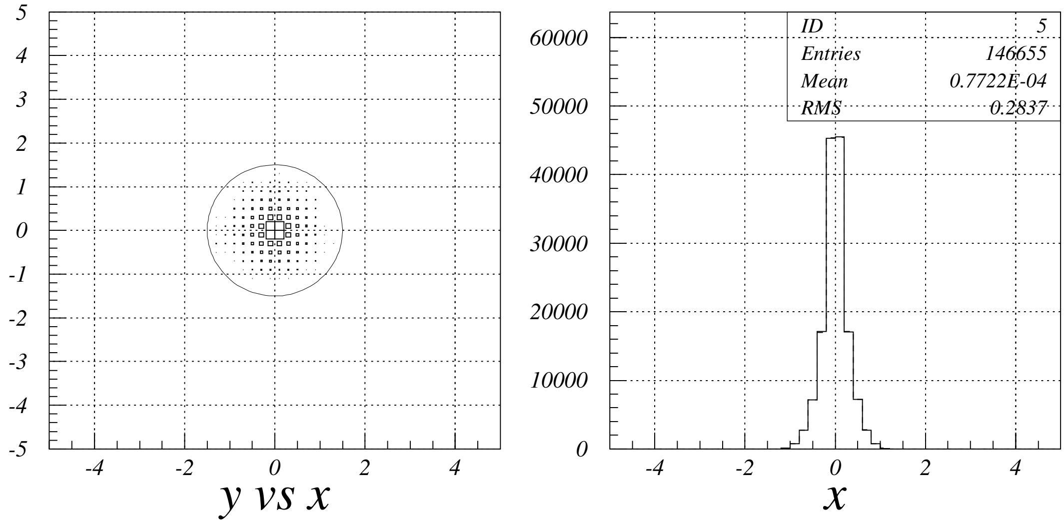

Figure 4: Image of a point like source placed 10 km above the telescope with Reflector 0.5 (units of x and y are cm). Camera plane placed at f = 1694 cm from the mirror. Now the spot is small, but the camera plane is not at the expected position (see text).

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 parameters 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). 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 $w = -\sqrt{u^2 + v^2}$, with a minus sign. However, in *ph2cph.c* we found exactly the 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]);
```

camera at 1700 cm camera at 1694 cm

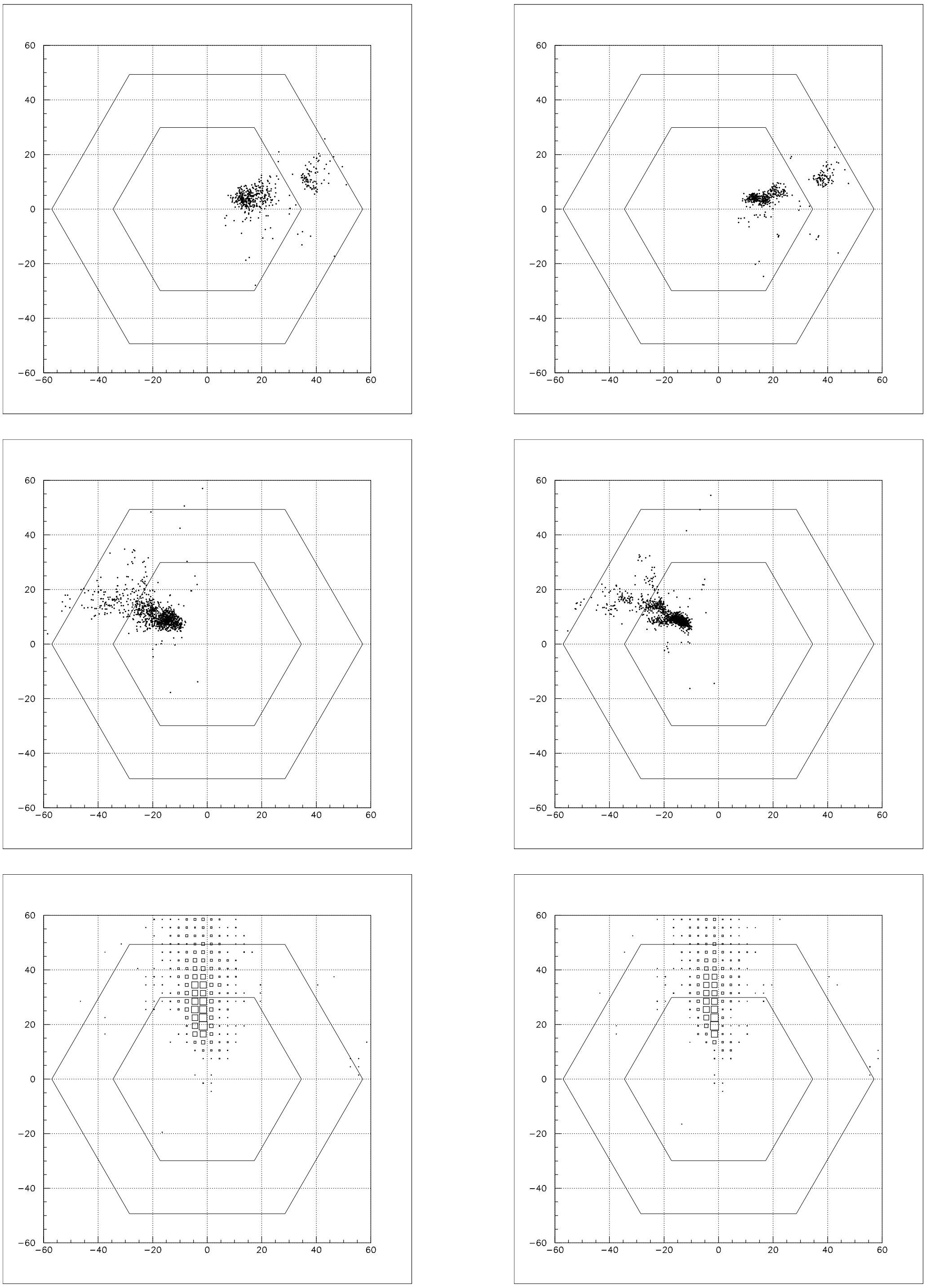

Figure 5: Gamma shower images obtained with Reflector 0.5 placing the camera at 1700 cm (left) and 1694 cm (right) from the mirror dish. Primary gammas at $\theta = 10°$, E = 16, 46 and 232 GeV.

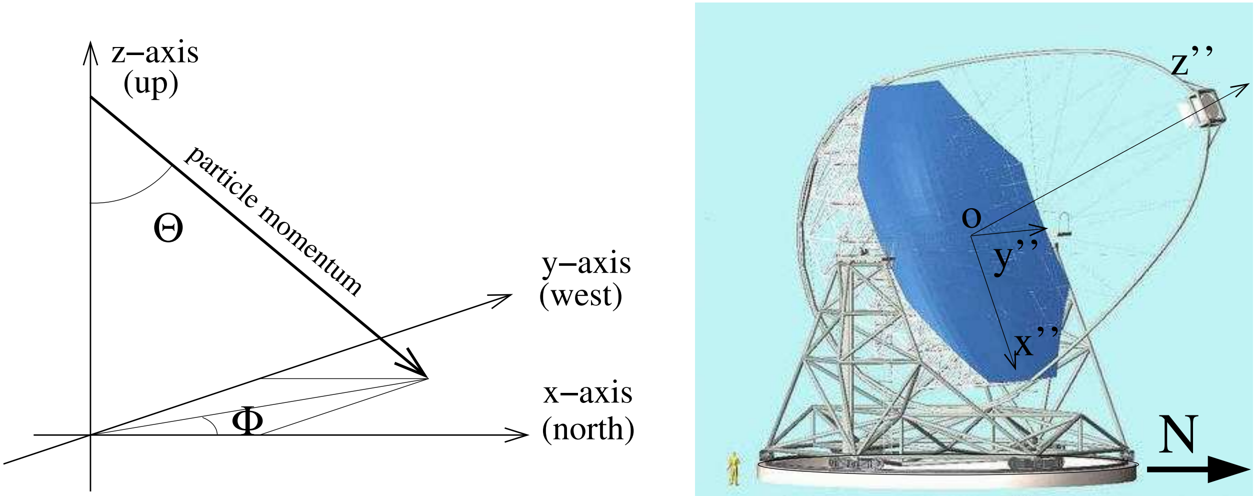

Figure 6: Coordinate systems of Corsika and MAGIC.

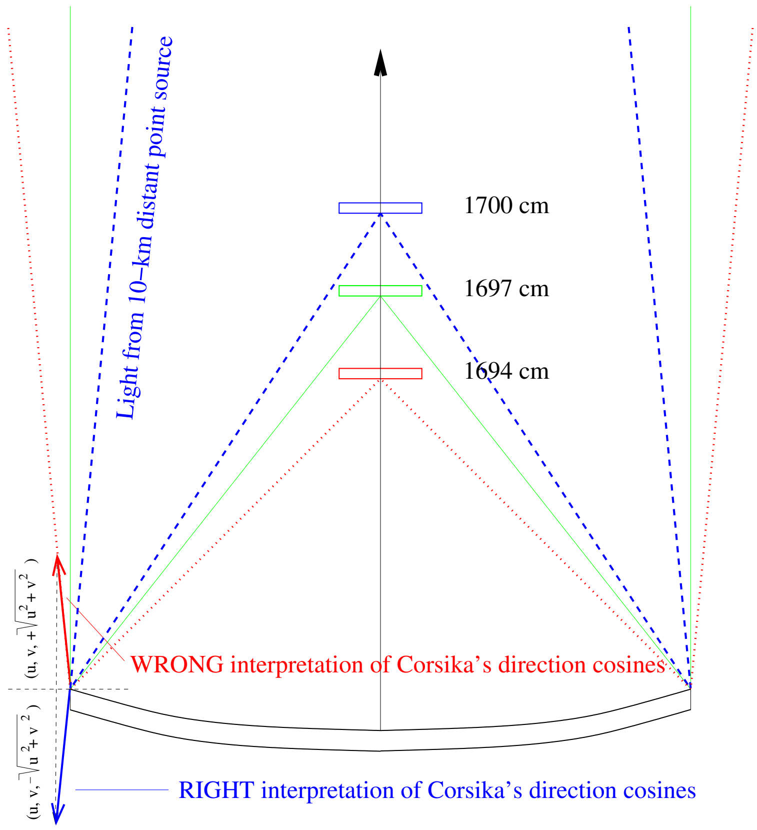

Figure 7: Misunderstanding of photon direction cosines from Corsika by Reflector v 0.5 and older versions.

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.

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. 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. 8. 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*.

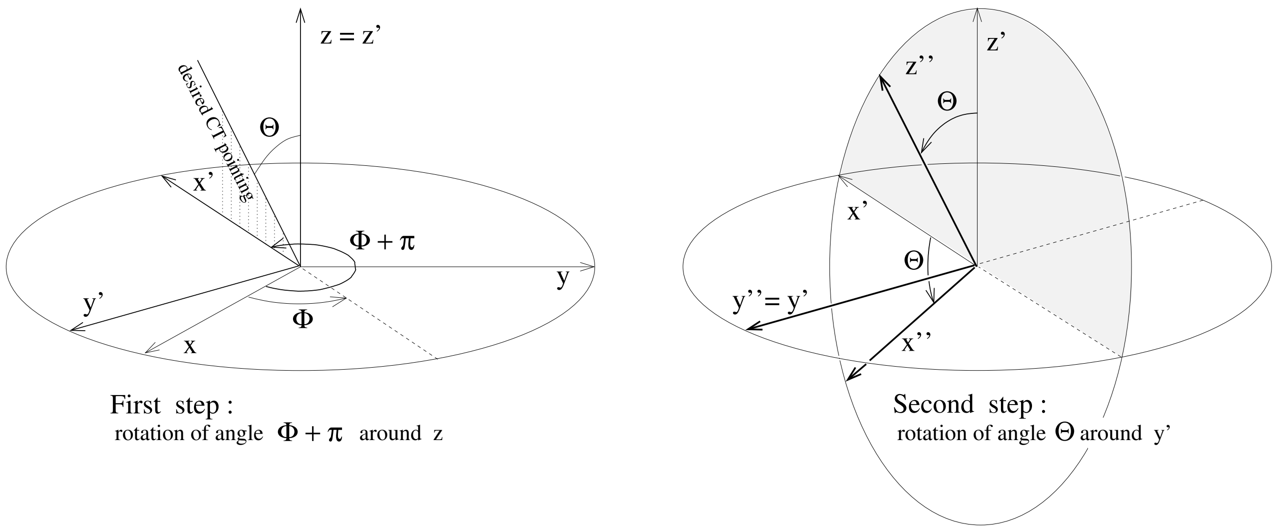

Figure 8: Transformation of coordinates between the Corsika and MAGIC coordinate systems.

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 (which must have been an early version of the present one) 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. 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. 10) has a less dramatic effect than the defocusing which the bug was producing.

3.2 Performance of the new version

In figure 9 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. 11. 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 10 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.

3.3 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.

3.4 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.

3.5 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): 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

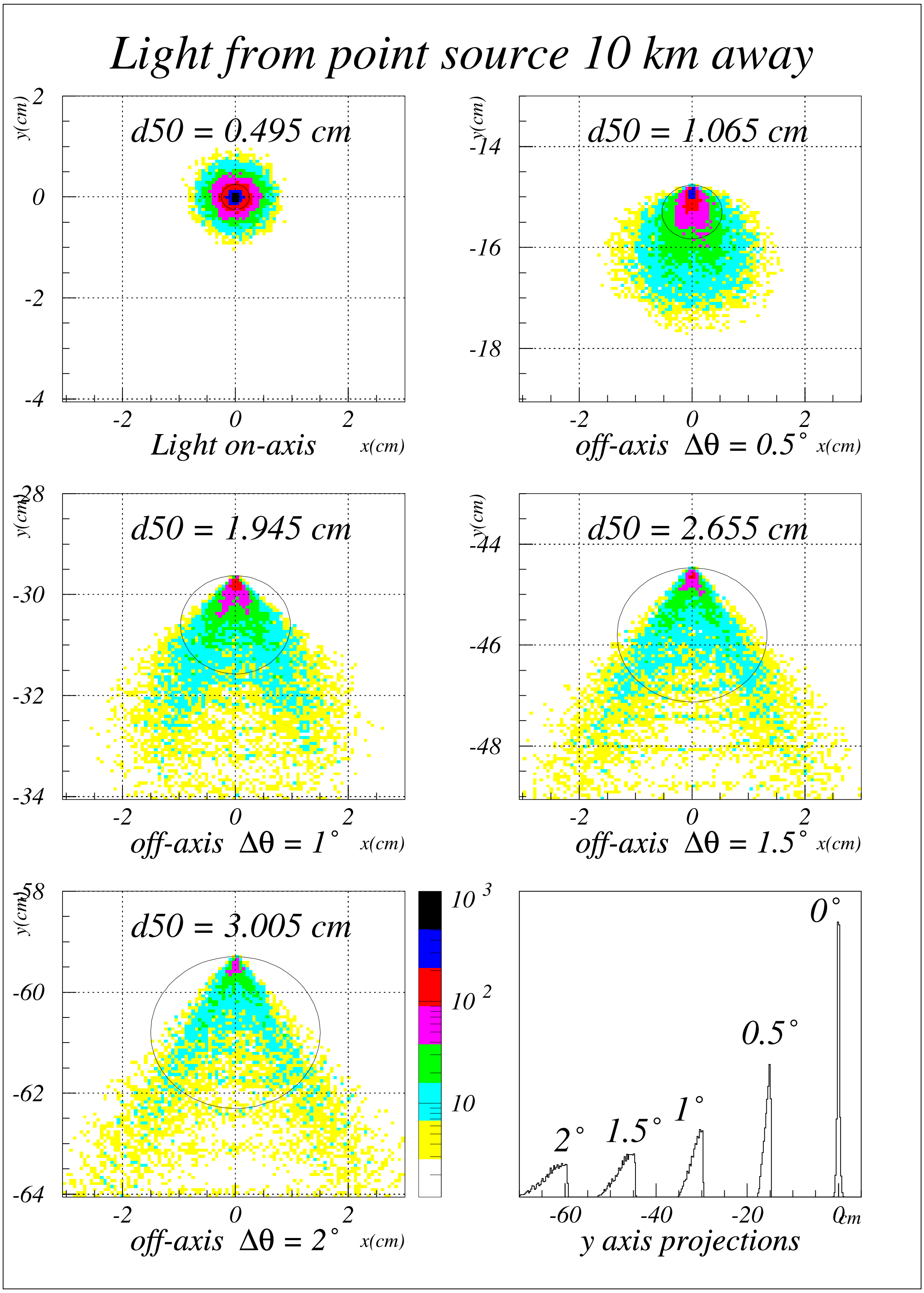

Figure 9: Reflector 0.6. Images of a point-like source at 10 km from the telescope for different incident angles (from on-axis to 2 degrees off-axis). The quantity d50 indicates the diameter of a circle (plotted) containing 50% of the reflected light. Note that the z-axis scale is logarithmic, and the same in the first five plots. The last plot shows the x-axis projections in linear scale.

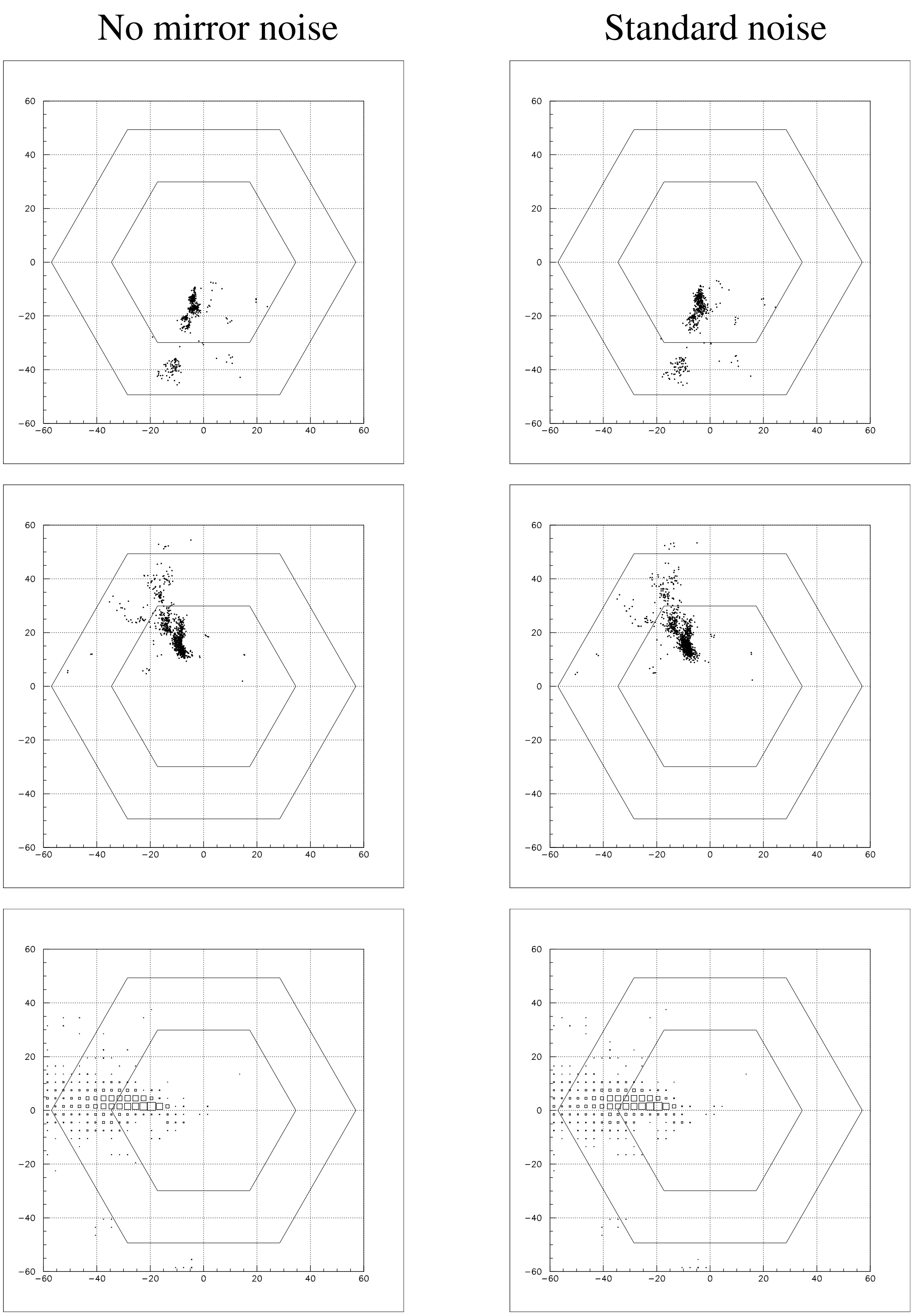

Figure 10: Reflector 0.6. Images of the three gamma showers shown in fig. 5, without noise added in the reflection (left) and with the standard noise (right) as described in section 2. Note that the orientation of the images has changed as a result of the introduction of a new camera coordinate system (see page 8).

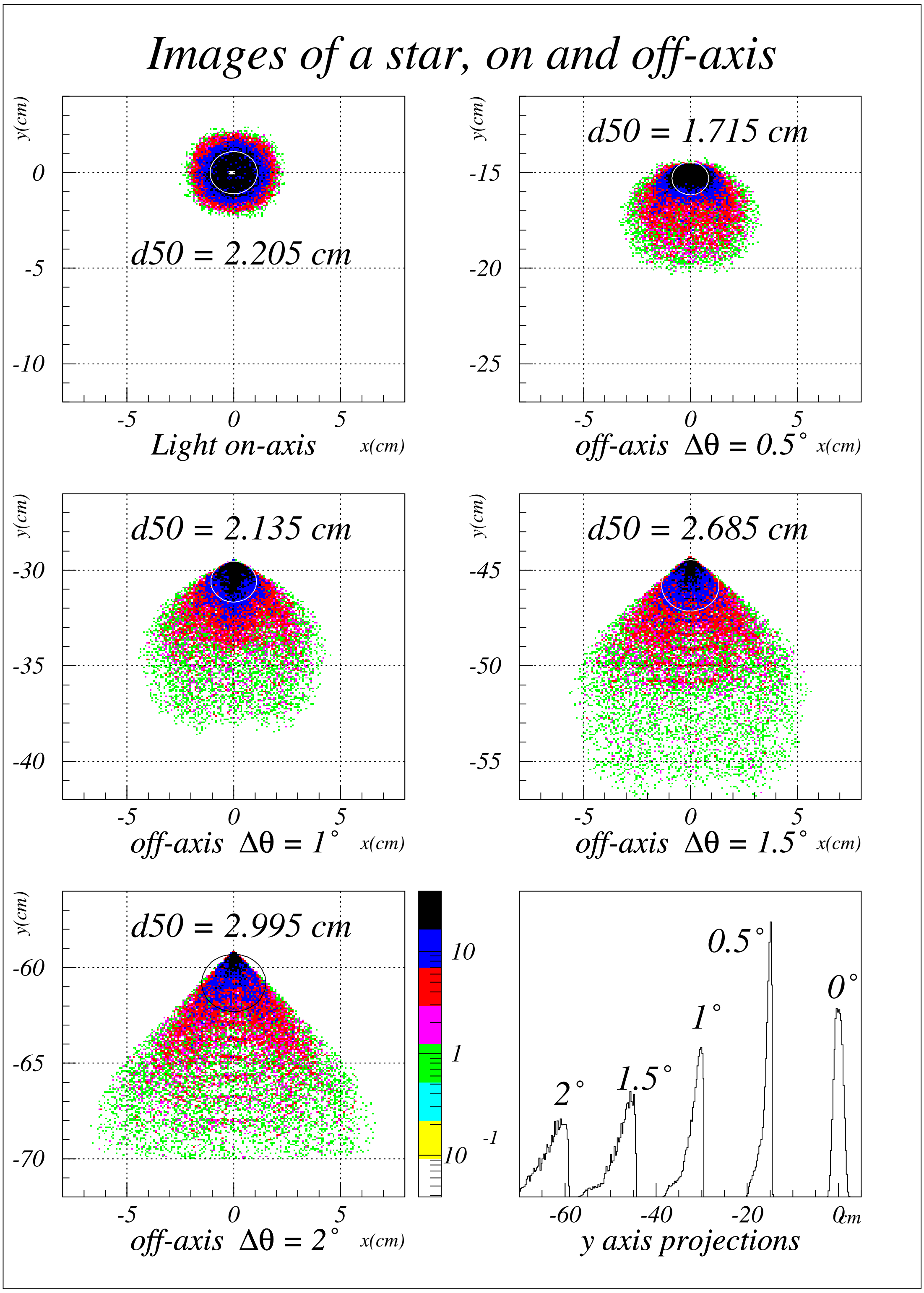

Figure 11: Reflector 0.6. Images of a star. The quantity d50 indicates the diameter of a circle (plotted) containing 50% of the reflected light.

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

Make symbolic links running the script: refl-install

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

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 are usually in the **MagicProgs/Simulation/Detector/Data/** 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 ../Data/magic.def The ct_file statement defines where the program can find the telescope characteristics. The path in the example above is correct to run reflector in MagicProgs/Simulation/Detector/Reflector_0.6/ directory. If you want to run it in a different directory you have to modify the path accordingly.

atm_model ATM_CORSIKA The atm_model statement says to the program what kind of atmospheric absorption model to use. Possible choices are: ATM_CORSIKA, ATM_ISOTHERMAL, ATM_90PERCENT and ATM_NOATMOSPHERE.

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 output file begins with two ascii lines:

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

After the `START---RUN` flag there is a carriage return, and then the 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 4-byte real numbers except the first, which is a 4 character string containing the run header ascii label from Corsika:

Variable	Description
ASCII Label	'RUNH'
RunNumber	
date	
Corsika_version	
NumObsLev	
HeightLev[10]	
SlopeSpec	
ELowLim	
EUppLim	
EGS4_flag	
NKG_flag	
Ecutoffh	
Ecutoffe	
Ecutoffg	
C[50]	
wobble_mode	Wobble mode with which the reflector was run (TDAS 01-05)
atmospheric_model	Atmospheric model used for the absorption simulation 0 = no atmosphere; 1 = atm_90percent; 2 = atm_isothermal; 3 = atm_corsika.
dummy1[18]	not used
CKA[40]	
CETA[5]	
CSTRBA[11]	
dummy2[104]	not used
AATM[5]	
BATM[5]	
CATM[5]	
NFL[4]	

Then there comes a "`START-EVENT`" flag, followed by a carriage return 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
ASCII label	'EVTH'
EvtNumber	Event Number
PrimaryID	Primary particle identification code
Etotal	Primary particle total energy (GeV)
Thick0	CORSIKA's starting altitude in g/cm2
FirstTarget	CORSIKA's number of first target if fixed
zFirstInt	Height of first interaction in cm
p[3]	Primary particle momentum in x,y,-z directions (GeV)
Theta	Primary particle zenith angle (rad)
Phi	Primary particle azimuth angle (rad)
NumRndSeq	Number of different CORSIKA random sequences (max. 10)
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.
RunNumber	Run number
DateRun	Date of run yymmdd
Corsika_version	Version of *CORSIKA*
NumObsLev	Number of observation levels (should be always 1 for us)
HeightLev[10]	Height of observation levels in cm
SlopeSpec	Energy spectrum slope
ELowLim	Energy lower limit (GeV)
EUppLim	Energy upper limit (GeV)
Ecutoffh	
Ecutoffm	
Ecutoffe	
Ecutoffg	
NFLAIN	
NFLDIF	
NFLPI0	
NFLPIF	
NFLCHE	
NFRAGM	
Bx	
By	
EGS4yn	
NKGyn	
GHEISHAyn	
VENUSyn	

Variable	Description
CERENKOVyn	
NEUTRINOyn	
HORIZONTyn	
COMPUTER	
ThetaMin	Minimum Theta of primaries (deg)
ThetaMax	Maximum Theta of primaries (deg)
PhiMin	Minimum Phi of primaries (deg)
PhiMax	Maximum Phi of primaries (deg)
CBunchSize	
CDetInX	
CDetInY	
CSpacInX	
CSpacInY	
CLenInX	
CLenInY	
COutput	
AngleNorthX	
MuonInfo	
StepLength	
CWaveLower	Wavelength lower limit (nm)
CWaveUpper	Wavelength upper limit (nm)
Multipl	
CorePos[2][20]	Core positions of randomized shower
SIBYLL[2]	
QGSJET[2]	
DPMJET[2]	
VENUS_cross	
mu_mult_scat	
NKG_range	
EFRCTHN[2]	
WMAX[2]	
rthin_rmax	
viewcone_angles[2]	Inner and outer angles of Corsika's VIEWCONE option.
telescopePhi	Telescope azimuth (rad). Measured from South, counter-clockwise
telescopeTheta	Telescope zenith angle (rad)
TimeFirst	Arrival time on camera of first photon (ns)
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)
longi_Nmax	Numer of charged particles at maximum
longi_t0	Atmospheric depth of shower starting point (N=0)
longi_tmax	Atmospheric depth of shower maximum (g/cm^2)
longi_a	
longi_b	For **longi_a**, **longi_b**, **longi_c**, see CORSIKA manual
longi_c	
longi_chi2	χ^2/dof of the fit

Variable	Description
CORSIKAPhs	Original photons written by *CORSIKA*
AtmAbsPhs	Photons absorbed by the atmosphere
MirrAbsPhs	Photons absorbed by the mirror
OutOfMirrPhs	Photons outside the mirror
BlackSpotPhs	Photons lost in the "black spot"
OutOfChamPhs	Photons outside the camera
CPhotons	Photons reaching the camera
elec_cph_fraction	Fraction of C-photons produced by electrons
muon_cph_fraction	Fraction of C-photons produced by muons
other_cph_fraction	Fraction of C-photons produced by electrons

The event header is followed by 8-word blocks, one for each photon that reaches the camera. A photon block contains the following variables:

Variable	Description
w	Wavelength (nm)
x, y	Impact point in camera coordinates (cm)
u, v	Director cosines
t	Arrival time on camera (ns)
h	Production height (cm)
phi	Incidence angle with respect to camera plane (rad)

After the last event photon block there is a blank line, an `END---EVENT` flag, another blank line and then the following event. After the last event in a run it appears the flag "`END-----RUN`", while after all the processed runs, a "`END----FILE`" flag is written. Finally, after this flag an "ascii tail" has been added to the file: it consists of the ascii files *magic.def*, *axisdev.dat* and *reflectivity.dat* one after the other separated by blank lines. In this way all the relevant parameters used to produce the output are kept together with the reflected events.

9 Appendix A

The list of all Reflector files follows.

```
refl-install
MagicProgs/Simulation/Detector/Reflector_0.6/Changelog
MagicProgs/Simulation/Detector/Reflector_0.6/Makefile
MagicProgs/Simulation/Detector/Reflector_0.6/Makefile.bak
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/reflector
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/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/parabola.eps
MagicProgs/Simulation/Detector/Reflector_0.6/doc/eps/refl06images.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/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
```