Simulation of High Energy Cosmic Ray Showers using CORSIKA

Mohammed Shadman Salam
I.D: 11315003

Thesis submitted in partial fulfilment of the requirements
for the degree of Bachelors of Science (BS) in Applied Physics and Electronics

Inspiring Excellence

Department of Mathematics and Natural Sciences
BRAC University
Dhaka, Bangladesh
January 2016

Declaration

I hereby declare that the thesis titled "Simulation of High Energy Cosmic Ray Showers using CORSIKA" is submitted to the Department of Mathematics and Natural Sciences of BRAC University as part of the requirements for the degree of Bachelors of Science (BS) in Applied Physics and Electronics. The presented work is a product of my own research and has not been submitted elsewhere for any other degree or diploma. All related research work that has been used as reference for this work has been properly cited.

Candidate

Mohammed Shadman Salam

ID:11315003

Certified by

Prof. Mohammed Arshad Momen

Chairperson

Department of Theoretical Physics

University of Dhaka

Acknowledgements

All it took was a week of classes conducted by Prof. Mohammed Arshad Momen for me to decide that he was going to be my thesis adviser. Since then he has been pushing my intellect to the limit by continually monitoring and encouraging my work. Although the research work conducted here was completely an unknown field for me in the beginning, but with his backing and motivation my confidence and self-belief never waned. For that and all the time he spent aiding me in my research, I am and shall be eternally grateful.

I would also like to convey my sincere gratitude to the Chairperson of the Department of Mathematics and Natural Sciences, Prof. A. A. Ziauddin Ahmad, for his lasting support and availability whenever I knocked on his door for a word. Furthermore, I am grateful to every faculty member in the department and specially would like to highlight the importance of
Dr. Md. Firoze H. Haque, Mr. Muhammad Lutfor Rahman, Mahabobe Shobahani, Ms Fardousi Ara Begum and the Late Prof. Mofiz Uddin Ahmed in the last four years of my undergraduate life at BRACU.

Last but certainly not the least, I would like to dedicate this research work to my beloved father who passed away seven years ago. Even in death, in all these years, I always felt he was watching over me. Father, hopefully I can make you proud some day.

The task is not so much to see what no one has yet seen; but to think what nobody has yet thought, about that which everybody sees.

-Erwin Schrödinger

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Origin and Energy Spectrum of Cosmic Rays

The surface of the Earth is continuously bombarded by very high energy particles, most of them relativistic and some non relativistic particles ranging in energy values from a few GeV to 10^{20}eV. The primary spectrum consists of charged particles, principally protons (86%) along with alpha particles, electrons (2%) and nuclei of some heavier elements up to uranium (1%). There are also small proportions of positrons and anti-protons which are believed to be of secondary origin, produced during the interaction of the charged primaries with interstellar gas. Neutral particles in the cosmic ray spectrum is conssist of γ rays, neutrinos and anti-neutrinos. The identified γ ray sources are the Crab Nebula and Active Galactic Nuclei (AGNs') while neutrinos are ejected from the Sun as a product of fusion reactions in its core [14].

Examining the energy density from the cosmic ray spectrum shows that although bulk of the incident radiation is of galactic origin, the energy spectrum rising to 10^{20}eV provides evidence of extra galactic origin [7]. The interstellar magnetic fields are very significant inside our own galaxy and they follow the spiral arms. The average magnitude of the galactic field is about 3 μG (0.3nT) with an energy density of about 0.2 $eVcm^3$ comparable to cosmic microwave radiation having energy density 0.26 $eVcm^{-3}$ and that of deep space cosmic radiation (1 $eVcm^{-3}$). Although the intergalactic field is only of the order of about 10^{-5} of the present galactic field, those very intense

fields associated with supernova explosions are considered to be the main accelerators of the cosmic rays that reach the surface of the Earth [14]. The following figure depicts the energy spectrum of cosmic ray protons showing the power law $E^{-2.7}$ dependence at energies below the so-called 'knee' region.

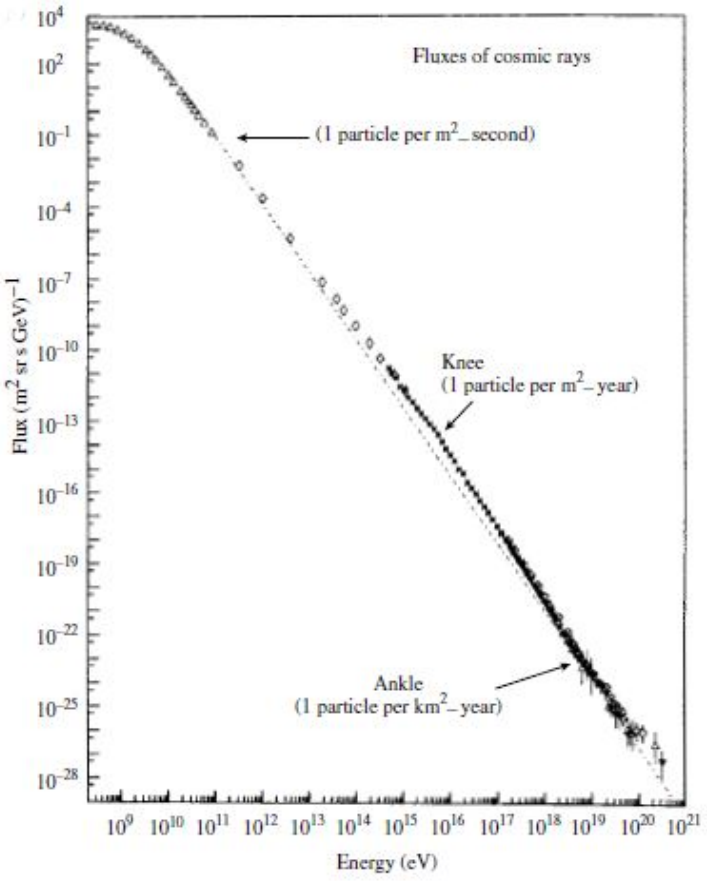

Figure 1.1: Cosmic Ray Spectrum

As the figure elucidates, the primary spectrum up to the knee at 10^{16}eV follows a simple power law but as energy increases further, the spectrum becomes steeper before flattening again at the ankle region indicated. To

summarise, at energies below the knee, the following power law is observed.

$$N(E)dE = \text{const.}E^{-2.7}dE \tag{1.1}$$

Above this knee but below the ankle, ($E_{ankle} > E > E_{knee}$), the index increases to -3.0 such that

$$N(E)dE = \text{const.}E^{-3.0} \tag{1.2}$$

and above the ankle, the index decreases to -2.69 as we reach the GZK cut-off above $4 \times 10^{19}\ eV$, i.e for ($E_{GZK} > E > E_{ankle}$)

$$N(E)dE = \text{const.}E^{-2.69} \tag{1.3}$$

Beyond this threshold, i.e $E > E_{GZK} = 4 \times 10^{19}\ eV$, the experimentally detected spectra (the AUGER and HIres experiments) is seen to flatten again. This region of the spectrum is parametrised by the following power law

$$N(E)dE = \text{const.}E^{-4.2}dE \tag{1.4}$$

Bulk of the radiation, at energies above 30 GeV in extensive air showers are isotropic as influence of the magnetic fields of the Earth or the Sun become unimportant. However, AUGER air shower experiment does detect anisotropies and significant correlation between the showers and Active Galactic Nuclei (AGN)'s at energies above 6×10^{19} within approximately 75 Mpc of Earth [14]

1.2 Extensive Air Showers (EAS)

The Extensive Air Shower (EAS) is a phenomenon that occurs when high energy cosmic ray primaries-protons or heavy nuclei-are incident on the Earth's atmosphere. The bombardment of the primaries with air nuclei results in the formation of huge numbers of charged and neutral secondary particles which in turn interacts with the atmospheric constituents or decays to give other particles [15]. Typically, two types of showers develop in the atmosphere; namely nuclear cascade and electromagnetic cascade.

The high energy primaries are hadrons which collide with air nuclei several times as they traverse the atmospheric depth. Each of these collisions

results in the production of large numbers of secondary particles such as π^0, π^+, π^-, K^+, etc. Some of the unstable hadrons decay into more penetrating particles, like μ mesons, while others show further interaction as they fall further downwards, giving rise to a series of interactions known as a nuclear cascade [15].

The π^0 produced in various interactions in the build-up to the nuclear cascade, decays almost instantaneously into high energy gamma rays which trigger huge electromagnetic cascades. The products of these cascades are photons, electrons and positrons which comprise the electromagnetic component of air showers.

$$\pi^0 \rightarrow \gamma + \gamma \tag{1.5}$$

Some of the other interactions that take place are :

$$K^+ \rightarrow \mu^+ + \nu \tag{1.6}$$

$$K^- \rightarrow \mu^- + \nu \tag{1.7}$$

The μ mesons form the muon component of the shower while the neutrinos, which have only weak interaction like muons, escape and carry away significant fraction of the primary energy with them. The surviving hadrons, all of them having strong interaction, make up the hadronic component of the shower [15].

In showers triggered by photon-electron collisions, the electrons lose most of their energies in one radiation length. In contrast, in nucleon induced showers the nucleons can penetrate several interaction lengths losing only about 25% of their energies in each encounter with other nucleons or mesons, making nuclear cascades more penetrating than electromagnetic ones [14].

Atmospheric density decreases exponentially with height due to the nature of the Earth's atmospheric gases and the gravitational field. As density decreases upwards, interaction lengths of the primaries are longer which leads to some of them decaying into secondaries. This exponential relation also allows the secondary particles to spread laterally as they descend towards Earth; the lower energy particles will have larger opening angles meaning these will spread more compared to their high energy counterparts. Hence, EAS is defined to have a dense core of high energy particles and the density

decreases with distance from the core. The lateral spread of density of particles away from the core is different for different species which enables the detection and identification of shower particles. Furthermore, the extended shape of the atmosphere introduces a characteristic arrival time for the different shower particles at detection level. It is found that mesons, with their lower mass, arrive first leading the heavier particles [15].

The defining factor of lateral spread of the shower particles is their transverse momentum which is typically about $0.3 GeV/c$ for nuclear cascades, much larger than for electromagnetic cascades of the same primary energy. Hence, EAS mostly consists of a high-energy core dominated by nucleons and a wider spread of particles outward containing the electron-photon component which gives rise to further electromagnetic cascades by means of pion production and decay [14].

1.3 Secondary Cosmic Radiation

The charged primaries that are incident on the Earth gives rise to air showers as elucidated in the previous section. The resulting particles of these atmospheric interactions are known as the secondary cosmic ray particles. These secondaries subsequently decays or interacts further along their trajectories. Secondary particles include mostly pions in all three, charged and neutral,

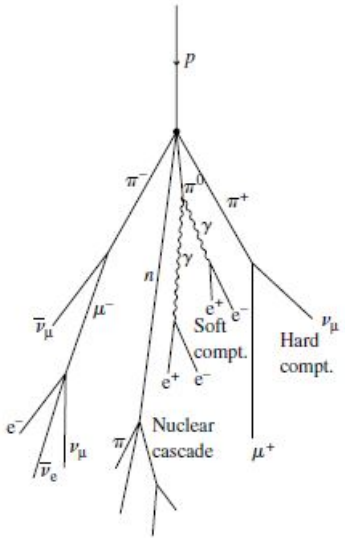

Figure 1.2: Decay of pions and muons in the atmosphere

states and some radioisotopes, most importantly $^{14}_{6}\mathbf{C}$. The nuclear interaction mean free path for a proton ($\lambda \approx 100 gmcm^{-2}$) which is very much less than total atmospheric depth ($X = 1030 gmcm^{-2}$), as a result of which the pions are produced in the stratosphere. The charged pions decay to form muons, neutrinos and anti-neutrinos with a proper half-life of $\tau = 26ns$.

$$\pi^+ \rightarrow \mu^+ + \nu_\mu \tag{1.8}$$

and

$$\pi^- \rightarrow \mu^- + \bar{\nu}_\mu \tag{1.9}$$

The daughter muons are unstable themselves, undergoing decay with $\tau = 2200ns$

$$\mu^+ = e^+ + \nu_e + \bar{\nu}_\mu \tag{1.10}$$

Mass of the muon is about $0.105GeV$ and a $1GeV$ energy muon has an average decay length of about $6.6km$ which is about the same as the scale height of the atmosphere. Hence, muons of energy $1GeV$ or less will decay in flight in the atmosphere without any competition from nuclear interaction as muons do not have strong interactions (Perkins,2009). However, muons of energy $3GeV$ have mean decay length of about $20km$ meaning that these energetic muons can traverse the atmosphere without decaying or being brought to rest and reach sea level on Earth. Even more energetic muons can reach deep underground which is why they are referred to as the 'hard component' of cosmic radiation [14].

Neutral pions undergo electromagnetic decay with $\tau = 8 \times 10^{-7}s$

$$\pi^0 \rightarrow 2\gamma \tag{1.11}$$

The resulting photons develop electron-photon cascades of relatively shorter absorption lengths, in the higher atmosphere. Due to the fact that these photons are easily absorbed in the upper atmosphere, the cascades are said to constitute the 'soft component' of cosmic radiation [14].

Among the products of nuclear interactions of the cosmic ray primaries are radioisotopes. $^{14}_{6}\mathbf{C}$ is of particular significance as it the presence of this isotope of a mean lifetime of 5600 years that makes radioactive carbon dating possible. It is formed by neutron capture in Nitrogen [14].

$$n + {}^{14}_{7}\mathbf{N} \rightarrow {}^{14}_{6}\mathbf{C} + {}^{1}_{1}\mathbf{H} \tag{1.12}$$

1.4 Detection of EAS

Primary particles that initiate EAS interact high in the atmosphere which makes it very challenging to detect them directly as it involves detectors being carried up by balloons or satellites. However, the very low flux at the highest energies of the cosmic ray primaries makes this method virtually impossible [13]. Hence, surface detection using array of ground detectors and mirror-photomultiplier systems to detect light due to charged particles traversing the atmosphere, have been employed to detect the secondary cosmic rays arriving and from these shower profiles, inferences can be made about which showers were initiated by electrons or which were initiated by nucleons [14].

1.4.1 Ground Detection

Ground detection of EAS is accomplished by using an extended array of ground detectors which samples the charged particles in the shower, typically by employing liquid scintillators or water Cerenkov counters. Showers detected in ground level are of primary energies, E_0 of about $1000TeV$ while at mountain heights, $E_0 \approx 100TeV$. As a result of the shower particles being relativistic ($v \approx c$), the shower front is well defined which enables accurate measurement of the direction of primary particles by timing the different parts of the shower front as it reaches the detector array [14].

1.4.2 Detection of Cerenkov Radiation

When a relativistic charged particle traverses a medium with a greater phase velocity than that of light in that medium, the loss in energy of the particle will give rise to coherent wavefronts in the path of the charged particle and an electromagnetic wave will be observed as a result, mainly in the UV or blue region of the spectrum [18].

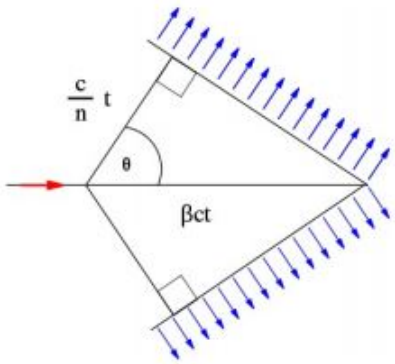

Figure 1.3: Huygen's construct for Cerenkov radiation

Following from the figure, we get

$$\cos\theta = \frac{1}{\beta n} \tag{1.13}$$

Here, θ is the opening angle, n is the refractive index of the medium and $\beta = \frac{v}{c}$ where v is the speed of the particle in the medium and c is the speed

of light in vacuum. The equation puts forward the notion of a threshold value for β below which no Cerenkov light is observed; however, most of the components of EAS have energies much greater than this threshold so that a significant amount of Cerenkov light is produced [14].

Cerenkov light can be detected by employing arrays of large spherical mirrors to direct the light on to photo-multipliers placed at the focus. The light is constricted in a narrow angular range, such that maximum opening angle, $\theta_{max} = \cos^{-1}\frac{1}{n}$, which results in the radius of the radiation to be restricted of around $100m$ around the shower axis which means that the mirrors and the photo-multipliers need to be in close proximity for effective recording of the radiation data [14].

1.4.3 Detection of Atmospheric Fluorescence

A healthy fraction of the energy of shower particles is dissipated in exciting and ionising air molecules. For shower detection, the excitation and subsequent emission of photons in the blue wavelength region is of vital importance and this phenomena is given the name atmospheric fluorescence [13]. In contrast with Cerenkov radiation which is emitted in a very narrow opening angle, this fluorescence light is given off isotropically in all directions [14].

A similar technique of imaging showers by photomultiplier tubes as in the detection of Cerenkov radiation is employed to detect atmospheric fluorescence. However, as fluorescence light is isotropic, the light pool is much more extensive than that received for Cerenkov radiation which enables the detection of distant shower particles which were not incident directly on the mirror-photomultiplier system [14].

Although, this mirror-photomultiplier technique is adept at detecting shower particles with significant accuracy, many limitations exist nonetheless. For instance, the incoming light signal is only appreciably detected on clear moonless nights when the optical background is low enough to be able to distinguish the very faint light signals from both fluorescence and Cerenkov radiations [13]. Even so, the detection system has a very poor duty cycle such that even in the most favourable environments, duty cycle can only improve to 10% [14]. A simulated shower profile obtained from a mirror photo-multiplier system is given below which validates the discussion above

that fluorescence light is more widely detected rather than Cerenkov light.

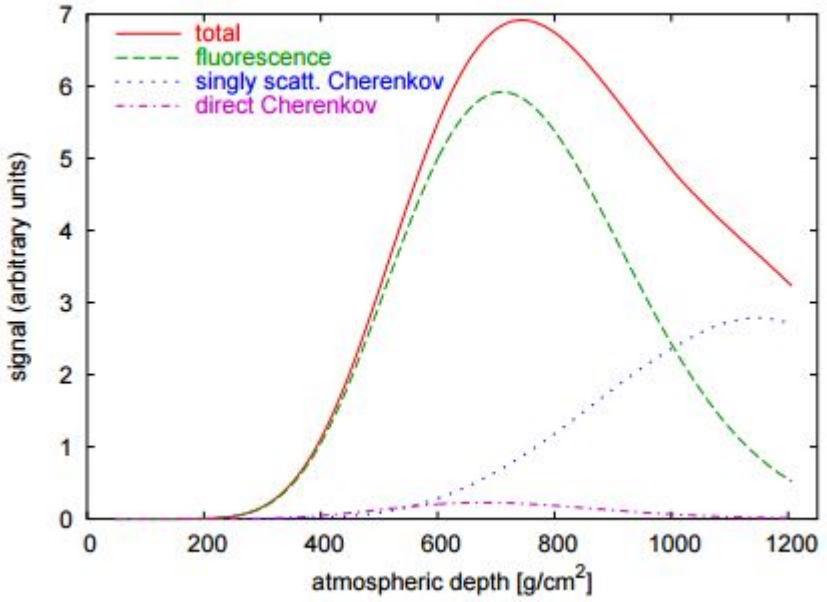

Figure 1.4: Simulated shower profile depicting various detected radiation

1.5 The GZK Cut-Off

The Greisen-Zatspein-Kuzmin (GZK) limit is a theoretical upper limit on the energy of cosmic rays and is taken to be about 5×10^{19}eV. This limiting value is approximately the same order of magnitude as the highest energy cosmic rays that have been detected on Earth [14].

Greisen, Kuzmin and Zatspein independently computed this upper limit based on the apparent interaction of cosmic ray protons and the photons of Cosmic Microwave Background (CMB) radiation. They predicted the interaction as

$$\gamma_{CMB} + p \rightarrow \Delta^{+} \rightarrow p + \pi^{0} \tag{1.14}$$

or as

$$\gamma_{CMB} + p \rightarrow \Delta^{+} \rightarrow n + \pi^{+} \tag{1.15}$$

where cosmic ray protons of energy over the threshold of $5 \times 10^{19}\ eV$ interact with CMB radiation resulting in pions via the Δ resonance. Eventually, the pions and neutrons decay resulting in a number of particles. π^{0} decay to photons, π^{+} decay to photons, positrons and various neutrinos while neutrons decay to give similar products. The combined effect of these decays is such that the energy of any cosmic ray proton is drained off by the production of high energy photons and high energy electron-positron and neutrino pairs so that an upper limit to the incident energy is implemented [6].

1.5.1 GZK Cut-Off Calculation

For calculation purposes, let's consider interaction of protons with CMB radiation to produce π^{+}.

$$\gamma_{CMB} + p \rightarrow \Delta^{+} \rightarrow n + \pi^{+} \tag{1.16}$$

Charge conservation is ensured by the presence of the positively charged pion. However, energy conservation mandates a maximum value of energy that the proton can have in order for the process to be possible.

Energy of an usual CMB photon, whose temperature is presently know to be about 3 kelvin, is given by

$$k_B T = 1.38 \times 10^{-23} \times 3$$
$$= 4.2 \times 10^{-23} J$$
$$= 2.63 \times 10^{-4} eV$$

This energy is very little compared to that of the incoming ultra-high energy proton. Because relativistic particles are on the agenda, the calculations must proceed by writing the 4-momentum of each particle and then squaring them to check for momentum conservation. The notations that will be used are

$\mathbf{p}_p$ = proton 4-momentum in lab frame before collision
$\mathbf{p}_\gamma$ = CMB photon 4-momentum in lab frame before collision
$\mathbf{p}_n$ = neutron 4-momentum in centre-of-mass frame after collision
$\mathbf{p}_\pi$ = pion 4-momentum in centre-of-mass frame after collision

Therefore, momentum is conserved if

$$(\mathbf{p}_p + \mathbf{p}_\gamma)^2 = (\mathbf{p}_n + \mathbf{p}_\pi)^2$$

π^+ and n are both produced at rest and so has total energy $(M_n + M_\pi)c^2$. In the centre-of-mass frame, their total energy is zero and hence, the right hand side of the equation equals

$$(\mathbf{p}_n + \mathbf{p}_\pi)^2 = -(M_n + M_\pi)^2 c^2 \tag{1.17}$$

Now, taking the dot product of the term on the left hand side gives,

$$(\mathbf{p}_p + \mathbf{p}_\gamma)^2 = (\mathbf{p}_p + \mathbf{p}_\gamma).(\mathbf{p}_p + \mathbf{p}_\gamma) = (\mathbf{p}_p)^2 + 2(\mathbf{p}_p)(\mathbf{p}_\gamma) + (\mathbf{p}_\gamma)^2 \tag{1.18}$$

where, all terms on the right are 4-vector dot products. Furthermore, we have

$$(\mathbf{p}_p)^2 = -M_p^2 c^2$$

and

$$(\mathbf{p}_\gamma)^2 = 0$$

Hence, combining equation (1.9) and (1.8) and using the above expressions,

$$-M_p^2c^2 + 2\mathbf{p}_p.\mathbf{p}_\gamma = -(M_n + M_\pi)^2c^2 \tag{1.19}$$

Also, the 4-momentum vectors for the proton and the CMB radiation are

$$\mathbf{p}_p = \begin{pmatrix} E_p/c \\ E_p/c \end{pmatrix} \tag{1.20}$$

and

$$\mathbf{p}_\gamma = \begin{pmatrix} E_\gamma/c \\ -E_\gamma/c \end{pmatrix} \tag{1.21}$$

Therefore, the dot product in (1.10) is evaluated using the matrices and we are left with

$$\mathbf{p}_p.\mathbf{p}_\gamma = -\tfrac{2E_pE_\gamma}{c^2}$$

Finally, by substituting this expression into (1.10), GZK cut-off can be calculated as

$$M_p^2c^2 + 4\frac{E_pE_\gamma}{c^2} = (M_n + M_\pi)^2c^2$$

$$\Rightarrow 4E_pE_\gamma = (M_nc^2 + M_\pi)^2 - (M_pc^2)^2$$

$$\Rightarrow E_p = \frac{(M_nc^2 + M_\pi c^2)^2 - (M_pc^2)^2}{4E_\gamma}$$

In units of c=1, $M_n = 939.6MeV/c^2$, $M_p = 938.3MeV/c^2$ and $M_\pi = 139.6MeV/c^2$ which gives the value of $E_p \approx 3 \times 10^{20}eV$.

However, the energetically favourable process is the interaction of the cosmic ray protons with the CMB radiation that results in π^0 because the final

state particles are lighter. Although it appears that the proton is reproduced in the interaction, energy is transferred to the π^0 and the proton will be about 15% less energetic, falling far below the GZK limit. Taking this and other factors into consideration, the threshold energy is calculated similarly for the $p + \gamma \rightarrow p + \pi^o$ and is found to be about $5 \times 10^{19} eV$ which is known as the GZK cut-off [6].

Theoretically, no cosmic radiation which exceeds this threshold energy, is expected to be detected on Earth as any such cosmic ray particle will scatter off the CMB photons.

1.6 Point Sources of Ultra-High Energy Cosmic Radiation (UHECR)

The highest energy cosmic rays, in and around the knee region in Figure 1.1, of energies exceeding 10^{15}eV, are given the name of Ultra-High Energy Cosmic Radiation (UHECR). UHECR is believed to have both galactic (energies below the knee) and extra-galactic (energies above the knee) origin and only a minute fraction of ultra-energetic particles which exceeds the ankle region are ever detected on Earth. The highest energy UHECR ever detected was by the Fly's Eye experiment in Utah, USA, whose shower energy is estimated to be around $3 \times 10^{20} eV$ which exceeds the GZK cut-off value, contradicting special relativity on which premise the GZK limit was calculated. On an average, detection of UHECR flux of order greater than $10^{20} eV$ is approximately estimated to be 0.5 to 1 event per square kilometre per century per steradians and even large detector arrays, covering tens of square kilometres are able to detect few such events in a span of ten years [16]. Judging by the very few number of events of such extreme energies, discrepancies in detectors and inaccuracy in measuring techniques could be used to argue against the validity of these events. Nonetheless, the possible sources of UHECR are of huge interest, mainly because these extreme energy values could potentially take physics to uncharted territories, some of which will be briefly discussed.

1.6.1 Criterion for Possible UHECR Sources

The uncertainty surrounding the existence of UHECR and their origins lead us to define possible UHECR sources with considerable precaution. Before an astrophysical entity can be touted as a possible candidate, it has to satisfy several prerequisites.

i) **Geometry** : The particle which is accelerated must be maintained withing the object during the accelerating process.

ii) **Power** : The source needs to have sufficient energy which it can transfer to accelerate the particle.

iii) **Radiation Loss** : Energy must be conserved which means that energy gained by the accelerating particle is equal to its radiation energy loss.

iv) **Interaction Loss** : Energy lost by the particle must always be balanced by the energy it loses due to its interaction with other particles.

v) **Emissivity** : The detected radiation flux of UHECR must be compatible with the density and power of the source.

vi) **Coexisting Radiation** : The observed UHECR flux must always be greater than the associated photon, neutrino and the low-energy cosmic radiation flux.

vi) **The Hillas Criterion** : The maximum energy that an accelerating particle can have before it leaves the region it is being accelerated in can be expressed as follows:

$$\varepsilon_{max} = qBR \tag{1.22}$$

where q is the particle's electric charge, B is the magnetic field it is subjected to and R is the size of the accelerating region. Equation (1.22) is subject to a restriction imposed by a general geometrical criterion, named the Hillas Criterion, which demands that the Larmor radius of the accelerating particle (also known as the gyroradius),

$$R_L = \frac{mv}{qB}$$

does not exceed the radius of the accelerating site,

$$R = \frac{\varepsilon}{qB}$$

.

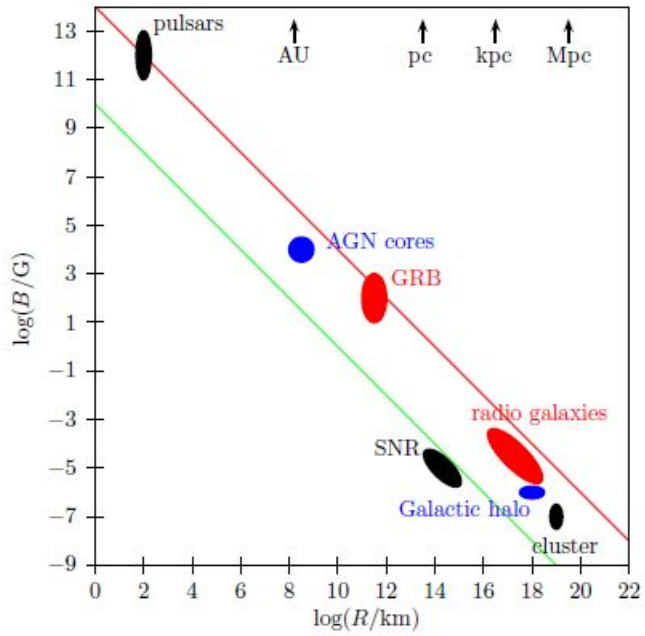

Figure 1.5: Hillas Plot depicting the relationship between magnetic field strength B and R and potential UHECR sites

Figure 1.5 shows how the size of an UHECR source, R, is related to the magnetic field strength, B, of the source for a specific value of ε_{max}. Sources marked above the top (red) line are able to accelerate protons up to $10^{21}eV$ while those marked above the bottom (green) line are capable of accelerating iron up to $10^{20}eV$.

Any potential UHECR source needs to satisfy the aforementioned prerequisites. A source satisfying all these requirements can be of galactic origin, which includes Type II supernovae, pulsars and shock acceleration in supernova remnants, while extragalactic sources consist of Active Galactic Nuclei (AGN) and γ ray bursts ([3]).

1.6.2 Sources of UHECR

This subsection is intended to briefly describe some of the astrophysical objects, both of galactic and extra galactic origin, that are touted as possible UHECR sources.

i) **Pulsars**: Pulsar, a highly magnetised and rotating neutron star, is the remains of a massive star that underwent gravitational collapse, that emits a beam of electromagnetic radiation. It is very dense and has a precise radiation interval and rotational period. The very high magnetic field associated with a pulsar is of order 10^{13} Gauss and has the ability to accelerate charged nuclei up tp energies of $10^{20}eV$ which is why it is a very possible source of UHECR [16].

ii) **Active Galactic Nuclei (AGN)**: A compact region at the centre of a galaxy with a greater than average luminosity, is called an Active Galactic Nucleus (AGN). Luminosity range of an AGN extends mostly all over the electromagnetic spectrum. Any galaxy that hosts an AGN at its centre is known as an Active Galaxy. Quasars, for instance, are associated with AGNs which has time-variant luminosities of enormous magnitudes, typically 10^{13} times that of our Sun (3.9×10^{26} W). Such high luminosities can only result from very high power generated in a very compact region, leaving black holes as the only conceivable object which can do so by accretion of inflowing matter. Hence, any AGN must be associated with a supermassive black hole at the galactic centre. The black hole is surrounded by an accretion disk of matter - gas, dust and stars - which contributes to its growth. The emitted energy is a result of release of gravitational energy when any material in the accretion disk is gobbled up by the black hole. In this manner, about half the mass energy of the accreted matter is released which is enough to generate the aforementioned energy output. In the course of the absorption process, the accretion matter undergoes violent oscillations resulting in them being ionised as a plasma. Plasma is a magneto-fluid and as a result the enormous number of charge carriers are accelerated to very high energies which generates a plasma current and that in turn induces a magnetic field. Such conditions allows the subsequent emission of radiation in the optical, infrared and X-ray frequencies. However, the accretion process cannot go on indefinitely and after the nearby matter is exhausted, the AGN reduces to a normal galaxy containing a quiet black hole at the centre ([14]).

iii) γ **ray bursts (GRB)**: γ ray burst (GRB) is one of the most powerful and yet the least understood explosion phenomena in the Universe, typically of energy of $\approx 10^{44}$ J suggestive of these bursts being associated with a Type II supernova explosion ($\approx 10^{46}$ J) (Perkins, 2009, p. 249). The bursts are quite common with one or two bursts detected everyday by NASA detectors (NASA, 2013). The burst can be categorised into two classes of different origins and durations; short-duration bursts (period < 2 s) and long-duration bursts (period $2 - 10$ s). Only vague ideas about the origin and mechanism have developed about short duration bursts but the very short burst lengths is indicative of extremely compact objects from whence these originate. For instance, a possible origin for short duration bursts are orbiting neutron stars in a binary system when they undergo gravitational collapse, forming black holes. In contrast, long duration bursts are better understood and they originate from events called *collapsars* which arise from the collapse of very massive, rapidly rotating and low-metallicity Wolf-Rayet (W-R) stars ($\approx 20 - 100$ solar masses). Due to their humongous mass, the lifetime of such stars are significantly shorter than that of our Sun. Thus, the W-R stars evolve rapidly, burning silicon at a very high rate, and collapses directly into a black hole when the fuel is exhausted. The resulting black hole sucks in the enormous amount of mass surrounding the accretion disk which is accompanied by a blast of γ-rays carrying out enormous amount of energy [14].

Chapter 2

Cosmic radiation detection and analysis at ALICE

ALICE (A Large Ion Collider Experiment) is one of the seven detector experiments at the Large Hadron Collider (LHC) at CERN. The primary objective of the ALICE experiment is to study the formation of quark-gluon plasma which is a state of matter formed at extreme energies. The protons and neutrons, that make up all matter, consist of particles called quarks which are held together by other particles named gluons. These quarks are confined inside protons and neutrons and due to this *quark confinement* no quark has ever been discovered in isolation. Collision experiments in the LHC are designed under extreme temperatures, greater than 100000 times hotter than the centre of the Sun, recreating conditions similar to those after the Big Bang. Under these extreme laboratory conditions, protons and neutrons *melt* which breaks the bond between the quarks and gluons and gives rise to a phase of matter known as the *quark-gluon plasma*. The existence of this phase and its properties are key to fully comprehend the phenomenon of confinement [4].

Additionally, ALICE contributes in the study of cosmic ray physics with the help of ACORDE(ALICE Cosmic Ray Detector). The ALICE underground cavern makes for a suitable place to enable the detection of high energy atmospheric muons coming from cosmic ray showers. ACORDE detects cosmic ray showers by triggering the arrival of muons to the top of the ALICE magnet. The cosmic ray trigger consists of 60 scintillator modules distributed on the three upper faces of the ALICE magnet. The array can

be configured to detect single to multi-muon events and ACORDE's high resolution enables the recording of cosmic events with very high multiplicity of parallel muon tracks known as 'muon bundles' [**?**].

There are two main aspects which motivated the use of ACORDE to contribute to cosmic ray physics. Firstly, study of cosmics would lead to a better understanding of nuclear interaction at high energies. Particle production at high energies can be estimated based on extrapolation of the accelerator data. Different interaction models are employed for identifying primary cosmic ray particles and interpreting the accelerator data. Currently some of the models used are the VENUS, DPMJET, QGSJET, EPOS, etc for high energy hadronic interactions and GHEISHA, FLUKA and URQMD for low energy hadronic interactions. Explanation of all such models will be given in this paper. The underground location of ALICE with 30 m of overburden composed of sub-alpine Molasses, is an ideal place for muon-based underground experiments. From geological surveys, it is known that the surface above ALICE location is flat within a radius of at least 200 m from the ALICE interaction point and the rock composition is almost uniform, enabling accurate measurements of muon absorption and energy loss for the muons reaching the central magnet with a threshold energy, $E_{\mu th} > 17$ GeV. The motive of these experiments is to observe underground multi-muon events and analyse the large muon-multiplicities of very high energy cosmic rays and to search for rare exotic cosmic ray events making a direct contribution to Cosmic Ray (CR) physics, the second objective of ACORDE [11].

2.1 About the CORSIKA Simulation Program

CORSIKA (Cosmic Ray SImulation for KAscade) is a detailed Monte Carlo program which uses only standard FORTRAN routines to study the evolution and properties of Extensive Air Showers (EAS) in the atmosphere. The program is able to analyse and model the cascade that develops when a high energy particle enters the atmosphere. It was first developed to perform simulations for the KASCADE experiment at Karlsruhe, Germany. It also gives the type, energy, location, direction and arrival times of all the secondary particles created in an EAS passing a specific observation level [5].

The CORSIKA program consists essentially of four parts. The first part

is a general program frame which identifies an input and produces an output, perform decays of unstable particles and tracks the particles taking into consideration ionisation energy loss and deflection by multiple scattering and the Earth's magnetic field. The second part simulates hadronic interactions of the nuclei and hadrons with air nuclei at high energies. The third part simulates hadronic interactions at lower energies while the fourth part enunciates the transport and interaction of electrons, positrons and photons. Varying options can be selected and the simulation steered by choosing different types of interaction models that best suites our interest [10].

2.2 The Monte Carlo Method

Monte Carlo (MC) method is a range of computational algorithms which has a range of uses in many fields including computational physics and astrophysics. The method is very useful in modelling phenomena with significant uncertainties in inputs or in simulating systems with many coupled degrees of freedom. In principle, MC method can be used to solve any problem having a probabilistic interpretation. In air-shower physics, the method is used to generate momentum and spatial coordinates of all shower particles produced in subsequent interactions and track all sub-showers along with the primary shower down to the ground [1].

MC method is employed in the CORSIKA simulation program because of the large uncertainties associated with shower simulations. The goal of the simulation programme is to produce MC data which exactly matches the shower profile from the actual detector. This is ensured by comparing distributions of many observables produced by the experimental data and the MC method [2]. This reconstruction and matching process is time consuming which is precisely why the CORSIKA simulation program requires a large CPU time in processing the output.

2.3 Some Relevant CORSIKA High-Energy Hadronic Interaction Models

2.3.1 DPMJET Version

DPMJET (Dual Parton Model with JETs) is a program developed to describe high energy hadronic interactions of hadron-nucleus and nucleus-neucleus collisions. It takes about the same CPU time as the VENUS model at primary energies of 10^15 eV and it is possible to use DPMJET model up-to the highest energies [10].

2.3.2 EPOS Option

EPOS (Energy conserving quantum mechanical multi-scattering approach, based on Partons, Off-shell remnants and Splitting parton ladders) combines features of the VENUS and QGSJET 01 models to safely extrapolate to higher energies. The most actual version is the EPOS LHC in which the LHC data are considered to constrain the model parameters. It takes about 7.5 times CPU time than the DPMJET version at similar energy range. EPOS also activates all inelastic hadron-nucleus cross-sections and these are calculated by built-in subroutines [10].

2.3.3 HDPM Routines

HDPM routines take a fairly smaller amount of CPU time and are adjusted to experimental data. Proton-proton interactions simulated with HDPM routines dovetail pretty well with the simulations using other models but HDPM fails when it comes to nucleus-nucleus and nucleon-nucleus interactions due to simpler modelling and shortage of experimental data [10].

2.3.4 QGSJET Option

QGSJET (Quark Gluon String model with JETs) is a programme developed to describe high-energy hadronic interactions using the Pomeron parametrization for the elastic hadron-nucleon scattering amplitude. The hadronization process is treated in the quark-gluon string model and the most actual version of the programme, QGSJET-II-04 includes LHC tuned Pomeron loops

and cross-sections. The run time for this routine is about 3 times that of HDPM routines [10].

2.3.5 VENUS Option

VENUS (Very Energetic NUclear Scattering) is a programme that simulates ultra-relativistic heavy-ion collisions. VENUS cross-section are defined automatically while extracting the FORTRAN code when the VENUS option is selected. CPU time is about 15 times more than that of the HDPM routines [10].

2.4 CORSIKA Low-energy Hadronic Interaction Models

2.4.1 FLUKA Option

FLUKA (FLUctuating KAscade) is a set of routines that follow energetic particles through matter by the Monte Carlo method and combined with CORSIKA, only tracks those hadronic interactions which are of low energy. FLUKA is employed in CORSIKA to calculate hadronic cross-sections with the components of air and perform their interactions and track the production of secondary particles including many details about the de-excitation of target nuclei. FLUKA option cannot be combined with the DPMJET option because of several identical subroutines used within FLUKA and DPMJET [10].

2.4.2 GHEISHA Option

GHEISHA (Gamma Hadron Electron Interaction SHower code) is a program designed capable of describing hadronic collisions up-to 100 GeV energy. Among other uses, GHEISHA routine is used in CORSIKA to calculate elastic and inelastic cross-sections of hadrons below 80 GeV in air and their interaction and particle production [10].

2.4.3 URQMD Option

URQMD (Ultra Relativistic Quantum Molecular Dynamics) is designed to describe hadron-nucleus interactions at low energies. Its use in CORSIKA program lies in performing elastic and inelastic hadronic interactions below 80 GeV energy in air [10].

Chapter 3

CORSIKA Simulation Details and Results

This chapter is intended to elucidate on the work carried out by myself and the results obtained. This will include detailed explanation of the CORSIKA parameters that were used and the analysis carried out using ROOT programme. Furthermore, shower profiles and particle distributions are also explained with the help of graphs generated by CORSIKA.

3.1 High-Energy Hadronic Interaction Model EPOS

Many elementary collisions occur in a nucleus-nucleus or proton-proton scattering. These scattering phenomena are called "parton ladder" or the cut Pomeron [12].

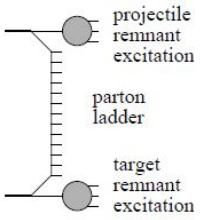

Figure 3.1: Parton-parton scattering

As a simple model, hadron-hadron interaction can be taken as an exchange of a "parton ladder" between the two hadrons. In EPOS, "parton ladder" consists of two parts; a hard scattering between the projectile and the target and a soft scattering component. The EPOS model takes into account energy conservations when calculating cross sections and strings unlike other models, which makes it a consistent quantum mechanical multiple scattering approach based on partons and strings. Furthermore, to enhance the predictive ability of the model, non-linear effects of hadron-hadron interactions have been made more consistent to describe both proton-proton, hadron-nucleus and nucleus-nucleus data as shown in Figure 3.2.[12].

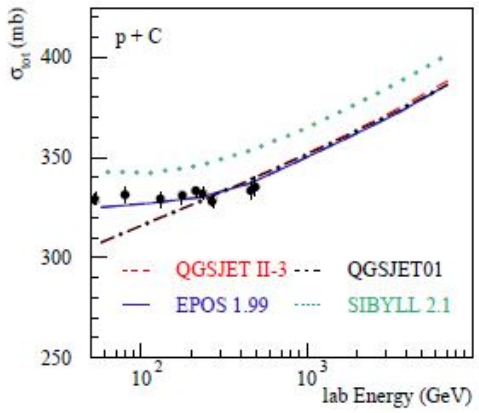

Figure 3.2: Total cross-section of proton-carbon interactions against Energy as compared with different hadronic interaction models

The EPOS option is improved from the older VENUS and QGSJET 01 options in that many technical problems have been solved and the screening effects made more efficient by including the recent Relativistic Heavy Ion Collider (RHIC) data. In addition, high density effects have been considered and the most actual version, used in this work, EPOS LHC has been incorporated with the LHC data in modelling constraint parameters [10].

3.2 Low-Energy Interaction Model UrQMD

The UrQMD model is a microscopic model which describes the phenomenology of hadronic interactions at low and intermediate energies ($\sqrt{s} < 5$ GeV)

in terms of interactions between known hadrons and their resonances. At higher energies, $\sqrt{s} > 5$ GeV, the excitations of colour strings and their subsequent disintegration into hadrons are taken into account in the UrQmd model [17].

The combination of high-energy and low-energy interaction models determine how accurately a shower profile is constructed. All the models mentioned above have their strengths and weakness and so a good combination is important for greater accuracy. UrQMD model takes a longer CPU time than FLUKA model but gives a more accurate ionisation profile [1].

3.3 CORSIKA Input File

After installing CORSIKA and setting EPOS LHC and UrQMD models as high-energy and low-energy interaction models respectively as the modelling parameters for showers incident on a horizontal array of detectors, an input file is generated. This file consists of a number of shower parameters that can be altered for getting different shower profiles. An example of such an input *card* is given in the table below.

Table 3.1: CORSIKA input card setup

Begin of input card		
Parameter	Value	Explanation
RUNNR	xxxx	run number
EVNTNR	x	number of first shower events
NSHOW	xxxx	number of showers to generate
PRMPAR	xx	particle type of primary particle
ESLOPE	-2.7	slope of primary spectrum
ERANGE	1.E5 1.E6	energy range of primary particle [GeV]
THETAP	0. 50.	range of zenith angle (in degrees)
PHIP	0. 360.	range of azimuth. angle (in degrees)
SEED	x 0 0	seed for 1. random
SEED	x 0 0	seed for 2. random
SEED	x 0 0	seed for 3. random
OBSLEV	479.E2	observation level CERN (in cm)
FIXHEI	0. 0.	first interaction height and target
FIXCHI	0.	starting altitude (g/cm^2)

Continuation of input card 3.1		
Parameter	Value	Explanation
MAGNET	22.1 41.6	magnetic field CERN
HADFLG	0 0 0 0 0 2	flags hadr. interact. and fragmentation
EPOS	T 0	EPOS as hadronic interaction model
EPOSIG	T	EPOS hadronic cross-sections
ECUTS	1.0 1.0 0.003 0.003	energy cuts had mu e photon
MUADDI	T	additional info for muons
MUMULT	T	muon multiple scattering angle
ELMFLG	T F	em. interaction flags
STEPFC	10.	mult. scattering step length factor
RADNKG	200.E2	outer radius for NKG lat.dens.distr.
ARRANG	0.	rotation of array to north
LONGI	F 20. T F	longit.distr. and step size and fit and out
ECTMAP	1.E4	cut on gamma factor for printout
MAXPRT	xxx	max. number of printed events
DIRECT	./	output directory
DATBAS	T	write .dbase file
PAROUT	T/F T/F	
USER	name	user
DEBUG	T/F x T/F xxxxxxx	debug flag and log.unit for out
End of Input card		

The values marked with "x's" are the parameters that can be changed to obtain different shower profiles. Note that, in the table, EPOS was marked as T (True). If any other hadronic interaction model were used, for instance QGSJET, that would have taken the place of EPOS in the input file.

A modified input file for simulation of 500 shower events using the EPOS LHC hadronic interaction model is given which was used in this work. Proton shower was simulated by inputing 14 in the PRMPAR portion of the input card. A wide range of other particles can also be chosen using appropriate entry numbers, for instance, for iron it is 5626.

Table 3.2: CORSIKA input card Used

Begin of input card		
Parameter	Value	Explanation
RUNNR	001300	run number
EVNTNR	1	number of first shower events
NSHOW	500	number of showers to generate
PRMPAR	14	particle type of primary particle
ESLOPE	-2.7	slope of primary spectrum
ERANGE	1.E5 1.E6	energy range of primary particle [GeV]
THETAP	0. 50.	range of zenith angle (in degrees)
PHIP	0. 360.	range of azimuth. angle (in degrees)
SEED	9 0 0	seed for 1. random
SEED	3 0 0	seed for 2. random
SEED	6 0 0	seed for 3. random
OBSLEV	479.E2	observation level CERN (in cm)
FIXHEI	0. 0.	first interaction height and target
FIXCHI	0.	starting altitude (g/cm^2)
MAGNET	22.1 41.6	magnetic field CERN
HADFLG	0 0 0 0 0 2	flags hadr. interact. and fragmentation
EPOS	T 0	EPOS as hadronic interaction model
EPOSIG	T	EPOS hadronic cross-sections
ECUTS	1.0 1.0 0.003 0.003	energy cuts had mu e phot
MUADDI	T	additional info for muons
MUMULT	T	muon multiple scattering angle
ELMFLG	T F	em. interaction flags
STEPFC	10.	mult. scattering step length factor
RADNKG	200.E2	outer radius for NKG lat.dens.distr.
ARRANG	0.	rotation of array to north
LONGI	F 20. T F	longit.distr. and step size and fit and out
ECTMAP	1.E4	cut on gamma factor for printout
MAXPRT	2	max. number of printed events
DIRECT	./	output directory
DATBAS	T	write .dbase file
PAROUT	T F	
USER	alessand	user
DEBUG	F 6 F 1000000	debug flag and log.unit for out
End of Input card		

3.4 Results

With the corresponding inputs discussed in the preceding section, CORSIKA generated three output files namely *DAT 001300*, *DAT 001300.dbase* and *p-1014-1015-100.out*. The first two files contain information about which models and parameters were used, while the third file is the file which contains information about the shower profile.

In the process of generating DAT 001300, a ROOT macro is subsequently generated by CORSIKA. This macro is used to read DAT 001300 using ROOT programme. This is the final process in the analysis of the shower and it generates a number of graphs which provides various features of the simulated shower. The ROOT macro is written in C++ language and the detailed coding is given in the appendix of this paper.

The graphs generated are discussed below:

i) Comparison of Detector Core at surface level and ALICE level:

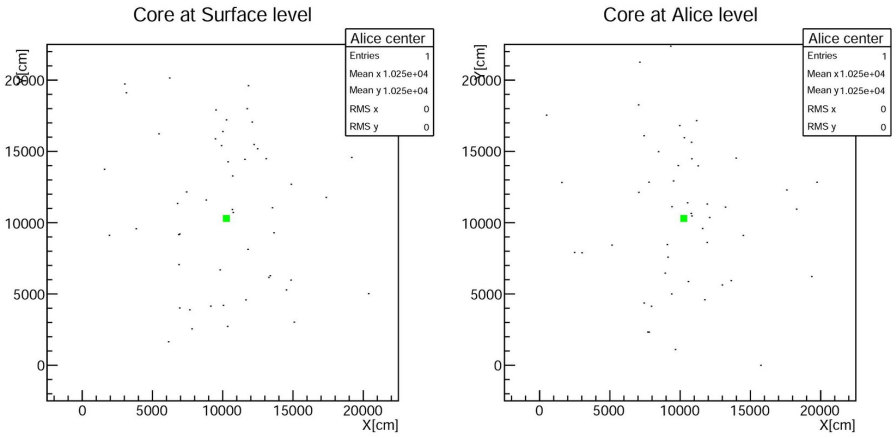

Figure 3.3: Surface level Vs ALICE level

This graph is a comparison of detected muons at surface level detectors and detectors at ALICE which is an underground facility. Both graphs show a similar profile which suggests that most muons detected at surface level are energetic enough to penetrate to the deep levels of the ALICE detectors.

ii) Muon distribution in correlation with detector area:

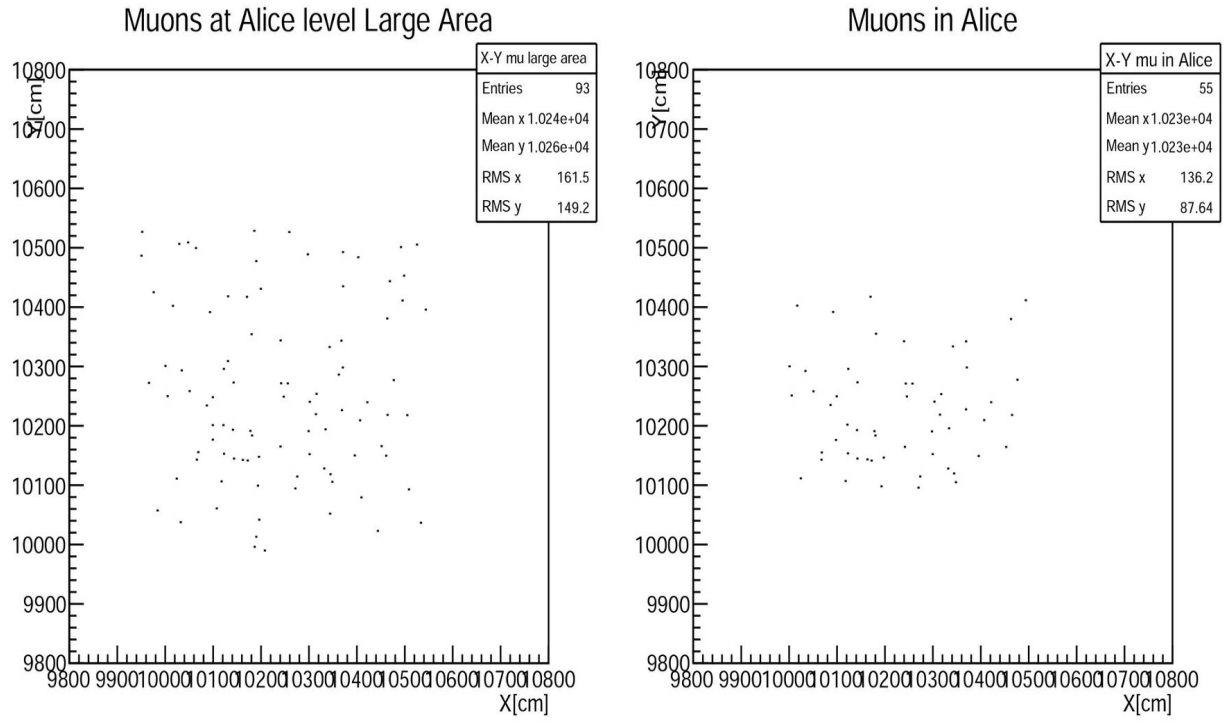

Figure 3.4: Muon distribution and relation with detector area

This graph is a comparison that between detection of muons in relation to detector area. In the ALICE experiment, there is a in-built muon reconstructor which can identify and track muons. Moreover, the ACORDE detector, used to detect energetic cosmic ray particles, is located over a vast area underground. The graphs demonstrate that larger the area, greater the number of muons detected.

iii) Relation of muon energy with number of events

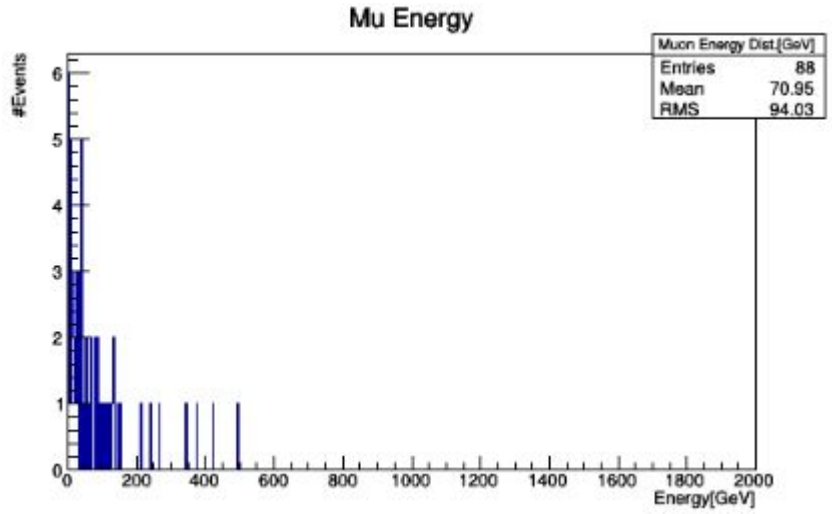

Figure 3.5: No. of Events vs Muon Energy

This graph essentially relates the incident muon energy with the number of interactions it is subjected to on its trajectory. It can be seen that, higher the muon energy, the lower number of events initiated by that muon. It is logical because the highest energy muons pass straight through to the detector without much interaction due to the fact that they do not show strong interaction.

iv) Muon multiplicities at energies above the knee

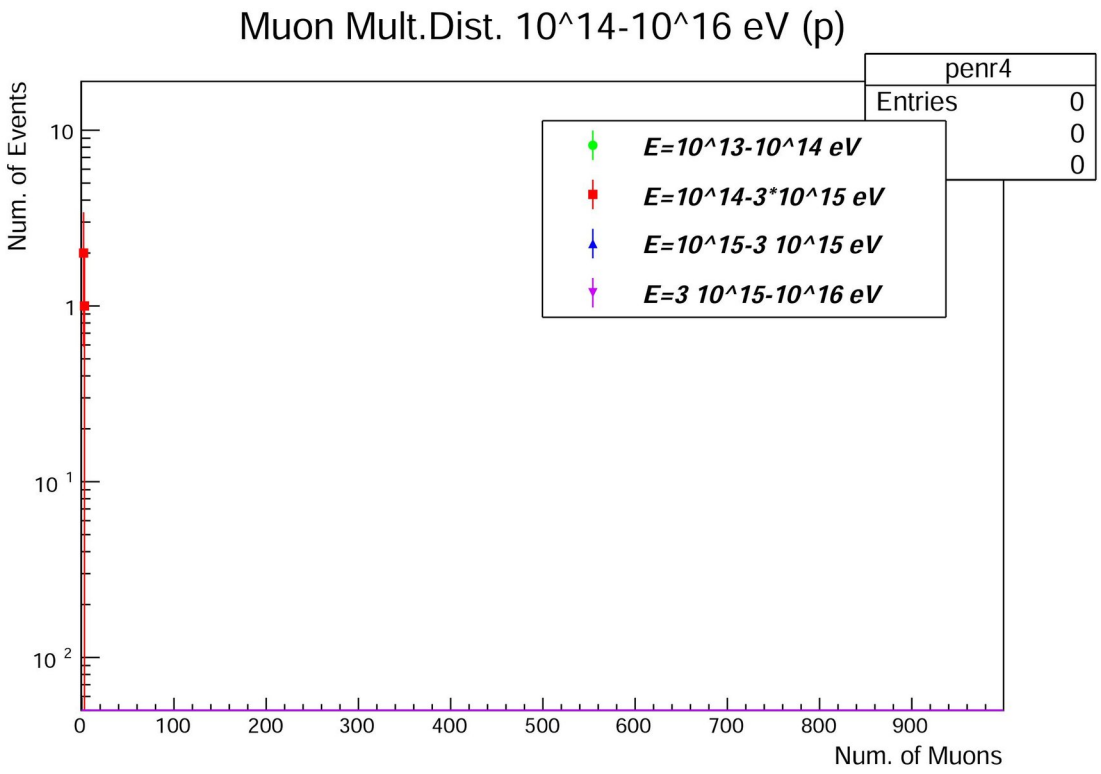

Figure 3.6: Muon multiplicity distribution generated by CORSIKA

This graph intends to establish a relation between the number of muons and the number of shower events for different energy ranges. Following from the previous graph, this graph is in good agreement. The greater the energy range of the muons, the least number of interactions it is subjected to and hence generates the lowest number of subsequent events.

Chapter 4

Conclusion

To summarise, the objectives of this thesis project were to simulate cosmic ray showers, investigate the muonic components in the shower profile and build a relation between the primary shower energy and the consequent events that occur as a result of particle interactions in the atmosphere. Simulation was done using CORSIKA, a complex Monte Carlo simulation program and the analysis of the shower profile was done using ROOT, a programme developed by CERN for detailed data analysis. The muons generated in the shower simulation were separated from the other other shower secondaries and a relation between their energies and the number of interaction events they give rise to.

Although a significant amount of time was taken in undertaking this research project, there were limitations to the generated shower data. The enormous amount of CPU time that is required to generate a single initiating event was nearly twelve hours, which was interrupted several times due to numerous power cuts. If the initiating event number were greater than one and if a larger number of secondary showers were used in the input command, the shower profiles would have looked more healthy. The subsequent data analysis would also have returned graphs more enriched than the existing graphs.

Appendix A: CORSIKA Code

```
#include <string>
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <TFile.h>
#include <TCanvas.h>
#include <TTree.h>
#include "aliana_corsika.h"
using namespace std;


extern void AnaEv(int n);

Int_t nn;
Float_t run,slope,nev,part_id,eneprim,emin,emax;
Float_t eneprimPeV,enerefPeV;  //ak
Float_t gammagen,gammaknee,deltagenknee,deltakneegen; //ak
Int_t tagg30;    //ak
Float_t rapene;  //ak
Int_t num_ev_tot,num_ev_run,num_ev_ene;
Int_t num_ev_tot_g30,num_ev_run_g30,num_ev_ene_g30; //ak
Int_t num_ev_test;
Float_t
    tetaprim,phiprim,tetaprimgrad,phiprimgrad,phiprimpos,elle,emme,enne;
Float_t xdeep,ydeep;
Int_t nmutdis[LOOPCORE];
```

```
29 Float_t primev;
30 Float_t
       nmu[ILOOP],tnmu[ILOOP],tdensmu[ILOOP],emur[ILOOP],temur[ILOOP];
31 Float_t tnmu_app_zero[ILOOP],tnmu_core_zero[ILOOP];
32 Float_t tnmu_app_ecut[ILOOP],tnmu_core_ecut[ILOOP];
33 Float_t dtnmu_app_zero[ILOOP],dtnmu_core_zero[ILOOP];
34 Float_t dtnmu_app_ecut[ILOOP],dtnmu_core_ecut[ILOOP];
35 Float_t distmuapp,distmu,distmureal;
36 Float_t nmutot,sizenkg,nmut15;
37 Float_t rag[ILOOP+1],area[ILOOP];
38 Float_t media_nmutot,media_nmut15,media_sizenkg;
39 Float_t err_media_nmutot,err_media_nmut15;
40
41 Float_t ltot=20500;    // total lenght in X and Y of the surface
       above Alice in cm.
42 const Int_t maxmu=4000; // max. number of muons forseen inside
       large area around Alice
43 Float_t nmuali;        // number of muons inside large area around
       Alice (6x6 m**2)
44 Int_t  idchmu[maxmu];     // identify charge of the muon
45 Int_t  tagmucell[maxmu];  // tagmu=1 muon in Alice
46 Float_t emucell[maxmu];
47 Float_t tmucell[maxmu];
48 Float_t xmucell[maxmu];
49 Float_t ymucell[maxmu];
50 Float_t xmucellsurf[maxmu];
51 Float_t ymucellsurf[maxmu];
52 Float_t pxmucell[maxmu];
53 Float_t pymucell[maxmu];
54 Float_t pzmucell[maxmu];
55 Float_t pmucell[maxmu];
56 Float_t tetamucell[maxmu];
57 Float_t phimucell[maxmu];
58 Float_t aemucell;
59 Float_t aemuev;
60 Float_t dtmucell;
61
62 Int_t  tagmu[maxmu];  // tagmu=1 muon in Alice
63 Float_t emuev[maxmu];
64 Float_t dtmuev[maxmu];
```

```
65  Float_t xmualiev[maxmu];
66  Float_t ymualiev[maxmu];
67  Float_t xmusurev[maxmu];
68  Float_t ymusurev[maxmu];
69  Float_t pxmuev[maxmu];
70  Float_t pymuev[maxmu];
71  Float_t pzmuev[maxmu];
72  Float_t pmuev[maxmu];
73  Float_t tetamuev[maxmu];
74  Float_t phimuev[maxmu];
75  Float_t tmuevmin;
76  Float_t adtmuev;
77  Int_t iasmin;
78  Int_t idcell,nmuev,nmuvec;
79  Float_t xco,yco; /* x core, y core */
80  Int_t nmuinali; // number of muons inside Alice (5 x 3 m**2)
81
82  Float_t a1;
83  Float_t rea[50];
84  Int_t i;
85
86  union heads{
87    Float_t header ;
88    Char_t headname[4] ;
89  }   head ;
90
91  //Float_t buf[21][273] defined in aliana_corsika.h
92  Float_t firstlast ;
93  FILE *bufana;
94
95  Char_t infilnam[150];
96
97  Int_t main(){
98
99    Int_t loopf;
100   Int_t loopf_ene;  //ak
101
102   //  Area = 205 x 205 m**2 0<teta<50 0<phi<360 solid angle~1.8 sr
103   //  All-particle gamma=-2.7 gammaknee=-3.0
104
```

```
105 //   Index gamma of the generation (CORSIKA) and our gamma of the
        knee
106   gammagen = -2.7; //ak  gamma used in Corsika
107   gammaknee = -3.0; //ak gamma used after the knee
108   deltagenknee = gammagen-gammaknee;  //ak
109   deltakneegen = gammaknee-gammagen;  //ak
110
111   //  Input for analyze some simulated runs
112   const Int_t nfile1=1; // first file to be analysed
113   const Int_t nfile2=1; // last file to be analysed
114   const Int_t nfilevec=30; // maximum number of files
115   Int_t nevf[nfilevec]; //max. number of events per file
116   Int_t nevf_ene[5]; //ak max num. of events for each range of
         energy
117
118
119     nevf_ene[0]=1000000; // ak gamma=-3.0 3 10**15-10**16 files:0-5
120     nevf_ene[1]=0; // ak 10**16-3 10**16       files:6-9
121     nevf_ene[2]=0; // ak 3 10**16-10**17       files:10-13
122     nevf_ene[3]=0;  // ak 10**17-3 10**17      files:14-15
123     nevf_ene[4]=0;   // ak 3 10**17-10**18     files:16
124
125   for(i=0; i<nfilevec; i++){
126     nevf[i]=10000000;
127   }
128
129
130 //  Ntupla:
131        // 1-loopf -> to identify the input file created by Corsika
132        // 2-part_id -> primary cosmic ray p=14, Fe=5626
133        // 3-eneprim -> energy of the primary [GeV]
134        // 4-tetaprimgrad -> zenith angle of the primary in degree
135        // 5-phiprimgrad -> azimuth angle of the primary in degree
136             // 6-num_ev_run -> number of the event in the actual
                   input file
137             // 7-xco -> X of the core at surface level [cm]
138             // 8-yco -> Y of the core at surface level [cm]
139             // 9-xcoali -> X of the core at Alice level [cm]
140             // 10-ycoali -> Y of the core at Alice level [cm]
```

```
141             // 11-nmuev -> num. of muons arriving in an area 6 x 6
                    m**2 around Alice
142             // 12-nmuinali -> num. of muons inside Alice 5 x 3.4 m**2
143             // 13-tagmu[nmuev] ->tagmu=0 outside Alice,tagmu=1
                    inside Alice(for each mu)
144             // 14-emuev[nmuev] -> mu energy (for each mu)
145             // 15-dtmuev[nmuev] -> arriving time of the mu (for each
                    mu)
146             // 16-xmualiev[nmuev] -> X of the mu at Alice level (for
                    each mu)
147             // 17-ymualiev[nmuev] -> Y of the mu at Alice level (for
                    each mu)
148             // 18-xmusurev[nmuev] -> X of the mu at surface level
                    (for each mu)
149             // 19-ymusurev[nmuev] -> Y of the mu at surface level
                    (for each mu)
150             // 20-pxmuev[nmuev] -> Px of the mu at surface (for each
                    mu)
151             // 21-pymuev[nmuev] -> Py of the mu at surface (for each
                    mu)
152             // 22-pzmuev[nmuev] -> Pz of the mu at surface (for each
                    mu)
153             // 23-pmuev[nmuev] -> P of the mu at surface (for each
                    mu)
154             // 24-tetamuev[nmuev] -> zenith angle of the mu (for
                    each mu)
155             // 25-phimuev[nmuev] -> azimuth angle of the mu (for
                    each mu)
156             // 26-aemuev -> average energy of the muon in the event
157             // 27-adtmuev -> average arrival time of the muon in the
                    event
158             // 28-idchmu[nmuev] -> charge of the muon (for each
                    muon) (added later)
159
160 //    Prepare the TREE
161   TFile *ftree1 = new
          TFile("ptest_tree_31015_1016_1.root","RECREATE","alicors_tree");
162   TTree *ev_tree = new TTree("AliCorsEv","AliCors_ev_tree");
163   ev_tree->Branch("loopf",&loopf,"loopf/I");
164   ev_tree->Branch("part_id",&part_id,"part_id/F");
```

```
165   ev_tree->Branch("eneprim",&eneprim,"eneprim/F");
166   ev_tree->Branch("tetaprimgrad",&tetaprimgrad,"tetaprimgrad/F");
167   ev_tree->Branch("phiprimgrad",&phiprimgrad,"phiprimgrad/F");
168 //  ev_tree->Branch("num_ev_run",&num_ev_run,"num_ev_run/I");
169   ev_tree->Branch("num_ev_run",&num_ev_run_g30,"num_ev_run_g30/I");
        //ak
170   ev_tree->Branch("xco",&xco,"xco/F");
171   ev_tree->Branch("yco",&yco,"yco/F");
172   ev_tree->Branch("xcoali",&xdeep,"xdeep/F");
173   ev_tree->Branch("ycoali",&ydeep,"ydeep/F");
174   ev_tree->Branch("nmuev",&nmuev,"nmuev/I");
175   ev_tree->Branch("nmuinali",&nmuinali,"nmuinali/I");
176   ev_tree->Branch("tagmu",&tagmu,"tagmu[nmuev]/I");
177   ev_tree->Branch("emuev",emuev,"emuev[nmuev]/F");
178   ev_tree->Branch("dtmuev",dtmuev,"dtmuev[nmuev]/F");
179   ev_tree->Branch("xmualiev",xmualiev,"xmualiev[nmuev]/F");
180   ev_tree->Branch("ymualiev",ymualiev,"ymualiev[nmuev]/F");
181   ev_tree->Branch("xmusurev",xmusurev,"xmusurev[nmuev]/F");
182   ev_tree->Branch("ymusurev",ymusurev,"ymusurev[nmuev]/F");
183   ev_tree->Branch("pxmuev",pxmuev,"pxmuev[nmuev]/F");
184   ev_tree->Branch("pymuev",pymuev,"pymuev[nmuev]/F");
185   ev_tree->Branch("pzmuev",pzmuev,"pzmuev[nmuev]/F");
186   ev_tree->Branch("pmuev",pmuev,"pmuev[nmuev]/F");
187   ev_tree->Branch("tetamuev",tetamuev,"tetamuev[nmuev]/F");
188   ev_tree->Branch("phimuev",phimuev,"phimuev[nmuev]/F");
189   ev_tree->Branch("aemuev",&aemuev,"aemuev/F");
190   ev_tree->Branch("adtmuev",&adtmuev,"adtmuev/F");
191   ev_tree->Branch("idchmu",&idchmu,"idchmu[nmuev]/I");
192
193
194 //        Plot Multiplicity, Energy, DTime events with Nmu>=0 or
      Nmu>0
195   TFile *froot = new
        TFile("ptest_plot_31015_1016_1.root","RECREATE","alicors_plot");
196   TH1F *h1=new TH1F("Muon Multiplicity Dist.","Muon Mult. Nmu>=0;
        Num.Muons; #Events",1002,-1.5,2000.5);
197   TH1F *h2=new TH1F("Muon Mult. Dist. Nmu>0","Muon Mult. Nmu>0;
        Num.Muons; #Events",1002,-1.5,2000.5);
198   TH1F *h6=new TH1F("Muon Mult. Dist. in ALICE Nmu>0","Muon Mult.
        in ALICE Nmu>0; Num.Muons; #Events",1002,-1.5,1000.5);
```

```
199   TH1F *h3=new TH1F("Average Muon Energy Dist.[GeV]","Av.Mu Energy;
          Energy[GeV];#Events",1000,0.,2000.);
200   TH1F *h5=new TH1F("Muon Energy Dist.[GeV]","Mu Energy;
          Energy[GeV];#Events",1000,0.,2000.);
201   TH1F *h4=new TH1F("Muon DTime Dist. [psec]","Time Of Flight;
          Dtime[ps]; #Events",1000,0.,10000.);
202 //       Energy of the primary all the events
203 TH1F *heneprim=new TH1F("Eneprim","Primary Energy Spectrum;
      E[PeV]; Num.of events",1000,0,100);
204 //  TH2F *hc1=new TH2F("Cell Distribution","Cell Distribution; X;
      Y",50,-5,45,50,-5,45);
205 //  TH2F *hc2=new TH2F("LEGO PLOT Nmu per
      cell","",50,-5,45,50,-5,45);
206   TH2F *hc3=new TH2F("X and Y Core Distribution","Core
          Distribution; X; Y",500,-2500.,22500.,500,-2500.,22500.);
207 //  TH2F *hmusd=new TH2F("X and Y Muons","Muon Spatial Distribution
      in Alice; X; Y",500,-2500.,22500.,500,-2500.,22500.);
208
209
210   //Some initializations for counting
211   num_ev_tot = 0;
212   num_ev_run = 0;
213   num_ev_test = 0;
214   num_ev_ene = 0;
215   num_ev_tot_g30 = 0; //ak
216   num_ev_run_g30 = 0; //ak
217   num_ev_ene_g30 = 0; //ak
218   media_nmutot = 0 ;
219   media_nmut15 = 0 ;
220   media_sizenkg = 0 ;
221
222 //  Initialization for lateral density of muons [in cm]
223   for(i=0; i<=ILOOP; i++){
224     rag[i]= 200*i;
225   }
226   for(i=0; i<ILOOP; i++){
227     area[i]=PIGR * ((rag[i+1]*rag[i+1])-(rag[i]*rag[i]));
228     tnmu_app_zero[i] = tnmu_core_zero[i] = 0;
229     tnmu_app_ecut[i] = tnmu_core_ecut[i] = 0;
230   }
```

```
231
232   // Initialize random seed (defined in library time.h)
233   srand(time(NULL));
234   Float_t randvalue;
235
236   ifstream inFileC;
237   bool eof();
238
239   bufana = fopen("ptest_bufana_31015_1016_1.dat","wt"); // info and
          summary run (input file Corsika)
240
241
242 //            Here START the loop on RUN (FILES)
243   //  Open Input File (binary file written from fortran) C++ style
244   for(loopf=nfile1;loopf<nfile2+1;loopf++){ //Starting loop on
          input files
245
246
247 //---------------- Example input : many files --------------------
248     if(loopf==1){
249       sprintf(infilnam,"DAT001300");
250       num_ev_ene=0; // Number of events of the same range of energy
251       num_ev_ene_g30=0; //ak Num. ev. same energy range (gamma=-3.0
             put knee)
252       loopf_ene=0;  //ak Range of energy
253       enerefPeV=3; //ak Reference energy for this range in PeV
             (lowervalue)
254     }
255     if(loopf==2){
256       sprintf(infilnam,"DAT001301");
257       loopf_ene=0;  //ak Range of energy
258       enerefPeV=3; //ak Reference energy for this range in PeV
             (lowervalue)
259     }
260
261
262      inFileC.open(infilnam,ios::binary);
263      fprintf (bufana,"\n %d) ====== OPEN FILE : %s ========
            \n",loopf,infilnam);
264      cout << loopf << ") OPEN FILE : " << infilnam << endl;
```

```
265       nev = 0 ; // Event number given by Corsika buf[n][1]
266       num_ev_run=0; // Number of events and event number in the run
              (file)
267       num_ev_test=0;
268       num_ev_run_g30=0; //ak Num. ev. with gamma=-3.0 in the run
              (put the knee)
269
270
271     if (inFileC.bad()){
272       fprintf (bufana,"Open input file ERROR unable to open it :
              loopf = %d \n",loopf);
273       return(1);
274     }
275
276
277     Int_t whilecount=0 ;
278
279     //  loop to read all the events in the open file DAT%%%%%% or
            input file
280
281     while(1) {
282
283       whilecount++;
284
285       //  Read a record that is 21 x 273 (n=0->20 l=0->272)
286       //  words. Each word 4 bytes
287       //  but in DAT%%%%%% first and last word
288       //  of a record has to be skipped and record
289       //  is 21 x 275 words (due to fortran write and c read).
290       //  In DAT%%%%%% buf[0][0] has to be read twice the first has
              no meaning
291       //  In DAT%%%%%% buf[20][272] has to be read twice the last
              has no meaning
292
293
294   //    -------- Here is the START to READ a record ------------
295
296       for (int n=0;n<21;n++) {
297         for (int l=0;l<273;l++) {
298
```

```
299     // Initialize buf
300          buf[n][l]=0;
301
302     if(n==0 && l==0){
303       // Read first element of the record that is without meaning
              and put it in variable firstlast
304       inFileC.read((char*)&firstlast,sizeof(Float_t));
305     }
306
307          // Read good data of DAT%%%%%% and put it in the variable
              buf[n][l]
308     inFileC.read((char*)&buf[n][l],sizeof(Float_t));
309
310     // Read last element of the record that is without meaning and
         put it in variable firstlast
311     if(n==20 && l==272){
312       inFileC.read((char*)&firstlast,sizeof(Float_t));
313     }
314
315          if(l==0)head.header = buf[n][l] ;
316
317      }     //   END for (int l=0;l<273;l++)
318      }      //    END for (int n=0;n<21;n++)
319
320      //     ---------- Here is the END to READ a record -------
321
322      //       ******* START  ANALYSING a record ***********
323      //   Analazing a record buf[21][273] buf(0:20-0:272) 21 x 273
             words in c
324
325      for (int n=0;n<21;n++) {
326
327        head.header = buf[n][0] ;
328
329
330        if(strncmp(head.headname,"RUNH",4)==0) {
331          run = buf[n][1];
332          slope = buf[n][15];
333          emin = buf[n][16];
334          emax = buf[n][17];
```

```
335       }       // end RUNH
336
337       //     if (blkt1==eEVTH) {   //fort.8
338
339       if(strncmp(head.headname,"EVTH",4)==0) {
340
341         if(nev==0) primev = buf[n][1];
342    nev  = buf[n][1];
343         num_ev_test ++;
344  // num_ev_run ++;
345
346
347    part_id = buf[n][2];
348    // Ev_prim.EvNum = (int)num_ev_run;          // event number
349    // Ev_prim.PrPart = (int)part_id;    // primary particle
350    // Ev_prim.PrEne = (double)buf[n][3]; // energy of the primary
351         eneprim = buf[n][3]; // primary energy
352    tetaprim = buf[n][10]; // teta primary in radiant
353    phiprim = buf[n][11]; // phi primary in radiant from -pigreco to
         +pigreco
354         tetaprimgrad = tetaprim/degree;
355         phiprimgrad = phiprim/degree;
356         phiprimpos = phiprimgrad;
357         if(phiprimpos<0)phiprimpos=phiprimpos+360;
358
359         heneprim->Fill(eneprim/1000000); // Primary Energy in [PeV]
360
361  //    This part is to get a energy spectrum above the knee (E=3
       PeV) with
362  //    a slope gamma=-3.0. The generation with Corsika has been
       done with
363  //    gamma=-2.7 so to obtain a correct spectrum with gamma=-3.0
       we have
364  //    to cut some events.
365
366  //              Part after the knee ak
367  //     Tag if the event is good for gamma=-3.0 cut some events to
       get the knee
368  //     Good event tagg30=1, cut the event tagg30=0
369         tagg30 = 0; //ak
```

```
370         eneprimPeV=eneprim/1000000; //ak Primary Energy in PeV
371         rapene=TMath::Power(eneprimPeV,deltakneegen)*TMath::Power(enerefPeV,deltagenknee
                //ak ratio to choose
372         // Take a random number between 0 and 1
373         randvalue = rand()%32000+1; //ak integer random number
                between 1-32000
374         randvalue /= 32000; //ak random value between ~0 and 1.
375       if(rapene>=randvalue){  //ak
376             tagg30=1;              //ak
377             num_ev_run_g30 ++;     //ak
378             num_ev_ene_g30 ++; //ak Num.ev.in the same energy range
                  g30
379 //          heneprim_g30->Fill(eneprim/1000000); //ak Primary Energy
        in [PeV]
380             if(num_ev_run_g30%100==0){
381               printf ("Loopf= %d, Loopf_ene= %d, NUM_EV_RUN_G30= %d,
                      nev = %f, NUM_SAME_ENE_G30= %d Max.n.ev.same
                      en.range= %d
                      \n",loopf,loopf_ene,num_ev_run_g30,nev,num_ev_ene_g30,nevf_ene[loopf_ene]
382               printf ("nevf_ene[0]= %d, nevf_ene[1]= %d, nevf_ene[2]=
                      %d, nevf_ene[3]= %d, nevf_ene[4]= %d
                      \n",nevf_ene[0],nevf_ene[1],nevf_ene[2],nevf_ene[3],nevf_ene[4]);
383
384              printf ("*****************************************\n
                     \n");
385         }
386           } // END if(rapene>=randvalue)
387 //              End part after the knee ak
388
389
390       elle = sin(tetaprim)*cos(phiprim);
391       emme = sin(tetaprim)*sin(phiprim);
392       enne = cos(tetaprim);
393
394           if (nev==primev) {
395         fprintf(bufana,"Event Number nev : %f \n",nev);
396         fprintf(bufana,"Run Number = %f, Slope en. spectrum = %f
              \n",run,slope);
397         fprintf(bufana,"Emin = %f, Emax = %f \n",emin,emax);
```

```
398           fprintf(bufana,"Range of theta: %f
                  %f\n",buf[n][80],buf[n][81]);
399       fprintf(bufana,"Range of phi : %f %f\n",buf[n][82],buf[n][83]);
400       fprintf(bufana,"NKG radial distr. range in cm
              %f\n",buf[n][146]); // 147 ?
401           fprintf(bufana,"Event Number = %d, Prim. Particle = %f,
                  Energy %f \n",num_ev_run+1,part_id,eneprim);
402           fprintf(bufana,"Theta = %f, Phi = %f
                  \n",tetaprimgrad,phiprimgrad);
403         }
404
405         // Take randomly the core coordinate xco,yco inside the
                sampling area
406         randvalue = rand()%32000+1; //integer random number between
                1-32000
407         randvalue /= 32000; /*random value between ~0 and 1.*/
408       xco = randvalue * ltot;
409         randvalue = rand()%32000+1; //integer random number between
                1-32000
410         randvalue /= 32000; /*random value between ~0 and 1.*/
411       yco = randvalue * ltot;
412       hc3->Fill(xco,yco,1.); // position of the core
413
414         // Initialize at the beginning of the event the vectors for
                muons
415         nmuali=0;
416         nmuinali=0;
417         for(Int_t imu=0;imu<maxmu;imu++){
418           idchmu[imu]=0;
419           tagmucell[imu]=0;
420           emucell[imu]=0;
421           tmucell[imu]=0;
422           xmucell[imu]=0;
423           ymucell[imu]=0;
424           xmucellsurf[imu]=0;
425           ymucellsurf[imu]=0;
426           pxmucell[imu]=0;
427           pymucell[imu]=0;
428           pzmucell[imu]=0;
429           pmucell[imu]=0;
```

```
430           tetamucell[imu]=0;
431           phimucell[imu]=0;
432         }
433
434        //   zdeep is the position of Alice in z (-4000 cm ==>
               underground)
435        // coordinates xdeep, ydeep of the core at z=zdeep cm. All
               in cm.
436         xdeep = elle * (zdeep/enne)+xco; // X coordinate of the
               core underground
437         ydeep = emme * (zdeep/enne)+yco; // Y "       "  "    "     "
438
439     // Initialize muon counters
440         nmutot=nmut15=0;
441         for (int k=0;k<LOOPCORE;k++)
442       nmutdis[k] = 0;
443     for (int k=0;k<ILOOP;k++)
444       nmu[k] = tnmu[k] = emur[k] = temur[k] = 0.;
445
446       }   // END if(strncmp(head.headname,"EVTH",4)==0) END EVTH
447
448       if(strncmp(head.headname,"EVTE",4)==0) {
449         //  fprintf(bufana," ============= Found EVTE ============
               Block \n");
450
451         sizenkg = buf[n][184]; // Size=electr.numb. at obs.level
               g/cm**2 (10 level)
452
453     // Fill the HISTO at the end of the event
454     // fprintf(bufana,"Real Event Number in the RUN %f
           \n",num_ev_run);
455         h1->Fill(nmuali,1.); // muon mult.distr. Nmu>=0
456         nmuev=int(nmuali); // number of muons in the events
457         nmuvec=nmuev-1; // num. of muons used as index vector
458         aemuev=0;    // average muon energy
459         adtmuev=0;   // average delta time among muons
460
461 //    CUT THE EVENT TO HAVE A POWER LAW GAMMA=gammaknee (tagg30=1)
462 //            instead gamma=-2.7 that is the generation
463       if(tagg30==1){ //ak
```

```
464
465
466         if(nmuev>0){
467       h2->Fill(nmuali,1.); // muon mult. distr. Nmu>0
468     }
469
470         if(nmuinali>0){
471       h6->Fill(nmuinali,1.); // muon mult. distr. Nmu>0 inside Alice
472     }
473
474         if(nmuev==0){
475           nmuvec=0;
476           tagmu[nmuvec]=0;
477           emuev[nmuvec]=-1000;
478           dtmuev[nmuvec]=-1000.;
479           aemuev=-1000;
480           adtmuev=-1000.;
481           xmualiev[nmuvec]=-999999;
482           ymualiev[nmuvec]=-999999;
483           xmusurev[nmuvec]=-999999;
484           ymusurev[nmuvec]=-999999;
485           pxmuev[nmuvec]=-1000;
486           pymuev[nmuvec]=-1000;
487           pzmuev[nmuvec]=-1000;
488           pmuev[nmuvec]=-1000;
489           tetamuev[nmuvec]=-1000;
490           phimuev[nmuvec]=-1000;
491         } // if(nmuev==0)
492
493         if(nmuev==1){
494           tagmu[nmuvec]=tagmucell[nmuvec];
495           emuev[nmuvec]=emucell[nmuvec];
496           dtmuev[nmuvec]=-1000.;
497           aemuev=emuev[nmuvec];
498           adtmuev=-1000.;
499           xmualiev[nmuvec]=xmucell[nmuvec];
500           ymualiev[nmuvec]=ymucell[nmuvec];
501           xmusurev[nmuvec]=xmucellsurf[nmuvec];
502           ymusurev[nmuvec]=ymucellsurf[nmuvec];
503           pxmuev[nmuvec]=pxmucell[nmuvec];
```

```
504           pymuev[nmuvec]=pymucell[nmuvec];
505           pzmuev[nmuvec]=pzmucell[nmuvec];
506           pmuev[nmuvec]=pmucell[nmuvec];
507           tetamuev[nmuvec]=tetamucell[nmuvec];
508           phimuev[nmuvec]=phimucell[nmuvec];
509         } // if(nmuev==1)
510
511     if(nmuev>1){ // aemucell & dtmucell for event with at least 2
          muons
512       aemucell=emucell[0];
513       dtmucell=tmucell[0];
514           tmuevmin=tmucell[0];
515           iasmin=0;
516           tagmu[0]=tagmucell[0];
517           emuev[0]=emucell[0];
518           xmualiev[0]=xmucell[0];
519           ymualiev[0]=ymucell[0];
520           xmusurev[0]=xmucellsurf[0];
521           ymusurev[0]=ymucellsurf[0];
522           pxmuev[0]=pxmucell[0];
523           pymuev[0]=pymucell[0];
524           pzmuev[0]=pzmucell[0];
525           pmuev[0]=pmucell[0];
526           tetamuev[0]=tetamucell[0];
527           phimuev[0]=phimucell[0];
528
529       h5->Fill(emucell[0],1.);
530       for(Int_t ias=1;ias<nmuev;ias++){
531         aemucell+=emucell[ias]; // average muon energy in the event
532         h5->Fill(emucell[ias],1.);
533              tagmu[ias]=tagmucell[ias];
534              emuev[ias]=emucell[ias];
535              xmualiev[ias]=xmucell[ias];
536              ymualiev[ias]=ymucell[ias];
537              xmusurev[ias]=xmucellsurf[ias];
538              ymusurev[ias]=ymucellsurf[ias];
539              pxmuev[ias]=pxmucell[ias];
540              pymuev[ias]=pymucell[ias];
541              pzmuev[ias]=pzmucell[ias];
542              pmuev[ias]=pmucell[ias];
```

```
543             tetamuev[ias]=tetamucell[ias];
544             phimuev[ias]=phimucell[ias];
545             if(tmucell[ias]<tmuevmin){
546               tmuevmin=tmucell[ias];
547               iasmin=ias;
548             }
549       }
550       for(Int_t ias=0;ias<nmuev;ias++){
551             dtmuev[ias]=-1000.; // initialize
552
553             if(ias!=iasmin){
554               dtmucell=(tmucell[ias]-tmuevmin*1000.); // Deltatime
                      in ps
555               dtmuev[ias]=dtmucell;
556          h4->Fill(dtmucell,1.);
557               adtmuev += dtmuev[ias]; // average muon dtime in the
                      event
558         }
559       } // close for(Int_t ias=0;ias<nmuev;ias++)
560
561          aemucell /=nmuev;
562          h3->Fill(aemucell,1.);
563               aemuev=aemucell;
564               if(nmuev>1)adtmuev /= (nmuev+1);
565               if(nmuev<=1)adtmuev=-1000.;
566
567     } // END if(nmuev>1)
568
569
570
571        //  Built here the entupl --> ev_tree only when nmuev>0
572        // 1-loopf, 2-part_id, 3-eneprim, 4-tetaprimgrad,
              5-phiprimgrad
573             // 6-num_ev_run, 7-xco, 8-yco, 9-xcoali, 10-ycoali
574             // 11-nmuev, 12-nmuinali
575             // 13-tagmu[nmuev], 14-emuev[nmuev], 15-dtmuev[nmuev]
576             // 16-xmualiev[nmuev] 17-ymualiev[nmuev]
577             // 18-xmusurev[nmuev] 19-ymusurev[nmuev]
578             // 20-pxmuev[nmuev] 21-pymuev[nmuev] 22-pzmuev[nmuev]
579             // 23-pmuev[nmuev] 24-tetamuev[nmuev] 25-phimuev[nmuev]
```

```
580             // 26-aemuev       27-adtmuev
581             // 28-idchmu[nmuev]
582
583             if(nmuev>=0)ev_tree->Fill();
584             num_ev_run ++;
585
586          if(num_ev_run%100==0||num_ev_run==nevf[loopf]){
587            printf ("Loopf= %d, nev = %f, NUM_EV_RUN= %d
                  Total_events_run= %d
                  \n",loopf,nev,num_ev_run,nevf[loopf]);
588            printf ("*****************************************\n
                  \n");
589     }
590
591
592        // Check on tree variabiles
593             if(nmuev>0&&loopf==1&&num_ev_run<=100){
594               fprintf(bufana,"********* Start Check on TREE
                     variables ******\n ");
595               fprintf(bufana,"Ev.number= %d, N.mu large area= %d
                     N.mu in Alice= %d \n",num_ev_run,nmuev,nmuinali);
596               fprintf(bufana,"loopf= %d, part_id= %f, eneprim= %f,
                     tetaprimgrad= %f, phiprimgrad= %f , phiprimpos= %f
                     \n",loopf,part_id,eneprim,tetaprimgrad,phiprimgrad,phiprimpos);
597               fprintf(bufana,"xcoresurface [m] = %f, ycoresurface
                     [m] = %f \n",xco/100,yco/100);
598               fprintf(bufana,"xcorealilevel [m] = %f, ycorealilevel
                     [m] = %f \n",xdeep/100,ydeep/100);
599
600
601            for(Int_t ias=0;ias<nmuev;ias++){
602                  fprintf(bufana,"Muon num.= %d, tagmu[ias]= %d
                        \n",ias,tagmu[ias]);
603               fprintf(bufana,"Muon num.= %d, xmusurface [m] = %f,
                     ymusurface [m] = %f
                     \n",ias,xmusurev[ias]/100,ymusurev[ias]/100);
604               fprintf(bufana,"Muon num.= %d, xmualilevel[m]= %f,
                     ymualilevel[m]= %f
                     \n",ias,xmualiev[ias]/100,ymualiev[ias]/100);
```

```
605                fprintf(bufana,"Muon num.= %d, pxmusurf= %f,
                       pymusurf= %f, pzmusurf= %f
                       \n",ias,pxmuev[ias],pymuev[ias],pzmuev[ias]);
606                fprintf(bufana,"Muon num.= %d, pmusurf= %f,
                       tetamusurf= %f, phimusurf= %f
                       \n",ias,pmuev[ias],tetamuev[ias],phimuev[ias]);
607              }
608            } // END if(nmuev>0)
609         } //ak END if(tagg30==1) do not write the event if tagg30=0
610       }   // END if(strncmp(head.headname,"EVTE",4)==0) END EVTE
              event end
611
612
613       if(strncmp(head.headname,"RUNE",4)==0) {
614
615    fprintf(bufana," ============= Found RUNE ============ Block
          \n");
616
617          fprintf(bufana,"N. events in the run = %d, Total N. events
                before this run = %d\n",num_ev_run,num_ev_tot);
618
619    fprintf(bufana,"==============END RUNE================== \n");
620
621       }   //  END  if(strncmp(head.headname,"RUNE",4)==0) end RUNE
              run end
622
623
624       //    Analysis of the particles function AnaEv
625
626       if(strncmp(head.headname,"RUNH",4)!=0 &&
             strncmp(head.headname,"EVTH",4)!=0 &&
627     strncmp(head.headname,"EVTE",4)!=0 &&
          strncmp(head.headname,"RUNE",4)!=0)
628    {AnaEv(n); /* n is the block */
629    }
630
631     } // END for (int n=0;n<21;n++) End Analazing a record
          buf[21][273]
632
633     // Stop when a proper number of events are analyzed for this run
```

```
634 //    if(num_ev_run==nevf[loopf]+1||inFileC.eof()){
635
636     if(num_ev_ene_g30>=nevf_ene[loopf_ene]+1||inFileC.eof()){ //ak
637
638       num_ev_tot += num_ev_run; //ak total number of events
639 //      num_ev_ene += num_ev_run; // total num. of ev. in the same
        energy range
640       num_ev_tot_g30 += num_ev_run_g30; //ak total number of events
            g30
641 //      num_ev_ene_g30 += num_ev_run_g30; // tot.num.of ev.same
        en.range g30
642
643       fprintf(bufana," ================ SUMMARY RUN
            ======================== \n");
644       fprintf(bufana,"Loopf = %d , Num.ev.run g30 = %d , Num.ev.run
            = %d \n",loopf,num_ev_run_g30,num_ev_run);
645       fprintf(bufana,"Loopf_ene = %d, MAX ev.ene range g30 = %d ,
            NUM.EV.SAME ENERGY RANGE g30 = %d, Num.ev.consumed same
            ene = %d
            \n",loopf_ene,nevf_ene[loopf_ene],num_ev_ene_g30,num_ev_ene);
646      fprintf(bufana," Tot.ev consumed = %d, TOT.EV g30 = %d
           \n",num_ev_tot,num_ev_tot_g30);
647       fprintf(bufana," ================ END SUMMARY RUN
            ======================== \n");
648
649       cout << "Loop = " <<loopf << " Num. ev run g30 = " <<
            num_ev_run_g30 << " Num. ev. run = " << num_ev_run << "
            Tot. ev consumed = " << num_ev_tot << " TOTAL EVENTS
            gammaknee = " << num_ev_tot_g30 << endl;
650
651       break;    // stop loop on this run (file)
652
653     }    // END if(num_ev_run==nevf[loopf]||inFileC.eof())
654
655   }   // end while(1) loop to read all the events in the open file
656
657   printf ("End Read Events\n");
658   inFileC.close();
659
660   }  // end loop on input files : loopf
```

```
661
662 //    Plot Primary Energy
663   TCanvas *e1=new TCanvas("e1","Primary Energy
          Spectrum",10,10,800,800);
664   e1->cd(1);
665   heneprim->SetLineColor(kRed);
666   heneprim->Draw();
667
668
669   // Draw here the global results of all the input files together
670
671   TCanvas *c1=new TCanvas("c1","Num. Muons",10,10,1000,800);
672   TCanvas *c2=new TCanvas("c2","Energy Time",10,10,1000,800);
673   TCanvas *c3=new TCanvas("c3","Core position",10,10,1000,800);
674 //  TCanvas *c4=new TCanvas("c4","Muon's position",10,10,1000,800);
675
676   c1->Divide(1,3);
677   c1->cd(1);
678   h1->Draw();
679   c1->cd(2);
680   h2->Draw();
681   c1->cd(3);
682   h6->Draw();
683
684   c2->Divide(1,3);
685   c2->cd(1);
686   h3->Draw();
687   c2->cd(2);
688   h5->Draw();
689   c2->cd(3);
690   h4->Draw();
691
692 //  c1->cd(6);
693 //  hc1->Draw();
694   //  hc2->Draw();
695 //  c1->cd(7);
696 //  hc2->Draw("lego");
697
698   c3->cd(1);
699   hc3->Draw();
```

```
700
701 //  c4->cd(1);
702 //  hmusd->Draw();
703
704   //  ev_tree->Print();
705   //  ftree1->cd();
706   //  ev_tree->Write();
707   //  ftree1->Close();
708   ftree1->cd();
709   ftree1->Write();
710   ftree1->Close();
711
712   froot->cd();
713   //  tree1->Print();
714   //  tree1->Write();
715
716   froot->Write();
717   froot->Close();
718
719
720 }
```

Bibliography

[1] Atri, D. (2011) *Terrestrial Effects Of High Energy Cosmic Rays* (Doctoral Dissertation)
https://kuscholarworks.ku.edu/bitstream/handle/1808/7671/Atri_ku_0099D_11432_DATA_1.pdf?sequence=1

[2] Barcikowski, E. L. (2011). *The Composition of Ultra High Energy Cosmic Rays Through Hybrid Analysis at Telescope Array* (Doctoral Dissertation)
http://www.cosmic-ray.org/thesis/thesis_barcikowski.pdf

[3] Bustamante, M. , ***et. al*** (2015). *High-Energy Cosmic Ray Acceleration*
https://cds.cern.ch/record/1249755/files/p533.pdf

[4] CERN, *Received on 2015* http://home.cern/about/experiments/alice

[5] CORSIKA, Karlshruhe Institute of Technology, *Received on 2015*
https://www.ikp.kit.edu/corsika/69.php

[6] Daw, E. (2012). *Lecture-5: Ultra High Energy Cosmic Rays and the GZK cut-off*
http://www.hep.shef.ac.uk/edaw/PHY206/Site/2012_course_files/phy206rlec5.pdf

[7] D'Urso, D. (2014). *Cosmic Ray Physics.*
http://arxiv.org/abs/1411.4642v2

[8] Dying Supergiant Stars Implicated in Hours-long Gamma-Ray Bursts. (2013). *NASA*
http://www.nasa.gov/mission_pages/swift/bursts/supergiant-stars.html

[9] Eckhard, J. (2014). *Cerenkov Radiation.*
http://www.thphys.uni-heidelberg.de/~wolschin/eds14_3s.pdf

[10] Heck, D. , Pierog, T. (2013). *Extensive Air Shower Simulation with CORSIKA: A User's Guide (Version 74xxx from September 3, 2015)*
https://web.ikp.kit.edu/corsika/usersguide/usersguide.pdf

[11] Pagliarone, C. , Fernandez-Tellez, A. (Received on 2015). *Cosmic Ray Physics with ACORDE at Alice*
http://arxiv.org/ftp/arxiv/papers/0709/0709.3066.pdf

[12] Pierog, T. , Werner, K. (2009). *EPOS Model and Ultra-High Energy Cosmic Rays*
http://arxiv.org/pdf/0905.1198.pdf

[13] Pękala, J. (2007). *Atmospheric Scattering of Light Emitted by Extensive Air Showers.*
http://www.ifj.edu.pl/msd/rozprawy_dr/rozpr_Pekala.pdf

[14] Perkins, D.H. (2009). *Particle Astrophysics.*

[15] Rao, M.V.S. , Sreekantan B.V. (1998). *Extensive Air Showers.*
https://books.google.com.bd/books?hl=en&lr=&id=t5UZ3mtJVzAC&oi=fnd&pg=PR9&dq=extensive+air+showers&ots=6jVoai4H6d&sig=VR8bytC05iYcTYaOor16DJmo_W0#v=onepage&q=extensive%20air%20showers&f=false

[16] Stanev, T. (2004). *Ultra High Energy Cosmic Rays*
http://www.slac.stanford.edu/econf/C040802/papers/L020.PDF

[17] Technical Progress Report for: PANDA – Strong Interaction Studies with Antiproton, (2005)
http://hepweb.jinr.ru/urqmd1_3/urqmd.pdf

[18] Watson, A. A. (2011). *The Discovery of Cerenkov Radiation and its Use in the Detection of Extensive Air Showers.*
http://arxiv.org/ftp/arxiv/papers/1101/1101.4535.pdf