DESY Report

Simulation of Cosmic Ray Particles in C++

Björn Sperling

14th September 2007

1 Introduction

During my stay at DESY in Hamburg, i have worked in the FLC-Group. In this subdivision they will be developing a TPC (Time Projection Chamber) for the ILC (International Linear Collider). My supervisor was Adrian Vogel. It was my task to write a program which generate cosmic particles. This can help to simulate perturbations for the TPC, because it is important to separate hits between cosmic ray particles and real e+/e- interactions. They planed some measurements with cosmic ray particles at a prototype. To simulate these experiments in Geant4, it is important to have cosmic rays at sealevel as input.

2 Cosmic Particles

Muons are the most numerous charged particles at sea level. Most muons are produced high in the atmosphere (typically 15km) and lose about 2 GeV to ionization before reaching the ground. Their energy and angular distribution reflect a convolution of production spectrum, energy loss in the atmosphere, and decay.

The integral intensity of vertical muons above 1 GeV/c at sea level is approx $70m^{-2}s^{-1}sr^{-1}$, with recent measurements tending to give lower normalization by 10-15%.

The overall angular distribution of muons at the ground is prop $\cos(theta)^2$, which is characteristic of muons with E approx 3 GeV. At lower energy the angular distribution becomes increasingly steep, while at higher energy it flattens, approaching a sec(theta) distribution for $\theta < 70$. [1]

Other particles are not important for the simulation, because they can't penetrate hard materials (e.g. electrons) or they are very rarely. (Or they don't interact with the detector.)

3 The Generator Program

I have written this program in C++ and divided it in different classes. The first main problem was to generate cosmic particles and save the data to a file. The main class is CosGun, which manage all particles. The constructor of CosGun will be accept three parameters: period of time in sec, a in mm, b in mm (a and b define the area of the detector / generator zone). With this three values can CosGun calculate a expected value of number of particles because the integral flux is known. This is a Poisson distribution. By side, the random number generator is used form CLHEP. This is a big library of math functions and methods for high energy physics. Until now there is a number of particles but no real particles are generated. So CosGun will create objects they represent particles. This objects are derived from CosPar which create a muon or anti-muon (this ratio is also a flat asymmetric distribution) with a stochastic value of energy by this distribution [1]

$$\frac{dN}{dE} \approx \frac{0.14 \cdot E^{-2.7}}{cm^2 \cdot s \cdot sr \cdot GeV} \cdot \left(\frac{1}{1 + \frac{1.1 \cdot E \cdot \cos(\theta)}{115 \cdot GeV}} + \frac{0.054}{1 + \frac{1.1 \cdot E \cdot cos(\theta)}{850 \cdot GeV}} \right)$$

and a stochastic direction. For this we need θ and ϕ. θ follow this

$$\frac{dN}{d\theta} \approx \cos(\theta)^{1.85}$$

and ϕ is evenly distributed [2]. Also it needs a randomly selected start position.

To produce random numbers of a common distributions like Poisson, it is easy to use a existing function of a commercial library. But to produce unequally distributed numbers like this one of energy, it is impossible to find a commercial function.

One solution is to use the so called 'hit and miss'-method.
Assumption: f(x) is a probability distribution function, $x \in [a, b]$

This algorithm consist of two steps:

- generate a pair of pseudo random numbers x and y which are evenly distributed with $x \in [a, b]$ and $y < \max(f(x))$
- if $y < f(x)$ then is x the new random number otherwise you need a new pair.

But this method is very slow, if the ratio between $\int_a^b f(x)dx$ and $(b-a) \cdot \max(f(x))$ is small. Because they need many trials for one new number with $y < f(x)$. To preventing this problem, there is a improvement. Get a function g with $g(x) > f(x)$ for all $x \in [a, b]$ and reverse the integral of g. Then you can produce random numbers they are distributed by g. Now you can use the hit and miss method on these new values to generate random numbers they are distributed by f. (The ratio between $\int_a^b f(x)dx$ and $\int_a^b g(x)dx$ should be small, if g is a adequate function.) Fig. 1 shows f for some θ's and g. This improvement save a lot of time.

For example:
Old version to generate about 5000 particles on this machine used 34sec of time.

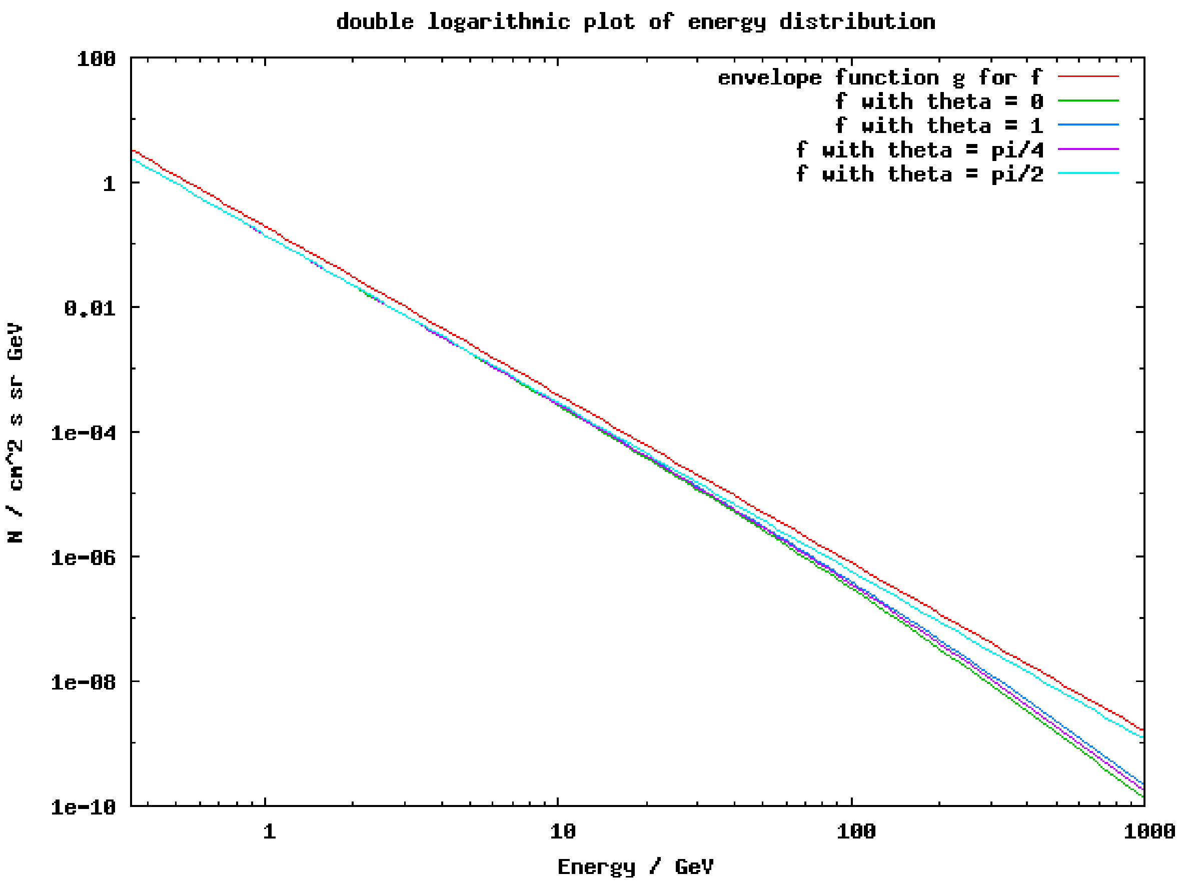

Figure 1: This plot shows the distribution function of energy for some θ values and the envelope function which is used by the improvement of hit and miss.

New version with same parameters takes about 1sec!!

Well, every particle need a creation time, too. But this is a flat distribution like ϕ. CosWrite is a class which write all generated particles to file in stdHEP format. For this it used the HEPEVT common block and some function of the stdhep library. stdhep is also used by other generators like pytha.

3.1 What aspects are not in the simulation?

- Only anti-muons and muons above 0.35 GeV energy will be simulated, because all other particles are very rare or don't penetrate hard materials.
- no geomagnetic effects are implemented which produce a systematic error at all muons below 1 GeV
- no variation of solar wind is implemented

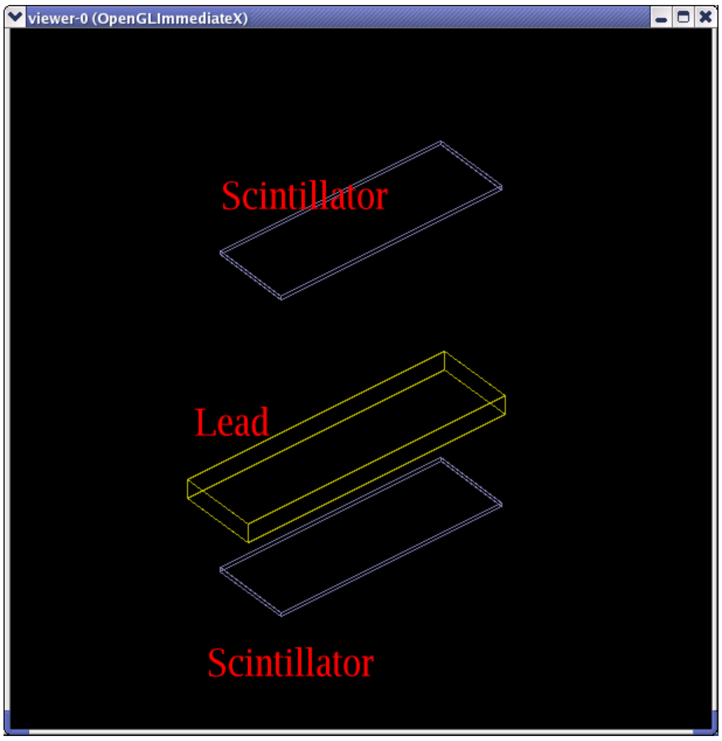

Figure 2: Picture of the simulation without any event

I think the last two aspects are all but impossible to inculde in such a simulation. One Reason is the varitation of such input parameters with the time. The other is the complexity of the problem.

4 Geant4 Simulation

Adrian Vogel had programmed a simulation based on geant4 which can simulate a experiment on a medi tpc. One scintillator is above of the medi tpc and the other below of them. Between the two scintillators is lead to reduce trigger events from electrons and other light particles (Fig. 2).

This experiment was done by Helena Stange in Juli 2005. But there were some problems and so real data didn't agree with simulated. She found an increase of the trigger rate after she put a second layer lead between the scintillators [6]. This is strong deviation of the normal expectation.

To investigate these problems, i have implemented a method in this simulation to read stdHEP files. Now it is possible to produce a file with the correct detector area and measurement period with my program. Then it is available in the simulation and produce trigger rates in the order of magnitude of the real data. For this i have modify the simulation and add a other mechanism to associate two trigger event on both scintillators to one event. To do this, the simulation save any event in the detector and the time when it happens. After the run it searches for all event-pairs (top and bottom detector)

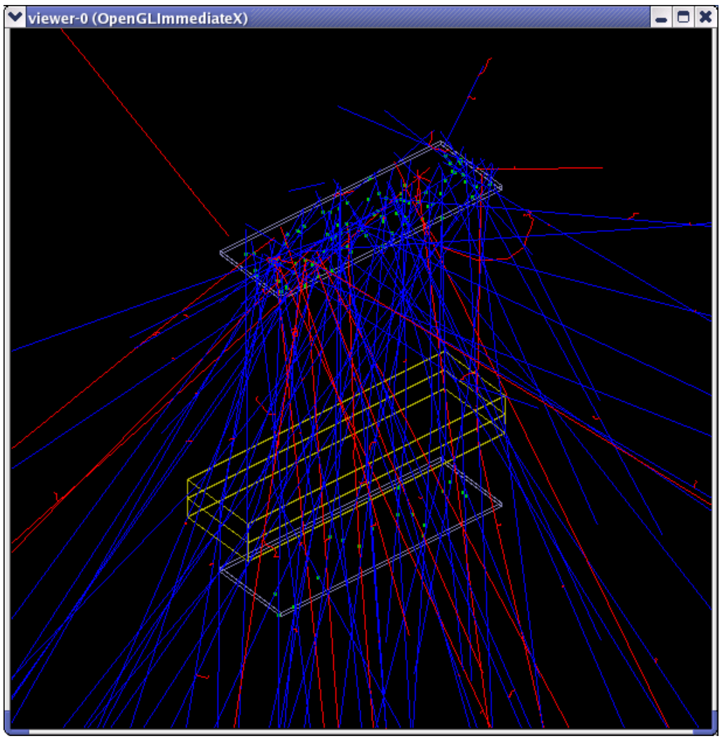

Figure 3: Picture of the simulation with cosmic particle events. The TCT simulation was a good test of my generator program.

which are inside a time frame. This is very close to the real experimental practice. Fig. 3 shows cosmic particle events in the TCT simulation.

My results show a reduction of trigger rates, if there is a second layer of lead. This is a expected, but a different result compared to the experiment of Helena Stange.
I don't know the details of the measurement, but one plausible explanation is the variation of cosmic rays after a CME (Coronal Mass Ejections). - If we know the date of the measurement, we can look at ACE for solar activities.

A Histograms of energy and angle distribution

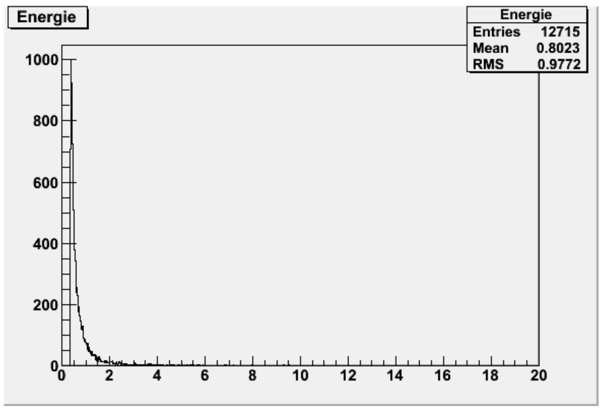

Figure 4: *Histogram of energy distribution*

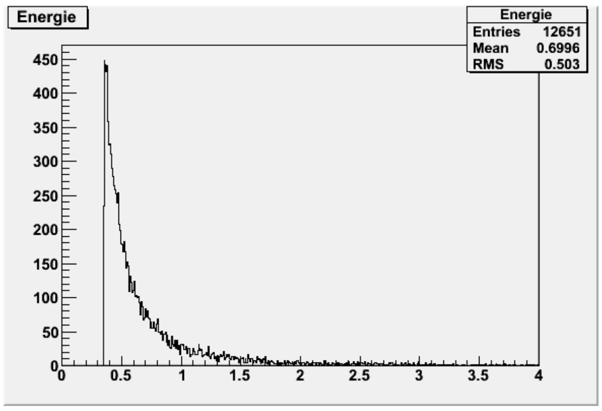

Figure 5: *Histogram of energy distribution*

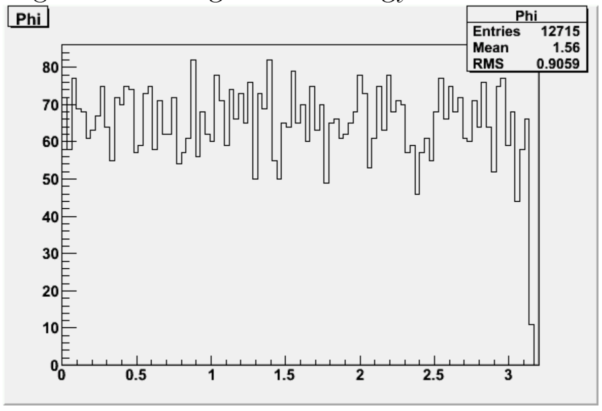

Figure 6: *Histogram of ϕ distribution*

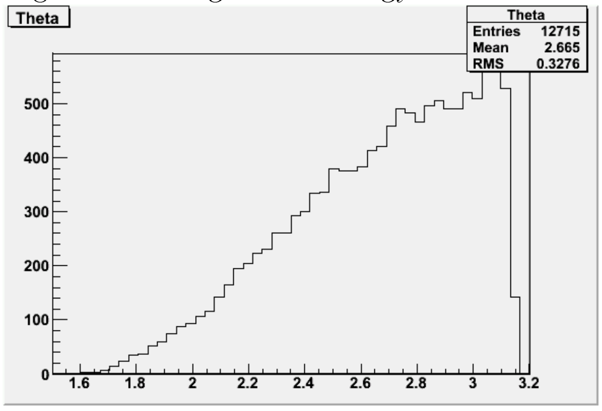

Figure 7: *Histogram of θ distribution*

References

[1] Journal of Physics G, Nuclear and Particle Physics Vol. 33 - Chapter 24

[2] Cosmic rays at earth: Researcher's reference manual and data book - Grieder (2001)

[3] C/C++ Reference - www.cppreference.com

[4] Geant4 User's Guide for Application Developers - Version: 9.0 (2007-06-29)

[5] C++ Resources Network - www.cplusplus.com

[6] Aufbau eines Messstandes mit Szintillationszaehlern zur Durchfuehrung von Experimenten mit kosmischen Myonen - Helena Stange (Juli 2005)