相対論的重イオン衝突実験PHENIXにおける Aerogel Cherenkov Counterのシミュレーションによる評価

筑波大学第一学群自然学類物理学専攻
平成13年度卒業論文

学籍番号 980355
高木 敏志

指導教官 :江角晋一

2002年2月4日

概要

最近、クォーク・グルーオン・プラズマ（QGP）生成のシグナルとして、『高横運動量粒子生成の抑制』が RHIC-PHENIX 実験において観測され、粒子の種類による違いが注目されている。そのため、高横運動量領域での粒子識別が重要となってくる。

現在、PHENIX 実験ではハドロン識別のための飛行時間測定器（ToF）をはじめ、粒子識別を目的とした様々な検出器が組み込まれていが、現在稼働中の ToF や RICH では高横運動量領域のハドロン識別には限界がある。そこで、識別出来る運動量領域を広げるために Aerogel Cherenkov Counter を PHENIX 検出器に組み込むことを検討している。ToF のみでは π 中間子と K 中間子の識別は 2.4GeV/c まで、K 中間子と陽子の識別は 5.0GeV/c まで可能であるが、これに Aerogel Cherenkov Counter を組み合わせることで π 中間子と K 中間子の識別が約 5GeV/c まで、K 中間子と陽子の識別が約 10GeV/c まで広がると期待されている。

設計のために Aerogel Cherenkov Counter 個々の性能は別途評価を進めているが、PHENIX 実験における他の検出器間の影響やバックグランド、そして動作環境等を評価することも重要となってくる。

本研究では、高エネルギー粒子が検出器等の物質中で引き起こす現象のシミュレーションを可能にしたシミュレーションコード（GEANT）を用いて、$\theta = 90^\circ$ 付近、ビーム軸から 450cm のところに ToF の 4 パネルの大きさ相当する約 200cm × 200cm の領域を覆うように縦 17 個、横 17 個（計 289 個）の Aerogel Cherenkov Counter（一個の大きさが 12cm×12cm×12cm）を組み込んだ PHENIX 検出器を計算機上に再現した。そして、核子あたり 100GeV の原子核・原子核衝突による粒子生成を再現するための event generator（HIJING）用いて、Aerogel Cherenkov Counter を組み込んだ PHENIX 検出器における核子あたり 100GeV の金・金衝突実験を計算機上に再現した。

その結果、バックグランドも含めた Aerogel Cherenkov Counter の占有率は ~ 5 % 程度であることがわかった。また、バックグランドは Dorift Chmber、Pad Chmber（PC1、PC2、PC3）、Ring Image Cherenkov Counter（RICH）等の検出器を用いることで、その 9 割以上を排除することが可能である。そして、バックグランドの影響として Aerogel Cherenkov Counter から誤った情報を得てしまうことがある。その確率はおよそ 10% 程度であることがわかった。

目 次

第1章　序論

我々の宇宙は約150億年前にビックバンを起し、非常に小さな宇宙として誕生した。この宇宙創成の極めて初期段階に、核子や中間子からの閉じ込めから解放され自由に動きまわるクォークやグルーオンの集団、即ちクォーク · グルーオン · プラズマ(**QGP**)が存在したと推定されている。QGPとは、エネルギー密度約$1 \sim 3$GeV/fm^{3}の超高温 · 高密度の未知なる物質状態であり、QGPの探求は原子核物理に限らず、宇宙創成に関わる重要な課題である。そして高エネルギー重イオン衝突実験には、地上でQGPを生成する唯一の可能性を秘めている。

現在、米国ブルックヘブン国立研究所(BNL)において、世界初の衝突型相対論的重イオン加速器(**RHIC**:Relativistic Heavy Ion Collider)を用いて2000年春から、核子あたり100GeVの金の原子核同士の衝突実験が行われている。従来の静止標的型加速器とは異なり、衝突型加速器であるRHICは衝突エネルギーが格段に大きく、QGP生成が大いに期待されている。そして、衝突中心部のエネルギーは、2 ~ 6 GeV/fm^{3}に達するであろうと予想されている。このエネルギー密度は、量子色力学(QCD)が予想しているQGP相転移に必要なエネルギー密度1.5 GeV/fm^{3}を大きく越えるものであり、それは人類がかつて手にしたことのない最高のエネルギー密度である。

我々の研究室ではRHICにおいて、QGPの研究を目的とした実験のひとつである**PHENIX**(the Pioneering High Energy Nuclear Interaction eXperiment)実験に参加しており、粒子識別のための飛行時間測定器(**ToF**:Time of Flight)の研究 · 開発を行っている。PHENIX実験では、我々の研究室が研究 · 開発を行っているハドロン識別用の飛行時間測定器をはじめとして様々な検出器が使われており、特に粒子識別能力が優れている。

QGP生成のシグナルとして、『ストレンジネスの増加』『J/ψ粒子の抑制』『ベクトル中間子の性質変化』『高横運動量粒子生成の抑制』等が考えられている。そして最近、PHENIX実験において『高横運動量粒子生成の抑制』が観測され、粒子の種類による違いが注目されている。そのため、横運動量で数GeV/cまでのハドロン識別及び、電子 · ミューオン · 光子の測定が重要になってくる。

現在稼働中の飛行時間測定器(ToF)ではπ中間子とK中間子の識別は2.4GeV/cまで、K中間子と陽子の識別は5GeV/cまで可能であるが、識別できる運動量領域をさらに広げるために、Aerogel Cherenkov CounterをPHENIX検出器に組み込むことを検討している。これにより、Aerogel Cherenkov Counter、ToFを組み合わせることでπ中間子とK中間子の識別は約5GeV/cまで、K中間子と陽子の識別は約9GeV/cまで識別が可能になると期待されている。

設計のために、Aerogel Cherenkov Counter 個々の性能は別途評価を行っているが、PHENIX 実験おける他の検出器間の影響やそのバックグランド、動作環境等も評価することが必要である。

本研究では、高エネルギー粒子が検出器等の物質中で引き起こす現象をシミュレーションし、検出器等の性能評価を可能としたシミュレーションコード（GEANT）と、核子あたり 100GeV の原子核・原子核衝突による粒子生成を再現するための event generator（HIJING）を用いて Aerogel Cherenkov Counter を組み込んだ PHENIX 検出器における核子あたり 100GeV の金・金衝突実験を計算機上に再現した。そして、PHENIX での Aerogel Cherenkov Counter の動作環境等をシミュレーションにより評価した。

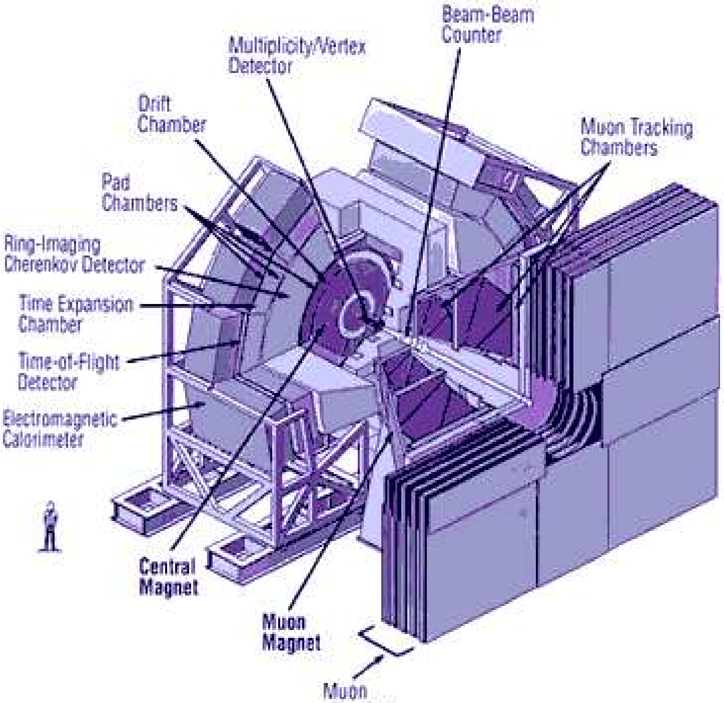

Fig. 1.1: PHENIX 検出器

第2章　本研究に関する基礎知識

2.1　高エネルギー重イオン衝突実験

従来の静止標的型実験において、重心系エネルギーは入射粒子（$\beta = 1$ とする）のエネルギーの $\frac{1}{2}$ 乗に比例して変化する。これは、入射粒子のエネルギーの相当の部分が運動量保存則を満たすのに使われるからである。これに対し、衝突型実験において、重心系エネルギーは入射粒子のエネルギーに比例して大きくなる。実験において要求されるエネルギーが大きくなるにつれ、衝突型加速器による衝突実験が主流となってくるだろう。

2.1.1　高エネルギー重イオン衝突実験の特徴

相対論的エネルギーにまで加速された原子核同士の衝突を重心系でみてみると、Fig.2.1 のように衝突前の原子核はローレンツ収縮により球というよりは碁石のようにみえる。一般に、入射粒子（projectile）と標的粒子（target）が重なりあった部分を反応関与部（participant）と呼び、直接には衝突には関与していない部分を反応傍観部（spectator）と呼ぶ。反応関与部では、激しい核子・核子衝突とそれによって生ずる π 中間子をはじめとする二次粒子群がさらに衝突を繰り返して高温・高密度状態が形成されると、期待されている。

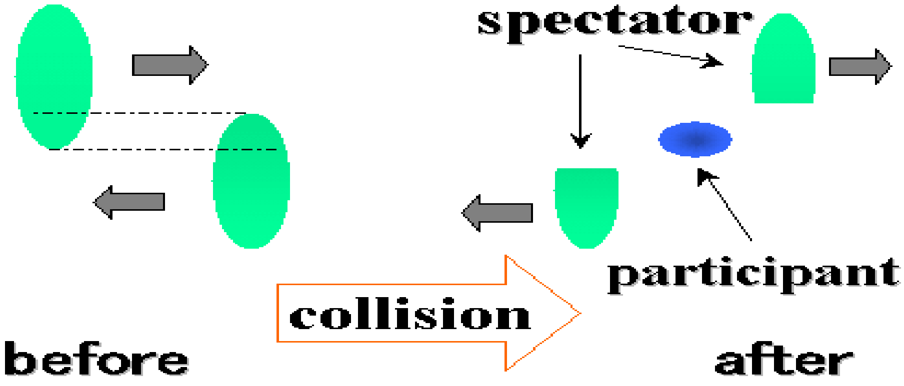

Fig. 2.1: 衝突前と衝突後の様子

原子核半径 R_A 及びドブロイ波長 λ_d は次式で表せる。

$$R_A \cong 1.2A^{1/3} \quad [fm] \tag{2.1}$$

$$\lambda_d = \frac{}{p} \quad [fm] \tag{2.2}$$

ここで、A は質量数、 $= 1.2$ [GeV·fm/c]、p は運動量 [GeV/c] である。本研究では重心系で核子あたり $E = 100\mathrm{GeV}$ のエネルギーであり、核子の静止質量より核子の運動量の方がずっと大きいと考えられるので、$p \cong 100\mathrm{GeV}$ とすると式 (2.2) より

$$\lambda_d \cong 0.012 \quad [fm] \tag{2.3}$$

となる。このとき、金原子核は $A = 197$ なので

$$R_A \cong 5.82 \quad [fm] \tag{2.4}$$

となり、原子核·原子核衝突は核子·核子衝突を素過程として考えて良いことが分かる。

2.1.2　高エネルギー重イオン衝突実験と QGP

通常、低温の原子において電子は、原子核のまわりのクーロン力によって束縛さている。このとき原子核は、その電荷の数と同数の電子をまとっているため、原子は全体として中性の状態にある。ところが、放電等によって加熱されると、電子は原子核の束縛を逃れることができ、原子は正イオンと自由電子とに分離する。これがよく知られたプラズマ状態である。

ところで、通常の原子核は核子（陽子と中性子の総称）から成り、またこの核子は 3 個のクォークから構成されている（陽子 :uud、中性子 :udd）。クォークから構成される粒子には核子のように 3 種類のクォークから成る重粒子の他にクォーク・反クォークからなる中間子がある。クォークはグルーオンと呼ばれる粒子によって強く結びつけられているため、通常はクォーク単体では飛び出すことはできない。

しかし、原子核を圧縮して核子間距離が核子の半径（約 1fm）以下になると核子の境界は意味を持たなくなり、クォーク・グルーオンは自由に動きまわると考えられている。この状態を原子のプラズマ状態との比較からクォーク・グルーオン・プラズマ（QGP）と呼んでいる（Fig. 2.2）。

この QGP を実験室で生成するためには、非常に高温高密度な状態を作ることが必要になるが、このときに用いられるのが原子核・原子核衝突である。原子核は、原子の約 1 万分の 1 の大きさでありながら、原子全体の質量のほとんどを占めている。つまり、原子核は数億トン /cm^3 という途方もない高密度の世界なのである。このような高密度の原子核同士を相対論的なエネルギーまで加速して正面衝突させると、その中心部では原子核の圧縮、及び高密度の粒子生成によって、さらなる高温高密度

な状態ができると期待される。PHENIX 実験においては、金の原子核同士が核子当り 100GeV まで加速され正面衝突する。このとき、反応中心部ではエネルギー密度が $2\sim6\mathrm{GeV}/fm^3$ にも達し、量子色力学（QCD）が予想する QGP 相転移に必要なエネルギー密度 $1.5\mathrm{GeV/fm}^3$ をはるかに越えるであろうと期待されている。

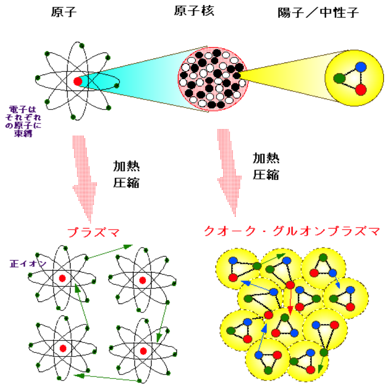

Fig. 2.2: クォーク · グルーオン · プラズマ（QGP）

2.1.3 高エネルギー重イオン衝突実験でよく用いられる物理量

impact parameter : b、reaction plane の角度 : Ψ_0

原子核同士が衝突する際、beam 軸と原子核の中心を通る beam 軸に平行な直線から決まる距離を impact parameter といい通常 b で表す。単位は一般に $fm(=10^{-15}m)$ であり、金の原子核の半径はおよそ $6fm$ であるので b の最大値はおよそ $12fm$ くらいになる。

また、impact parameter と beam 軸から 1 つに決まる平面を reaction plane または反応面と呼び、この面の水平方向からのずれを Ψ_0 で表す。(Fig.2.3 参照)

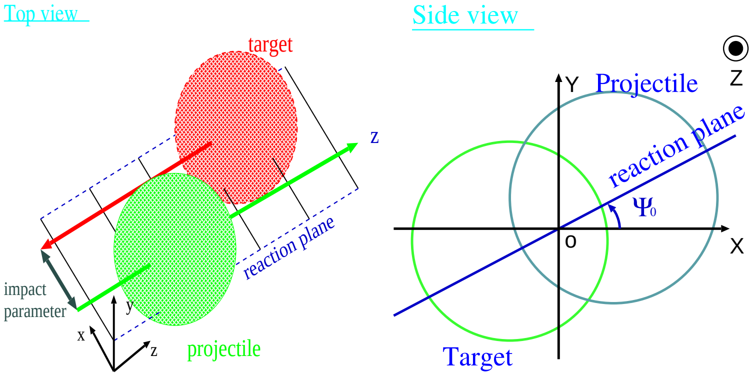

Fig. 2.3: impact parameter:b 及び reaction plan の定義

rapidity：y

非相対論では粒子の速度は概念として分かりやすく、速度の合成は単なるベクトルの和で表される。しかし、粒子の速度が大きくなり相対論的運動になると単なるベクトルの和というわけにはいかない。たとえば、慣性系Kに対してz軸正の向きに速度uで走る座標系K'を考える。K'系でz方向の粒子の速度をvとし、vをK系で見たときの速度をwとすると、

$$w = \frac{u+v}{1+uv/c^2} \tag{2.5}$$

となる。つまり、ローレンツ変換によって速度は非線型の合成則に従い、その上限が光速cで抑えられてしまう。このように、相対論的な運動では速度という概念は必ずしも便利な量とは限らない。そこで、ローレンツ変換にお対して線形の合成則に従い、beam軸方向のある種の速度を表す量として、次のようなrapidityという量を定義し、これをyで表すことにする。

$$y = \frac{1}{2}\ln\left(\frac{E+P_z}{E-P_z}\right) \tag{2.6}$$

ここでEは全エネルギー、P_zはbeam軸方向の運動量である。

また高エネルギーにおいては、静止質量mと全運動量Pとの関係は$P \gg m$となるので、

$$y \approx \frac{1}{2}\ln\left(\frac{P+P_z}{P-P_z}\right) = \frac{1}{2}\ln\left(\frac{1+\cos\theta}{1-\cos\theta}\right) = -ln\left(\tan\left(\frac{\theta}{2}\right)\right) \tag{2.7}$$

となり、粒子の放出角だけで定義されるようになる。このyをηで表し擬ラピディティーと呼ぶ。

横方向運動量 (transverse momentum)： P_t

高エネルギー重イオン衝突ではbeam軸方向にはローレンツ変換するが、beam軸に垂直な方向にはローレンツ変換されない。そこでbeam軸方向に平行なローレンツ変換(例えば、実験室系から重心系への変換)に対して不変な量として、横運動量P_tを用いる。P_tは以下の式で与えられる。

$$P_t = \sqrt{{P_x}^2 + {P_y}^2} = P\sin\theta \tag{2.8}$$

2.2 PHENIX 実験とその検出器

PHENIX 実験とは、米国ブルックヘブン国立研究所（BNL）において世界初の衝突型高エネルギー重イオン加速器（RHIC）を用いておこなわれている 4 つの実験（PHENIX、STAR、PHOBOS、BRAHMS）の一つである。その目的は、QGP を生成・検出し、探求することである。この章では、PHENIX 検出器において本研究と特に関係の深い検出器について述べる。

2.2.1 飛行時間測定器（Time of Flight）

飛行時間測定器（ToF）は $\theta = 90^\circ$、$\phi = 180^\circ$ 付近、ビーム軸から 500cm のところに位置しており、その主な目的はハドロン（π 中間子、K 中間子、陽子）の識別である。その構造は、1cm × 1cm × 63cm のシンチレーターが 64 本、1cm × 1cm × 43cm のシンチレーターが 32 本の計 96 本のシンチレーターと、その両端に信号読み取りのための光電子増倍管（PMT）が計 192 個取り付けられている。1 パネルは 200cm × 50cm であり、全部で 10 パネルが配置されている。飛行時間測定器（ToF）の時間分解能は 80ps 以下である。

ハドロン識別の方法は、飛行時間法（TOF 法）と呼ばれる方法を用いておこなわれている。それは、電磁石や飛跡検出器を用いて磁場中での曲率半径から粒子の運動量 p を求める。これと、スタートカウンターに Beam-Beam Counter（BBC）を、ストップカウンターに飛行時間測定器（ToF）を用いて一定の距離を通過するのに要する時間を測定し、粒子の速度 v をもとめる。そして、

$$mc^2 = cp\sqrt{\beta^{-2} - 1} \qquad (c : \text{真空中の光速度、} \beta = \frac{v}{c}) \tag{2.9}$$

の関係式から粒子の質量 m をもとめ、粒子の種類を特定するという方法である。

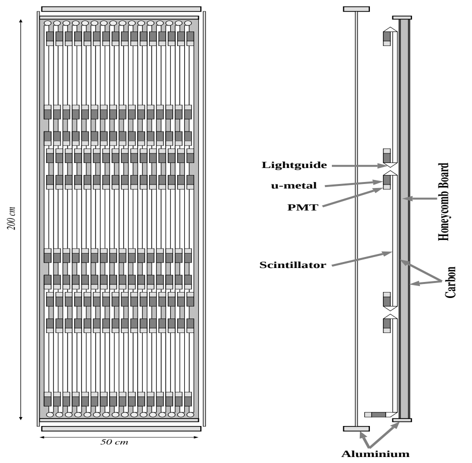

Fig. 2.4: 飛行時間測定器（ToF）

2.2.2　Beam-Beam counter

ビーム軸の周り、前後方にBeam-Beam counterが設置されている。これは、それぞれ64本のradiatorとして石英を使用したCherenkov Counterである。
その主な目的は、

- ビーム・ビーム衝突を検出し、トリガーを作り出す
- 反応開始時間を測定し、飛行時間測定による粒子識別のためのスタート時間を測定する
- 前後のBeam-Beam counterへのシグナルの時間差から衝突点の位置を決定する

ということである。

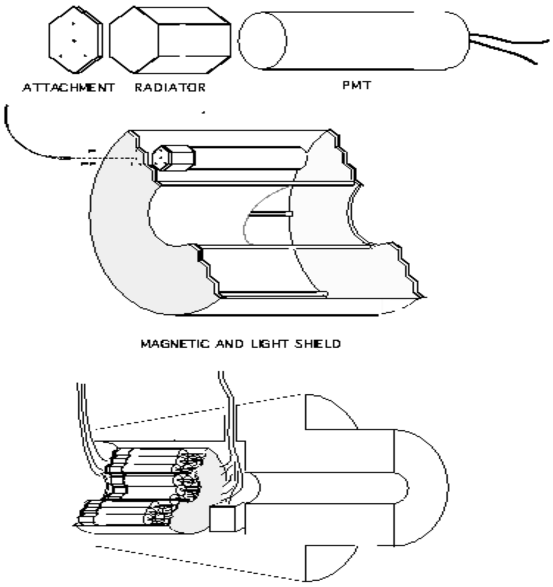

Fig. 2.5: Beam-Beam Counter(BBC)

2.2.3　Drift Chamber

PHENX検出器の一番内側に、ビーム軸から202cm $\sim$ 246cmにかけてDrift Chamberが配置されている。Drift Chamberの内径が202cmと大きいのは、これより内側では飛跡密度が高くなりすぎて飛跡再構成が困難になるためである。
その主な目的は、

- 荷電粒子の飛跡を測定する
- 荷電粒子の運動量を測定する

ということである。

入射荷電粒子によって最初に作られたイオン対の電子がワイヤーの位置まで移動するのに要する時間を測定して、ワイヤーから入射粒子の位置までの距離を測定するものである。そして、Drift Chamber の位置分解能は 150μm （$r-\phi$ 方向）である。

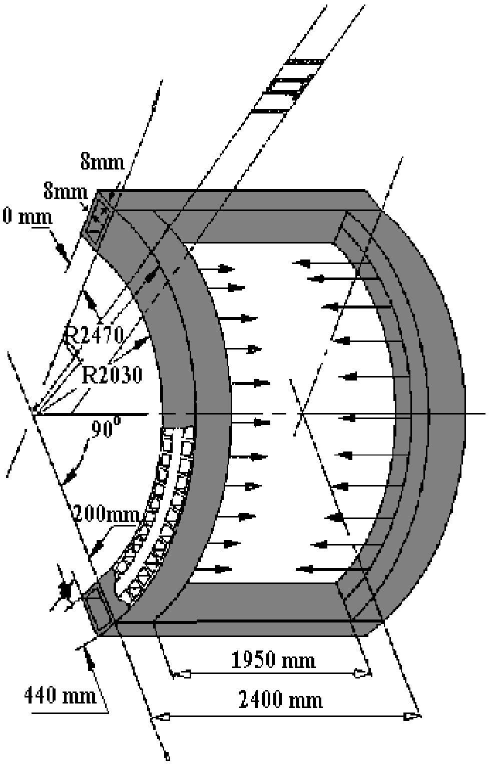

Fig. 2.6: Drift Chamber(DC)

2.2.4 Pad Chamber

PHENIX 検出器には Pad Chamber が合計 6 個配置されている（PC1,PC2,PC3）。PC1 は Drift Chamber（DC）の後ろで RICH（Ring Image CHerenkov detecter）の前に、そして PC2 と PC3 は RICH の後ろに配置されている。
その主な目的は、

- 粒子飛跡上の 3 次元点を測定して飛跡再構成上でのパターン認識を助ける

ということである。

また、PC1,PC2,PC3 の情報と、DC と TEC(Time Expansion Chamber) からの情報を組み合わせることで粒子の tracking の精度をあげることができ、interaction vertex を調べることができる。

2.2.5 RICH (Ring Image CHerenkov detecter)

PHENIX実験において、RICHは電子識別のための主な検出器の一つであり、ビーム軸から255cm $\sim$ 401cmにかけてRICHが配置されている。RICHは容積約40m^{3}のガス容器と、その中に置かれた超薄型球面反射鏡、及びその焦点面に敷き詰められた多数の光電子増倍管（PMT）からなる。RICH全体では、5120本のPMTが使用されている。

本質的には、電子識別用の閾値型Cherenkov測定器である。リングイメージングの光学を用いているが、Cherenkovリングの直径の測定により粒子識別をしているわけではない。10MeV/c以上の電子がRICH中のgas radiator（二酸化炭素（CO_2、n $\sim$ 1.00041）内でCherenkov光を発生するのに対して、ほとんどのハドロンはCherenkov光を出さないことを利用して電子識別を行う。

磁場がほぼゼロの領域に置かれているため電子はRICH内を直進して、そこから発生するCherenkov光は球面反射鏡で反射され、結像する。Cherenkov Ringの中心位置から、親の電子の進行方向がわかるので飛跡検出器で測定された荷電粒子飛跡と組み合わせて電子の識別をおこなう。

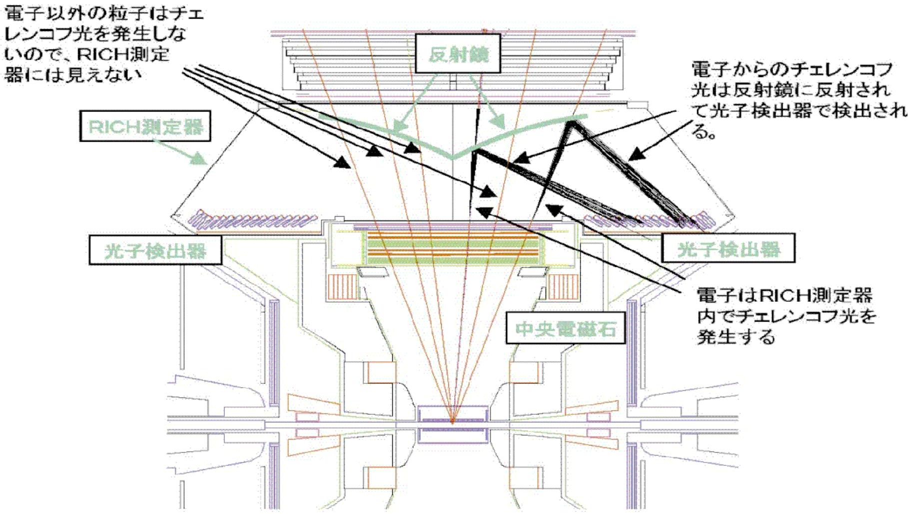

Fig. 2.7: RICH (Ring Image CHerenkov detecter)

2.3 Aerogel Cherenkov Counter

2.3.1 Aerogel

Aerogelは固体でありながら限りなく空気に近い物質とされている。シリカ（ガラス、土等の主成分）酸化物をゼリー状にしたところで、高温・高圧の超臨界状態で乾燥させることで作ることができる。加工性は非常にもろくあまりよくない。密度は約$40\mathrm{mg/cm}^3$という低密度性と、屈折率が極めて小さく（$\mathrm{n} = 1.006 \sim 1.07$）、表面にはほとんどまわりのものは映らない。そのために、Aerogelは輪郭をもたない固体ともいわれている。その光学的性質として光の吸収はほとんどないが、散乱効果（Rayleigh散乱）が著しくみられる。

特に注目すべき光学的性質としては、吸収長λ_{absp}と散乱長λ_{scat}の光の波長依存性ということがあげられる。現在、それらの間には以下のような依存性があると考えられている。

$$\lambda_{scat} \propto \lambda_{photo}^4 \tag{2.10}$$

$$\lambda_{absp} \propto \lambda_{photo}^2 \tag{2.11}$$

2.3.2 Cherenkov放射

物質中を通過する荷電粒子の速度vが、その物質中での光速$c' = c/n$ (c:真空中での光速、n:物質の屈折率) を越えると粒子の飛跡に沿って物質が発する弱い光のことをCherenkov光という。これは、荷電粒子の作る電場によって物質中に分極が起こり、粒子の通過後分極が消滅する際に発光するためである。

Fig.??に示すように時間tの間に荷電粒子が点Aから点Bへ進むとき、Aから出たCherenkov光はAを中心とする半径$(c/n)t$の球面上に達する。この時、すべてのCherenkov光はBを頂点とする頂角$\phi = \sin^{-1}(c/nv)$の円錐面上に到達し、波面を形成する。

シンチレーション光が等方的に放出されるのに対し、Cherenkov光は荷電粒子の進行方向と角度θをなす方向に放出される。ただし、

$$\cos\theta = \frac{1}{n\beta} \qquad (\beta = v/c) \tag{2.12}$$

である。

荷電粒子の飛跡の単位長さ、単位エネルギー幅当りのCherenkov光子の発生数は、

$$\begin{aligned} \frac{d^2N}{dEdx} &= \frac{\alpha z^2}{c}\sin^2\theta_c \\ &= \frac{\alpha^2 z^2}{r_e m_e c^2}\left(1 - \frac{1}{\beta^2 n^2}\right) \\ &\approx 370\sin^2\theta_c \qquad eV^{-1}cm^{-1} \qquad (z=1) \end{aligned} \tag{2.13}$$

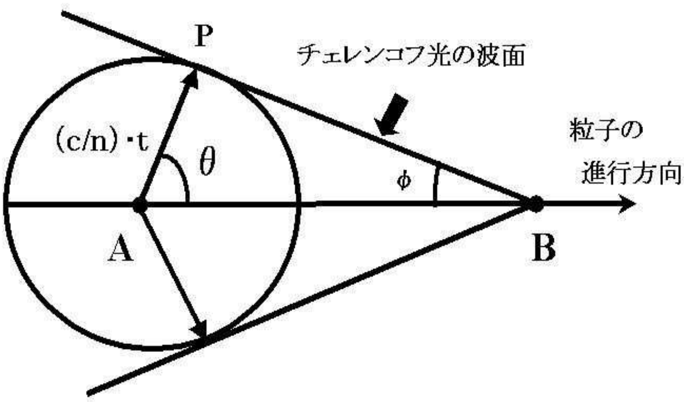

Fig. 2.8: cherenkov 放射の様子

もしくは、

$$\frac{d^2N}{dxd\lambda} = \frac{2\pi\alpha z^2}{\lambda^2}\left(1 - \frac{1}{\beta^2 n^2}\right) \tag{2.14}$$

で与えられる。荷電粒子は物質中を通過するときに Cherenkov 放射によってエネルギーを失うが、その量は一般的には電離損失や放射損失に比べてはるかに小さい。

(2.14) 式を見ればわかるように、波長 λ に対する Cherenkov 光子の収率は $1/\lambda^2$ に比例している。したがって、放射光のスペクトルは短波長領域に集中していることがわかる。

2.3.3 Cherenkov Counter

Cherenkov 光の検出を目的とした検出器である Cherenkov Counter は、高エネルギー物理学実験において幅広く用いられている。Cherenkov Counter は、Cherenkov 光が Cherenkov 媒体と光学的に接触している光電子増倍管 (PMT) で電気信号に変換されるという点で、シンチレーション・カウンターとたいへん類似しているが、その発光原理はまったく異なったものである。

Cherenkov Counter では粒子の放出する光を観測するので、使われる媒体は光に対して透明でなければならない。それに加え、シンチレーション成分を持たないということも大事である。

Cherenkov 光は、その発光時間が固体や液体中で通常 ps 程度である。シンチレーション光の発光時間は通常 ns 程度であることを考えると、Cherenkov Counter は非常に高速のものにできる可能性があるが、このような時間特性は PMT によって制限される。また、最大の欠点は発生する光のレベルが低いことである (高性能のシンチレーターの場合の 1/100 程度)。

Cherenkov Counter は、その使用目的により以下のタイプに分類することができる。

- 閾値型:Cherenkov 放射の条件が $\beta > 1/n$ ということを利用して、$v > c/n$ の速度の粒子をすべて検出するもの。
- 微分型:Cherenkov 放射の方向性を利用して、ある決まった光学系を用いて、特定の半径の Cherenkov Ring を放出した粒子 (特定の速度の粒子) だけを測定するもの。
- RICH:Cherenkov Ring の半径を測定することで、粒子の速度を測定するもの。

2.3.4 Aerogel Cherenkov Counter

Cherenkov 媒体として Aerogel を用いた Cherenkov Counter である。PHENIX 実験において、ハドロン識別のための ToF と Aerogel Cherenkov Counter を組み合わせることで π 中間子と K 中間子の識別は約 5GeV/c まで、K 中間子と陽子の識別は約 9GeV/c まで識別が可能になると期待されている。

現在、Aerogel Cherenkov Counter の形状として

- 散乱光型 ：Fig. 2.9 （左）
- 直接光型 ：Fig. 2.9 （右）

を考えている。

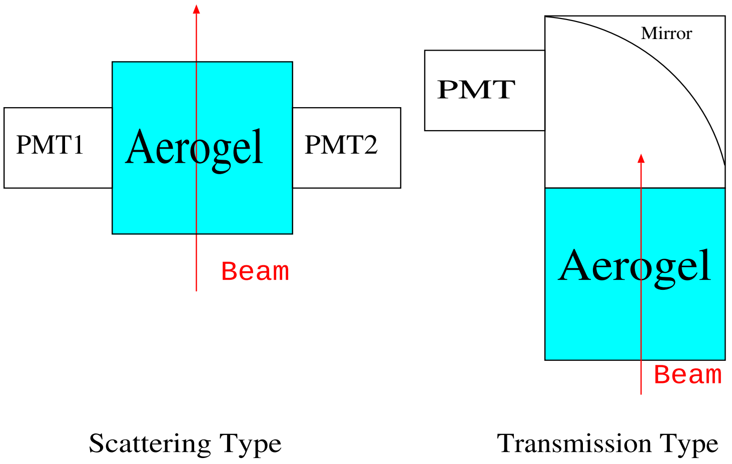

Fig. 2.9: Aerogel Cherenkov Counter の形状

本研究では Aerogel Cherenkov Counter の形状を散乱光型とし、シミュレーションを行った。

第3章　シミュレーションの方法

3.1　セットアップ

3.1.1　PHENIX 検出器

本研究では、GEANT3 を用いて PHENIX 検出器を計算機上に再現してた。GEANT とは、高エネルギー粒子が検出器等の物質中で引き起こす現象をシミュレーションし、検出器の性能評価を可能にするシミュレーションコードのことであり、高エネルギー原子核実験をはじめとして様々な分野にわたって利用されている。

それにくわえ、核子あたり 100GeV の原子核・原子核衝突による粒子生成を再現するための event generator である HIJING を使用し、Aerogel Cherenkov Counter が組み込まれた PHENIX 検出器における核子あたり 100GeV の金・金衝突実験を計算機上に再現した。以下、本研究にて計算機上に再現した PHENIX 検出器を Fig.3.1 に示す。

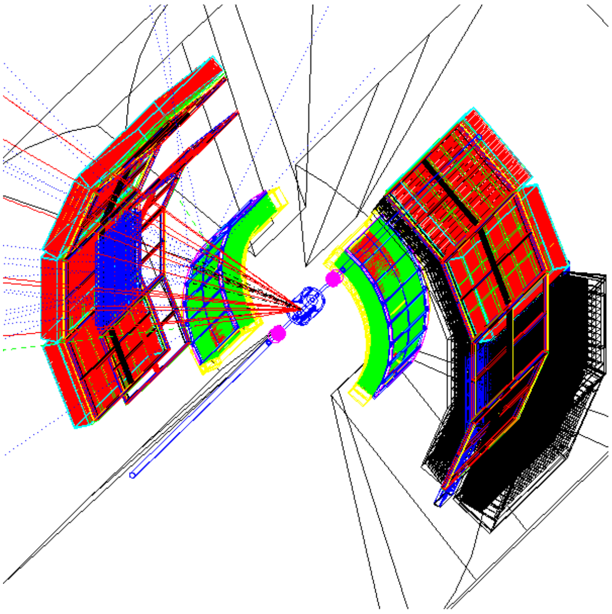

Fig. 3.1: 計算機上に再現された PHENIX 検出器

本研究では、飛行時間測定器（ToF）の 4 パネル分に相当する縦 200cm × 横 200cm の領域に、1 個の大きさが 12cm × 12cm × 12cm の Aerogel Cherenkov Counter を Fig.3.2 のように敷き詰めた。そして、Aerogel Cherenkov Counter を組み込む場所は $\theta = 90^\circ$、$\phi = 0^\circ$ 付近、ビーム軸から 450cm のところにした。

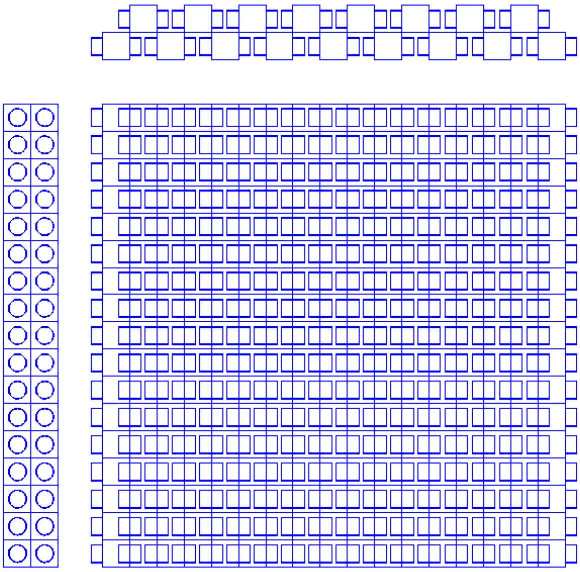

Fig. 3.2: Aerogel Cherenkov Counter の敷き詰め方

また、シミュレーションコード GEANT では、Cherenkov 放射による光量は計算されないので Cherenkov 放射を再現するようなプログラムと Aerogel 中の光の伝搬を再現するようなプログラムを別途作成する必要があった。

3.2 Aerogel Cherenkov Counterの設定

3.2.1 cherenkov放射の計算方法

GEANTでは、cherenkov放射による光量は計算されない。そのため、本研究ではcherenkov放射を再現するようなプログラムを作成した。以下、その計算方法を説明する。

放出される光子数

放出される光子数は、

$$\theta_c = \cos^{-1}\left(\frac{1}{n\beta}\right) \tag{3.1}$$

$$\frac{d^2N}{dEdx} = \frac{\alpha z^2}{c}\sin^2\theta_c \tag{3.2}$$

により、その計算をおこなった。

まず、(3.2)式をエネルギーEから、波長λへと変数変換をおこない、

$$\frac{d^2N}{d\lambda dx} = \frac{2\pi\alpha z^2}{\lambda^2}\sin^2\theta_c \tag{3.3}$$

とした。

そして、波長λとpath length xを積分し、

$$N = 2\pi\alpha z^2\sin^2\theta_c\int_{\lambda_1}^{\lambda_2}\frac{1}{\lambda^2}d\lambda\int_{x_1}^{x_2}dx \tag{3.4}$$

を、cherenkov放射により放出される光子数とした。

本研究では光電子増倍管(PMT)がカバーする波長領域を考慮し、積分する波長領域は$\lambda = 300\mathrm{nm} \sim 650\mathrm{nm}$とした。

放出位置

荷電粒子がAerogel Cherenkov Counterに入射した位置と出た位置を直線で結び、これをAerogel Cherenkov Counter中での粒子の飛跡とみなした。そして、放出される光子の位置は、この直線上で一様になるように決めた。

放出方向

θ方向に関しては、式(3.1)より、入射粒子のβと入射した物質の屈折率nにより決定できるが、ϕ方向に関しては特に依存性はない。そこで、ϕ方向に関しては、ϕ=0$\sim$2πで一様となるように決定した。

光子の波長

式 (3.3) より、放射光子数 N と波長 λ の間には

$$\frac{dN}{d\lambda} \propto \frac{1}{\lambda^2} \tag{3.5}$$

という比例関係があることがわかる。つまり、放射光のスペクトルは短波長領域に集中している。そこでシミュレーションにおいて放射される光子の波長は、Fig.3.3 にあるように長波長の光子よりも短波長の光子が多く発生するようにした。

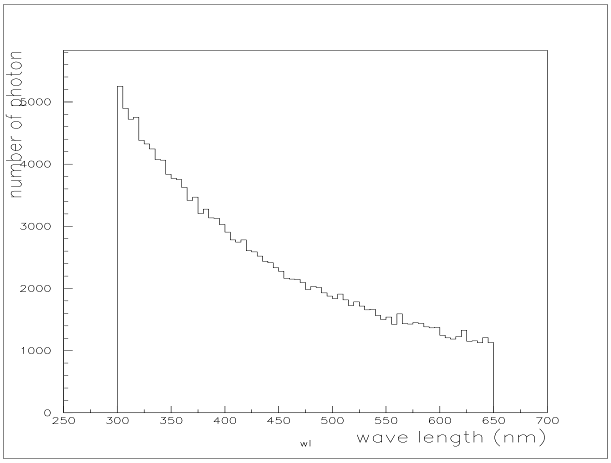

Fig. 3.3: シミュレーションにおいて放射される光子数の波長依存性

3.2.2 Aerogel Cherenkov Counter の光学系

Aerogel 中での光の伝搬は GEANT では計算されない。そのため、本研究では Aerogel 中での光の伝搬を再現するようなプログラムを作成した。

Aerogel の吸収長と散乱長

Aerogel の散乱長 λ_{scat} と吸収長 λ_{absp} には、以下のような光の波長依存性があると考えられている。

$$\lambda_{scat} \propto \lambda^4 \tag{3.6}$$

$$\lambda_{absp} \propto \lambda^2 \tag{3.7}$$

本研究では、光の波長 λ が 415nm のときの散乱長 λ_{scat} を 3cm、吸収長 λ_{absp} を 20cm という実験的に求められた値を用いた。そして、これを満たすように式 (3.6) と (3.7) の比例定数を決めた。

Fig.3.4 に、シミュレーションにおける Aerogel の散乱長 λ_{scat} と吸収長 λ_{absp} の光の波長依存性依存性を示す。

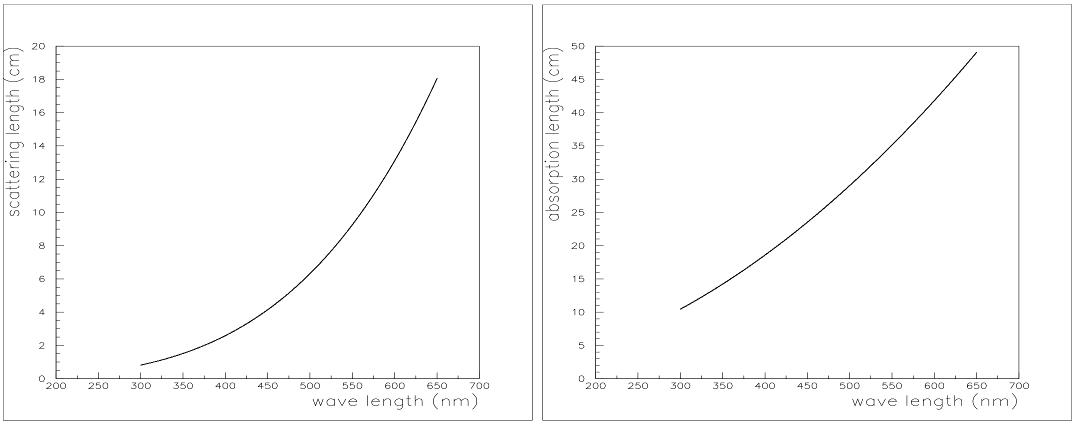

Fig. 3.4: シミュレーションのおける Aerogel の散乱長 λ_{scat} （左）と吸収長 λ_{absp} （右）の光の波長依存性

Aerogel 中での光子の散乱と境界での反射

散乱後の光子の進む方向は、空間的な偏りが起こらないようにその散乱方向が一様となるように決めた。また、境界での吸収はなく反射は乱反射とし、散乱と同様に空間的な偏りが起こらないようにその反射方向を一様に決めた。このとき、境界での反射における吸収はないものとしている。

以上のようなことを考慮し、Aerogel 中での光の伝搬を計算した結果を Fig.3.5 に示す。

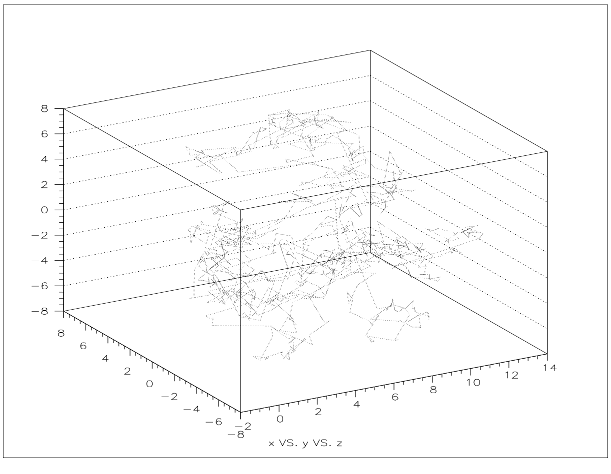

Fig. 3.5: Aerogel 中の光の伝搬の様子

3.2.3　構造

本研究で入力したAerogel Cherenkov Counterの構造をFig.3.6に示す。これは、昨年製作されたAerogel Cherenkov Counterのプロトタイプをに再現したものである。

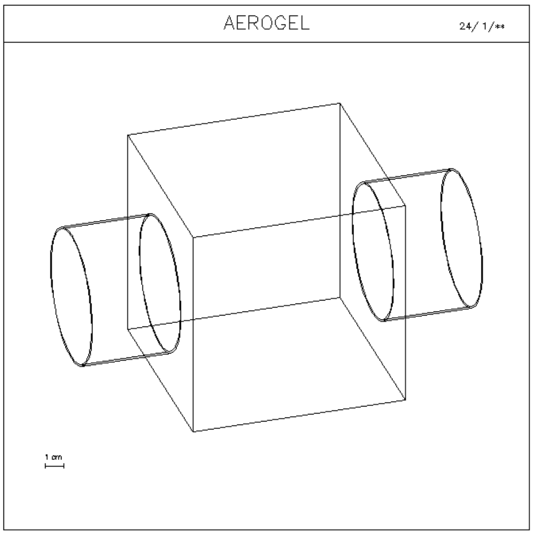

Fig. 3.6: Aerogel Cherenkov Counter

本研究ではAerogelの屈折率はn=1.015とした。そして、その大きさは12cm × 12cm × 12cmとし、その側面には3インチの光電子増倍管とμメタルシールドがそれぞれついた構造をしている。光電子増倍管はその構造を正確に入力することが困難であるため、その代わりとして新たな物質を定義し、その円筒状の物として光電子増倍管を入力した。また、μメタルシールドは鉄の円筒状として入力した。光電子増倍管は、昨年KEK-π2ビームラインでの実験で使われたものを参考にして、量子効率は波長が300nm ~ 500nmのときに25%、500nm ~ 650nmのときに10 %となるように決めた。

3.2.4　フローチャート

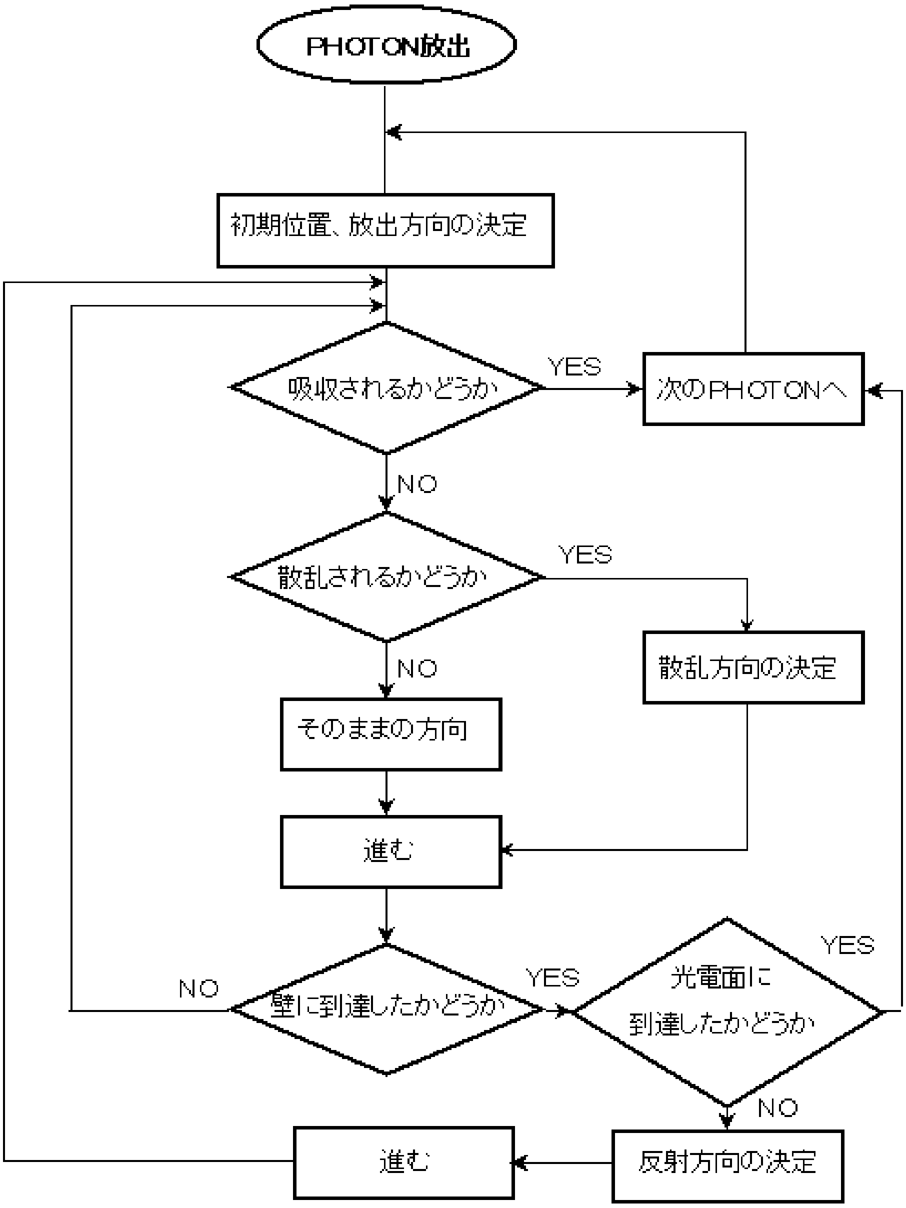

Fig. 3.7: Aerogel Cherenkov Counter 中での光子を追う計算方法

第4章　シミュレーションの結果

4.1　Aerogel Cherenkov Counter の評価

ここでは一個の Aerogel Cherenkov Counter に着目し、その性能評価をおこなった。

4.1.1　入射ビームの位置依存性

シミュレーション中で運動量 10GeV の π^+ 中間子を入射し、再現された Aerogel Cherenkov Counter の入射ビーム位置と観測される Photoelectron 数の位置依存性を調べ、昨年 KEK-π2 ビームラインでおこなわれた実験データと比較をした。シミュレーションでの入射ビームの位置と観測される Photoelectron 数の依存性を以下の Fig.4.1 に、実験での入射ビームの位置と観測される Photoelectron 数の依存性を 4.2 に示す。

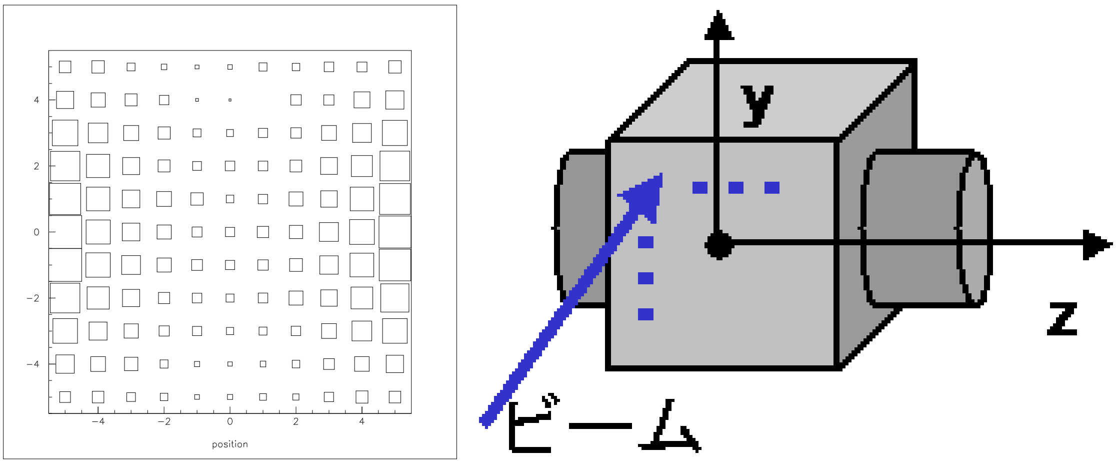

Fig. 4.1: シミュレーションにおける入射ビームの位置依存性（四角の大きさが大きいほど観測される Photoelectron 数が多い。）

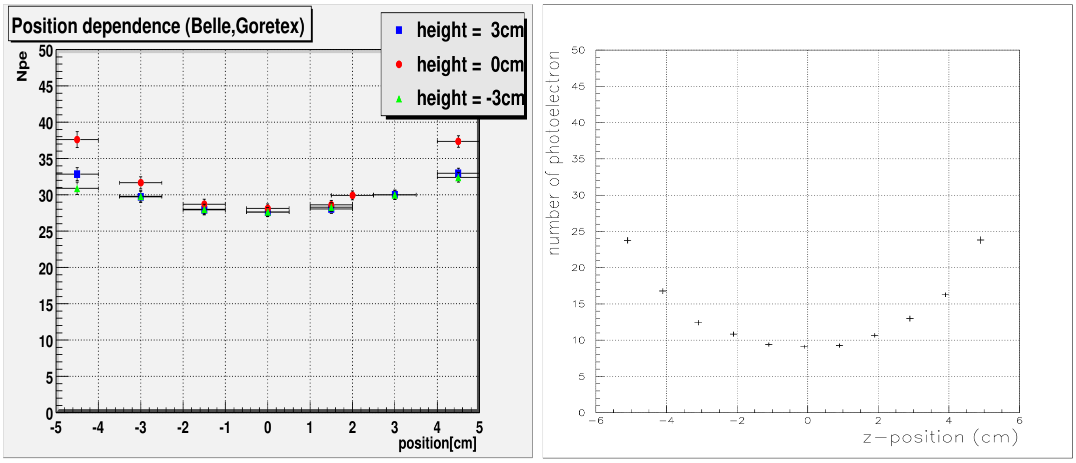

Fig. 4.2: 実験による z 方向依存性 (左) とシミュレーションによる z 方向依存性 (右)

シミュレーションの結果、入射ビームが PMT に近いほど観測される Photo electron 数が多くなっていることがわかる。この依存性は実験データにもみられるものである。

4.1.2　入射粒子の種類と運動量による依存性

π 中間子、K 中間子、陽子をそれぞれ 0.1GeV/c ∼ 10.9GeV/c まで運動量を変化させながら Aerogel Cherenkov Counter に入射し、入射粒子とその運動量による依存性を調べた (Fig.4.3 参照)。

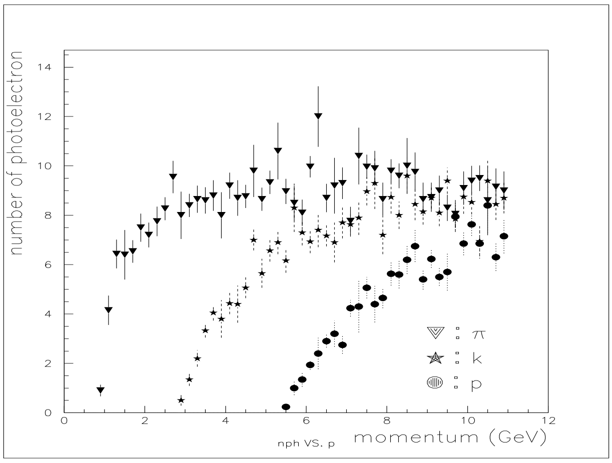

Fig. 4.3: 入射粒子の運動量と Photoelectron 数の依存性

この結果、入射した粒子の種類と運動量の違いにより Aerogel Cherenkov Counter の responce の違いがはっきりと見て取れる。

4.2 PHEIX 実験における Aerogel Cherenkov Counter

ここでは、以上のような入射した粒子の位置、種類、運動量に対する依存性を持った Aerogel Cherenkov Counter を PHENIX 検出器に組み込みシミュレーションをおこなった結果を評価する。

以下、シミュレーションにより得られた結果として、イベントあたり縦 17 個、横 17 個の Aerogel Cherenkov Counter において荷電粒子が Hit した Counter の場所と Cherenkov 光が観測された（以下、これを” 光った” と呼ぶ）Counter の場所を示す。

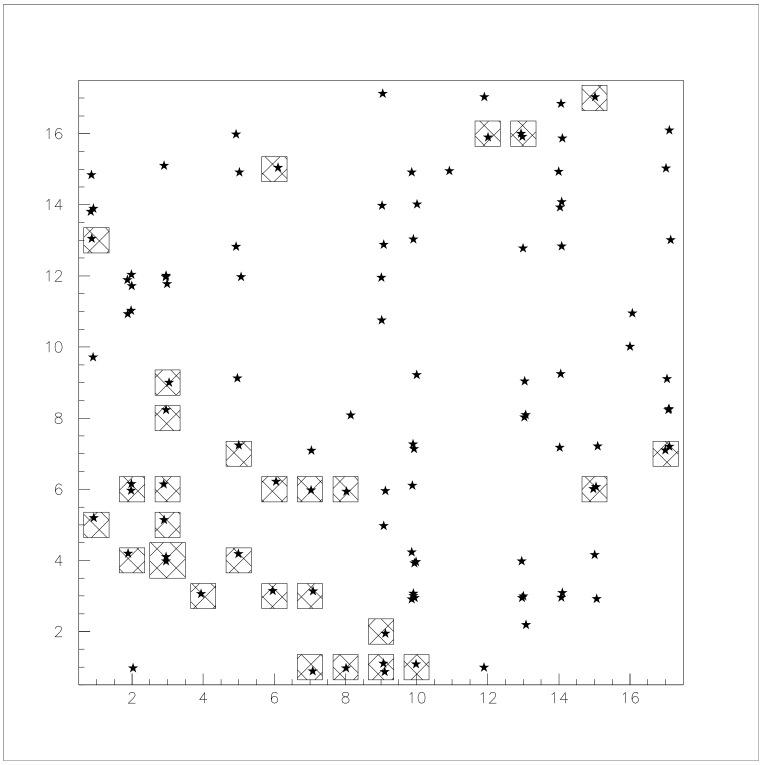

Fig. 4.4: 1 イベントにおける荷電粒子の Hit した Counter の場所と光った Counter の場所（★ :荷電粒子が Hit した Counter 場所、 :光った Counter の場所）:縦 17 個、横 17 個の Aerogel Cherenkov Counter の場所に縦軸、横軸は相当

4.2.1 占有率

イベントあたり、荷電粒子の Hit があった Aerogel Cherenkov Counter の数と、Aerogel Cherenkov Counter の数を Fig.4.5 に示す。

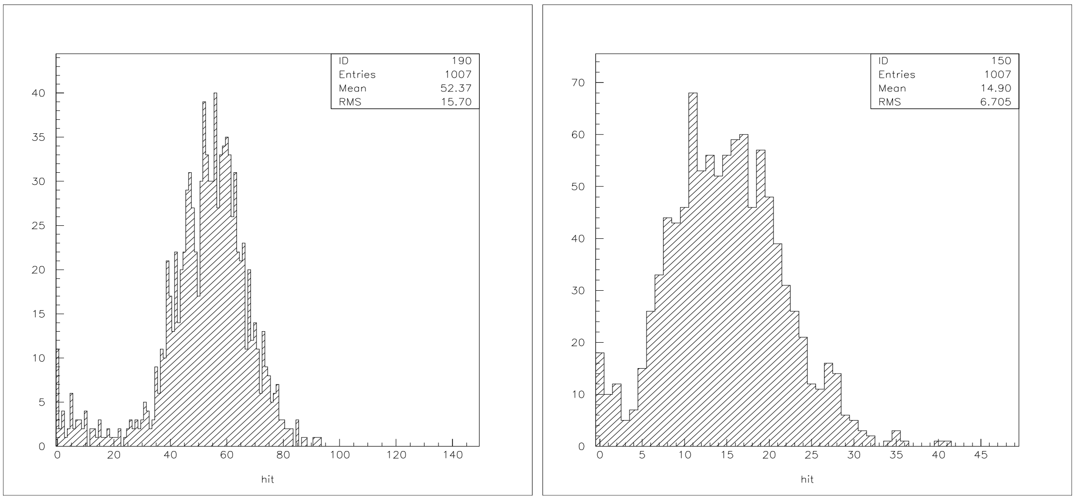

Fig. 4.5: イベントごとに荷電粒子の Hit した Counter の数 (左) と光った counter の数 (右)

この結果、イベントあたり光った Aerogel Cherenkov Counter の数は平均 15 個であることがわかる。したがって、$\theta = 90^\circ$ 付近、ビーム軸から 450cm のところに縦 17 個、横 17 個の計 289 個の Aerogel Cherenkov Counter を敷き詰めたときの占有率は、約 5% である。

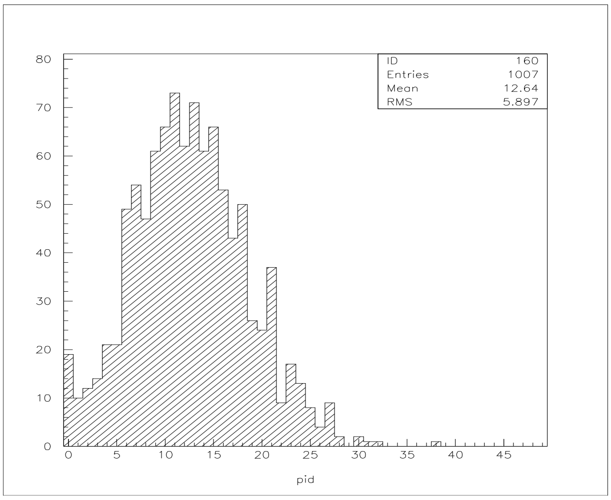

Fig. 4.6: event あたり、single hit の counter の数

Cherenkov 光を放出するような運動量の荷電粒子が一個だけ通過した（以下、single hit と呼ぶ）Aerogel Cherenkov Counter の数を求めた結果を Fig.4.6 に示す。

Aerogel Cherenkov Counter 個々の大きさには、Cherenkov 光を放出するような運動量の荷電粒子が同時に入射しない程度の大きさが求められている。占有率が約 5% ということは、Cherenkov 光を放出するような運動量の荷電粒子が同じ Counter に同時に複数入射する確率が約 5% ということでもある。

4.2.2 バックグランド

核子あたり 100GeV の金・金衝突実験において、Aerogel Cherenkov Counter は、様々な検出器からバックグランドを受けていることがわかる（Fig.4.7 参照）。

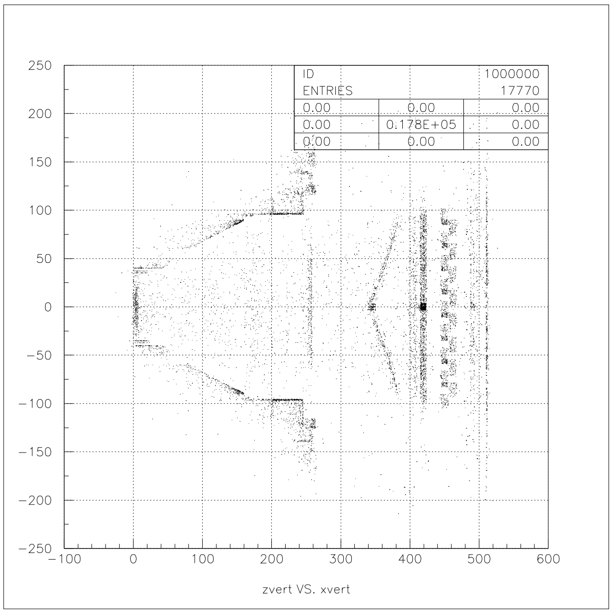

Fig. 4.7: background source

バックグランドは原子核・原子核衝突により生成された粒子が崩壊もしくは検出器等と相互作用した結果、生成した 2 次粒子であることが Fig. 4.7 から見て取れる。

Aerogel Cherenkov Counter のバックグランド総数における各粒子の割合は以下の通りである。

粒子	割合 (%)
e^+、e^-	96.3
μ^+、μ^-	2.0
π^+、π^-	1.1
p、$\overline{p}$	0.6

Aerogel Cherenkov Counter の受けるバックグランドのほとんどが e$^+$、e$^-$ であることがわかる。また Fig.4.8 に示す通り、バックグランドのほとんどは低エネルギーの粒子であった。

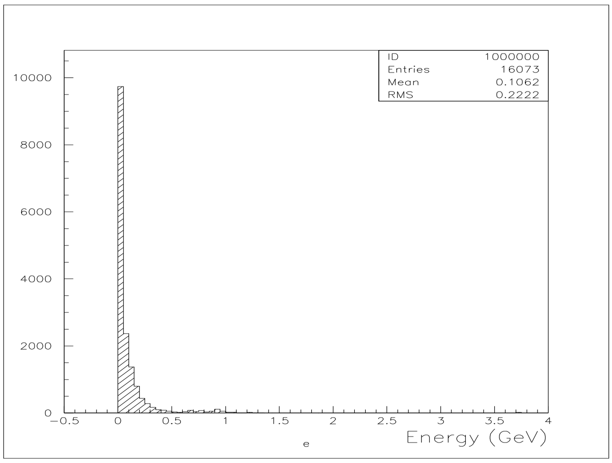

Fig. 4.8: bachground （バックグランド粒子のエネルギー）

以下では、Drift Chmber 等による粒子の Tracking や RICH による電子の識別により Aerogel Cherenkov Counter のバックグランドがどの程度まで排除可能であるかを評価した。

Drift Chmber に Hit していない粒子については、原子核・原子核衝突により生成された粒子が崩壊、もしくは検出器等と相互作用を起して生成した二次粒子（バックグランド）と考えられる。そして、これらは Drift Chmber に Hit していないので Tracking によって排除することが可能である。しかし Drift Chmber の特性上、それより以前に生成された粒子については、それが原子核・原子核衝突により生成された粒子であるのか、バックグランドであるのかを Tracking により判別することは難しいといえる。バックグランドの総数に対して Tracking によって判別することが難しい粒子の割合は約 14% であった。

Fig.4.9 に Drift Chmber よりまえ（r<202cm）に生成されたバックグランドの生成点を示す。これらの各粒子の割合は以下の通りである。

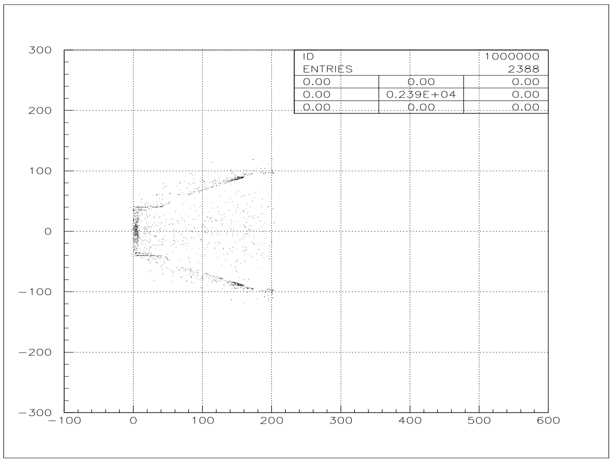

Fig. 4.9: bachground source（r<202cm）

粒子	割合 (%)
e^{+}、e^{-}	88.6
μ^{+}、μ^{-}	6.1
π^{+}、π^{-}	5.0
p、$\overline{p}$	0.3

ここで、 e^{+}、e^{-} は、RICH による電子識別で排除することが可能である。

以上より、Drift Chmber 等による粒子の Tracking や RICH による電子の識別によってその 9 割以上が排除可能である。

4.2.3 バックグランドの粒子識別への影響

以下では、バックグランド等の影響により、同じ Counter に複数の荷電粒子が同時に入射することで Aerogel Cherenkov Counter のみの情報では粒子識別が出来なくなることがある。ここでは、その可能性について評価をした。

π 中間子

π 中間子が入射した Aerogel Cherenkov Counter おいて、π 中間子のみが通過する確率を Fig.4.10 に示す。

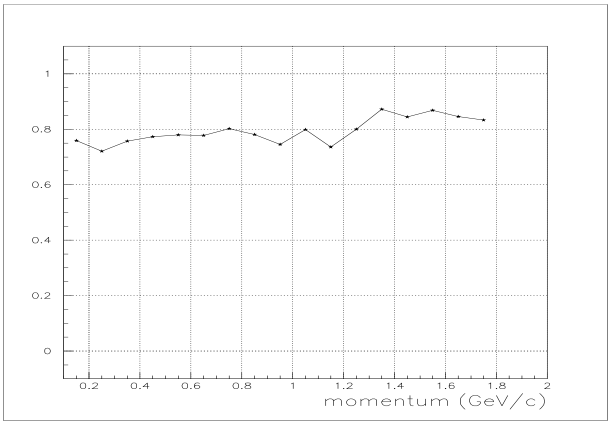

Fig. 4.10: π 中間子が入射した Counter において π 中間子のみが入射した確率

この結果、Aerogel Cherenkov Counter に入射した π 中間子のうち約 80% 程度が容易に識別可能である。しかし、残りの約 20% がすべて粒子識別が不可能になるのではなく、入射した荷電粒子すべてが Drift Chamber より前に生成された粒子であったら、Drift Chamber、RICH、ToF、Aerogel Cherenkov Counter からの情報を組み合わせることで識別が可能になる。

そこで、Aerogel Cherenkov Counter に入射した荷電粒子のすべてが Drifet Chamber 以前に生成された粒子である確率を Fig.4.11 に示す。

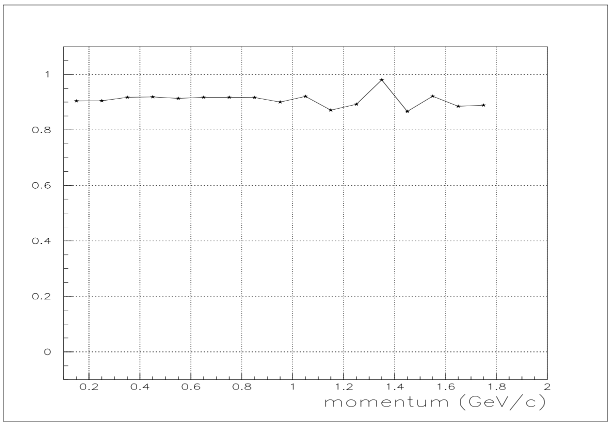

Fig. 4.11: PHENIX 検出器における Aerogel Cherenkov Counter による π 中間子識別の確率

したがって、PHENIX 検出器における Aerogel Cherenkov Counter での π 中間子の識別は約 90% の確率で可能であるといえる。

K 中間子

K 中間子が入射した Aerogel Cherenkov Counter おいて、K 中間子のみが入射する確率を Fig. 4.12 に示す。

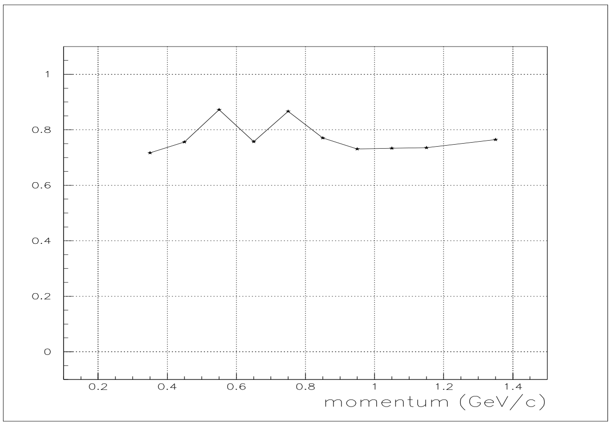

Fig. 4.12: K 中間子が入射した Counter において、それが K 中間子のみである確率

この結果、Aerogel Cherenkov Counter に入射した K 中間子の約 75% が容易に識別可能である。しかし、残りの約 25% が識別不可能というわけでなく、入射した荷電粒子すべてが Drift Chamber より前に生成されたものであったら、Drift Chamber、RICH、ToF、Aerogel Cherenkov Counter からの情報を組み合わせることで識別が可能になる。Aerogel Cherenkov Counter に入射した荷電粒子のすべてが Drifet Chamber 以前に生成された粒子である確率を Fig.4.13 に示す。

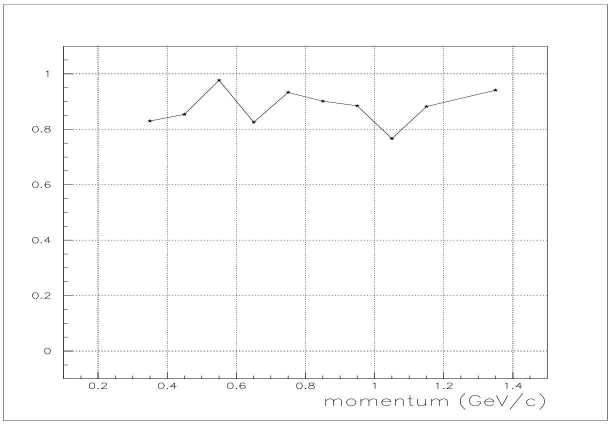

Fig. 4.13: PHENIX 検出器における Aerogel Cherenkov Counter による K 中間子識別の確率

したがって、PHENIX 検出器における Aerogel Cherenkov Counter による K 中間子の識別は約 90% の確率で可能であるといえる。

陽子

陽子が入射したAerogel Cherenkov Counterおいて、陽子のみが入射する確率をFig. 4.14に示す。

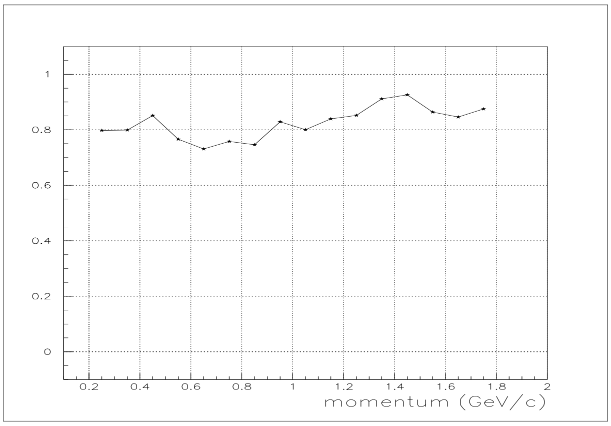

Fig. 4.14: 陽子が入射したCounterにおいて、それが陽子のみである確率

この結果、Aerogel Cherenkov Counterに入射した陽子の約85%が容易に識別可能である。しかし、残りの約20%が識別不可能というわけでなく、入射した荷電粒子すべてがDrift Chamberより前に生成されたものであったら、Drift Chamber、RICH、ToF、Aerogel Cherenkov Counterからの情報を組み合わせることで識別が可能になる

Aerogel Cherenkov Counterに入射した荷電粒子のすべてがDrifet Chamber以前に生成された粒子である確率をFig.4.15に示す。

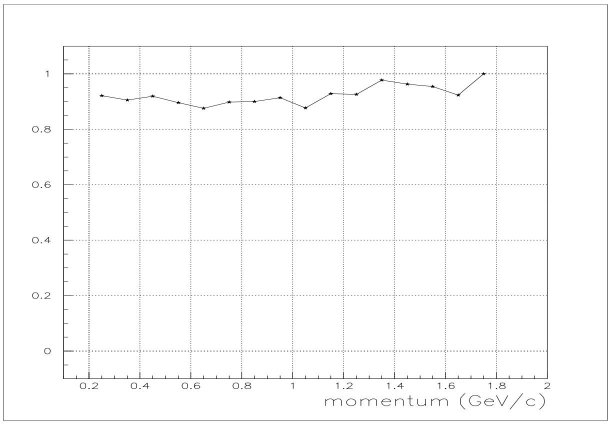

Fig. 4.15: PHENIX検出器におけるAerogel Cherenkov Counterによる陽子識別の確率

したがって、PHENIX検出器におけるAerogel Cherenkov Counterによる陽子の識別は約90%の確率で可能であるといえる。

4.2.4 検出効率

ここでは、Aerogel Cherenkov Counterのπ中間子、K中間子、陽子の検出効率について述べる。

π中間子

π中間子の各運動量に対するAerogel Cherenkov Counterの検出効率をFig.4.16に示す。

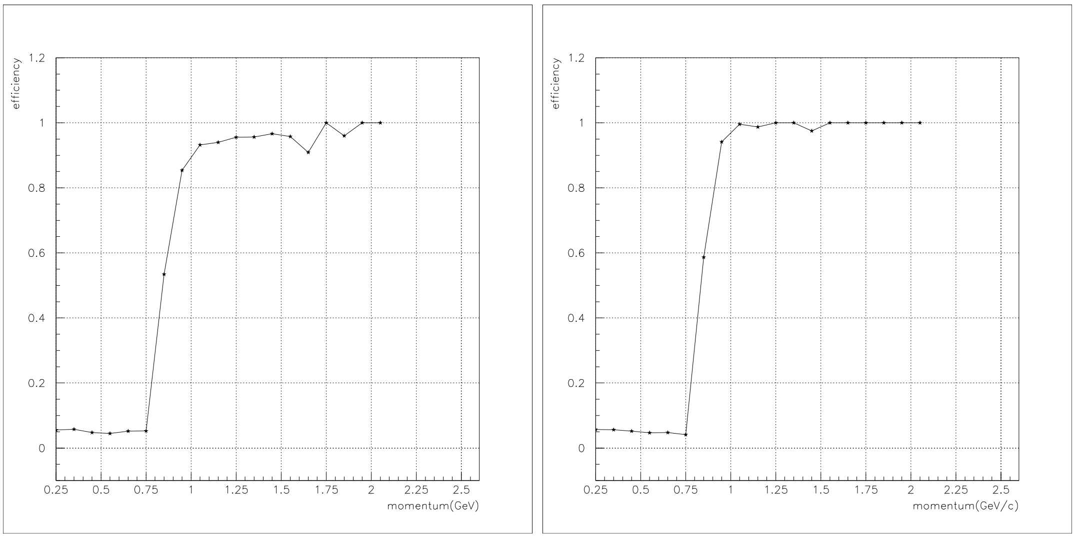

Fig. 4.16: π中間子の検出効率（左：no cut）とπ中間子の検出効率（右：path length>10cm）

Aerogel中での粒子の通過距離（path length）が十分長いものだけを選ぶと、π中間子の検出効率はCherenkov光を出すような運動量領域において、ほぼ100%を達成知ることができる（Fig.4.16（右）参照）。

Cherenkov光を出さないような運動量領域のπ中間子検出されることがあるが、これはπ中間子と同時にCherenkov光を出すような運動量領域の粒子がAerogel Cherenkov Counterに入射したためである。そのような確率は約5%程度であることがわかる。つまり、Cherenkov光を出さないような運動量のπ中間子について、Aerogel Cherenkov Counterは5%以下の確率で識別を間違えるということである

K 中間子

K 中間子の各運動量に対する Aerogel Cherenkov Counter の検出効率を Fig.4.17 に示す。

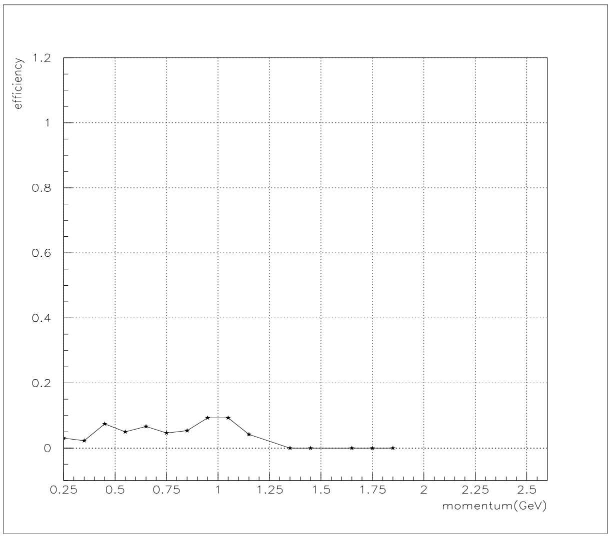

Fig. 4.17: K 中間子の検出効率 (no cut)

Cherenkov 光を出さないような運動量領域の K 中間子が検出されることがあるが、これは K 中間子と同時に Cherenkov 光を出すような運動量領域の粒子が Aerogel Cherenkov Counter に入射したためである。そのような確率は約 10% 以下であることがわかる。つまり、Cherenkov 光を出さないような運動量の K 中間子について、Aerogel Cherenkov Counter は 10% 以下の確率で識別を間違えるということである。

また、π 中間子と同様に Aerogel 中での K 中間子の path length が十分に長いという条件を加えることで、ほぼ 100% の検出効率を達成できる。

陽子

陽子の各運動量に対する Aerogel Cherenkov Counter の検出効率を Fig.4.18 に示す。

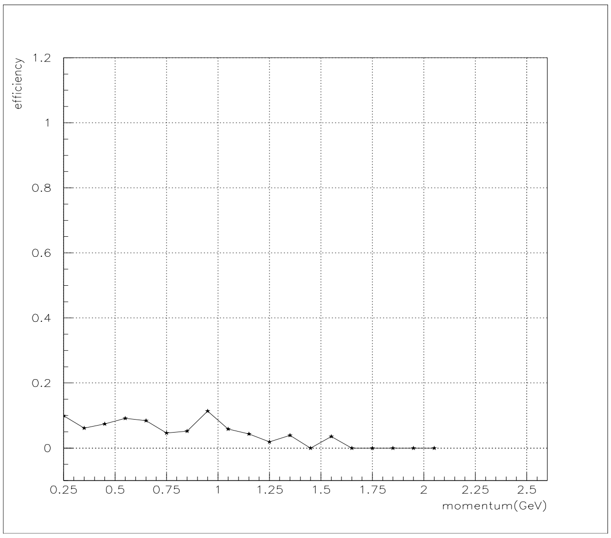

Fig. 4.18: 陽子の検出効率 （no cut ）

Cherenkov 光を出さないような運動量領域の陽子が検出されることがあるが、これは陽子と同時に Cherenkov 光を出すような運動量領域の粒子が Aerogel Cherenkov Counter に入射したためである。そのような確率は約 10% 以下であることがわかる。つまり、Cherenkov 光を出さないような運動量の陽子について、Aerogel Cherenkov Counter は 10% 以下の確率で識別を間違えるということである。

また、π 中間子と同様に Aerogel 中での陽子の path length が十分に長いという条件を加えることで、ほぼ 100% の検出効率を達成できる。

第5章　考察

Aerogel Cherenkov Counter 個々には、同じ Counter に複数の荷電粒子が同時に入射しないほど小さく、そして十分な光量が得られる程度の大きさが求められる。本研究では、Aerogel Cherenkov Counter の大きさを 12cm × 12cm × 12cm とし、計算をおこなった。
計算結果を以下にまとめる。

- Aerogel Cherenkov Counter のバックグランドも含めた占有率は約 5% であった。
- バックグランドのほとんどが低エネルギーの粒子であり、そのほとんどが e^+、e^- であった。
- π 中間子が入射した Aerogel Cherenkov Counter のなかで約 80% が π 中間子のみの入射であった。また、約 90% が通過した荷電粒子がすべて Drift Chamber より前で生成されたものであった。
- K 中間子が入射した Aerogel Cherenkov Counter のなかで約 75% が K 中間子のみの入射であった。また、約 90% が通過した荷電粒子がすべて Drift Chamber より前で生成されたものであった。
- 陽子が入射した Aerogel Cherenkov Counter のなかで約 85% が陽子のみの入射であった。また、約 95% が通過した荷電粒子がすべて Drift Chamber より前で生成されたものであった。
- 検出効率は Aerogel 中の荷電粒子の path length が十分長ければ、ほぼ 100% を達成できる。

計算の結果より、占有率が約 5% であるので Aerogel Cherenkov Counter の大きさとして 12cm × 12cm × 12cm は十分な大きさだといえる。

また、PHENIX 実験においてこの大きさの Aerogel Cherenkov Counter では π 中間子、K 中間子、陽子について誤った結果を出力してしまう確率が、それぞれ 10% 程度になることがわかった。このような事がおこる原因として考えられるのは

1. 複数の荷電粒子が同時に通過する確率が高い
2. 個々の Aerogel Cherenkov Counter が大きい
3. 配置の仕方

である。1 は 2 と 3 から決定できる要素ではあるが基本的にはバックグランドの影響であるため、これは PHENIX 検出器全体に関係する事である。

第6章　まとめ

本研究では、シミュレーションコードGEANTを用いて、Aerogel Cherenkov Counterを組み込んだPHENIX検出器の動作環境を計算機上に再現した。そして、核子あたり100GeVの金・金衝突実験におけるAerogel Cherenkov Counterの動作環境の評価をおこなった。

PHENIX検出器における核子あたり100GeVの金・金衝突実験ではAerogel Cherenkov Counter（12cm × 12cm × 12cm）によるハドロン（π中間子、K中間子、陽子）識別において、バックグランド等の影響によって約10%程度の確率で粒子識別が困難になるという結果を得た。

今後、Aerogel Cherenkov Counterの大きさ・配置の仕方を変えて検討していく必要がある。そして、Aerogel Cherenkov Counterにとって最適になるような運動量もしくはPhotoelectron数の閾値を見積もるということも必要となってくる。

謝辞

本研究を行うにあたり大変興味深いテーマを提案して頂くとともに、御指導・御助言を頂いた三明康郎教授、江角晋一講師には深く感謝致します。

特に江角晋一講師には本研究を進めるにあたり様々な相談に乗ってもらい、非常に有用な助言を数多く頂きました。そして、加藤純雄技官には、計算機に関する多くの面で支えて頂きました。深く感謝するとともにお礼を申し上げます。

研究室の先輩である清道明男氏、小野雅也氏、鶴岡裕士氏、黒木善昭氏、坂井真吾氏、団村絢子氏、益井宙氏には本研究を進めるうえで多くの助言と研究室内外の様々な面でその支えとなって頂きました。。特に、清道明男氏、小野雅也氏、益井宙氏には本研究の目的であるシミュレーションについて多くの助言をして頂き、大変参考になりました。そして、黒木善昭氏には計算機に関することやその他多くの助言をして頂きました。また、同じ4年生の大木俊和氏、神長誉人氏、金野正裕氏には様々な面で数多く支えて頂きました。

本研究を行うにあたりその支えとなってくれた数多くの人達に深く感謝するとともに、この場を借りてそのお礼を申し上げます。

参考文献

[1] A Primer Manual for the PHENIX Simulation Code PISA
Brookhaven National Laboratory

[2] CERNLIB -CERN Program Library Short Writeups
CERN Program Library

[3] GEANT -Detector Description and Simulation Tool
CERN Program Library

[4] PAW -Physics Analysis Workstation
CERN Program Library

[5] HBOOK -Statistical Analysis and Histogramming
CERN Program Library

[6] PHYSICAL REVIEW D - Particle and Fields
The American Society

[7] 八木 浩輔 ：原子核物理学
基礎物理科学シリーズ4 (朝倉書店、1995)

[8] 神吉 健 ：クォーク · グルーオン · プラズマ
パリティ物理学コース (丸善株式会社、1992)

[9] Glenn F. Knoll ：放射線計測ハンドブック [第2版]
木村 逸郎、阪井 英次 訳 (日刊工業新聞社、1990)

付 録A　program

A.1　プログラムリスト

本研究で作成したプログラムの主なものを載せる。

aerogel.f :**GEANT** による **Aerogel Cherenkov Counter** の **geometry** を定義する **program**

aer_gustep.f :**Cherenkov** 放射により放出された光子数計算と **GEANT** による **Aerogel Cherenkov Counter** に関する情報を計算する **program**

aer_photon.f :**Cherenkov** 放射を再現し、**Aerogel** 中の光の伝搬を再現する **program**

A.1.1 aerogel.f

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     ======================                                         c
c     File Name :  aerogel.f                                         c
c     ======================                                         c
c                                                                    c
c                                                                    c
c     Description :                                                  c
c     =============                                                  c
c                                                                    c
c     This subroutine defines the geometry of PHENIX - aerogel.      c
c                                                                    c
c     Author:- Satoshi Takagi                                        c
c                                                                    c
c     Creation Date: 10-Jan-2002                                     c
c     ==========================                                     c
c                                                                    c
c     Revisions:-                                                    c
c     =========                                                      c
c     Date      Name            Description                          c
c     ----      ----            -----------                          c
c    1/10/21    S.takagi        Therd Version                        c
c                                                                    c
c                                                                    c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c=====================================
      subroutine AEROGEL(FULL,NH)
c=====================================
      implicit none
      include 'gcflag.inc'
      include 'gclist.inc'
      include 'gconst.inc'
      include 'gugeom.inc'
      include 'guphnx.inc'
      include 'sublink.inc'
      include 'fpflink.inc'
      include 'fstore.inc'
c
      integer  aer_med,aer_vol,gla_med,gla_vol,
```

```
     &         sd_vol,vc_vol
c
      real  aer_dim(3) /6.0,6.0,6.0/
      real  box_dim(3) /25,110,110/
      real  ppp_dim(3) /3.7,3.81,2.5/
      real  sd_dim(3) /3.95,4.0,2.5/
      real    pl1(3)   /1.0,105.0,105.0/
      real    pl2(3)   /1.0,105.0,105.0/
      real    aa(2)    /28.09,16.0/
      real    za(2)    /14.,8.0/
      real    wa(2)    /1.0,2.0/
      real    da(1)    /0.030/
      real    aag(4)    /16.0,28.086,10.81,22.99/
      real    zzg(4)    /8.0,14.0,5.0,11.0/
      real    wwg(4)    /2.01,0.8,0.24,0.1/
      real    ddg(1)    /2.23/
      integer idtyp    /999/
      data  aer_med    / 5000  /  ! AEROGEL
      data  aer_vol    / 6000/
      data gla_med   / 7003/
      data gla_vol   / 7001/
      data sd_vol      / 7002/
      data vc_vol      / 7004/
c
c ---> working parameters
c
      character*4  v_m_name
      character*4  v_i_name
      character*4  pm1_name
      character*4  pm2_name
      character*4  pl1_name
      character*4  pl2_name
      character*4  sd1_name
      character*4  sd2_name
      character*4  v_b_name
      real gpos(3)
      integer      ivolu
      integer      nr
      integer*4    nh           ! set before call in gugeom
```

```
      integer      npar
      integer      nmed
      integer      i1
      integer      j1
      integer      k
      integer      i2
      integer      j2
      integer      h
      integer      irot_aer
      character*20 chform
      character*4  full         ! set before call in gugeom
      character*50 par_file
      character*4  set_id
      character*8  name
      character*8  name1
c
      integer nmat,isvol,ifield,nwbuf
      real    fieldm,tmaxfd,dmaxms,deemax,epsil,stmin,ubuf(10)
c
c ---> define tracking media for AEROGEL
c
      call gmate

      isvol = 1      ! sensitive
      ifield = 1     ! no magnetic field
      fieldm = 5.0   ! max field
      tmaxfd = 45.0  ! maximum angle due to field (one step) in degrees
      dmaxms = 0.50  ! max disp. due to mulsct. in one step (cm)
      deemax = 0.05  ! max fractional energy loss in one step
      epsil = 0.01   ! tracking precision (cm)
      stmin = 0.1    ! min step due to e loss or mulsct. (cm)
      ubuf(1) = 0.   ! tracking stop switch
      nwbuf=0
c
c ---> Define new material (Aerogel,PMT glass...)
c
      call gsmixt(aer_med,'SiO2 $',aa,za,da,-2,wa)
      call gstmed(aer_vol,'AEROGEL $',aer_med,isvol,ifield,
     &     fieldm,tmaxfd,dmaxms,deemax,epsil,stm in,ubuf,nwbuf)
```

```
c
      call gsmixt(gla_med,'Borosilicate$',aag,zzg,ddg,-2,wwg)
      call gstmed(gla_vol,'PMT glass $',gla_med,-1,ifield,
     &     fieldm,tmaxfd,dmaxms,deemax,epsil,stm in,ubuf,nwbuf)
c
      call gstmed(sd_vol,'m_metal $',10,-1,ifield,
     &     fieldm,tmaxfd,dmaxms,deemax,epsil,stm in,ubuf,nwbuf)
c
      call gstmed(vc_vol,'vacuum $',16,-1,ifield,
     &     fieldm,tmaxfd,dmaxms,deemax,epsil,stm in,ubuf,nwbuf)

c
c
c---> Definition of rotation matrix
c
      irot_aer = irot+1
      irot = irot+10
      call GSROTM(irot_aer,90.,0.,90.,90.,0.,0.) ! no rotation
      call GSROTM(irot_aer+1,90.,90.,90.,180.,0.,0.) ! +90 deg around z-axis
      call GSROTM(irot_aer+2,90.,270.,135.,0.,45.,0.) ! -90z x -45x
      call GSROTM(irot_aer+3,90.,90.,90.,0.,180.,0.) ! +90z x +180x
      call GSROTM(irot_aer+4,0.,0.,90.,0.,90.,-90.) ! x->z y->x z->-y
      call GSROTM(irot_aer+5,0.,0.,90.,0.,90.,90.) ! x->z y->x z->y
      call GSROTM(irot_aer+6,90.,180.,90.,270.,0.,0.) ! +180z
      call GSROTM(irot_aer+7,90.,0.,180.,0.,90.,90.) ! +90x for 90deg PMT
      call GSROTM(irot_aer+8,90.,180.,90.,90.,180.,0.) ! +180y
      call GSROTM(irot_aer+9,90.,0.,90.,90.,180.,0.) ! z -> -z
c
c---> Create PHENIX-AEROGEL pseudo volums
c---> Positioning media into mother volumes
c
      v_m_name  =  'WEST'
c

      npar = 3
c
      v_b_name  =  'ARGL'
      gpos(1)   =  456.0
      gpos(2)   =  0.0
```

```
      gpos(3)   =  0.0
      call GSVOLU(v_b_name,'BOX ',16,box_dim,npar,ivolu)
      call gsatt(v_b_name,'SEEN',0)
      call GSPOS(v_b_name,nr,v_m_name,gpos(1),
     &           gpos(2),gpos(3),irot_aer,'ONLY')

      pl1_name  =  'PLT1'
      gpos(1)   =  -20.0
      gpos(2)   =  0.0
      gpos(3)   =  0.0
      call GSVOLU(pl1_name,'BOX ',vc_vol,pl1,npar,ivolu)
      call GSATT(pl1_name,'SEEN',0)
      call GSPOS(pl1_name,nr,v_b_name,gpos(1),
     &           gpos(2),gpos(3),irot_aer,'ONLY')
c
      pl2_name  =  'PLT2'
      gpos(1)   =  20.0
      gpos(2)   =  0.0
      gpos(3)   =  0.0
      call GSVOLU(pl2_name,'BOX ',vc_vol,pl2,npar,ivolu)
      call GSATT(pl2_name,'SEEN',0)
      call GSPOS(pl2_name,nr,v_b_name,gpos(1),
     &           gpos(2),gpos(3),irot_aer,'ONLY')


c
c ---> Aerogel Cherenkov Counter (geometry, position )
c
      k    = 0
      npar = 3
      do 20 j1=1,17
         do 10 i1=1,17
            gpos(1) = mod(i1+1,2)*12.0 -6.0
            gpos(2) = 96.0 - (j1-1)*12.0
            gpos(3) = (i1-1)*12.0 - 96.0
            k = k + 1
            write(name,*)1000+k
            v_i_name = 'A'//name(3:5)
c            v_i_name = name(2:5)
```

```
            call GSVOLU(v_i_name,'BOX ',aer_vol,aer_dim,npar,ivolu)
            call GSATT(v_i_name,'SEEN',1)
            call GSATT(v_i_name,'COLO',4)
            call GSPOS(v_i_name,nr,v_b_name,gpos(1),
     &                 gpos(2),gpos(3),irot_aer,'ONLY')
 10      continue
 20   continue
c
c ---> PMT (geometry,position)
c
      h    = 0
      do 30 j2=1,17
         do 40 i2=1,17
            gpos(1) = mod(i2+1,2)*12.0 -6.0
            gpos(2) = 96.0 - (j2-1)*12.0
            gpos(3) = (i2-1)*12.0 - 104.5
            h = h + 1
            write(name1,*)1000+h
            pm1_name = 'Q'//name1(3:5)
            call GSVOLU(pm1_name,'TUBE',gla_vol,ppp_dim,npar,ivolu)
            call GSATT(pm1_name,'SEEN',1)
            call GSATT(pm1_name,'COLO',4)
            call GSPOS(pm1_name,nr,v_b_name,gpos(1),
     &                 gpos(2),gpos(3),irot_aer,'ONLY')
c
            sd1_name = 'S'//name1(3:5)
            call GSVOLU(sd1_name,'TUBE',sd_vol,sd_dim,npar,ivolu)
            call GSATT(sd1_name,'SEEN',1)
            call GSATT(sd1_name,'COLO',4)
            call GSPOS(sd1_name,nr,v_b_name,gpos(1),
     &                 gpos(2),gpos(3),irot_aer,'ONLY')
c
            gpos(1) = mod(i2+1,2)*12.0 -6.0
            gpos(2) = 96.0 - (j2-1)*12.0
            gpos(3) = (i2-1)*12.0 - 87.5
            pm2_name = 'L'//name1(3:5)
            call GSVOLU(pm2_name,'TUBE',gla_vol,ppp_dim,npar,ivolu)
            call GSATT(pm2_name,'SEEN',1)
            call GSATT(pm2_name,'COLO',4)
```

```
            call GSPOS(pm2_name,nr,v_b_name,gpos(1),
     &                gpos(2),gpos(3),irot_aer,'ONLY')
c
            sd2_name = 'D'//name1(3:5)
            call GSVOLU(sd2_name,'TUBE',sd_vol,sd_dim,npar,ivolu)
            call GSATT(sd2_name,'SEEN',1)
            call GSATT(sd2_name,'COLO',4)
            call GSPOS(sd2_name,nr,v_b_name,gpos(1),
     &                gpos(2),gpos(3),irot_aer,'ONLY')


 40      continue
 30    continue
c
c
      return
      end
```

A.1.2 aer_gustep.f

```
cccccccccccccccccccccccccccccccccccccccccccccccccc
c     ========================                   c
c     File Name : aer_gustep.f                   c
c     ========================                   c
c                                                c
c     Author:- Satoshi Takagi                    c
c                                                c
c     Creation Date: 10-Jan-2002                 c
c                                                c
c     Revisions:-                                c
c     =========                                  c
c     Date       Name           Description      c
c     ----       ----           -----------      c
c    1/10/02     S.Takagi       eighth Version   c
c                                                c
c                                                c
c                                                c
cccccccccccccccccccccccccccccccccccccccccccccccccc
c================================
      subroutine AER_GUSTEP
c================================
c---------------------------------------------------------------------
c  This routine called at the end of each tracking step
c---------------------------------------------------------------------
      include ’gcking.inc’
      include ’gckine.inc’
      include ’gcbank.inc’
      include ’gctrak.inc’
      include ’gcflag.inc’
      include ’gctmed.inc’
      include ’gcsets.inc’
      include ’gcvolu.inc’
      include ’gugeom.inc’
      include ’namlvl.inc’
      include ’guphnx.inc’
      include ’secubuf.inc’
      include ’gcunit.inc’
      include ’pisa_parts.inc’
```

```
      include ’gclist.inc’
      include ’subevt.inc’
      include ’gcjloc.inc’
      include ’gconst.inc’
c
      common/udata1/x0,y0,z0,x2,y2,z2,ax1,ay1,az1,Nckov
      common/udata2/len,beta,theta,phi,the1,phi1,Nph,wl
      common/udata3/pid,p,Nhit,Nph1,Nph2
      common/ndata4/Ckov,Ph1,Ph2,Phit
c
      common/udata5/p1_data,p2_data,ac_data,
     &              npt1,npt2,nacc
c
      real x,y,z,x0,y0,z0,x1,y1,z1,x2,y2,z2,beta,beta1
      real ax,ay,az,ax1,ay1,az1,p,e,st,len,ra,ah,teta
      real xps1,yps1,zps1,xdir,ydir,zdir,th,ph
      real xps2,yps2,zps2,leng,xgnt,ygnt,zgnt,tra1,tra2
      real xvert,yvert,zvert
      real nt_data(50)
      real p1_data(100,13),p2_data(100,10),ac_data(100,24)
      integer pid,i,j,VOLU,id,Nckov,Npe,Nph,track,track1
      integer Ckov,Pe,Ph1,Ph2,Nhit,Phit,flag,flag1,flag2
      integer flag3,flag4,npt1,npt2,nacc,trknb1,trknb2
c
      character*4 ch_VOLU
      character*4 ch_ARGL
      character*3 ch_name
      character*8 name
      equivalence (VOLU, ch_VOLU)
      VOLU = names(nlevel)
c
      if(ngkine.ne.0)then
         call gsking(0)
      end if
c
      if((inwvol.eq.1).and.(ch_VOLU.eq.’PLT1’).and.(charge.ne.0))then
         npt1             = npt1 + 1
         flag1            = 0
         p1_data(npt1,1)  = itra     ! track number
```

```
         p1_data(npt1,2)  = nsub_evt ! subevent number
         p1_data(npt1,3)  = ipart    ! particle number
         xps1             = vect(1)
         yps1             = vect(2)
         zps1             = vect(3)
         xdir             = vect(4)
         ydir             = vect(5)
         zdir             = vect(6)
         p1_data(npt1,8)  = xps1
         p1_data(npt1,9)  = yps1
         p1_data(npt1,10) = zps1
         p1_data(npt1,11) = xdir
         p1_data(npt1,12) = ydir
         p1_data(npt1,13) = zdir
c
         if(xdir.lt.0)then
            xdir        = -1.0*xdir
            ydir        = -1.0*ydir
            zdir        = -1.0*zdir
            flag1       = 1
         end if
c
         th          = atan2(zdir,ydir)
         ph          = atan2(ydir,xdir)
         p1_data(npt1,4) = 476.0
         p1_data(npt1,5) = (40.0*tan(ph)) + yps1
         p1_data(npt1,6) = ((40.0*tan(ph))*tan(th)) + zps1
         p1_data(npt1,7) = flag1
      end if
c
      if((inwvol.eq.1).and.(ch_VOLU.eq.'PLT2').and.(charge.ne.0))then
         npt2        = npt2 + 1
         p2_data(npt2,1) = itra       ! track number
         p2_data(npt2,2) = nsub_evt   ! subevent number
         p2_data(npt2,3) = ipart      ! particle number
         p2_data(npt2,4) = vect(1)
         p2_data(npt2,5) = vect(2)
         p2_data(npt2,6) = vect(3)
      end if
```

```
c
c
c ====> read out in Aerogel Cherenkov Counter
c
c
      if((numed.eq.6000).and.(vect(7).ne.0.0)
     +                              .and.(charge.ne.0))then
c
         if(inwvol.eq.1)then
            write(6,*)'vx=',vert(1),'vy=',vert(2),'vz=',vert(3)
            nacc         = nacc + 1
            ac_data(nacc,1)  = itra           ! track number
            trknb2           = itra
            ac_data(nacc,2)  = nsub_evt       ! subevent number
            ac_data(nacc,17) = ntru_evt       ! event number
            ac_data(nacc,3)  = ipart          ! particle number
            ac_data(nacc,4)  = vect(1)        ! x-position
            ac_data(nacc,5)  = vect(2)        ! y-position
            ac_data(nacc,6)  = vect(3)        ! z-position
            ac_data(nacc,7)  = vect(7)        ! momentum
            ac_data(nacc,8)  = getot          ! total energy
            beta             = vect(7)/getot  ! beta
            ac_data(nacc,9)  = beta
            ac_data(nacc,21) = vert(1)        ! x-coordinate of vertex
            ac_data(nacc,22) = vert(2)        ! y-coordinate of vertex
            ac_data(nacc,23) = vert(3)        ! z-coordinate of vertex
            ch_ARGL = ch_VOLU
            ch_name = ch_VOLU(2:4)
            read(ch_name,*)id
            j    = int((id-1)/17)+1
            i    = mod((id-1),17)+1
            ac_data(nacc,10) = i
            ac_data(nacc,11) = j
c
            write(6,*)'i=',i,':','j=',j,':',ipart
            write(6,*)ch_ARGL,':','TRACK=',itra
            x0   = ac_data(nacc,4) - 444.0 - mod(i+1,2)*12.0
            y0   = ac_data(nacc,5) + (j - 9.0)*12.0
            z0   = ac_data(nacc,6) - (i - 9.0)*12.0
```

```
            ac_data(nacc,18) = x0            ! hit position acc
            ac_data(nacc,19) = y0            ! hit position acc
            ac_data(nacc,20) = z0            ! hit position acc
         end if
c
         if((ch_ARGL.eq.ch_VOLU) )then
            trknb1 = itra
            if(trknb1.eq.trknb2)then
               beta1 = vect(7)/getot
               ra    = ((1.0/300.0)-(1.0/650.0))*(10**7.0)
               ah    = 1.0/137.0
               st    = abs(step)
               Pe    = 2.0*pi*ah*(charge**2.)*st*ra
     +               *(1.-1./((1.015*beta1)**2))
               if(1./(1.015*beta1).le.1.)then
                  Ckov = Ckov + Pe          ! number of Cherenkov Photon
               end if
            end if
         end if
c
         if(inwvol.eq.2)then
            ac_data(nacc,24) = itra         ! track number
            x1           = vect(1)          ! x-position
            y1           = vect(2)          ! y-position
            z1           = vect(3)          ! z-position
            x2           = x1 - 444.0 - mod(i+1,2)*12.0
            y2           = y1 + (j - 9.0)*12.0
            z2           = z1 - (i - 9.0)*12.0
            ax           = x1 - ac_data(nacc,4)
            ay           = y1 - ac_data(nacc,5)
            az           = z1 - ac_data(nacc,6)
            len          = sqrt(ax**2.+ay**2.+az**2.)
            ac_data(nacc,16) = len
            ax1          = ax/len
            ay1          = ay/len
            az1          = az/len
c
            call aer_photon
c
```

```
            ac_data(nacc,12) = Ckov   ! total number of photon
            ac_data(nacc,13) = Ph1    ! photoelectron
            ac_data(nacc,14) = Ph2    ! photoelectron
            ac_data(nacc,15) = Phit
c


c            write(6,*)'x0=',x0,':','y0=',y0,':','z0=',z0
c            write(6,*)'beta=',beta,':','n*beta=',beta*1.015
c            write(6,*)'beta1=',beta1,':','n*beta1=',beta1*1.015
c            write(6,*)'x2=',x2,':','y2=',y2,':','z2=',z2
            write(6,*)'Nckov=',Ckov
            write(6,*)'Nph=',Ph1+Ph2
c            write(3,*)Nckov,Nph,p,e,pid,x,y,z
c            open(3,file='aero.dat',status='old')
c            write(6,*)'len=',len
c            write(6,*)'ax1=',ax1,':','ay1=',ay1,':','az1=',az1
c            write(6,*)'rot=',rot
c            write(6,*)'the1=',the1,':','phi1=',phi1


            Ph1    = 0
            Ph2    = 0
            Ckov   = 0
            Phit   = 0

         endif
      end if
c
      return
      end
```

A.1.3 aer_photon.f

```
cccccccccccccccccccccccccccccccccccccccccccccccccc
c      ========================                  c
c      File Name : aer_photon.f                  c
c      ========================                  c
c                                                c
c      Author : Satoshi Takagi                   c
c                                                c
c      Creation Date : 10-Jan-2002               c
c                                                c
c      Revisions:-                               c
c      =========                                 c
c      Date       Name           Description     c
c      ----       ----           -----------     c
c     1/10/02     S.Takagi       fifth Version   c
c                                                c
cccccccccccccccccccccccccccccccccccccccccccccccccc
c================================
      subroutine AER_PHOTON
c================================
      include 'gcking.inc'
      include 'gckine.inc'
      include 'gcbank.inc'
      include 'gctrak.inc'
      include 'gcflag.inc'
      include 'gctmed.inc'
      include 'gcsets.inc'
      include 'gcvolu.inc'
      include 'gugeom.inc'
      include 'namlvl.inc'
      include 'guphnx.inc'
      include 'secubuf.inc'
      include 'gcunit.inc'
      include 'pisa_parts.inc'
      include 'gclist.inc'
      include 'subevt.inc'
      include 'gcjloc.inc'
      include 'gconst.inc'
c
```

```
      common/udata1/x0,y0,z0,x2,y2,z2,ax1,ay1,az1,Nckov
      common/udata2/len,beta,theta,phi,the1,phi1,Nph,wl
      common/udata3/pid,p,Nhit,Nph1,Nph2
      common/ndata4/Ckov,Ph1,Ph2,Phit
c
      real x0,y0,z0,x3,y3,z3,x4,y4,z4,x5,y5,z5
      real ax1,ay1,az1,dx,dy,dz,beta,len,rot,la,ls,d
      real pa,ps,l1,l2,ap,sp,pmt_dim,wl
      real the1,phi1,the2,phi2,the3,phi3,theta,phi
      real the4,phi4,ax2,ay2,az2,ax3,ay3,az3,lg,phax
      real thprim,phprim,ef,ta1,ta2,ta3,ta4,ta5,ta6
      real aer_data(3)
      real pho_data(4)
      real box_data(3)
      integer i1,Ckov,t,Phit,Ph1,Ph2
c
      t       = 0
      d       = 0.1             ! step length (cm)
      pmt_rad = 3.81            ! pmt radius (cm)
c
c ->  define the propability of absorpsion or scattering
c
      do 10 i1=1,Ckov
         t    = t + 1
c
c ===> define wavelength of cherenkov photon
c
         wl   = 1.0/(rand()*(7.0/3900.0)+(1.0/650.0)) !wave lenght (nm)
c
c ===> define  absorpsion length or scattering length
c
         la   = (20.0/(415**2.0))*(wl**2.0)
         ls   = (3.0/(415**4.0))*(wl**4.0)
c
c ===> define the propability of absorpsion or scattering
c
         pa   = exp(-1.0*d/la)
         ps   = exp(-1.0*d/ls)
c
```

```
         aer_data(1) = wl
         aer_data(2) = la
         aer_data(3) = ls
         call hfn(778,aer_data)
c
c ===>  define the position of radiation
c
         l1   = rand()*len
         z3   = x0 + ax1*l1
         x3   = y0 + ay1*l1
         y3   = z0 + az1*l1
c
c ===> define the dirction of radiation  and rotation of axis
c
         ta1 = 1.0/(1.015*beta)
         ta2 = ay1
         ta3 = az1
         ta4 = ax1
         ta5 = 2.0* pi*rand()
         ta6 = atan2(ta3,ta2)
c
         if(ta4.ne.0)then
            dx = ((sqrt(1.0 - (ta1*ta4)**2.0))*cos(ta5)) - (ta1*ta2)
            dy = ta4*(sqrt((1.0 - (ta1*ta4)**2.0))*sin(ta5)) - (ta1*ta3)
            dz = ta1*ta4
         else
            dx = (ta2*ta1)*cos(ta6)-sqrt(1.0-(ta2*ta1)**2.0)*sin(ta5)*sin(ta6)
            dy = (ta2*ta1)*sin(ta6)+sqrt(1.0-(ta2*ta1)**2.0)*sin(ta5)*cos(ta6)
            dz = sqrt(1.0-(ta2*ta1)**2.0)*cos(ta5)
         end if
c
         x4 = x3
         y4 = y3
         z4 = z3
c
c
c ===> calculate trajectory of cherenkov photon in aerogel
c
c
```

```
 20      ap = rand()
         if(ap.le.pa)then       ! photon is not absorbed
            sp = rand()
         else
            go to 10            ! photon is absorbed
         end if
c
         if(sp.ge.ps)then       ! photon is scattered
            the3 = acos(2.0*rand()-1.0)
            phi3 = (rand()*2.0)*pi
            dx   = sin(the3)*cos(phi3)
            dy   = sin(the3)*sin(phi3)
            dz   = cos(the3)
         end if
c
         pho_data(1) = t
         pho_data(2) = x4
         pho_data(3) = y4
         pho_data(4) = z4
c         call hfn(779,pho_data)
c
         x4 = x4 + d*dx
         y4 = y4 + d*dy
         z4 = z4 + d*dz
c
c ===> calculate the reflection of cherenkov photon at wall
c
         if(x4.le.-6.0)then
            y5   = (dy/dx)*(-6.0-x4)+y4
            z5   = (dz/dx)*(-6.0-x4)+z4
            x5   = -6.0
            box_data(1) = x5
            box_data(2) = y5
            box_data(3) = z5
c            call hfn(800,box_data)
c
            the4 = acos(2.0*rand()-1.0)
            phi4 = (rand()+1.5)*pi
            dx   = sin(the4)*cos(phi4)
```

```
            dy   = sin(the4)*sin(phi4)
            dz   = cos(the4)
            l2   = sqrt((x4-x5)**2.0+(y4-y5)**2.0+(z4-z5)**2.0)
            x4   = x5 + l2*dx
            y4   = y5 + l2*dy
            z4   = z5 + l2*dz
            go to 20
c
         else if(x4.ge.6.0)then
            y5   = (dy/dx)*(6.0-x4)+y4
            z5   = (dz/dx)*(6.0-x4)+z4
            x5   = 6.0
            box_data(1) = x5
            box_data(2) = y5
            box_data(3) = z5
c            call hfn(800,box_data)
c
            the4 = acos(2.0*rand()-1.0)
            phi4 = (rand()+0.5)*pi
            dx   = sin(the4)*cos(phi4)
            dy   = sin(the4)*sin(phi4)
            dz   = cos(the4)
            l2   = sqrt((x4-x5)**2.0+(y4-y5)**2.0+(z4-z5)**2.0)
            x4   = x5 + l2*dx
            y4   = y5 + l2*dy
            z4   = z5 + l2*dz
            go to 20
c
         else if(y4.le.-6.0)then
            x5   = (dx/dy)*(-y4-6.0)+x4
            z5   = (dz/dy)*(-y4-6.0)+z4
            y5   = -6.0
            box_data(1) = x5
            box_data(2) = y5
            box_data(3) = z5
c            call hfn(800,box_data)
c
            if(((z5-6.0)**2.0+x5**2.0).le.((pmt_rad)**2.0))then
               ef   = rand()
```

```
               Phit = Phit + 1
               if((wl.le.500.0).and.(ef.le.0.25))then
                  Ph1 = Ph1 + 1
                  go to 10
               else  if((wl.gt.500.0).and.(ef.le.0.1))then
                  Ph2 = Ph2 +1
                  go to 10
               endif
            end if
c
            the4 = acos(2.0*rand()-1.0)
            phi4 = rand()*pi
            dx   = sin(the4)*cos(phi4)
            dy   = sin(the4)*sin(phi4)
            dz   = cos(the4)
            l2   = sqrt((x4-x5)**2.0+(y4-y5)**2.0+(z4-z5)**2.0)
            x4   = x5 + l2*dx
            y4   = y5 + l2*dy
            z4   = z5 + l2*dz
            go to 20
c
         else if(y4.ge.6.0)then
            x5   = (dx/dy)*(-y4+6.0)+x4
            z5   = (dz/dy)*(-y4+6.0)+z4
            y5   = 6.0
            box_data(1) = x5
            box_data(2) = y5
            box_data(3) = z5
c            call hfn(800,box_data)
c
            if(((z5-6.0)**2.0+x5**2.0).le.((pmt_rad)**2.0))then
               ef   = rand()
               Phit = Phit + 1
               if((wl.le.500.0).and.(ef.le.0.25))then
                  Ph2 = Ph2 + 1
                  go to 10
               else  if((wl.gt.500.0).and.(ef.le.0.1))then
                  Ph2 = Ph2 +1
                  go to 10
```

```
               endif
            end if
c
            the4 = acos(2.0*rand()-1.0)
            phi4 = (rand()+1.0)*pi
            dx   = sin(the4)*cos(phi4)
            dy   = sin(the4)*sin(phi4)
            dz   = cos(the4)
            l2 = sqrt((x4-x5)**2.0+(y4-y5)**2.0+(z4-z5)**2.0)
            x4 = x5 + l2*dx
            y4 = y5 + l2*dy
            z4 = z5 + l2*dz
            go to 20
c
         else if(z4.le.0.0)then
            x5 = (dx/dz)*(-z4)+x4
            y5 = (dy/dz)*(-z4)+y4
            z5 = 0.0
            box_data(1) = x5
            box_data(2) = y5
            box_data(3) = z5
c            call hfn(800,box_data)
c
            the4 = asin(rand())
            phi4 = (rand()*2.0)*pi
            dx   = sin(the4)*cos(phi4)
            dy   = sin(the4)*sin(phi4)
            dz   = cos(the4)
            l2 = sqrt((x4-x5)**2.0+(y4-y5)**2.0+(z4-z5)**2.0)
            x4 = x5 + l2*dx
            y4 = y5 + l2*dy
            z4 = z5 + l2*dz
            go to 20
c
         else if(z4.ge.12.0)then
            x5 = (dx/dz)*(12.0-z4)+x4
            y5 = (dy/dz)*(12.0-z4)+y4
            z5 = 12.0
            box_data(1) = x5
```

```
            box_data(2) = y5
            box_data(3) = z5
c            call hfn(800,box_data)
c
            the4 = acos(rand())+pi
            phi4 = (rand()*2.0)*pi
            dx   = sin(the4)*cos(phi4)
            dy   = sin(the4)*sin(phi4)
            dz   = cos(the4)
            l2 = sqrt((x4-x5)**2.0+(y4-y5)**2.0+(z4-z5)**2.0)
            x4 = x5 + l2*dx
            y4 = y5 + l2*dy
            z4 = z5 + l2*dz
            go to 20
c          else
      endif
         go to 20
c          end if
 10   continue
c
      return
      end
```