

Particle fluxes and spectrum

Often in physics we have to deal with particle spectra. A particle spectrum is defined as the variation of the particle fluxes as function of the particle energy. Analytically, a particle spectrum is represented by the equation

$$S(E) = N / (T \times G \times E \times \varepsilon) \text{ [s cm}^2 \text{ sr GeV]}^{-1},$$

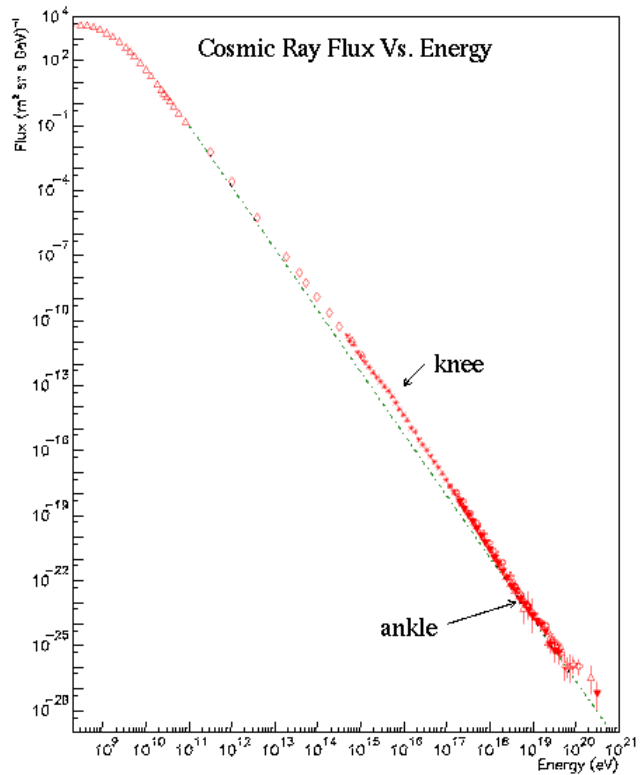
where N is the number of particles of energy E [GeV] passing through the solid angle G [cm² sr] in the integrated time T [s] and detected with efficiency ε .

Experimentally, precise determination of particle spectra requires an excellent knowledge of the detector – that is of the acquisition time (T), geometrical factor (G), energy determination and efficiencies – plus a clean selection of a data sample.

On the right is shown, as an example, the spectrum of cosmic rays. Notice the double log scale and notice that, in the case of cosmic rays, the spectrum can be roughly described with power law functions, that is : $F(E) = A E^{-\gamma}$,

where A is the normalization factor and γ is usually called “spectral index”.

Spectra determination is of crucial importance in many fields of physics for example to determine sources composition, looking for new particles, study the amount of traversed material and its distribution.



Goal of these exercises

The aim of these exercises is to determine the energy spectrum for (simulated) electrons crossing the detectors of the PAMELA apparatus.

Basic informations: you will work on a set of events coming out from a PAMELA experiment simulation. The given file

```
/home/mocchiut/pamela/data/pamsimu2013.root
```

contains the TTree pamcalotree, storing data with the PamCalo class, header file:

```
/home/mocchiut/pamela/PamCalo/inc/PamCalo.h
```

so library:

```
/home/mocchiut/pamela/PamCalo/lib/Linux/libPamCalo.so.
```

The ROOT file contains a mixture of different particles: protons, antiprotons, electrons and positrons in an energy range from about 1 to about 20 GeV.

In these exercises, we will create a variable to select electrons from the sample ($qtotene \equiv qtot/energy$) and we will save this and other variables on a different file for negative particles. Then we will use a script to select electrons and to plot their number as function of energy. Finally, we will calculate the spectrum (given T, G and ε) and we will fit the energy spectrum with a given function.

Exercise 1

Write an executable compiled program which reads the input file

```
/home/mocchiut/pamela/data/pamsimu2013.root
```

and gives as output a new ROOT file containing a TTree with two variables (a TBranch for each one):

- `abs(energy)`
- `qtotene (\equiv qtot/abs(energy))`

Save into the new file events which satisfy the following condition:

1. each event has energy lower than zero.

Hints:

- to compile, remember to add also the compilation flags:

```
-I/home/mocchiut/pamela/PamCalo/inc
```

```
-L/home/mocchiut/pamela/PamCalo/lib/Linux/
```

```
-lPamCalo
```

- to run, remember to export `LD_LIBRARY_PATH`:

```
export LD_LIBRARY_PATH=/home/mocchiut/pamela/PamCalo/lib/Linux/:$LD_LIBRARY_PATH
```

Exercise 2

Write a ROOT-CINT script which reads the output file of exercise 2 (should be similar to this one: /home/mocchiut/scripts/EM_output_030914.root use this file if you are not able to complete or run exercise 1) and gives as output on the screen and on the disk (pdf format) a TCanvas divided into two pads (one column, two rows – hint: TCanvas::Divide) which contain from top:

1. one scatter plot (TH2D, use nbins=1000 for both axis, Y range [0.,1000.]) of `qtotene` vs `energy`;
2. the distribution of events (TH1D, X range [0.1,20.] with both X and Y log scales) which pass the selection `qtotene>150.` and `qtotene<400.` (i.e. event by event fill the histogram at energy “energy” only if the condition on `qtotene` is satisfied).
3. Get the number of entries (hint: look at TH1/2 methods in the class reference) from both the histograms and determine the selection efficiency

```
 $\varepsilon = \text{entries}(\text{histo point 2}) / \text{entries}(\text{histo point 1.})$ 
```

Print the resulting efficiency on the STDOUT.

Exercise 3

Update the script of exercise 2 in order to draw a new TCanvas divided into two pads (one columns, two rows – hint: `TCanvas::Divide`) which contain:

1. A TH1D with the number of selected entries (like the one of point 2, exercise 2) where the histogram is created with a custom binning given by this array: `x[9] = {1,2,3,4,5,7,9,13,18}` .

(hint: look at examples shown during the course and at the TH1D manual).
2. A TH2D histogram (X range [0.5,20.] Y range [1e-4,2.]) with over-imposed (hint `Draw("same")`) a TGraphErrors with the flux.

Hints: create the three TGraphErrors input arrays (x value, y flux and ey error on flux) with a loop over the number of bins of TH1D of previous point 3.1. For each bin of the TH1D with custom binning get the bin content (look at `TH1D::GetBinContent`). This will be the number of selected events (N) for every element of y, that is: the y array will be filled with N divided by the efficiency determined in ex. 2.3 (in case use 0.77), divided by the bin width (Gev, use `TH1D::GetBinWidth`), divided by a time of 1000 (seconds) and a geometrical factor of 21 (cm² sr). The array ey, error on the flux, will be filled like y where square root of N is used instead of N. Use the bin centers as values for the X array needed by TGraphErrors (`TH1D::GetBinCenter`). WARNING: pay attention at bin numbering in TH1 histograms! Look at TH1 class reference “convention for numbering bins”.

3. Fit the TGraphErrors with the function $f(x) = A (E-B)^C$ and plot the fit results in the same histogram of ex. 3.2, over the measured flux. Before fitting set the following starting values for the three parameters:

(A,B,C)=(6000.,-4.,-5.).

Print on the STDOUT the value of C.

Save the output on the disk as pdf file.

Preparing the output

- create a directory and put inside this directory ALL the files you want me to correct and look at.
- create a README text file (named like EM_README.txt), inside the file write:
 - **your name and surname**
 - a list of the files you are submitting
 - **in details** how to compile and run the programs
 - any other comment and answer to question(s)
- create a compressed tarfile containing the directory:

```
bash> ls
Mocchiutti/
bash> tar zcf Mocchiutti.tar.gz Mocchiutti/
```
- copy the tarzipped file on the USB key I will circulate

Timing and rules

- You have four hours time to do your work.
- You can search the web, look at manuals, look at any note you wrote during the course, etc.
- We will discuss what you have written at the oral examination on 2014/09/05, until that (if needed) you can change and improve your programs. In that case prepare an electronic version we can look at during the oral examination, we will compare it to the one handed in today and we will discuss any change and/or correction.