



СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

Дубна

95-449

E13-95-449

S.Trusov\*, L.Groza

SOFTWARE FOR THE SIMULATION  
OF THE THRESHOLD CHERENKOV COUNTERS

---

\*Skobel'syn Institute of Nuclear Physics of Moscow State University,  
Moscow, Russia

1995

## Introduction

To accomplish our task of calculating the Cherenkov counter efficiency, we need to know the number of photons produced per unit path length by a particle with charge  $Ze$  and per unit energy interval of the photons. We used [1,2]:

$$\frac{d^2 N}{dE dx} = \frac{\alpha Z^2}{\hbar c} \sin^2 \theta_c = \frac{\alpha^2 Z^2}{r_e m_e c^2} \left(1 - \frac{1}{\beta^2 n^2(E)}\right) \quad (1)$$

or, equivalently per unit wavelength  $\lambda$ :

$$\frac{d^2 N}{dx d\lambda} = \frac{2\pi\alpha Z^2}{\lambda^2} \left(1 - \frac{1}{\beta^2 n^2(\lambda)}\right) \quad (2)$$

Where:

$\theta_c = \arccos(1/n\beta)$  - Cherenkov angle

$n$  - the refraction index;

$\alpha$  - fine-structure constant;

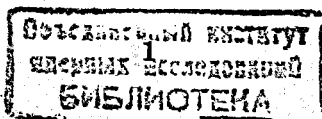
$r_e$  - classical electron radius;

$m_e$  - electron mass.

For this type of counters the light produced in the radiator is usually collected by an optical system formed by spherical or parabolic mirrors on a photomultiplier where it is transformed in an electric signal. In order to calculate the efficiency we have to evaluate the number of photoelectrons collected on the first dynode of the photomultiplier.

$$N_{pe} = L \frac{\alpha^2 Z^2}{r_e m_e c^2} \int \eta_c(E) \varepsilon_{coll}(E) \varepsilon_{det}(E) \sin^2 \theta_c(E) dE \quad (3)$$

where  $L$  is the path length in the radiator,  $\varepsilon_{coll}$  is the efficiency of collecting the Cherenkov light;  $\varepsilon_{det}$  is the quantum efficiency of the photocathode,  $\eta_c(E)$  is the collection efficiency of the first stage of the photomultiplier.



To achieve a high efficiency for the whole counter one have to design an optical system with a very good collection efficiency  $\epsilon_{coll}$ . For this purpose and to be able to calculate the arrangement of the mirrors and of the photomultipliers one have to simulate the optical system. The software extensively used for simulations in high energy physics, "GEANT" [3], does not handle with low energy photons, where the photomultipliers are sensitive, and other programs for optics, such as "Ray Tracer", are not able to simulate the production of photons. For these reasons it was necessary to develop a program able to calculate the trajectory of particles and photons and to simulate the Cherenkov photon production.

#### The structure of the software:

The simulation program was written in the "C" language and developed at three levels. At the beginning we developed a library that includes structures and functions for the analytical geometry. The structures describe the basic elements: points, lines, planes and a number of functions for constructing intersections, parallels and perpendiculars. Other functions help to define different coordinate systems and to recalculate coordinates in these new systems. This library contains around 2000 lines of code. Based on this library, some specific structures were defined at the second level. These structures are intended to describe mirrors, photomultipliers, entrance and exit windows of the detector, particles and photons. In order to have a model reasonably close to reality, the description of the mirrors includes:

- the focal length;
- the coordinates of the focus;
- two angles which describe the axis inclination relative to the coordinate system;
- the reflection index of the surface;
- the rectangular area covered by the mirror.

The photomultipliers are described by:

- the coordinates of the centre of the window;
- the useful diameter of the photocathode;
- a structure which describes the window plane;
- two angles that describe the inclination relative to the coordinate system.

At the last level all these entities were assembled and some additional functions for generation of particles and photons, for reflection on the mirrors and for intersection with the photomultiplier window were developed. Until here we used only simple instructions, independent of the operating system so these libraries are highly portable.

Finally some functions for input-output, data handling and efficiency calculation were built.

#### The input of the program

In order to describe a counter one have to input the coordinates of the entrance and exit windows, the number and the description of the mirrors and photomultipliers. The mirrors may be positioned by entering the coordinates and the axis inclination or using a function which calculates the inclination to obtain the best light collection on a given photomultiplier. The mirror segmentation for one of our counters may be seen in Fig 1. In this figure Z is the coordinate along the beam. Three photomultipliers were used and it may be seen on which of them the mirrors are collecting the light. The efficiency may be calculated during the simulation if the quantum efficiency of the photocathode averaged over the Cherenkov spectra is input and if a threshold of detection ( $tr$ ) in photoelectrons which reach the first dynode is given.

A function allows one to modify parts of the counter and another one to simulate for a part of entrance window. A routine was created to compute a "map" of efficiency on the entrance window. The entrance window is divided into

rectangular bins: the simulation is carried out and the efficiency is calculated for each bin. Based on the output, the efficiency may be calculated after the simulation for various thresholds  $\tau$  and efficiency values of the photocathode.

#### The simulation. Some approximations

After some uniformity tests we decided to use the random number generator available from the programming environment (which has a period of  $2^{31}$ ), but improved with the Bays-Durham algorithm [1.4].

The first step of the simulation is generation of the coordinates  $x, y$  where the particles enter the detector. The particles are uniformly distributed in the entrance window and there is a correlation between the position of the particle and its inclination according to the conditions of the "DIRAC" experiment [5]. Let us consider the axis  $Oz$  of the detector. Then "ver" and "hor" are the angles (in degrees) between  $Oz$  and the projections of the particle trajectory in a vertical and horizontal plane ( $xOz$  and  $yOz$ ) as follows:

$$\text{ver} = -1.5 + 3(h - x) + \text{rnd}(-1.7, 1.7);$$

$$\text{hor} = -22 + 11(w - y) + \text{rnd}(-1.7, 1.7);$$

where  $h$  and  $w$  are the height and the width of the entrance window;  $\text{rnd}$  is a function which returns a random value uniformly distributed between its two arguments.

After the calculation of the particle trajectories, the Cherenkov photon production is simulated. The specific conditions allow us to do some approximations. Because the radiator we intend to use is the air at normal conditions, the maximum Cherenkov angle is rather small ( $1.5^\circ = 26 \text{ mrad}$ ). The electrons have quite a large momentum so we can suppose that all the photons are generated at the maximum Cherenkov angle relative to the particle trajectory (i.e. all electrons have  $\beta=1$ ). Because the light collection efficiency is sensitive to the

inclination of the photons, this first approximation leads us to a smaller efficiency than in real life.

In our case the length of the radiator is about 3 m and the number of photons that are generated is rather large ( $\approx 280$  for 3 m of air). For these reasons we could consider the number of photons produced as a constant instead of a distribution. The coordinates of the places where the photons were produced were calculated by dividing the path of the electron by the number of photons produced. The large number of photons generated also allow us to simulate the conversion in photoelectrons using a binomial distribution with the probability  $p$ , where  $p$  is the photocathode spectral sensitivity  $S_{ke}(E)$  multiplied by the number of photons produced per unit energy interval, integrated over the sensitivity domain of the photocathode and divided by the total number of photons:

$$P = \frac{N_{pe}}{N_{photons}} = \frac{\int SK_e(E) \sin^2 \theta_c(E) dE}{\int \sin^2 \theta_c(E) dE}$$

For each generated photon its trajectory and the coordinates of the intersection with the photomultiplier's window (if it hits any window) are calculated. These coordinates are used to fill two dimensional histograms (100\*100 channels) which will represent the light distribution in the photomultiplier window.

#### The calculation of the efficiency

For each generated particle and for each photomultiplier, the number of photons that reach a photomultiplier is counted. For each photomultiplier a histogram that contains the number of photons coming from one particle which hit its window is filled. Not all the photons produced reach a photomultiplier window, that is why, in Fig 2, where a distribution for one of the photomultipliers is shown, we obtained a long tail. The number of photons reaching a photomultiplier window is decreased in correspondence with the mirror reflection

index. The number of photoelectrons emitted by the cathode is calculated using a binomial distribution with the probability  $p$  and the number  $n$  of photons. The photomultipliers employed usually have large windows. For this reason the collection of photoelectrons on the first dynode is smaller for those produced at the cathode's margins than for those coming from the centre. The spectrum of electrons that reach the first dynode is computed using a binomial distribution with probability  $d=0.8$  (from [6]) applied to the emitted photoelectron spectrum. The other 3-4 dynodes may also contribute to losses of events [7]. In our case we found enough to compute the photoelectron spectra applying a Poisson distribution to the spectrum of the photoelectrons that reach the first dynode, but for other cases one has to employ single electron spectra to find the coefficients of a Polya distribution which might be used several times.

The efficiency may be computed for two cases, corresponding to two different ways of processing of the photomultiplier signals. This is realised by an option named "Photoelectron sum". When this option is "yes", the signal coming from the photomultipliers is summarised, if it is "no" only the largest signal is considered. Another histogram is filled with the total number of the photoelectrons that reach the first dynode (in all photomultipliers) if the option "Photoelectron sum" is "yes" or with the maximum number of the photoelectrons that reach the first dynode (in one of the photomultipliers) if the option "Photoelectron sum" is "no".

The distributions computed for our case are shown in Fig 3. The corresponding efficiency computed for  $tr=3$  is 99.62%. The efficiency is computed as the probability for the number of photoelectrons to be greater than or equal to the threshold  $tr$ . In order to obtain a useful signal  $tr=1$  might be enough but for a good signal/noise discrimination  $tr=2$  or 3 is needed. Anyway, the histogram with the number of photoelectrons that reach the first dynode in the photomultipliers is

part of the output, so the efficiency may be computed after the simulation for various thresholds ( $tr$ ).

### The output of the program

The program creates a directory with a symbolic name of the simulated detector and places all the files there. If the detector is modified, then it is considered as a new one. The minimal output contains a file with the extension .stp which describes the parameters of the detector (mirror, PM window); a "report" file with the .rep extension which provides a report of the simulation and its parameters  $p$ ,  $tr$ ,  $d$ . There are 3 ways to present the output and any of them can be selected at the same time.

The N-tuple format. The file "Name.dat" contains a record for each photon. This record contains the number of the electron that produced it, the number of the mirror that reflected it, the number of the photomultiplier reached by the photon and the coordinates of the intersection with the window. The information is very detailed, but the volume of the data is very large. This format is useful only if another program gets this record by a pipeline and handles it.

The "1d" format is used for storage of 1d histograms with the photon yield in the PM window and with the total photon yield presented above.

The "2d" format is used for storage of 2d histograms with the light distribution for each PM window. The window is divided into  $100 \times 100$  bins; for a photomultiplier with a large window (11 cm) the granularity is low enough ( $1.1 \times 1.1 \text{ mm}^2$ ). An example is shown in Fig 4. One can see that the main part of the light falls on the central part of the photomultiplier.

The simulation might be performed for different areas of the entrance window and for each area (bin) the output may be stored in one of these forms: the name of the files begins with the number of the bin. A file called "identify.bin" contains the number of the bin and the coordinates of its area in the entrance window.

When the optical program creates a map of efficiency for the entrance window the results are stored in a file called "eps.map". Such a file was used to produce Fig 5: here the option "Photoelectron sum" was set "No". The inefficiency appears mainly at the edges of the mirrors and where the light from one electron hits two different photomultipliers.

All the results are stored in a text mode to be accessible for an analysis program as PAW.

We choose the "C" language mainly for the possibility of working with structures. The compiling options were set to produce the fastest code. This increases the size of the executable code which is anyway no larger than 150 KB. Some different compilations were done to generate instructions for microprocessors 386 and Pentium. When running a Pentium -75 MHz, one can simulate 4.5 events/sec.: 10-12 hours are enough to obtain a detailed result.

### Conclusions

The software allows one to describe easily and fast different arrangements of mirrors and PM's. It is possible to compute the efficiency during the simulation and to recalculate it for different parameters. Also, one may obtain the light distribution on the PM window, this distribution helps to find the optimum position of the PM's.

We successfully applied this software and developed a number of variants of threshold Cherenkov counters for which the efficiency was calculated considering different parameters.

### Acknowledges:

We would like to thank our colleagues in the "DIRAC" collaboration for their help and support. We particularly wish to thank L.L. Nemenov, Valeriu Zoran and Titus Ponta for their suggestions, and Andre Braem for his advice.

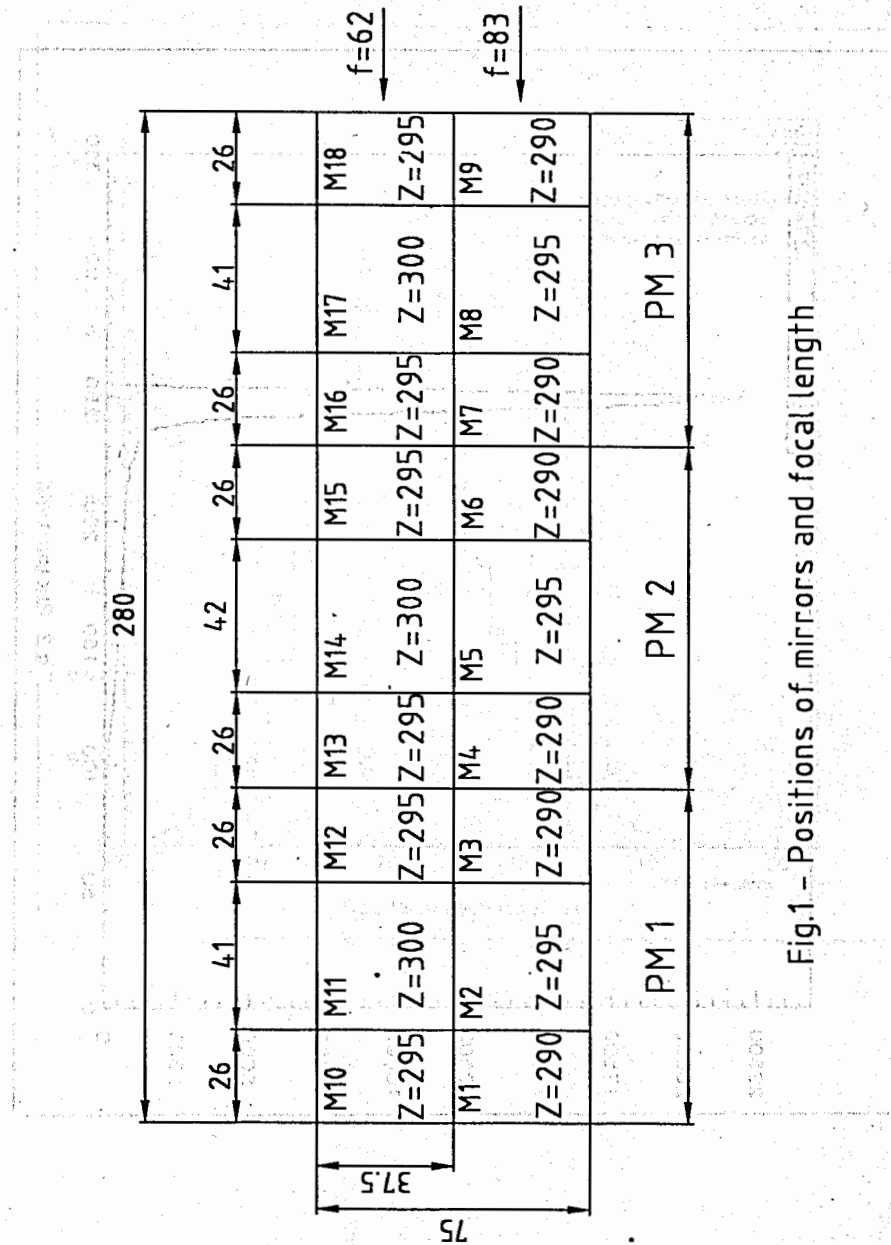
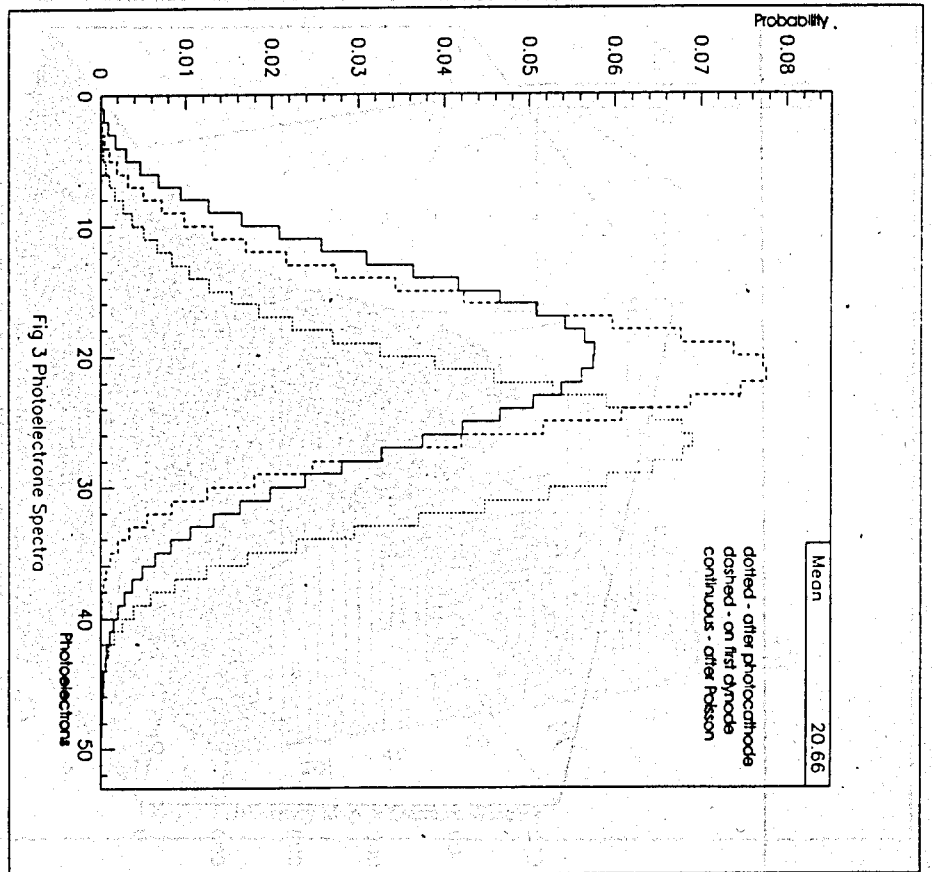
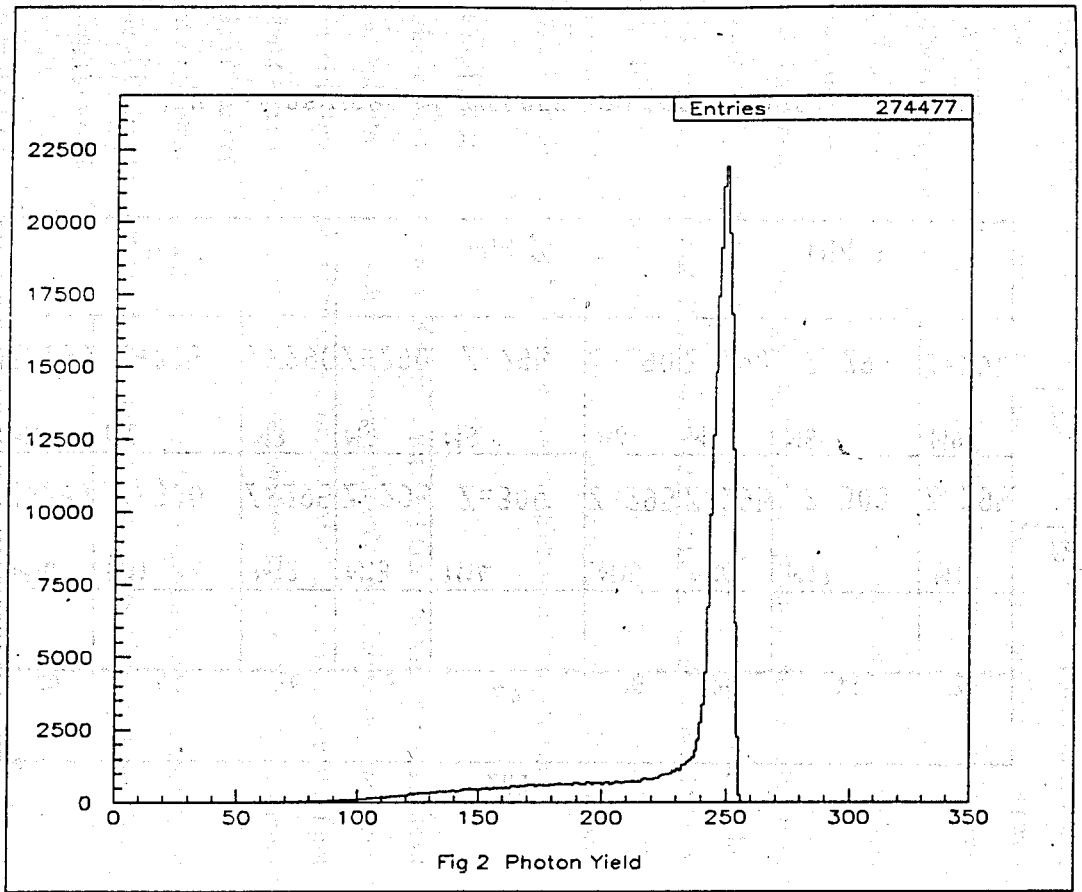
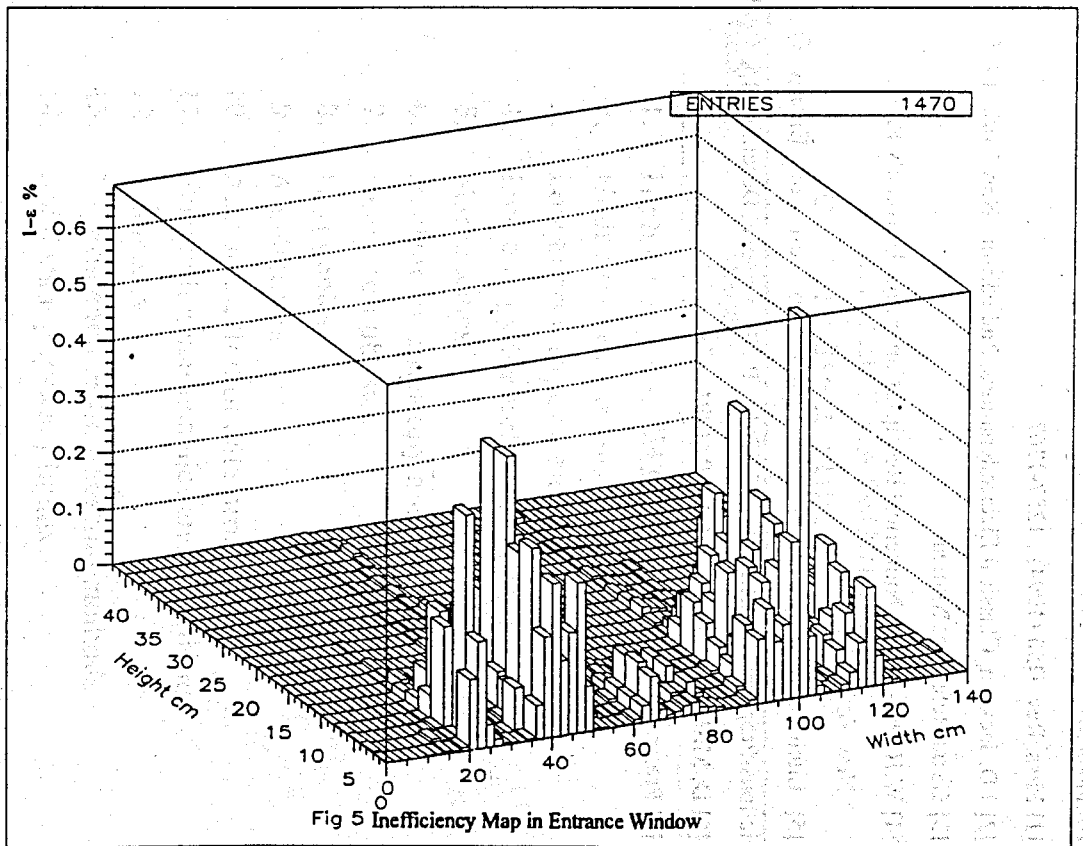
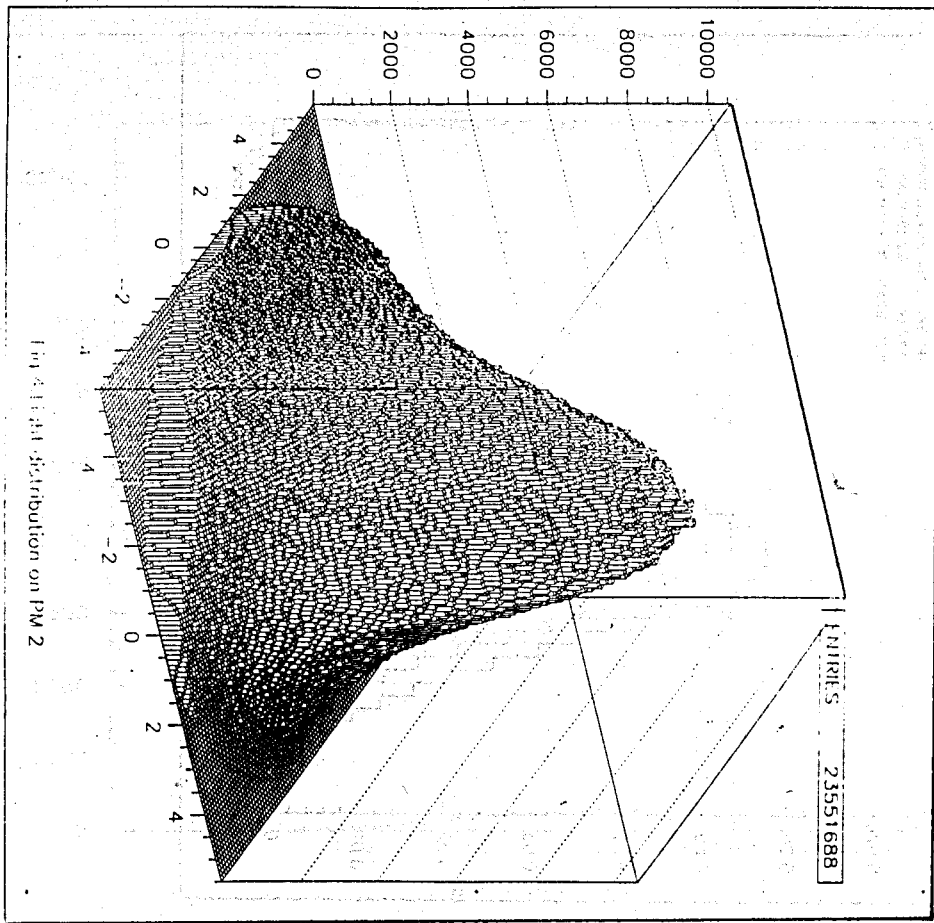


Fig.1 - Positions of mirrors and focal length









## References:

- [1] Phys. Rev. D50 (1994) , 1257-1262
- [2] J.D. Jackson, Classical Electrodynamics, 2nd edition, New York, 1975
- [3] Geant Reference Manual
- [4] W.H. Press et al, Numerical Receipts. Cambridge University Press.  
New York, 1986
- [5] Lifetime Measurement of  $\pi^+ \pi^-$  Atoms to Test Low Energy QCD  
Predictions-Proposal to CERN/SPSLC SPSLC/P 284, Geneva, December 15, 1994
- [6] D. Miskowicz et al. NIM A350 (1994) 174
- [7] Prescott J.R. NIM A 39 (1966) 173

Received by Publishing Department  
on November 3, 1995.