

目次

第 0 章 はじめに	3
第 1 章 先行研究	4
第 2 章 宇宙線	5
2.1 宇宙線について	5
2.2 宇宙線の歴史	5
2.3 空気シャワー	6
2.4 超高エネルギー宇宙線	7
2.4.1 GZK カットオフ	7
第 3 章 宇宙線の観測	8
3.1 宇宙線検出器の概要	8
3.2 シンチレーターと光電子増倍管	9
3.2.1 プラスチックシンチレーター	9
3.2.2 光電子増倍管	9
3.3 観測機器	10
第 4 章 CORSIKA シミュレーター	11
4.1 CORSIKA について	11
4.2 解析方法	13
4.3 解析結果	14
第 5 章 空気シャワーの広がり分布モデル	17
5.1 空気シャワーの広がり分布モデル	17
5.2 宇宙線の同時測定確率の算出	19
5.3 2つの検出器間の距離による同時測定確率の依存性	22
第 6 章 結論	23

第 7 章 参考文献-----23

第 8 章 謝辞-----24

第 9 章 付録-----24

第0章 はじめに

宇宙空間を高エネルギーで飛び交っている極めて小さな粒子を宇宙線という。早稲田大学本庄高等学院では、QuarkNet Japan(日本国内の中学・高校と大学・研究機関によって構成された宇宙線計測・放射線計測のコラボレーション)の活動の一環として、シンチレータや光電子増倍管、FPGA ボードからなる宇宙線検出器を用いた宇宙線の同時測定を計画している。これは 10^{18} eV より大きなエネルギーを持つ宇宙線である、超高エネルギー宇宙線の観測を目的としたものである。超高エネルギー宇宙線の観測には十分な距離を離して設置された複数の宇宙線検出器が必要である。そのため、広尾学園高等学校と連携した遠距離間の同時測定を行っている。

しかし、超高エネルギー宇宙線が実際に大気に到達し生成される空気シャワーがどれほどの範囲に広がるかといった要素が未だに検証されていない。早稲田大学本庄高等学院と広尾学園高等学校での同時測定が超高エネルギー宇宙線の観測に妥当な実験であるか確認するには、超高エネルギー宇宙線の空気シャワーの広がる範囲を推定し、ある地点に置かれた2つの検出器が同時に同一の宇宙線を検出する確率を求めることが必要である。

そこで、CORSIKA という Linux 環境上で動作するシミュレーションソフトウェアを用いる。CORSIKA を用いると、超高エネルギー宇宙線による空気シャワーが広がる範囲を解析することができる。

また、CORSIKA シミュレーションにより作成できる空気シャワー粒子の到来個数の水平方向分布から、フィッティングを用いて宇宙線の持つエネルギーごとに空気シャワーの広がり分布モデルを作成することができる。作成した空気シャワーの広がり分布モデルとシミュレーションプログラムを使用すれば、ある地点に置かれた2つの検出器が同時にあるエネルギーを持った宇宙線を検出する確率を計算することが可能となる。

宇宙線の同時測定確率を算出した後、2つの宇宙線検出器間の距離がどのように同時測定確率に影響しているかを確かめる。これを表すグラフを超高エネルギー宇宙線とそれ以下のエネルギーを持つ宇宙線についてそれぞれ作成することで、超高エネルギー宇宙線の同時測定に適した距離の範囲を確認することができる。

本論文は、こうしたプロセスを踏まえて超高エネルギー宇宙線の同時測定に適した2つの検出器間の距離を確かめるとともに、早稲田大学本庄高等学院と広尾学園高等学校間での超高エネルギー宇宙線の同時測定が妥当なものであるかの検証を目的としたものである。

第1章 先行研究

Telescope Array 実験 (TA 実験)

1020eV を超えるエネルギーを持つ最高エネルギー宇宙線の観測を目的として立案された日本、米国、韓国、ロシア、ベルギーの国際共同実験であり、地表検出器と大気蛍光望遠鏡を同じサイトに設置したハイブリッド型検出器を観測施設として用いるものである。初めて本格的な大気蛍光望遠鏡を宇宙線観測に用いたのは、ユタ大学のグループによるフライズアイ (Fly's Eye) 実験であった。フライズアイ実験をグレードアップし、1990年代の超高エネルギー宇宙線研究をリードした High-Resolution Fly's Eye (HiRes) 実験の後継となる実験が TA 実験である。

地表検出器による観測においては、地表に到達した空気シャワーの粒子数と到達時間を測定し一次宇宙線のエネルギーの大きさ及び到来方向を推定する。また大気蛍光望遠鏡による観測においては、空気シャワーが大気を通ったときに生成されるシンチレーション光の軌跡を測定し空気シャワーの大気中でのエネルギー損失及び到来方向を推定する。測定原理の異なる二つの観測方法を用いることで、より高精度なハイブリッド観測を実現している。

TA 実験は、大気が乾燥しており大気蛍光望遠鏡での観測に適している、米国ユタ州の西部砂漠地帯に建設された施設で行われている。有効検出面積は 700km² であり、地表検出器として面積 3m² のシンチレーション検出器が 1.2km 間隔で 507 台設置されている。各検出器は厚さ 1.2cm のシンチレーター上下 2 層構造で、各層には波長変換マイクロファイバーが敷設されており、荷電粒子のシンチレーター入射位置による信号のばらつきを軽減するとともに各層にある光電子増倍管に集光されて検出される。また、38 台の大気蛍光望遠鏡も地表検出器の周りを 3 か所に分割して設置されている。

TA 実験で特筆すべきなのは、大気蛍光望遠鏡の前方に設置された線形加速器 (Electron Light Source, ELS) の存在である。線形加速器は電子ビームを大気中に射出することができるため、エネルギーの判明した空気シャワーを検出器の前で人工的に作り出し、発生する大気蛍光量を測定することができる。これにより大気蛍光の発光効率の測定とエネルギーの校正を同時に行うことが可能となり、実験室で測定された大気発光効率を使用してデータ解析を行っていた HiRes 実験と比較してより精度の高い大気蛍光検出による宇宙線観測ができるようになった。

主な TA 実験の目標は、宇宙線のエネルギー分布スペクトルを正確に測定して、スペクトルの形に超高エネルギー宇宙線と宇宙背景放射の衝突の痕跡 (GZK カットオフ) が存在するかを確かめること、超高エネルギー宇宙線が来た方向に発生源となりうる天体現象があるか探ること、そしてどのような素粒子 (陽子、電子、原子核、ミューオン、ニュートリノなど) が超高エネルギーに加速されているのかを調査することである。

これまでの TA 実験の観測データから、 5×10^{19} eV 付近で宇宙線のエネルギースペクトル

の折れ曲がりが存在することが確かめられている。この折れ曲がり GZK カットオフであることを確認するためには、宇宙線の組成を観測したデータを元に決定し、その組成から予測されるカットオフの位置と観測値が一致した上でカットオフ以上のエネルギーを持つ宇宙線の到来方向分布に大きな異方性が見られる必要がある。

2017 年の時点では、有効検出面積を現在の 4 倍に拡張する” TA×4” や、高仰角望遠鏡と高密度地表検出器アレイを増設する低エネルギー側拡張である” TALE” が建設中である。
[2]

第 2 章 宇宙線

2.1 宇宙線について

宇宙空間には、核子当たり数十 MeV 以上の高いエネルギーを持つ陽子や原子核、電子などが飛び交っており、その一部が地球大気にも到来してきている。これらの粒子は、大気を作る酸素や窒素の原子核と衝突してそれらを破壊し、パイオンやミュオン、電子や光子といった 2 次粒子を生み出し、空気シャワーとなって地表に降り注ぐ。これらを総称して宇宙線と呼ぶ。特に地球外から到来する宇宙線を 1 次宇宙線と呼び、大気中で生成される 2 次粒子を 2 次宇宙線と呼んで区別している。[1]

2.2 宇宙線の歴史

1910 年代初頭まで、人類は高エネルギーを持った粒子が宇宙空間から地表に降り注いでいることを知らなかった。ウランやポロニウム、ラジウムなどから放射線が放出されていることが分かっていたが、これらの鉱石から離れた遠方でも放射線はゼロにはならず、この放射線の成分は解明されていなかった。そこでヘス (V. Hess) は、ゼロではない放射線の成分を調べるため、1911-1912 年に電離計を用いた電離度計測を気球に乗って行った。高度 4000m まで上昇し計測した結果、高度 1000m より上空で、電離計は高空に上昇するにつれてより強くなる放射線の存在を示した。コルエルスター (W. Kolhorster) はこの強い放射線に対して” Hohen-strahlung” (高空放射線) と名付けた。” Hohen-strahlung” は” 宇宙線” と和訳されるが、このときヘスはまだ高空放射線が宇宙空間から飛来していることに気がついていなかった。ヘスはこの放射線の正体を浸透力のあるガンマ線であると考え、これを” 超ガンマ線” と呼んだ。クレイ (J. Clay) は、1929 年緯度により放射線強度が変動することを発見した。これにより地球に降ってくる放射線は磁場に影響されることが分かり、磁場に影響されることがないガンマ線ではなく、電荷を持った粒子線であることが分かった。

ロッシ (B. Rossi) とジョンソン (T. Johnson) は地球の磁力線に対し正の電荷を持った粒子は西から多く、負の電荷を持った粒子は東から多く飛来すると予測した。1930 年に行った観測の結果、透過力の高い放射線が西から多く飛来することがわかり、粒子線は正

の電荷を持っているという結論になった。これを”宇宙線の東西効果”と呼ぶ。

ヘスは高空放射線の起源が太陽ならば、日食で月に太陽が遮蔽されるとき放射線が減少すると考え、1931年に日食時に気球を飛ばす実験を行った。その結果、日食では高空放射線の強度は変化せず、高空放射線の起源は太陽ではないことがわかった。その後ヘスはオーストリア・インスブルックの山頂 2300m で高空放射線の連続観測を行った。コンプトン(A.H. Compton)とゲッティング(I.A. Getting)は、このヘスの連続観測データをもとに天球座標に依存する高空放射線の強度変動を研究した。

この高空放射線を最初に”宇宙線”と命名したのはミリカン(R.A. Millikan)である。1925年の論文で”cosmic rays”という言葉が使用されており、宇宙から地球の大気を浸透する高エネルギーの放射線が飛来しているに違いないと記述している。ヘスはそれまで”超ガンマ線”や”超放射線”と呼んでいたが、1940年ミリカンが名付けた”宇宙線”(cosmic rays)という言葉に同意したため、これ以降”宇宙線”という名前が定着することになった。[2]

2.3 空気シャワー

大気中の宇宙線の相互作用長は大気的全厚さに比べて非常に小さいため、宇宙線は大気中で原子核との相互作用を起こす。これによりパイオンの多重発生が起こる。生成されたパイオンがまた相互作用を起こすことで、大気を通過していくにつれて粒子数が増大していく。これを空気シャワー現象という。

宇宙線と大気中の原子核との相互作用で生成されるパイオンには主に π^+ 、 π^- 、 π^0 の3種類が存在する。パイオンの他にロー中間子やケイオン中間子が生成されることもあるが、パイオンの生成数が最も多い。静止した荷電パイオン π^\pm は寿命 2.6×10^{-8} 秒でミューオン μ^\pm に崩壊するが、最初の相互作用で生成されるパイオンは高いエネルギーを持っているため、ローレンツ因子の分だけ寿命が延びる。そのため、ミューオンに崩壊する前にさらに次の相互作用を発生させる。これをハドロンカスケードと呼ぶ。やがてエネルギーを失うと、パイオンは最終的にミューオンに崩壊する。

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

$$\pi^+ \rightarrow \mu^+ + \nu_\mu$$

静止したミューオンは寿命 2.2×10^{-6} 秒で e^\pm に崩壊するが、ローレンツ因子により寿命は 10^{-5} 秒程度に延びる。また、ミューオンは電子の約 200 倍の質量を持つため、電子と比較すると制動輻射が起こりにくい。また、制動輻射のエネルギー損失は粒子の質量の2乗に反比例するためエネルギー損失もゆっくりとしたものとなり、ミューオンの多くは崩壊前に地表に到達する。

原子核の相互作用により生成されたパイオンのうち、中性パイオン π^0 は寿命が 8.4×10^{-17} 秒と非常に短く、高いエネルギーを持っていたとしてもほぼ即時に崩壊して2個の

ガンマ線となる。元の中性パイオンが高いエネルギーを持っていれば崩壊によって生成されるガンマ線も高いエネルギーを持つ。物質中のガンマ線は電子対生成を起こし、また物質中の電子や陽電子は制動放射によってガンマ線を生成する。これが繰り返し発生することにより、ガンマ線と電子・陽電子の大群が生成される。これを電磁カスケードと呼ぶ。

$$\pi^0 \rightarrow \gamma + \gamma$$

$$\gamma \rightarrow e^+ + e^-$$

$$e^\pm \rightarrow e^\pm + \gamma$$

空気シャワーとはハドロンカスケードと電磁カスケードの重ね合わせである。[2]

2.4 超高エネルギー宇宙線

エネルギーが 10^{18} eV を超える宇宙線のことを超高エネルギー宇宙線 (ultra-high energy cosmic rays, UHECRs) と呼ぶ。このような高いエネルギーに到達するまで陽子や原子核などの粒子を加速できる加速できるメカニズムは未だ解明されておらず、その起源は謎であるが、ガンマ線バーストやクラスター銀河団、活動銀河核、超強磁場中性子星など、宇宙における天体现象の中でも最も激しい事象と関連していると考えられている。

通常、 10^{12} eV までのエネルギーを持つ宇宙線は大気上層や大気圏外で宇宙ステーションや気球による直接観測を行う。しかし宇宙線のエネルギーが高くなるにつれてその到来頻度は急激に低くなるため、同じように直接観測を行うのは難しい。高いエネルギーを持つ宇宙線が生成する空気シャワーは広範囲にわたり地表に到達するため、超高エネルギー宇宙線はこれを利用して高山や地表で観測する。

超高エネルギー宇宙線の起源を解明するために重要な情報として、宇宙線のエネルギースペクトル、到来方向分布の異方性、原子核組成がある。これらの重要な情報を高精度で得るためには、観測する事象数の蓄積が必要となる。しかし、超高エネルギー宇宙線ではエネルギーが 1 桁上がるにつれて到来頻度は 2 桁下がる。たとえば、 10^{20} eV のエネルギーを持つ宇宙線の到来頻度は面積 100km^2 あたり 100 年で 1 回程度と超低頻度である。そのため広大な有効検出面積を持つ観測装置を用いた長時間の観測が必要となる。[2]

2.4.1 GZK カットオフ

この宇宙は、温度 $T = 2.73\text{K}$ に相当する黒体放射のエネルギースペクトルを持った光子で満ちている。これを宇宙背景放射 (Cosmic Microwave Background radiation, CMB) と呼び、過去の宇宙が高温高密度の熱平衡状態にあったことの名残であると解釈されている。宇宙背景放射の光子は 10^{-3} eV 程度の極めて低いエネルギーであるが、宇宙背景放射が 10^{-3} eV にみえる観測者の系を実験室系と考え、この系を 10^{20} eV の超高エネルギー宇宙線が止まってみえる系にローレンツ変換すると、その系では宇宙背景放射は 100MeV を超えるガンマ線に見える。計算によれば、宇宙線のエネルギーが $4-6 \times 10^{19}$ eV を超えると

宇宙背景放射光子との相互作用で π 粒子生成によるエネルギーの減衰が起こるようになる。これにより、観測される宇宙線では $E = 4-6 \times 10^{19} \text{eV}$ 付近で急激な強度の減少が起きることが期待される。これをGZKカットオフ (Greisen-Zatsepin-Kuzmin cut-off)と呼ぶ。[2]

黒体放射光子が高いエネルギーを持つ陽子と相互作用してパイオンを生成するには陽子がGZKカットオフのしきいエネルギーを超える必要がある。このしきいエネルギーで、光子によるパイオンの生成が光子-陽子の質量系で励起状態を通じた反応を運動学的に可能にする。陽子がこのしきいエネルギーを超えると陽子は急速にエネルギーを失い、しきいエネルギーより下に落ちる。これにより、超高エネルギー宇宙線の平均自由行程が数十メガパーセクよりも小さいとする制限がつけられる。また、光子も陽子と同様に黒体放射光子と相互作用して、電子-陽子対を $\gamma \gamma \rightarrow e^+e^-$ で生成する。この過程によって 10^{15}eV 以上のエネルギーを持つ光子の平均自由行程は数十キロパーセクに制限される。これらにより、超高エネルギー宇宙線の平均自由行程に、銀河系外スケールからすると非常に短い距離の制限がつけられる。[3]

第3章 宇宙線の観測

3.1 宇宙線検出器の概要

宇宙線の観測には、粒子の検出が必要不可欠である。荷電粒子は、その粒子の持つエネルギーに応じて、物質中を通過する際に近接する原子や分子をイオン化し、半導体の中では伝導電子と空孔の対を形成する。これらの現象を利用した荷電粒子の検出器として、シンチレーション検出器と半導体検出器がある。シンチレーション検出器は大きな立体角や0.1ミリ秒以下の高速応答を必要とする場合に適している。一方、半導体検出器は粒子のエネルギーの持つエネルギーを高精度で測定する場合に適しているが、限られた用途のみ使用される。宇宙線の観測には多くの場合シンチレーション検出器が使用される。

シンチレーション検出器はシンチレーターと光電子増倍管、高速論理回路を組み合わせたものが主に使用されている。ミューオンなどの荷電粒子がシンチレーターの中を通過した際に発生するシンチレーション光を光電子増倍管でとらえ、光電効果により放出された電子を増幅して高速論理回路に送り、データとして記録するという仕組みとなっている。シンチレーション検出器は粒子が通過した瞬間に高速の電気信号を送信するが、このような高増幅作用を持つ装置ではノイズの問題が発生するためこれに対応しなければならない。ノイズと信号を分別するために、尖頭電圧値を用いたディスクリミネーター (Discriminator) 回路を使用する。この回路を通過すると、設定した波高のしきい値よりも大きい入力パルスに対しては規格化されたパルスを出力端子から返し、しきい値よりも小さい入力パルスに対しては出力パルスを発生しないように作られている。このように荷電粒子の通過が電気信号に変換された後、複数のシンチレーション検出器を荷電粒子がど

のような角度分布、同時係数を有した状態で通過したかを高速論理回路で判断する。この高速論理回路の基本となるのがコインシデンス (Coincidence) 回路である。コインシデンス回路は電子計算機などに使用されている AND-GATE を高速化したもので、ある時間間隔の間に 2 つ以上の入力端子にパルスが入力されたときのみ出力パルスを返すように作られている。[4]

空気シャワー中の荷電粒子は正負電荷の電子とミュオンが主成分であるが、シンチレーション検出器はどちらも検出できるため電子とミュオンの分離は難しい。これをとらえるために、遮蔽物を利用してミュオンのみ検出することができるようにしたミュオン検出器が併設されることもある。同じエネルギーの宇宙線による空気シャワーが発生した場合は、生成されるミュオンの粒子数は重い原子核による空気シャワーの方が多くなる。また、1 次ガンマ線による空気シャワーはほぼ純粋な電磁カスケードであるため、電磁カスケードとハドロンカスケードの重ね合わせである陽子・原子核による空気シャワーよりもミュオンの粒子数は少なくなる。このため、ミュオン検出器の情報は 1 次宇宙線の原子核種の推定や、高エネルギーの 1 次宇宙ガンマ線の測定などに用いられる。[2]

3.2 シンチレーターと光電子増倍管

3.2.1 プラスチックシンチレーター

空気シャワーに含まれる荷電粒子の測定に用いる粒子検出器として、代表的なものに光電子増倍管とプラスチックシンチレーターを組み合わせたシンチレーション検出器がある。シンチレーターにはプラスチックシンチレーターに代表される有機シンチレーターと、NaI 結晶に代表される無機シンチレーターに大きく分けられる。一般には発光時間は有機シンチレーターのほうが短く、発光量は無機シンチレーターのほうが大きい。空気シャワーアレイに用いるシンチレーション検出器はナノ秒オーダーの高い時間決定精度を要求されるほか、軽く、衝撃に強く、不燃性・非毒性を備えているなど取り扱いが容易いため、多くの場合プラスチックシンチレーターが用いられる。プラスチックシンチレーターはポリスチレンと p-ターフェニルのシンチレーション性溶質に POPOP などを加えた材質を用いて製作されている。ミュオンなどの荷電粒子がプラスチックシンチレーターの中を通過すると、シンチレーション光という波長 400nm 前後の蛍光を発生する。このシンチレーション光は微弱な発光であるため、光電子増倍管でとらえるためにはシンチレーターを完全に遮光する必要がある。[2]

3.2.2 光電子増倍管

光電子増倍管は高速応答・超高感度の光センサーである。光電子増倍管の入射窓に光子が入ると、光電効果が起こりやすい物質でできた光電面に光子が当たり電子に変換される。光電効果が発生しダイノードまで到達する確率は 20-30%であり、これを量子効率という。光電面によく用いられる物質として、可視波長域で高い量子効率を持つ GaAsP (Cs) や、2 種類のアリカリ金属を使う Sb-Rb-Cs、Sb-K-Cs などが知られている。光電面の直径は 2 インチから 15 インチ程度であり、その先にダイノードと呼ばれる電圧の印加された電極が並んでいる。[2]

3.3 観測機器

QuarkNet Japan の活動の一環として広尾学園高等学校と連携して行っている、超高エネルギー宇宙線の観測を目的とした遠距離間の同時測定に実際に使用している観測機器は Figure 1 に示したような構成となっている。このため、第 5 章の宇宙線の同時測定確率の算出においても、宇宙線の観測に以下のようなシンチレーターや光電子増倍管、FPGA ボードからなる宇宙線検出器を 2 つ使用したという想定で行う。

1. プラスチックシンチレーター	2 台
光電子増倍管	2 台
2. BNC 信号延長ケーブル	2 本
3. QuarkNet FPGA ボード	1 台
4. CAT-5 LAN ケーブル	1 本
5. GPS モジュール	1 個
6. GPS アンテナ	1 個
7. 温度センサー	1 個
8. 5V DC 電源	1 個
9. PDU 電源ケーブル	1 本
10. PDU	1 個
11. 電源延長ケーブル	2 本
12. USB ケーブル	1 本
13. PC	1 台

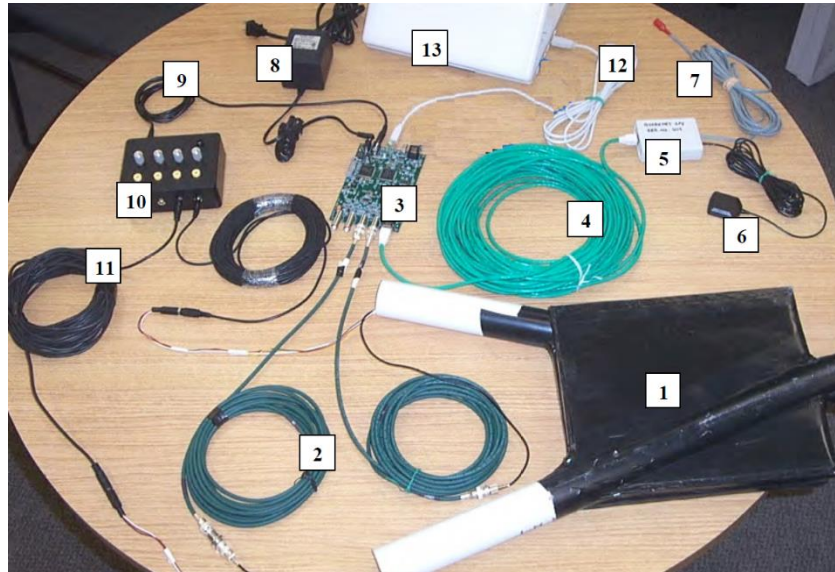


Figure 1 宇宙線ミュオン検出器 (CRMD) の構成¹

第4章 CORSIKA シミュレーター

4.1 CORSIKA について

CORSIKA (Cosmic Ray Simulations for Cascade) は高エネルギー宇宙線による大規模空気シャワー (Extensive Air Shower, EAS) のシミュレーションを行うことのできるコンピュータソフトウェアである。100EeV までのエネルギーを持った宇宙線による空気シャワーのシミュレーションを実行することができる。元々はドイツ・カールスルーエ工科大学 (Karlsruhe Institute of Technology, KIT) がカスケード実験を行うために開発されたプログラムであるが、チェレンコフ放射に関する実験だけでなく、標準的な空気シャワーシミュレーションプログラムとして宇宙線に関する実験全般に広く利用されている。[5]

CORSIKA への入力パラメータは all-inputs (付録に記載) という入力ファイルに記載されており、利用するユーザーにより自由に定義することができる。以下に all-inputs 内で定義される主要なパラメータを示す。

- EVTNR
一次宇宙線となる粒子の数
- NSHOW
生成する空気シャワー数
- PRMPAR
陽子や電子、ミュオンなどの入射粒子種別 (Figure 2 に詳細する)

¹ Cosmic Ray e-Lab Resources

(http://quarknet.i2u2.org/sites/default/files/cf_6000crmdusermanual-small.pdf)

- ERANGE
入射粒子のエネルギー幅
- THETAP
入射粒子の天頂角 θ
- PHIP
入射粒子の方位角 ϕ
- OBSLEV
観測地の標高
- MAGNET
観測地における地磁気の大きさ
- ECUTS
ERANGE で定義できるエネルギーの下限值

Particle identifications			
Identification	Particle	Identification	Particle
1	γ	50	ω
2	e^+	51	ρ^0
3	e^-	52	ρ^+
		53	ρ^-
5	μ^+	54	Δ^{++}
6	μ^-	55	Δ^+
7	π^0	56	Δ^0
8	π^+	57	Δ^-
9	π^-	58	$\bar{\Delta}^{--}$
10	K_L^0	59	$\bar{\Delta}^-$
11	K^+	60	$\bar{\Delta}^0$
12	K^-	61	$\bar{\Delta}^+$
13	n	62	K^{*0}
14	p	63	K^{*+}
15	\bar{p}	64	\bar{K}^{*+}
16	K_S^0	65	\bar{K}^{*0}
17	η	66	ν_e
18	Λ	67	$\bar{\nu}_e$
19	Σ^+	68	ν_μ
20	Σ^0	69	$\bar{\nu}_\mu$
21	Σ^-		
22	Ξ^0	71	$\eta \rightarrow \gamma\gamma$
23	Ξ^-	72	$\eta \rightarrow 3\pi^0$
24	Ω^-	73	$\eta \rightarrow \pi^+\pi^-\pi^0$
25	\bar{n}	74	$\eta \rightarrow \pi^+\pi^-\gamma$
26	$\bar{\Lambda}$	75	μ^+ add. info.
27	$\bar{\Sigma}^-$	76	μ^- add. info.
28	$\bar{\Sigma}^0$		
29	$\bar{\Sigma}^+$	85	decaying μ^+ at start ⁸⁹
30	Ξ^0	86	decaying μ^- at start ⁸⁹
31	Ξ^+		
32	$\bar{\Omega}^+$	95	decaying μ^+ at end ⁸⁹
48	η'	96	decaying μ^- at end ⁸⁹
49	Φ		

Particle identifications (continued)			
Identification	Particle	Identification	Particle
116	D^0	155	Ξ_c^{*+}
117	D^+	156	Ξ_c^{*0}
118	\bar{D}^-	157	$\bar{\Omega}_c^0$
119	\bar{D}^0		
120	D_s^+	161	Σ_c^{*++}
121	\bar{D}_s^-	162	Σ_c^{*+}
122	η_c	163	Σ_c^{*0}
123	D^{*0}		
124	D^{*+}	171	$\bar{\Sigma}_c^{*-}$
125	\bar{D}^{*+}	172	$\bar{\Sigma}_c^{*0}$
126	\bar{D}^{*0}	173	$\bar{\Sigma}_c^{*+}$
127	D_s^{*+}		
128	\bar{D}_s^{*-}	176	B^0
		177	B^+
130	J/ψ	178	\bar{B}^-
131	τ^+	179	\bar{B}^0
132	τ^-	180	B_s^0
133	ν_τ	181	\bar{B}_s^0
134	$\bar{\nu}_\tau$	182	B_c^+
		183	\bar{B}_c^-
137	Λ_c^+	184	Λ_b^0
138	Ξ_c^+	185	Σ_b^-
139	Ξ_c^0	186	Σ_b^+
140	Σ_c^{*+}	187	Ξ_b^0
141	Σ_c^+	188	Ξ_b^-
142	Σ_c^0	189	Ω_b^0
143	Ξ_c^{*+}	190	$\bar{\Lambda}_b^0$
144	Ξ_c^0	191	$\bar{\Sigma}_b^+$
145	Ω_c^0	192	$\bar{\Sigma}_b^0$
		193	$\bar{\Xi}_b^0$
149	$\bar{\Lambda}_c^-$	194	$\bar{\Xi}_b^+$
150	$\bar{\Xi}_c^-$	195	$\bar{\Omega}_b^+$
151	$\bar{\Xi}_c^0$		
152	$\bar{\Sigma}_c^{*-}$		
153	$\bar{\Sigma}_c^-$		
154	$\bar{\Sigma}_c^0$		
$A \times 100 + Z$	nucleus of Z protons and A - Z neutrons ($2 \leq A \leq 56$)		
8888jjj	weights of preceding particle (MULTITHIN option)		
9900	Cherenkov photons on particle output file		

Figure 2 CORSIKA における入射粒子種別²

² Extensive Air Shower Simulation with CORSIKA: A User's Guide
(<https://web.ikp.kit.edu/corsika/usersguide/usersguide.pdf>)

4.2 解析方法

Oracle VirtualBox を用いて、Ubuntu 仮想環境上で CORSIKA を起動しシミュレーションを行う。シミュレーションに使用するソフトウェアは以下の通りである。

- CORSIKA
- ROOT

CORSIKA シミュレーションは all-inputs という入力ファイルに書かれた宇宙線のパラメータを読み込み行われる。CORSIKA を用いてシミュレーションをする際には、Ubuntu 仮想環境上で Terminal を用いてコマンドを打ち込む必要がある。手順は以下の通りである。

1. 入力ファイルである all-inputs にシミュレーション対象とする宇宙線のパラメータを書き込む
2. `~/corsika-76900/run/` (CORSIKA がインストールされているディレクトリ名) でターミナルを開く
3. `./corsika76900Linux_QGSJET_gheisha_inclined <all-inputs &` とコマンドを打ち CORSIKA を実行、シミュレーション開始
4. END OF RUN の表示を確認しシミュレーション終了

以上の手順をもってシミュレーションが終了すると演算結果が記された出力ファイルと DAT ファイルが生成される。演算結果を図やグラフとして表示するためにはファイル形式をデータ解析の行いやすいものに変更する必要があるため、拡張子 `.root` で表されるファイル形式に変換し、データ解析ソフトである ROOT に読み込めるようにする。また、図やグラフを表示するためにグラフの種類や使用するデータを C 言語で記述したプログラム (`colz.C`、付録に記載) を用いる。ROOT を用いて演算結果を図やグラフで表すためには追加で以下の手順を行う。

1. `./CorsikaPlotter DAT00000x` (`x` は all-inputs の RUNNR の値) とコマンドを打ち root ファイルを生成
2. `root DAT00000x_1.root` とコマンドを打ち ROOT を起動
3. `.x colz.C("DAT00000x_1.root")` とコマンドを打ち `x:y` グラフを描画

4.3 解析結果

宇宙線の持つエネルギーごとに空気シャワーの広がり分布モデルを作成するために、CORSIKA シミュレーションを宇宙線のエネルギーごとに行い、空気シャワーが広がる範囲を解析する。CORSIKA シミュレーションの結果から空気シャワー粒子の到来個数の水平方向分布を作成することができ、フィッティングを用いて宇宙線の持つエネルギーごとに空気シャワーの広がり分布モデルを作成することができる。今回シミュレーションする宇宙線の主要なパラメータとして以下のものを使用した。その他のパラメータは第 10 章内の all-inputs の項目に記載してある。

- EVTNR:1

一次宇宙線となる粒子の数は 1 個

- NSHOW:1

生成する空気シャワー数は 1 個

- PRMPAR:2

入射粒子種別は陽子

- ERANGE:1. E5, 1. E6, 1. E7, 1. E9

入射粒子のエネルギー幅は 10^{14}eV 、 10^{15}eV 、 10^{16}eV 、 10^{18}eV

- THETAP:20

入射粒子の天頂角 θ は 20°

- PHIP:-180 180

入射粒子の方位角 ϕ は -180° から 180°

Figure 3-15 は地表に到達する空気シャワーの粒子の個数を示した 2 次元ヒストグラムである。x 軸と y 軸は地表における距離を表している。最初に宇宙線のエネルギーのみを 10^{14}eV から超高エネルギー宇宙線に相当する 10^{18}eV までの範囲で変更し、それぞれの値でシミュレーションを行った。Figure3-6 までは 10^{14}eV から 10^{18}eV までの各エネルギーにおける 2 次元グラフを示している。

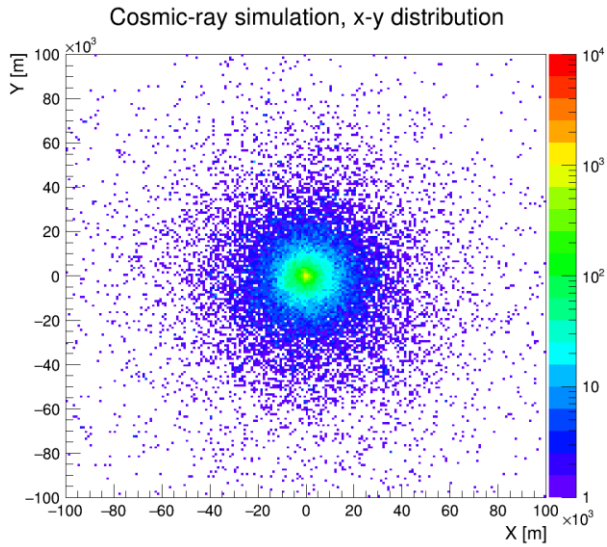


Figure 3 10^{14}eV

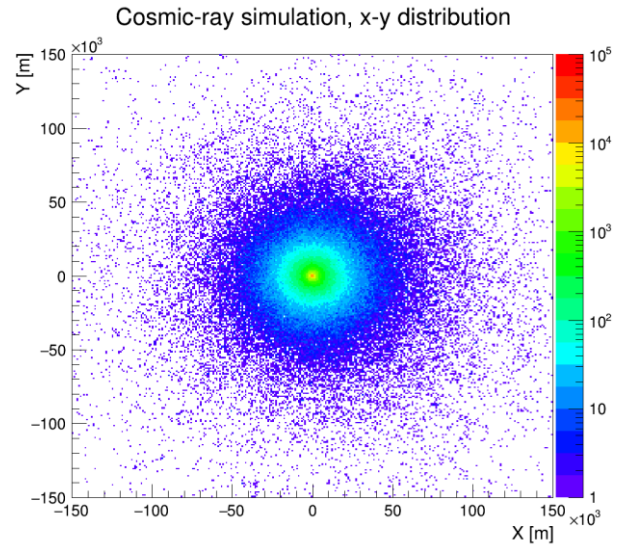


Figure 4 10^{15}eV

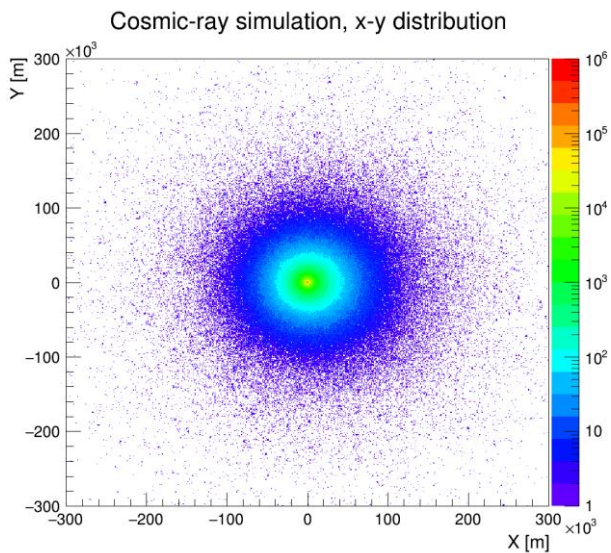


Figure 5 10^{16}eV

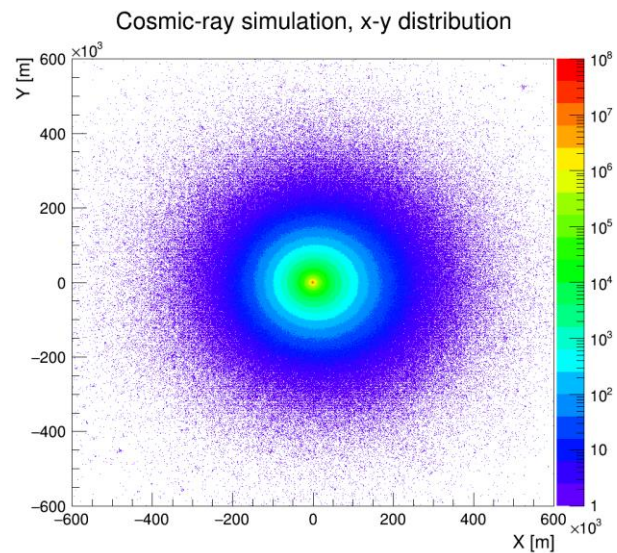


Figure 6 10^{18}eV

次に宇宙線のエネルギーを 10^{15}eV に固定し、到来してくる宇宙線の天頂角のみを変えてシミュレーションを行った。これは宇宙線が大気に突入する角度と空気シャワーの粒子の個数の関係を探るためである。天頂角 20° の際のグラフは Figure 4 ですすでに示したため省略した。Figure 7-13 が示す通り、宇宙線の天頂角が大きくなるにつれて空気シャワーの粒子の個数は減少した。これは宇宙線の天頂角が大きくなるにつれて空気シャワーが通過する大気量も多くなり、空気シャワーの粒子の持つエネルギーが減衰したことで地表に届かなくなったことが原因として考えられる。

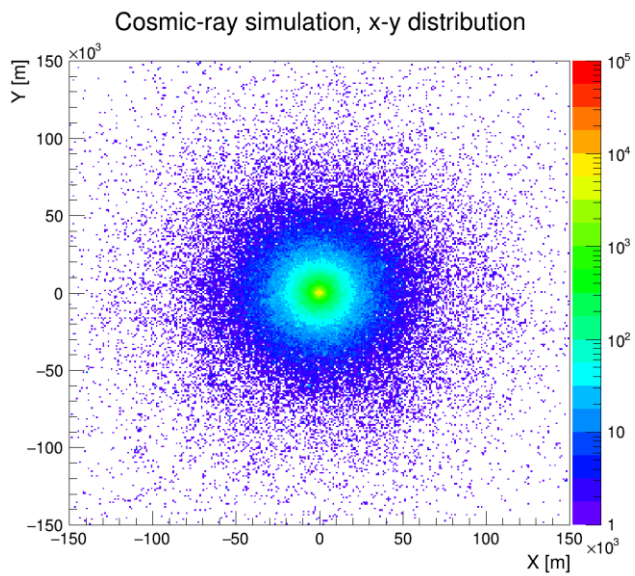


Figure 7 天頂角 0°

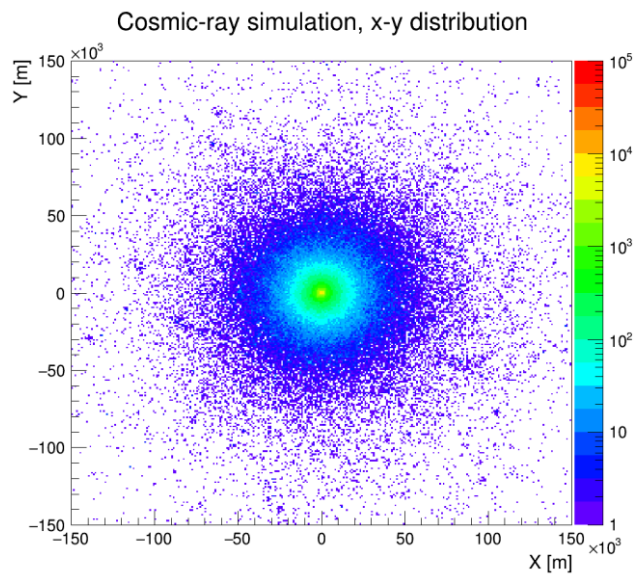


Figure 8 天頂角 10°

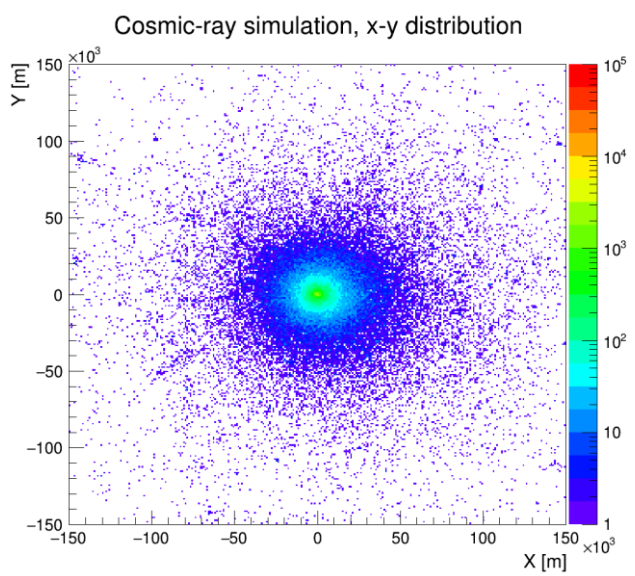


Figure 9 天頂角 30°

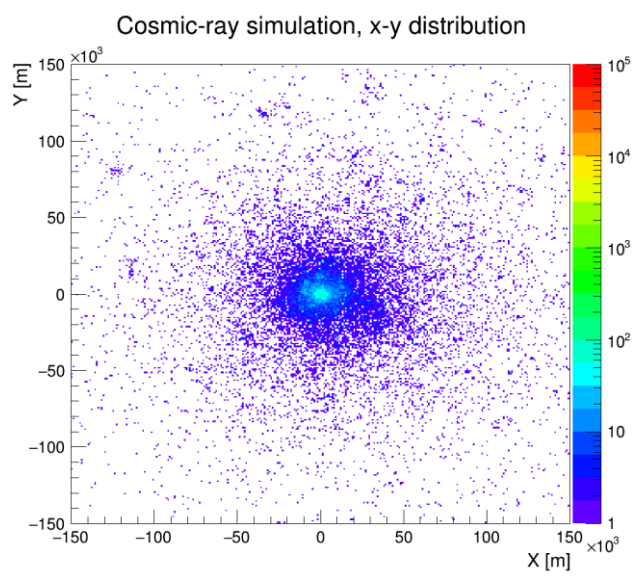


Figure 10 天頂角 40°

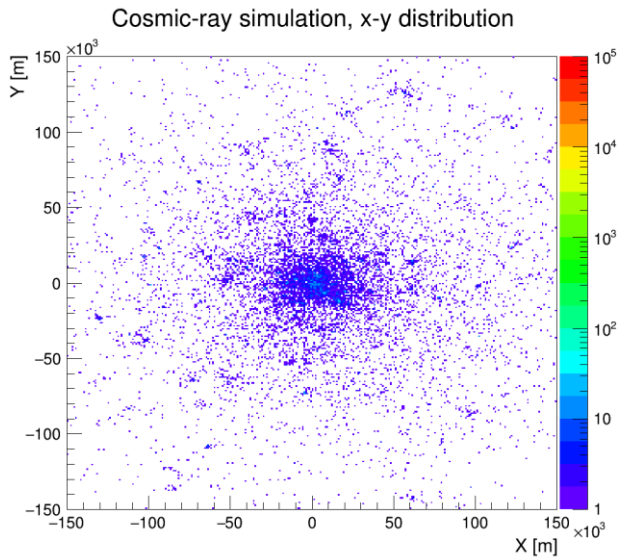


Figure 11 天頂角 50°

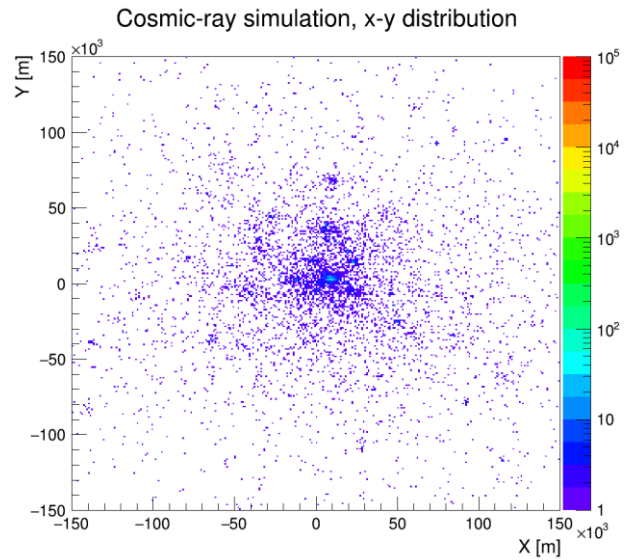


Figure 12 天頂角 60°

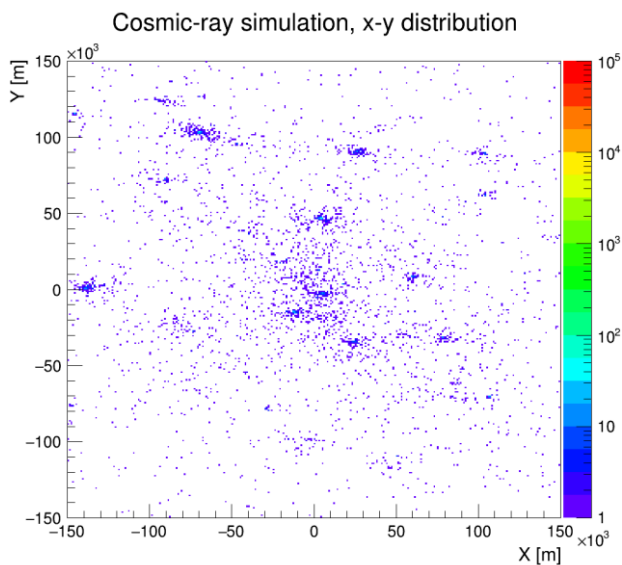


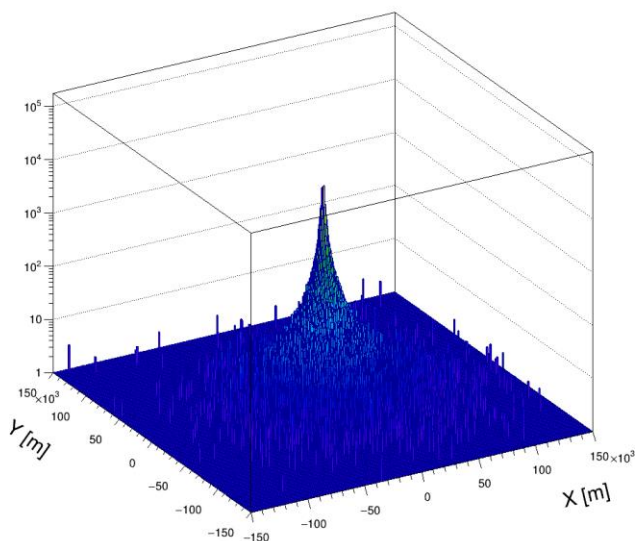
Figure 13 天頂角 70°

第5章 空気シャワーの広がり分布モデル

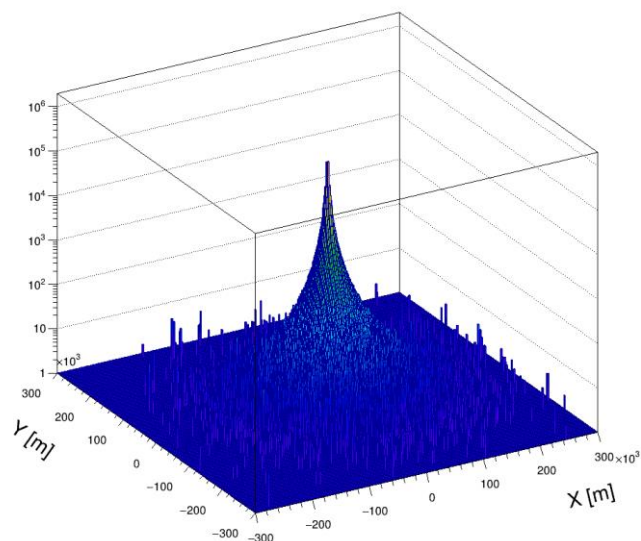
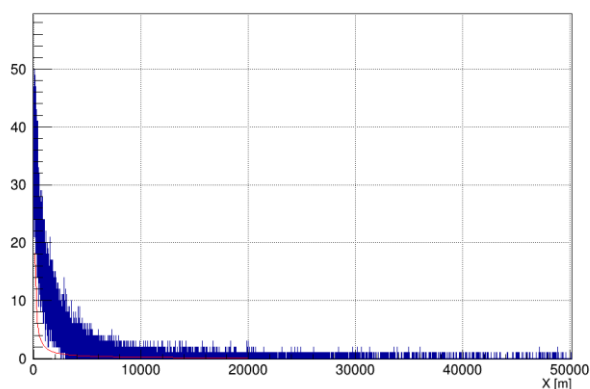
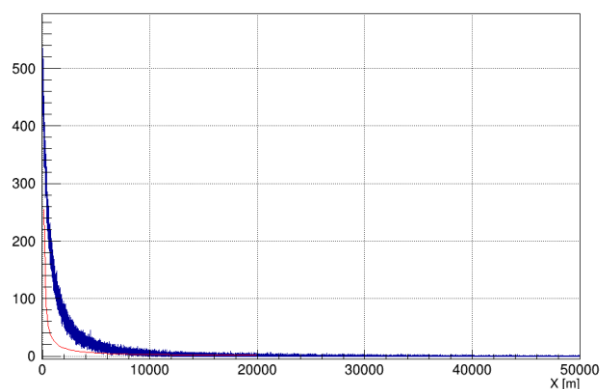
5.1 空気シャワーの広がり分布モデル

CORSIKA を用いたシミュレーション結果を使用して宇宙線のエネルギーごとの空気シャワーの広がり分布モデルを作成する。分布モデルをわかりやすく説明するために、Figure 14-15 を示す。Figure 14-15 は 10^{15} eV と 10^{16} eV のエネルギーにおける地表に到達する空気シャワーの粒子の個数を示した 3 次元グラフであり、用いられている CORSIKA シミュレーションのデータは Figure 4-5 のものと同様である。

Cosmic-ray simulation, x-y distribution

Figure 14 10^{15}eV

Cosmic-ray simulation, x-y distribution

Figure 15 10^{16}eV Figure 16 10^{15}eV Figure 17 10^{16}eV

この3次元グラフを縦方向の地表の距離 10^3 から -10^3m までの範囲で切り取り、宇宙線が到来してきた中心地点を原点とする、地表に到達する空気シャワーの粒子の個数と距離に関する2次元ヒストグラムを新たに作成する。Figure 16-17は 10^{15}eV と 10^{16}eV における2次元ヒストグラムであり、用いられている CORSIKA シミュレーションのデータは Figure 14-15 のものと同様である。この2次元ヒストグラムは空気シャワー粒子の到来個数の水平方向分布を示しており、縦軸はビンあたりの地表に到来する空気シャワーの粒子の個数、横軸は横方向の地表の距離をとっている。また、ROOT を用いてフィッティングすることで、空気シャワーの広がり分布モデルを構成する2つのパラメータ p_0 、 p_1 を算出することができる。これを宇宙線のエネルギーが 10^{14}eV 、 10^{15}eV 、 10^{16}eV 、 10^{18}eV の CORSIKA シミュレーション結果でそれぞれ行い、 10^{14}eV 、 10^{15}eV 、 10^{16}eV 、 10^{18}eV の空気シ

シャワーの広がり分布モデルを作成する。また、2次元ヒストグラムを作成するにあたり、横方向のビン幅が1mとなるようにx軸のビン数を設定する必要がある。たとえば、横方向の地表の距離が200kmであればx軸のビン数は200000に設定する。ただし、 10^{14} eVと 10^{15} eVの条件下では空気シャワー粒子の到来個数が少ないため、ビン幅が1mとなるように設定するとフィッティングするために十分なグラフを作成することができない。そのため、宇宙線のエネルギーが 10^{14} eVと 10^{15} eVの条件下では横方向のビン幅が10mとなるようにx軸のビン数を設定し、ビンごとの空気シャワーの粒子個数を増やしてグラフを作成できるようにする必要がある。この場合、後述するビンごとの面積を用いた計算の際にビン幅が10mである面積を代入することで調整する。y軸のビン数は2次元ヒストグラムに直接影響しないため、x軸のビン数よりも少ない100に設定する。また、パラメータ p_0 は 1m^2 あたりの値である必要があるため、フィッティングにより算出した p_0 の値を各ビンの面積 m^2 で割る必要がある。フィッティングは以下に示す関数を ROOT 上の FitPanel で入力し行った。

$$([0]/x) * \exp(-(1 * (x/[1])))$$

これらの方法を用いて宇宙線のエネルギーごとの空気シャワーの広がり分布を表す2つのパラメータ p_0, p_1 を算出した。Table 1 はこれを示したものである。

Table 1 空気シャワー分布に関するパラメータ

	10^{14}eV	10^{15}eV	10^{16}eV	10^{18}eV
p_0	0.153	2.14	12.2	1194
p_1	9.78E+11	3.06E+12	4.99E+12	1.01E+13

5.2 宇宙線の同時測定確率の算出

フィッティングを用いて算出したパラメータ p_0, p_1 と Python で記述したプログラム (ObservingCosmicRaySimulation.py、付録に記載) を用いて、宇宙線のエネルギーごとに2検出器間の距離による同時測定確率の算出を行う。このプログラムは、入力したパラメータ p_0, p_1 により定義される空気シャワーの広がり分布を持った宇宙線を、指定したエリアの範囲内のランダムな地点に打ち込み、指定したエリア内にある検出器が検出する空気シャワーの粒子数の期待値を算出するものである。エネルギーごとの空気シャワーの分布に関するパラメータである p_0 と p_1 の他に、宇宙線検出器の大きさや数、異なる検出器間の距離などを自由に定義することができる。本節の最終的な目的は早稲田大学本庄高等学院と広尾学園高等学校間での高エネルギー宇宙線および超高エネルギー宇宙線の同時測定確率の算出であるため、宇宙線検出器の数は2つに設定する。以下にプログラム内で定義される主要なパラメータと、実際に2検出器間の距離による同時測定確率を算出するにあたり設定した値を示す。

• detector_arr.append(Detector("det0", [10000, 1], 1.0, [0.2, 0.2]))

Detector()内のパラメータは左から検出器名、検出器の座標、検出効率、検出器の有効検出範囲の大きさを表す。検出効率以外の値の単位は m である。検出器の座標は宇宙線の同時測定確率を調べたい、2 つの検出器間の距離となるような座標をそれぞれ設定する。たとえば、2 つの検出器間の距離が 1m の場合で宇宙線の同時測定確率を調べる場合には、1 つ目の検出器の座標を [2, 1]、2 つ目の検出器の座標を [1, 1] とする。有効検出範囲の値については、第 3 章の観測機器の節で示したプラスチックシンチレーターの有効検出範囲を参考にする。有効検出範囲は約 0.2m 四方であるため、検出器の有効検出範囲を [0.2, 0.2] という値に設定する。

• cosmicray = CosmicRayApprox(1.5, 1e12, [0, 0])

CosmicRayApprox()内のパラメータは左から p0、p1、到来する宇宙線の中心座標を表す。たとえば、 10^{15} eV のエネルギーの宇宙線の同時測定確率を調べる場合には、Table 1 で示した p0、p1 の値より (2.14, 3.06e12) という数値を入力する。

• run = 100000

シミュレーションの試行回数を表す。これは打ち込む宇宙線の個数と捉えることもできる。今回は各エネルギーおよび各 2 検出器間の距離ごとに 100000 回ずつシミュレーションを行うため、100000 という値に設定する。

• area=[-10000, 10000]

宇宙線が到来するエリアの範囲を表す。2 つの検出器間の距離の最大値よりも広い範囲を設定する必要がある。今回行うシミュレーションは早稲田大学本庄高等学院と広尾学園高等学校との間の宇宙線の同時測定確率の算出が目的であるため、2 つの検出器間の距離の最大値は早稲田大学本庄高等学院-広尾学園高等学校間の距離である 79150m である。そのため、宇宙線が到来するエリアの範囲はこの距離よりも広い 100000m 四方に設定する。

このプログラムを実行すると、シミュレーションの試行回数と同じ数だけ 2 つの検出器名と数字のデータが表示される。この数字はシミュレーションごとの、各検出器が検出する空気シャワーの粒子数の期待値を表している。この期待値は以下のポアソン分布に従う。

$$P(k) = e^{-\lambda} \frac{\lambda^k}{k!}$$

宇宙線の 2 検出器間の同時測定確率を求めるためには、検出器ごとに空気シャワーの粒子が 1 個でも検出できる確率を求める必要がある。この確率は空気シャワーの粒子が 1 個も検出できない確率の余事象である。空気シャワーの粒子が 1 個も検出できない確率はポアソン分布に $k = 0$ を代入した式で求められる。よって、空気シャワーの粒子が 1 個でも検出できる確率を求める式は以下の通りとなる。

$$P = 1 - e^{-\lambda}$$

また、検出器 0 で検出される粒子数の期待値を λ^0 、検出器 1 で検出される粒子数の期待値を λ^1 とおくと、シミュレーションで宇宙線を打ち込む回数 λ のうち 2 つの検出器で同時測定できる回数 N を求める式は以下の通りとなる。

$$N = \sum_{n=1}^{\lambda} (1 - e^{-\lambda^0})(1 - e^{-\lambda^1})$$

この式を用いて、宇宙線を 100000 回打ち込んだ際に 2 つの検出器で同時測定できる回数を、2 つの検出器間の距離ごとに算出した。Table 2 はこれを示したものである。

Table 2 2 検出器間の距離による同時測定個数

	10^{14}eV	10^{15}eV	10^{16}eV	10^{18}eV
1m	3.059E-06	0.0004045	0.01453	92.46
2m	2.312E-06	0.0003979	0.01471	92.89
3m	2.637E-06	0.0003867	0.01444	91.63
500m	2.256E-06	0.0004367	0.01362	90.82
1250m	2.159E-06	0.0003461	0.01275	89.25
2500m	1.960E-06	0.0003629	0.01217	88.48
5000m	1.549E-06	0.0003097	0.01070	81.82
7500m	1.445E-06	0.0002893	0.009988	76.60
10000m	1.380E-06	0.0002725	0.008698	73.08
20000m	1.145E-06	0.0002282	0.007005	63.39
30000m	1.017E-06	0.0001959	0.006272	55.46
40000m	8.868E-07	0.0001716	0.005602	49.94
50000m	7.945E-07	0.0001556	0.005035	45.30
60000m	7.228E-07	0.0001407	0.004592	41.68
70000m	6.544E-07	0.0001316	0.004158	37.65
79150m	6.281E-07	0.0001190	0.003862	34.85

5.3 2つの検出器間の距離による同時測定確率の依存性

Table 2 に示した宇宙線を 100000 回打ち込んだ際の 2 検出器間の距離による同時測定個数をグラフに表すことで、宇宙線のエネルギーごとの 2 つの検出器間の距離による同時測定確率の依存性を確認しやすくなる。Figure 18-21 は 10^{14}eV から 10^{18}eV までの各エネルギーにおける宇宙線を 100000 回打ち込んだ際の 2 検出器間の距離による同時測定個数を示したグラフであり、縦軸に宇宙線を 100000 回打ち込んだ際の 2 つの検出器による同時測定個数、横軸に 2 つの検出器間の距離をとる。 10^{14}eV 、 10^{15}eV では 2500-5000m の範囲において同時測定個数が急激に減少しており、傾きが大きくなっているため、この範囲の間に感度の高い場所があると考えられる。 10^{16}eV では 7500-10000m の範囲で同時測定個数が急激に減少しているため傾きが大きくなっており、10000m よりも長い範囲では傾きが小さくなっている。このため、7500-10000m の範囲に感度の高い場所があると考えられる。一方で超高エネルギー宇宙線に相当する 10^{18}eV では、 10^{14}eV 、 10^{15}eV 、 10^{16}eV の場合と比較して 10000m までの範囲で同時測定個数は緩やかに減少していることがわかる。

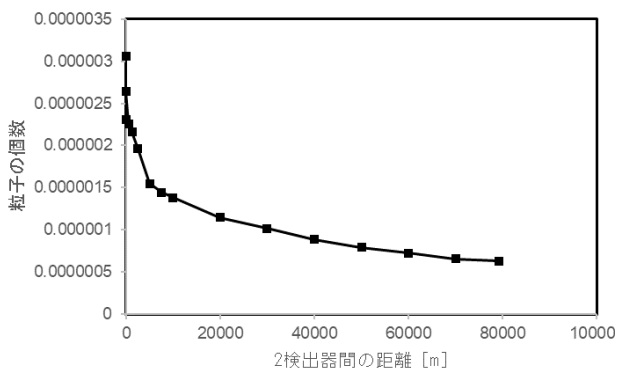


Figure 18 10^{14}eV

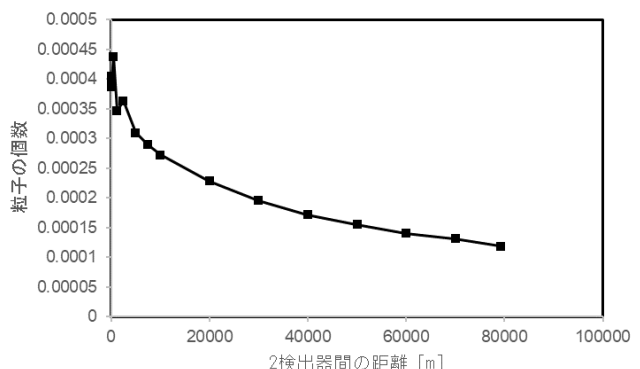


Figure 19 10^{15}eV

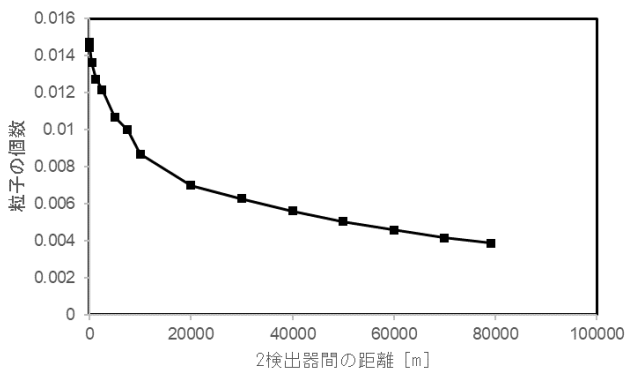


Figure 20 10^{16}eV

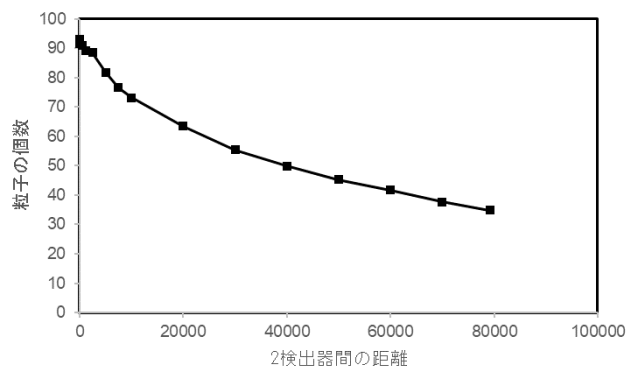


Figure 21 10^{18}eV

第 6 章 結論

第 3 章の観測機器の節で示した 2 つの宇宙線検出器を用いて 10^{18} eV より大きなエネルギーを持つ超高エネルギー宇宙線の同時測定を行う場合、2 つの検出器間の距離としては、10000m よりも長く、かつその範囲内でできるだけ短い距離が適していると考えられる。

Figure 18-20 で示してあるように、10000m よりも長い距離では 10^{18} eV 以下のエネルギーを持つ宇宙線の 2 つの検出器による同時測定確率が急激に下がるため、10000m よりも長い距離が 2 つの宇宙線検出器を使用した超高エネルギー宇宙線の同時測定には適していることがわかる。また、2 つの検出器間の距離について、早稲田大学本庄高等学院と広尾学園高等学校間の距離である 79150m の距離をとった場合と、10000m の距離をとった場合で 10^{18} eV のエネルギーを持つ宇宙線を 100000 回打ち込んだ際の宇宙線の同時測定個数を比較すると、Table 2 で示してあるように、79150m の距離をとった場合の個数は 10000m の距離をとった場合の個数の半分以下となっている。これにより、2 つの検出器間の距離が 10000m よりも長い範囲では、より短い距離をとったほうが 2 つの検出器による超高エネルギー宇宙線の同時測定確率が上がり、同時測定に適していることがわかる。

以上より、早稲田大学本庄高等学院と広尾学園高等学校間での超高エネルギー宇宙線の同時測定は、 10^{18} eV 以下のエネルギーを持つ宇宙線の同時測定確率が低くなるため、超高エネルギー宇宙線の同時測定には妥当な距離を 2 つの検出器間で確保できているものの、2 つの検出器間の距離を 10000m までの範囲でできるだけ近付けた方が超高エネルギー宇宙線の同時測定確率が上がり、より超高エネルギー宇宙線の同時測定に適していると結論付けられる。

第 7 章 参考文献

[1] 『高エネルギー宇宙物理学』

桜井邦朋 朝倉書店 1990 年 2 月 15 日

[2] 『第 2 版 宇宙の観測Ⅲ -高エネルギー天文学』

井上一、小山勝二、高橋忠幸、水本好彦 日本評論社 2019 年 10 月 20 日

[3] 『宇宙素粒子物理学』

C. グルーペン 丸善出版 2012 年 1 月 20 日

[4] 『物理学実験 I : 宇宙線』

神戸大学 大学院理学研究科 物理学専攻 粒子物理学研究室

[5] 『モンテカルロ計算を用いたガンマ線天文台 Cherenkov Telescope Array 大口径望遠

第8章 謝辞

本論文を執筆するに当たり、指導教員である早稲田大学本庄高等学院物理科教諭の大塚未来先生からは適切な指導を賜りました。感謝いたします。また、東北大学サイクロトロン・ラジオアイソトープセンター測定研究部の田中香津生助教授からは解析プログラムの使用方法などの多大な助言を賜りました。厚く感謝を申し上げます。

第9章 付録

all-inputs

RUNNR	2									run number
EVTNR	1									number of first shower event
NSHOW	1									number of showers to generate
PRMPAR	14									particle type of prim. particle
ESLOPE	-2.7									slope of primary energy spectrum
ERANGE	1.E5	1.E5								energy range of primary particle
THETAP	20.	20.								range of zenith angle (degree)
PHIP	-180.	180.								range of azimuth angle (degree)
SEED	1	0	0							seed for 1. random number sequence
SEED	2	0	0							seed for 2. random number sequence
OBSLEV	110.E2									observation level (in cm)
FIXCHI	0.									starting altitude (g/cm**2)
MAGNET	20.0	42.8								magnetic field centr. Europe
HADFLG	0	0	0	0	0	2				flags hadr.interact.&fragmentation
ECUTS	0.3	0.3	0.003	0.003						energy cuts for particles
MUADDI	T									additional info for muons
MUMULT	T									muon multiple scattering angle
ELMFLG	T	T								em. interaction flags (NKG,EGS)
STEPFC	1.0									mult. scattering step length fact.
RADNKG	200.E2									outer radius for NKG lat.dens.distr.
LONGI	T	10.	T	T						longit.distr. & step size & fit & out
ECTMAP	1.E4									cut on gamma factor for printout
MAXPRT	1									max. number of printed events
DIRECT	./									output directory
USER	you									user
DEBUG	F	6	F	1000000						debug flag and log.unit for out
EXIT										terminates input

colz.C

```
//////////////////////////////////////  
//colz by Kazuo Tanaka //  
//////////////////////////////////////  
//////////////////////////////////////  
//////////////////////////////////////  
//////////////////////////////////////  
//include
```



```

#include <fstream>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
/////////////////////////////////////////////////////////////////
//for 2D histogram
const int  xnbin = 200000;      // number of bins in x-axis
const double xmin = -100000;   // minimum of x-axis
const double xmax = 100000;    // maximum of x-axis
const int  ynbin = 100;        // number of bins in y-axis
const double ymin = -100000;   // minimum of y-axis
const double ymax = 100000;    // maximum of y-axis
const double zmin = 1;         // minimum of z-axis
const double zmax = 1e5;       // maximum of z-axis
const double emin = 0;         // selection for minimum "e" (might be energy)
const double emax = 5e3;       // selection for maximum "e" (might be energy)
const double ycut_min = -1000; // minimum of y-axis
const double ycut_max = 1000;  // maximum of y-axis
/////////////////////////////////////////////////////////////////
const string ene_selection = Form("%lf<e&&e<%lf", emin, emax);
const string y_selection = Form("%lf<y&&y<%lf", ycut_min, ycut_max);

/////////////////////////////////////////////////////////////////
//titles for histogram
const string xtitle = "X [m]"; // Title of x-
axis
const string ytitle = "Y [m]"; // Title of y-
axis
const string title = "Cosmic-ray simulation, x-y distribution"; // Title of
histogram
/////////////////////////////////////////////////////////////////

TCanvas *c1;
TFile *fF;
TTree *fT;
TH2D *h2;

void Draw()
{
    //二次元プロファイルの Tpad を設定
    c1 = new TCanvas("c1", "profile image", 10, 10, 750, 900);
    double sepa = 0.65; //上下の比
    TPad *pad0 = new TPad("pad0", "pad0", 0.0, 1 - sepa, 1, 1, 0, 0, 0);
    pad0->SetGrid(1, 1);
    pad0->SetTopMargin(0.1);
    pad0->SetBottomMargin(0);
    pad0->Draw();
    pad0->SetNumber(1); //番号をつけて id する
    //射影の Tpad を設定
    TPad *pad1 = new TPad("ratio", "ratio", 0.0, 0, 1, 1 - sepa, 0, 0, 0);
    //divide (bottom)
    pad1->SetGrid(1, 1);
    pad1->SetTopMargin(0);
    pad1->SetBottomMargin(0.1);
    pad1->Draw();
    pad1->SetNumber(2);
}

```

```

c1->cd(1);
gStyle->SetOptFit(1111);
h2->SetTitle(xtitle.c_str());
h2->SetTitle(ytitle.c_str());
h2->GetZaxis()->SetRangeUser(zmin, zmax);
h2->Draw("colz");
h2->GetXaxis()->SetLabelSize(0.03); //x 軸の目盛りサイズを設定
h2->GetYaxis()->SetLabelSize(0.03); //y 軸の目盛りサイズを設定
h2->GetXaxis()->SetTitleSize(0.03); //x 軸タイトルの文字サイズを設定
h2->GetYaxis()->SetTitleSize(0.03); //y 軸タイトルの文字サイズを設定
c1->Update();
gPad->SetLogz();

c1->cd(2);
gStyle->SetOptFit(1111);
h1->SetTitle(xtitle.c_str());
h1->GetZaxis()->SetRangeUser(zmin, zmax);
h1->Draw("");
h1->GetXaxis()->SetLabelSize(0.06); //x 軸の目盛りサイズを設定
h1->GetYaxis()->SetLabelSize(0.06); //y 軸の目盛りサイズを設定
h1->GetXaxis()->SetTitleSize(0.06); //x 軸タイトルの文字サイズを設定
h1->GetYaxis()->SetTitleSize(0.06); //y 軸タイトルの文字サイズを設定
c1->Update();
}
void Fit()
{
    f1 = new TF1("f1", "[0] / x * exp (-1 * x/[1])");
    //f1 = new TF1("f1", "[0] / ( 2 * 3.141592 * [1] * [1]) * pow( x / [1] , [2]-
2 ) * pow(1 + x / [1] , [2] - 4.5 ) * TMath::Gamma(4.5-[2]) / ( TMath::Gamma([2])
* TMath::Gamma(4.5-2*[2]))");
    f1.SetParameter(0,100);
        f1.SetParameter(1,1000000);
        f1.SetParameter(2,2);
    h1->Fit("f1", "", "", 10, 20000);
    h1->GetFunction("f1")->SetLineWidth(1);
}
void colz_v3(string file_name)
{
    fF = new TFile(file_name.c_str(), "read");
    fT = (TTree *)fF.Get("data_1");
    h2 = new TH2D("h2", "", xnbins, xmin, xmax, ynbins, ymin, ymax);
    fT->Project("h2", "y:x", ene_selection.c_str());
    h1 = new TH1D("h1", "", xnbins, xmin, xmax);
    fT->Project("h1", "x", y_selection.c_str());
    Draw();
    Fit();
}

```

ObservingCosmicRaySimulation.py

```

# -----
# import
# -----
import numpy as np
import random
import math

```

```

from matplotlib.figure import Figure
import matplotlib.pyplot as plt
import matplotlib as mpl
# -----
# Class
# -----

#Detector Class
class Detector():
    def __init__(self, name, pos, eff, size):
        self.name =name
        self.pos =pos
        self.eff = eff
        self.size = size

    def detectionProb(self, rho):
        return self.eff * rho

    def detection(self, rho):
        if random.random() < detectionProb(rho):
            return 1
        else:
            return 0

#CosmicRay Class
class CosmicRayApprox():
    def __init__(self, A, B, core_pos):
        self.core_pos = core_pos
#         self.r_M = 1
#         self.s = 1
#         self.N = 10000000
        self.A =A
        self.B = B
    def lateralDistribution(self, pos):
        r = np.sqrt((pos[0]-self.core_pos[0])*(pos[0]-self.core_pos[0])+(pos[1]-self.core_pos[1])*(pos[1]-self.core_pos[1]))
        rho = (self.A/r)*math.exp(-r/self.B)
#         rho = self.N / ( 2 * math.pi * self.r_M * self.r_M) * math.pow( r / self.r_M , self.s-2 ) * math.pow(1 + r / self.r_M ,self.s - 4.5 ) * math.gamma(4.5-self.s)/ ( math.gamma(self.s) * math.gamma(4.5-2*self.s))
        return rho

    def refresh(self, core_pos):
        self.core_pos = core_pos

# -----
# main
# -----
detector_arr =[]
detector_arr.append(Detector("det0", [300, 1], 0.8, [0.02, 0.02]))
detector_arr.append(Detector("det1", [1, 1], 0.8, [0.02, 0.02]))

cosmicray = CosmicRayApprox(1.5, 1e12, [0, 0])
run = 10
area=[-1000, 1000]

```

```
for i in range (1,run):

cosmicray.refresh([random.uniform(area[0], area[1]), random.uniform(area[0], area
[1])])
    for detector in detector_arr:

print(detector.name, detector.detectionProb(cosmicray.lateralDistribution(detec
tor.pos)))

r_arr = np.arange(10, 1000, 1)
rho_arr = []
for r in r_arr:
    rho_arr.append(cosmicray.lateralDistribution([r, 0]))
plt.plot(r_arr, rho_arr)
plt.show()
```