衝突型高エネルギー重イオン実験における飛行時間測定器と電磁カロリメーターのコンピューターシミュレーションによる評価

清道 明男

筑波大学自然学類物理学専攻
学籍番号 920316

卒業論文

指導教官 :八木浩輔、三明康郎

平成 8 年 2 月 6 日

Abstract

米国のブルックヘブン国立研究所(BNL)において建設中の衝突型高エネルギー重イオン加速器(RHIC)ではクォーク・グルーオン プラズマ(**QGP**)の探索を目的とした**PHENIX**と呼ばれる実験が行われる予定である。**PHENIX**実験のような衝突型加速器実験では各種の検出器がぎっしり詰め込まれた状態で測定を行うことになる。そのため、このような実験では測定に先立ち、検出器の配置、形状やバックグランドの評価などシミュレーションによってあらかじめ計算しておく必要がある。

本研究では、この装置より我々の研究室で開発、製作している飛行時間測定器と一番近くにあり検討の必要のある電磁カロリメーターをピックアップし、シミュレーションコード**GEANT**を用いてこれらの検出器の動作を計算機上で実現した。そして電磁カロリメーターの存在による飛行時間測定器の測定への影響、飛行時間測定器の存在による電磁カロリメーターのエネルギー測定への影響を調べ検出器間の動作環境を評価した。

目次

図 目 次

第 1 章

序論

我々の宇宙は約 150 億年前のビックバンから始まったとされる。この宇宙創成のきわめて初期段階に、核子や中間子の閉じ込めから解放され自由に動き回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る。QGP とは、エネルギー密度約 1–2 GeV/fm^3の非常に高温高密度の状態であり、地上では唯一高エネルギー重イオン衝突実験にて生成する可能性を持っている。これにより QGP を生成且つ検出することは原子核物理の重要な課題の一つである。

米国のブルックヘブン国立研究所（BNL）において現在、1999 年実験開始を目標に衝突型高エネルギー重イオン加速器（RHIC）が建設中である。完成の暁には、核子あたり 100 GeV の金の原子核同士の衝突実験が可能となり、反応の中心部の数百 fm^3 の領域において、2 ~ 6 GeV/fm^3 ものエネルギー密度が達成されると予測されている。これは、最高のエネルギー密度であるとともに、量子色力学の予測する QCD 相転移に必要なエネルギー密度を十分に越えていると考えられている。

RHIC において QGP の研究をする実験計画は大きなものが二つあり、STAR と呼ばれる実験と PHENIX と呼ばれる実験がある。STAR 計画では特に、稀に起こる現象よりも個々の衝突で発生する粒子すべてを観測し、大まかに事象全体を把握することを目的としている。一方、PHENIX 計画では、なるべく多種多様な QGP 生成のシグナルを同時に正確に測定することを戦略としている。カバーする領域は STAR 計画に比べ狭くなるが、粒子識別能力に優れ、かつ、稀な現象を数多くとらえる設計となっている。測定対象に由来して、**PHENIX**(Photon, Hadron and Electron in Nuclear Interaction eXperiment) と名付けられている。我々の研究室では、この PHENIX 計画に参加しており、粒子識別のための検出器として飛行時間測定器（**ToF**:Time of flight）の研究、開発を行っている。

原子核同士の衝突実験の場合、QGP 生成の有無に関わらず、実際に観測されるのはおびただしい数のπ中間子等の既知の粒子であり、仮に QGP ができていたとしてもその寿命は非常に短く、たちまち崩壊してもとの粒子群に戻ってしまう。このような複雑な現象では、一つ二つのシグナルを測定しても不十分である。PHENIX 実験では、QGP の生成時に予測される約 7 ~ 8 種のシグナルを同一条件下で見る必要があり、原子核同士の衝突で生成されるレプトン対（μ粒子対、電子対）、光子、ハドロ

ン (K 中間子、π中間子、ϕ中間子、j/ψ中間子、反陽子等々) をそれぞれ分離して測定する必要がある。そのため PHENIX では、ハドロン識別用である我々の飛行時間測定器をはじめとして、μ粒子を測定するミュースペクトロメーター、飛跡認識および運動量分析のための飛跡検出器、電子識別用のリングイメージチェレンコフカウンター、エネルギーを測る電磁カロリメーター等がぎっしりと詰め込まれた状態で測定を行うことになる。個々の検出器の性能は各研究グループで研究されているが、検出器間の影響についての検討はいまだ不十分な状況である。

PHENIX において飛行時間測定器は衝突点から 5 m の位置、ϕ方向に 30 度覆っている。そしてすぐ後ろに鉛ガラスの電磁カロリメーターが配置されている。飛行時間測定器に使用した様々な物質が電磁カロリメーターにどのような影響を与えるか、また逆に電磁カロリメーターが飛行時間測定器に与える影響はあるか、詳しく検討する必要がある。

本研究では、PHENIX の装置より飛行時間測定器と電磁カロリメーターをピックアップし、シミュレーションコード **GEANT** を用いてこれらの検出器の動作を計算機上で実現した。そしてそれらの影響など、動作環境の評価を行なった。

1–1

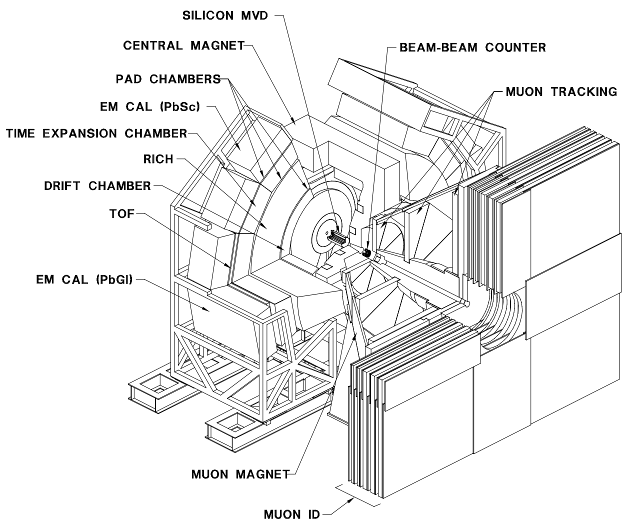


Figure 1.1: Schematic of PHENIX detector.

図 1.1: PHENIX detector

1–2

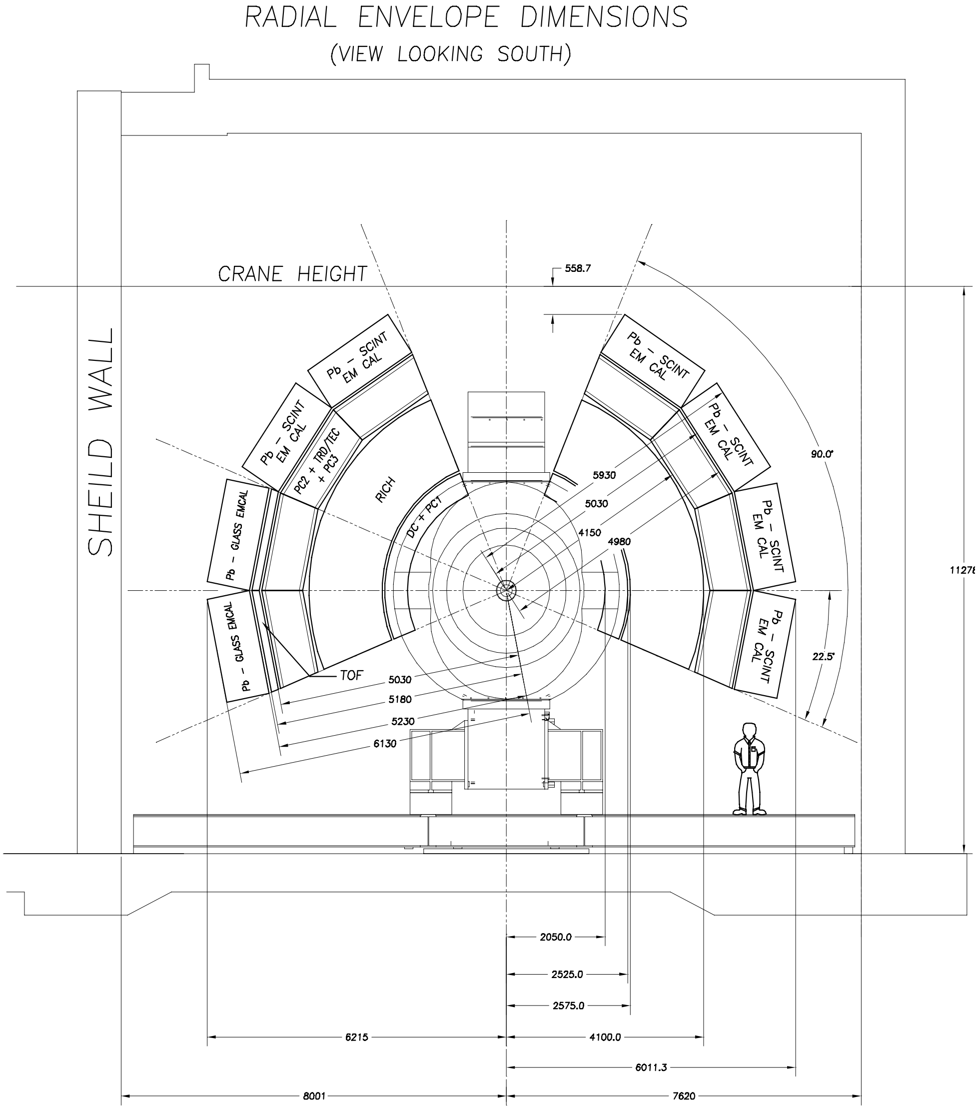


Figure 1.2: Radial cross section of PHENIX central arm.

図 1.2: PHENIX detector

第 2 章

基本原理

シミュレーションの対象となる飛行時間測定器と電磁カロリメーターについてその動作原理を述べる。

2.1 飛行時間測定器（ToF:Time-of-Flight）について

2.1.1 飛行時間法

一定の運動量を持った粒子が一定の距離を通過するのに要する時間を測定することにより粒子の速度を知り、これによって粒子のエネルギーを知ることが出来る。これを飛行時間法（ToF 法）という。高エネルギー原子核実験において荷電粒子の測定には、電磁石と飛跡検出器により粒子の磁場中での曲率半径を測定し、運動量 p を求める。スタートカウンター、ストップカウンターの二つの検出器を一定の距離をおいて設置し、粒子が通過する時間を測定することにより速度 v が求まる。そして公式

$$mc^2 = cp\sqrt{\beta^{-2} - 1} \tag{2.1}$$

より粒子の質量 m が求められ、粒子の種類が決定される。

この飛行時間法による粒子識別能力は時間分解能によるところが大きく、検出器として検出効率や時間分解能に優れたシンチレーションカウンターが用いられることが多い。我々の研究室では PHENIX 実験用の飛行時間測定器としてプラスチックシンチレーターを用いたシンチレーションカウンターについての研究、開発を行っている

2.1.2 シンチレーションカウンター

シンチレーションカウンターは放射線の計数およびエネルギーを測定するのに用いられる。シンチレーター（蛍光体）に光電子増倍管を組み合わせた検出器である。

シンチレーターとは荷電粒子が当ったときに蛍光を発する物質のことで、この発光現象をシンチレーション、光のことをシンチレーション光と呼ぶ。実際に利用されるシンチレーターの条件として、次のことが上げられる。

1. 荷電粒子の運動エネルギーを高いシンチレーション効率で検出可能な波長が可視光領域付近（$300 \sim 600nm$）にあること。
2. できるだけ広い範囲の光の収量が付与エネルギーに比例すること。
3. 良好な集光特性を得るためにその材料に対して透明であること。
4. 粒子の通過後短時間内に光が放出されること、シンチレーション光の減衰時間が短いこと。
5. シンチレーション材料は光学的性質が良好で、実際の検出器として十分な大きさのものが制作できること。
6. シンチレーション光を光電子増倍管へ効率よく導くため、シンチレーション光の屈折率がガラスの値（約 1.5）に近いこと。

これらすべてを満足する材料はないので、シンチレーターを選定するに当っては目的に合わせてこれらの条件とその他の因子を妥協させることになる。大きく分けてシンチレーターには無機材料によるものと有機材料によるものの二種類あり、シンチレーション機構に違いがある。有機物質中の蛍光過程は単一分子のエネルギー準位間での遷移によって生じる。これは分子の種類によって定まるものでその物理的状態には依存しない。一方無機結晶シンチレーターは規則的な結晶格子がシンチレーション過程の基礎になっている。有機シンチレーターは無機材料によるシンチレーターに比べると光の集率はあまり良くないが、応答時間がはるかに速く（ns 程度、一方無機シンチレーターは μs 程度）、計数率の高い測定により向いている。我々の開発している飛行時間測定器は時間の測定を目的としているので、有機（プラスチック）シンチレーターを使用している。

使用するプラスチックシンチレーター（BICRON BC-404）の物理定数は以下の通りである。

立ち上がり時間（Rise Time） : $0.7\ ns$
立ち下がり時間（Decay Time） : $1.8\ ns$
パルス半値幅（Pulse Width,FWHM） : $2.2\ ns$
光減衰長（Light Attenuation Length） : $140\ cm$
最大発光波長（Wavelength of Max.Emission） : $408\ nm$
反射係数（Refractive Index） : 1.58
密度（Density） : $1.032\ g/cm^3$

入射粒子のエネルギーが高いとき、シンチレーターによる蛍光量とエネルギー損失はほぼ比例する。**GEANT** では蛍光量は計算できないので、エネルギー損失の測定によりシンチレーターの評価をする。

シンチレーターによって発した蛍光は光電子増倍管 (PMT) によって検出される。光電子増倍管は、真空容器 (ガラス管) の中に光を金属または半導体より成る光電面にあてて光電子を放出させる光電陰極と、この電子を何段かのダイノード（二次電子

放出面）にあて二次電子を増殖させて電流として取り出す電子増倍器より構成されている。

光電子増倍管にも時間分解能に優れているタイプ、エネルギー分解能に優れているタイプ等があり、大きさも様々なものがある。飛行時間測定器に使用している光電子増倍管は浜松フォトニクス社の R3478 というもので、時間分解能に優れている。

光電子増倍管の使用にあたり磁場をシールドすることが重要になる。これがμ メタルシールドというもので、鉄や Ni,Cu,Co の合金でできている。

2.1.3 物質中でのエネルギー損失

ここではシンチレーターの中で起こる粒子のエネルギー損失の様子について述べる。エネルギー損失の過程は粒子の種類やエネルギーによって異なるが、その主要なものは以下のとおりである。 陽子や π 中間子といった重い荷電粒子は物質との電磁的な相互作用をするので、物質原子内の電子の励起や電離などによってエネルギーを失う。このエネルギー損失は電離損失と呼ばれ、粒子の速度が極めて遅い場合を除いて、すべてのエネルギー領域で最も重要な過程である。このエネルギー損失の値はベーテ- ブロッホの公式で与えられる（式 2.2 参照）。物質 $1g \cdot cm^{-2}$ を通過するとき、（スピンをもたない粒子の場合）

$$-\frac{dE}{dx}=\frac{4\pi N_0 Z z^2 e^4}{m_e v^2 A}\left[\ln\frac{2m_e v^2}{I(1-\beta^2)}-\beta^2\right] \tag{2.2}$$

ここで x は $g \cdot cm^{-2}$ 単位の物質の厚さ、 I はすべての電子についての有効電離ポテンシャルで、$v,\beta\left(=v/c\right)$ および ze は入射荷電粒子の速度と電荷、 N_0 はアボガドロ定数、Z および A は物質の原子番号および質量数、m_e は電子の質量である。スピンをもつ粒子の場合も、電離損失の平均値にはベーテ- ブロッホの式があてはまる。

この式からわかるように、電離損失によるエネルギー損失の特徴は次のとおりである。

1. 入射粒子の荷電 (z) の二乗に比例、質量に依存しない。
2. 電荷 z の入射粒子の電離損失は、非相対論的速度 $(v \ll c)$ のとき $1/v^2$ に従う。また相対論的速度 $(\gamma \gg 1)$ においては $\gamma=1/\sqrt{1-\beta^2}$ の増加に伴いゆっくり上昇する。
3. 物質の Z/A に近似的に依存している。単位電荷の入射粒子に対しては原子番号の小さな物質の場合 $1g \cdot cm^{-2}$ あたり約 $2MeV$ であり、原子番号の大きな物質の場合は約 $1MeV$ である。

2.1.4 飛行時間測定器の構造

PHENIX 実験における飛行時間測定器は、$1.5cm \times 1.5cm$ のシンチレーターが $63cm$ のものが 64 本、$43cm$ のものが 32 本の計 96 本が図 2.1 のようにぎっしり集まって１パネルを構成する。それぞれのシンチレーターの両端に読み出し用の光電子増倍管と μ メタルシールドが 192 個取りつけられている。その他、これらを取りつけるための台として $200cm \times 50cm \times 3cm$ のハニカムボード（蜂の巣状の段ボールを $3mm$ 幅のカーボンの板で囲ったもの）、設置のためのアルミのフレームで構成されている。また、光電子増倍管の数だけの同軸ケーブルがある。

PHENIX 実験では、このパネルを 8 つ並べて使用することになる。

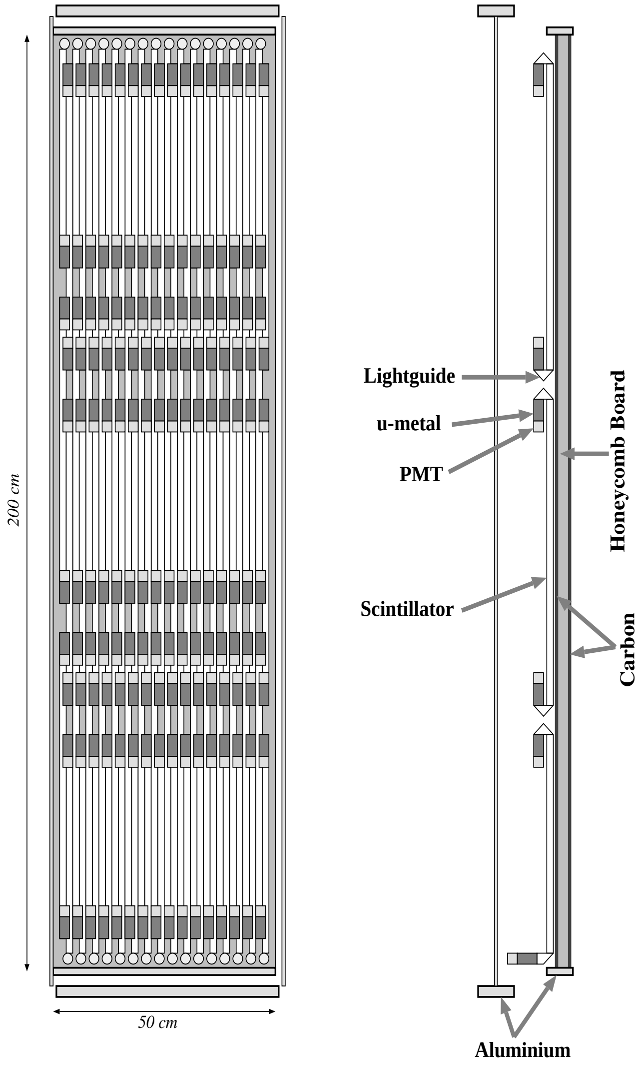


図 2.1: 飛行時間測定器 [ToF] （1 パネル）

2.2 電磁カロリメーターについて

2.2.1 カロリメーターとは

原子核や素粒子の実験において、高エネルギー粒子のカスケードシャワー現象を測る測定器をカロリメーターと呼ぶ。シャワーは物質中での二次粒子生成反応の繰り返しによって入射エネルギーが消費される現象で、発達しきったシャワーの場合には二次荷電粒子による電離または分子励起の総量および全粒子数が、平均として入射エネルギー E に比例する。カロリメーターはシャワーを吸収できるだけの厚い物質で構成され、そこでの電離量やシンチレーション光量などを測る機能をもち、エネルギーを求めることに使われる測定器である。

カロリメーターは以下に示すような 4 つの重要な特徴を持つ。

1. シャワーは反応ごとの変動が大きい素過程の積み重ねであり、大きな統計的変動がつきまとう。しかしエネルギーが高くなると、この相対的変動の大きさは $\sqrt{E}$ に逆比例するのでカロリメーターの性能は向上する。

2. 高密度物質中でシャワーの横への広がりは小さく、横方向に分割した構成により、多数粒子の同時検出とともに、個々の粒子の入射位置を求めることもできる。

3. シャワーは入射粒子の電荷の有無にかかわらず起きるので、γ 線や中性子などの中性粒子のエネルギーと位置を測定できる。

4. 素過程の種類によってシャワー発達の大局的なようすに違いを生ずるため、粒子の種別、特にハドロンとレプトンを区別する重要な手段ともなる。

これらの特徴を兼ね備えたカロリメーターは、高エネルギーの実験において不可欠な測定器である。

カロリメーターに必要な厚さ、つまりシャワーの奥行きはエネルギーとともに対数的に増える。電子や γ 線による電磁シャワーは電磁相互作用を素過程とするので、放射長が厚さの単位になる。中間子や核子がおこすハドロンシャワーでは原子核を相手にした強い相互作用が素過程となるので、核吸収長を単位とする。だいたい電磁シャワーには $15 \sim 20$ 放射長、ハドロンシャワーには $6 \sim 10$ 吸収長が必要となる。測定したい粒子によって設計は違ってくる。

カロリメーターには、大別して 2 つの構造があり、PHENIX における電磁カロリメーターもこの 2 種類ある。一つは、シャワーを吸収できるだけの厚い物質のみで構成されているものであり、PHENIX 実験においては重量比で $50 \sim 70$ % の酸化鉛を含んだ透明な鉛ガラスが用いられている。この鉛ガラスによるカロリメーターは、二次電子によるチェレンコフ光を検出手段とするもので分解能が優れている。しかしこのように均質な物質で構成される測定器は高価であるうえ実用的な厚さに限度がある。その代りにシャワーを効率よく発達させるための金属板と薄い粒子検出器とを交互に積み重ね、シャワーエネルギーを一定間隔でサンプルする方法もよく使われ、こちらは主にサンプリングカロリメーターと呼ばれている。金属板（吸収板）中での

シャワー部分は観測しないので、金属の厚さ（サンプルする頻度）に制限された性能の低下は避けられないが全体の厚さを自由に設定できること、シャワー発達の様相を記録できること、いろいろな種類の検出器を使えることなどの利点がある。PHENIX 実験においては、電磁カロリメーターとして鉛とシンチレーティング・タイル（プラスチックシンチレーター）のサンドイッチ構造が使用される。

PHENIX 実験において飛行時間測定器の後ろに配置される電磁カロリメーターは鉛ガラスによるカロリメーターであるので、これについてのみ説明する。

2.2.2 電磁シャワー

電磁カロリメーターに高エネルギーの電子や光子が入射すると、電子の制動放射による光子生成や光子による電子・陽電子対生成が起こり、新たに光子や電子・陽電子の対が生成される。二次的に生まれた光子や電子も、物質の中を走行中、さらに電子・陽電子対生成や制動放射を起こす。上の過程がカロリメーター中で繰り返し起こり、その深さにつれて、電子、陽電子、光子の数は増大していく（シャワーの発達）。この現象を電磁カスケードシャワーという。入射時に電子や光子が持っていたエネルギーは多数の生成粒子に分配されるので、１つ１つの粒子のエネルギーは粒子数の増加に伴い減少する。こうして電子や陽電子のエネルギーが物質によって決まる臨界エネルギー (E_c) まで減少すると一放射長当たりの電離損失エネルギーが制動放射によるエネルギー損失を上まわるようになり、制動放射やそれにより発生する光子による電子・陽電子対生成を通しての粒子増殖も止まり、粒子の数は物質の深さとともに減少する（シャワーの減衰）。

臨界エネルギー (E_c) は粒子の電離損失と放射損失が等しくなるエネルギーと定義され、近似的に $E_c \sim 800/(Z-1.2)[MeV]$ と表される。シャワーの発達のだいたいの見当は次の式により与えられる。電子がエネルギー (E_0) をもって入射したときシャワー粒子の数が最大となる深さ T_0 は放射長単位で

$$T_0 = 1.01\left(\ln\frac{E_0}{E_c} - 1\right)\left[\text{放射長}\right] \tag{2.3}$$

となり、そこでの電子の数は

$$N_0 = \frac{0.137}{(\ln(E_0/E_c) - 0.37)^{1/2}}\frac{E_0}{E_c} \tag{2.4}$$

で近似的に表される。

光子がエネルギー (W_0) をもって入射したとき、シャワーの T_0 は

$$T_0 = 1.01\left(\ln\frac{W_0}{E_c} - \frac{1}{2}\right)\left[\text{放射長}\right] \tag{2.5}$$

と近似され、そこでの電子の数は

$$N_0 = \frac{0.137}{(\ln(W_0/E_c) - 0.183)^{1/2}}\frac{W_0}{E_c} \tag{2.6}$$

となる。この式では電子の電離損失を無視している。

シャワーの横方向の広がりは

$$\left\langle X^2 \right\rangle = 0.60\left(\frac{E_s}{E_c}\right)^2 \left[\text{放射長}^2\right] \tag{2.7}$$

となる。ここで $E_s = 21MeV$ である。

2.2.3 チェレンコフ光

鉛ガラスの電磁カロリメーターは、二次電子によるチェレンコフ光を検出手段とするので、別名全エネルギー吸収型チェレンコフカウンターという。放射体が鉛ガラスのような高密度の物質では二次電子によるチェレンコフ光のみであるが、NaI 結晶等ではチェレンコフ光にシンチレーション光が加わり大きな出力が得られる。

チェレンコフ光は荷電粒子の速度 v が物質中の光速度 c/n （n は物質の屈折率）より大きいとき粒子の飛跡に沿って物質が発する弱い光のことである。このとき発生する光電子数 $N_{p.e.}$ は次の式で表せる。

$$N_{p.e.}(E) = L\frac{\alpha^2 Z^2}{r_e m_e c^2}\int \varepsilon_{col}(E)\varepsilon_{det}(E)\sin^2\theta_c(E)dE \tag{2.8}$$

$$\cos\theta_c = \frac{c}{nv} = \frac{1}{n\beta}$$

$$\frac{\alpha^2 Z^2}{r_e m_e c^2} = 370cm^{-1}eV^{-1}$$

ここで、

L : 粒子が物質中を通過した長さ
ε_{col} : チェレンコフ光の集光効率
ε_{det} : 光電面における量子効率

である。

θ_c がほぼ一定ならば、

$$N_{p.e.}(E) \approx LN_0\left\langle \sin^2\theta_c \right\rangle \tag{2.9}$$

$$N_0 = \frac{\alpha^2 Z^2}{r_e m_e c^2}\int \varepsilon_{col}(E)\varepsilon_{det}(E)$$

で表せる。

2.2.4 電磁カロリメーターの構造

PHENIX 実験における鉛ガラスの電磁カロリメーターは、一つのモジュールの構成が $4.1cm \times 4.1cm \times 50cm$ の鉛ガラスと発生した光電子の読み出し用に光電子増倍管が一つずつついている。これが 4×6 の 24 個集まって図 2.2 のような 1 ユニットになる。

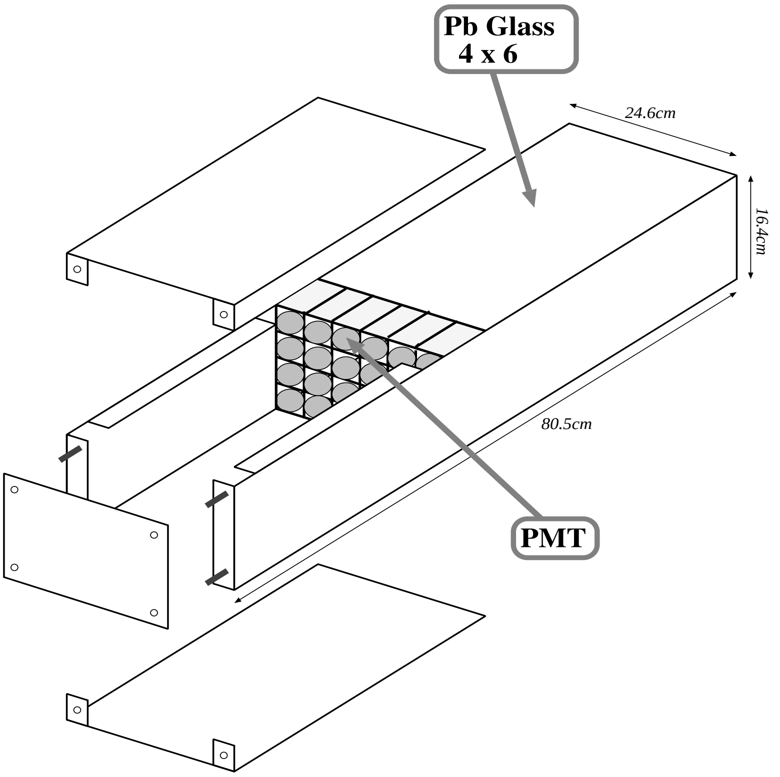


図 2.2: PbGl 電磁カロリメーター

このユニットがビーム軸方向に 16、ファイ方向に 12 の 192 ユニット、つまりモジュールの数ではビーム軸方向に 96 個、ファイ方向に 48 個の 4608 個が一つの側面に配置される。これが飛行時間測定器 8 パネルのすぐ後ろにおかれるカロリメータである。

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　環境設定

シュミレーション・コード GEANT に入力した飛行時間測定器及び電磁カロリメーターの構造を図 3.1 に示す。

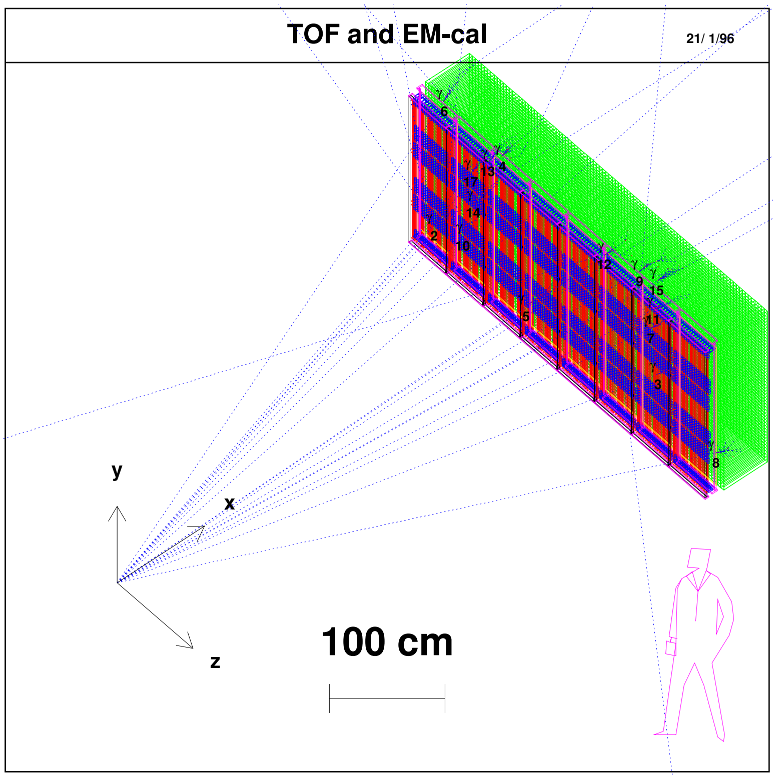


図 3.1: GEANT に入力した飛行時間測定器と電磁カロリメーター

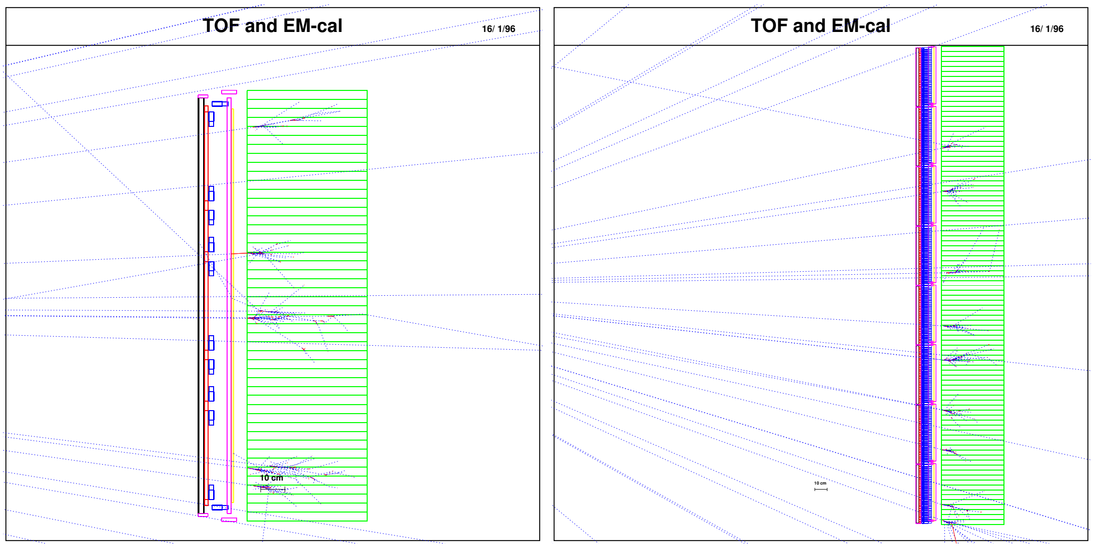


図 3.2: 側面から見た飛行時間測定器と電磁カロリメーター

図 3.1 で z 軸に沿ってビーム軸があり原点で正面衝突する。この原点を衝突によって発生した粒子の発生点とする。飛行時間測定器は衝突点（原点）より 500cm の位置、電磁カロリメーターは 520cm の位置にある。

それぞれの物質は以下の通り。サイズは 2.1.4,2.2.4 節を参照。

新たに定義した物質					
物質	原子量	原子番号	密度	放射長 (cm)	吸収長 (cm)
シンチレーター (BC-404)	6.221	3.373	1.032	0.424×10^2	0.820×10^2
鉛ガラス (SF-2)	116.162	47.045	4.010	0.252×10^1	0.384×10^2

その他の物質		
物質	成分	放射長 (cm)
カーボン	C 100%	0.188×10^2
アルミニウム	Al 100%	0.890×10^1
ライトガイド	$C_5H_8O_2$ 100%	0.231×10^2
PMT のガラス	SiO_2 80% + B_2O_3 12% + Na_2O 5%	0.101×10^2
μ メタル	Fe100%	0.177×10^1
同軸ケーブル	Cu 39.3% + C_2H_3Cl 29.6% + C_2H_4 5%	0.832×10^1

ハニカムは材質が紙であるため粒子と相互作用する確率が低いと考え、シミュレーションには入れなかった。同軸ケーブルはその構造を入力することは困難なため、材質の比から物質を定義し同軸ケーブルは一枚の板として入力した。

また、飛行時間測定器と電磁カロリメーター以外の影響は極力なくして考えたので、それぞれ真空の中に配置した。

3.1.2 相互作用の設定

発生粒子と物質との相互作用として以下のものを考える。必要に応じて独立に切ることもした。

- 粒子対生成
- コンプトン散乱
- 光電効果
- 光子による原子核崩壊
- デルタ線放射
- 陽電子対消滅
- 制動放射
- ハドロン間の相互作用
- ミューオン・原子核相互作用
- 粒子の崩壊
- エネルギー損失 (ランダウのゆらぎ)
- 多重散乱 (モリエール散乱)

3.2 飛行時間測定器の設定

3.2.1 構造

飛行時間測定器 1 パネルの構造は図 3.3のようになる。2.1.4 節で述べた構造になるようできるだけ正確に入力した。

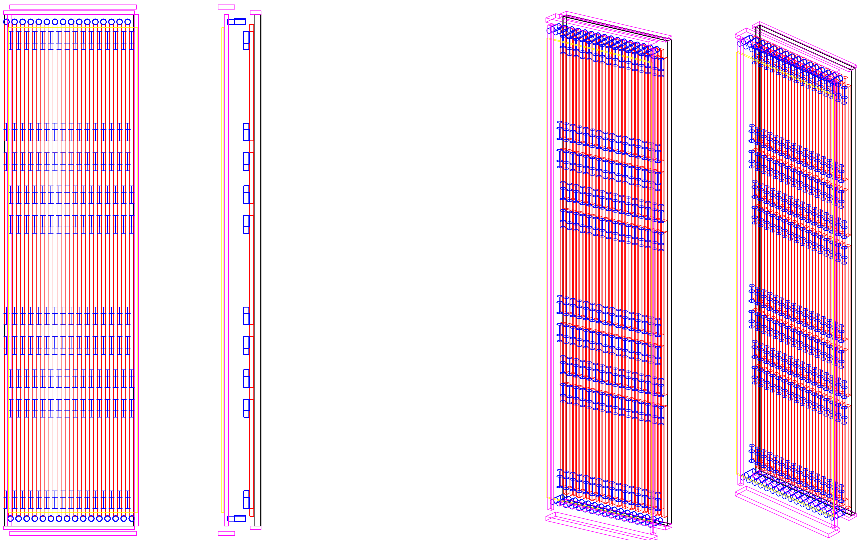


図 3.3: **GEANT** に入力した飛行時間測定器

飛行時間測定器はその構造より放射長を単位として 0.1 ∼ 0.3[放射長] の厚さがある。図 2.1,3.3 よりシンチレーター、カーボン、同軸ケーブルは飛行時間測定器の全体に置かれ、PMT、アルミフレームは一部に配置されている。それぞれの厚さは以下の通り。

飛行時間測定器の構造		
	領域	厚さ [放射長]
横にした PMT のある部分	20.0%	0.24
縦にした PMT がある部分	1.0%	0.78
アルミフレームの部分	6.0%	0.34
その他	73.0%	0.15

3.2.2 エネルギー損失の計算

２章で述べた通り **GEANT** ではシンチレーターの蛍光量は計算しない。その代わりにシンチレーターにおけるエネルギー損失の総量で評価した。図 3.4 は π^- 中間子を入射したときの飛行時間測定器のシンチレーターにおけるエネルギー損失である。

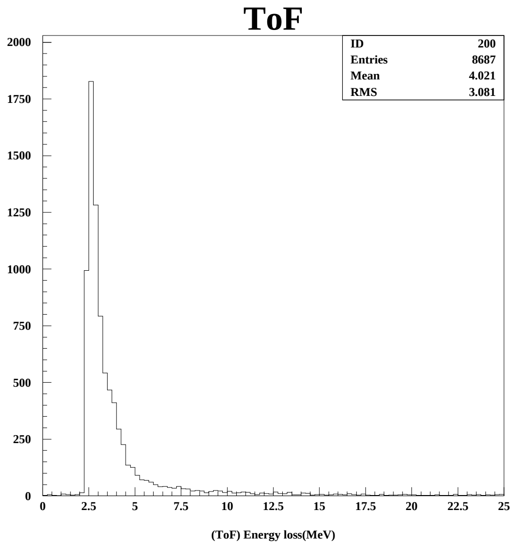


図 3.4: 飛行時間測定器におけるエネルギー損失

図 3.4 をみるとエネルギー損失のピークは約 $3MeV$ であり厚さ $1.5cm$ のシンチレーターとしては適当な値である。また、分布は Gauss 分布から外れて高い側に延びており、典型的な Landaw 分布を示している。

高速荷電粒子が薄い物質中で起こすエネルギー損失の分布は Landaw 分布 （または Landaw ゆらぎ、電離ゆらぎと呼ばれる) になる。Landaw 分布になる理由は、一回の衝突によるエネルギー損失の確立分布

$$P(E)dE \propto \frac{1}{E^2}dE \tag{3.1}$$

が広い、シンチレーターが薄いため物質中での全衝突回数が少ない、等があげられる。もし通過する物質が厚く一回の衝突によるエネルギー損失が小さければ Gauss 分布に近くなる。

3.3 電磁カロリメーターの設定

3.3.1 チェレンコフ光の計算

GEANT ではチェレンコフ光の放射による光量は計算しない。この代わりになる量として、式 2.9 より計算した光電子数 $N_{p.e.}$ で評価した。この時、光電子増倍管による検出量ではなく、鉛ガラス中のみの光電子数を測ることにしたので、集光効率 ε_{col} 、量子効率ε_{det} は 1 として式 3.2 にて計算した。

$$
\begin{aligned}
N_{p.e.}(E) &= LN_0 \left\langle \sin^2\theta_c \right\rangle \\
&= LN_0 \left[1 - \frac{1}{n^2\beta^2} \right] \qquad (3.2) \\
N_0 &= 370 \\
L &= \mathrm{STEP} \\
\beta &= \frac{p}{E} = \frac{\mathrm{VECT(7)}}{\mathrm{GETOT}}
\end{aligned}
$$

ここで、STEP, VECT(7),GETOT は **GEANT** の parameter でそれぞれ 1 ステップあたりの粒子が通過した距離、運動量、全エネルギーである。また鉛ガラスの屈折率 n は $n = 1.5$ である。これをチェレンコフ光発生の条件 （$n\beta > 1$、荷電粒子）のもとで計算した。

比較のために入射粒子を電子と μ 粒子、運動量を $100MeV \sim 5GeV$ の範囲でいくつか設定しそれぞれ 1000 イベントずつ測定した。

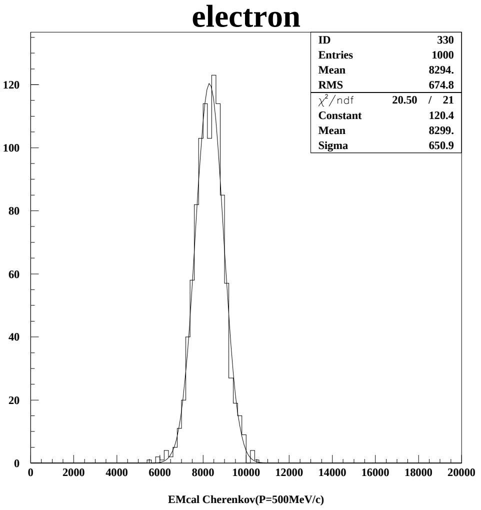


図 3.5: カロリメーターで測定した光電子数の代表例

図 3.6 左は電子及び μ 粒子の入射エネルギーと計算して求めた光電子数の関係である。電子の場合、入射エネルギーと光電子数に比例関係があり正常に電磁シャワーの量を計算していることが分かる。また図 3.6 右は電子のエネルギー分解能を表しており

$$\frac{dE}{E} = \frac{dN_{p.e.}}{N_{p.e.}} = \frac{5.8\%}{\sqrt{E}} \tag{3.3}$$

と求められる。この値は鉛ガラスのカロリメーターのエネルギー分解能としては適当な値である。

μ 粒子の場合、電離によるエネルギー損失が主な反応であるため $50cm$ の鉛ガラスに対して約 $400MeV$ のエネルギーを落とす。よって約 $500MeV(= 400 + m_{\mu})$ 以上では突き抜けるので一定値になる。

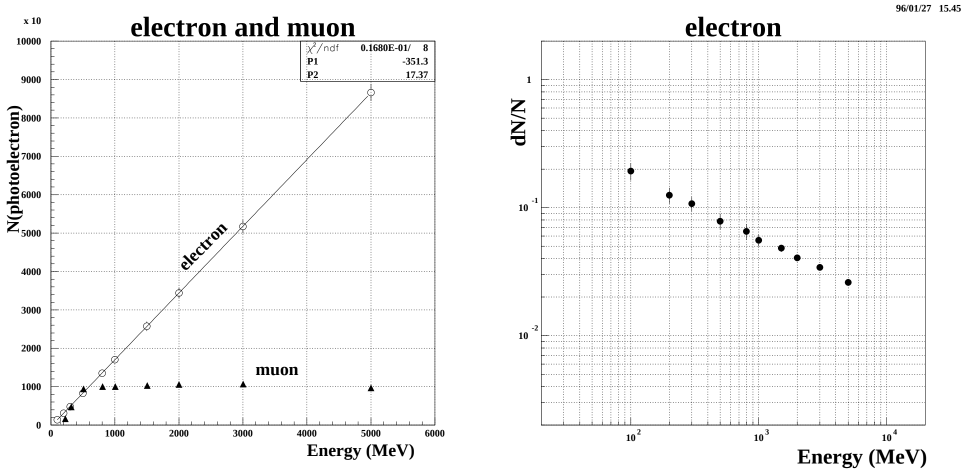


図 3.6: 電子、 μ 粒子の測定と分解能

3.3.2 クラスターアルゴリズム

実際の実験では同時に発生する粒子が多数になる。これらの粒子が同時に一カ所に入射した場合、検出器はそれらを識別することができない。この同時入射の確率をできる限り小さくし、かつ電磁シャワーの起きる十分な領域を確保しなければならない。そのためカロリメーターでは幾つかのモジュールを組み合わせて、1 粒子による測定としている。これをクラスタリング という。エネルギー分解能と位置分解能の兼ね合いよりその範囲を決定する。

式 2.7より電磁シャワーの横方向の広がりは臨界エネルギー $(\simeq 17MeV)$ のあたりで $2.6cm$ 、 $1MeV$ まで追うと $30cm$ まで広がる。また、PHENIX においてこのカロリメーターの領域に入射する粒子の数は理論上 90 個程度と予想されている。1 粒子が 1 つのクラスターに入射する確率を 10% とすると、2 粒子が入射する確率は 1% になる。この確率を生むためのクラスターの面積は約 $100cm^2$ と求められる。

これらの兼ね合いから 1 つのクラスターあたりのモジュール数は 9 個とした。そしてそのアルゴリズムは、すべてのモジュールから光電子数ピークの位置を選び、それに周りの 8 つのモジュールからの数を足し合わせて 1 粒子の検出とした。図 3.7及び図 3.8は γ 線を 1 個、100 個発生させたときの測定例とエネルギー分解能である。エネルギー分解能はともに

$$\frac{dE}{E} = \frac{dN_{p.e.}}{N_{p.e.}} = \frac{6.5\%}{\sqrt{E}} \tag{3.4}$$

とやや落ちるが粒子識別は実現している。

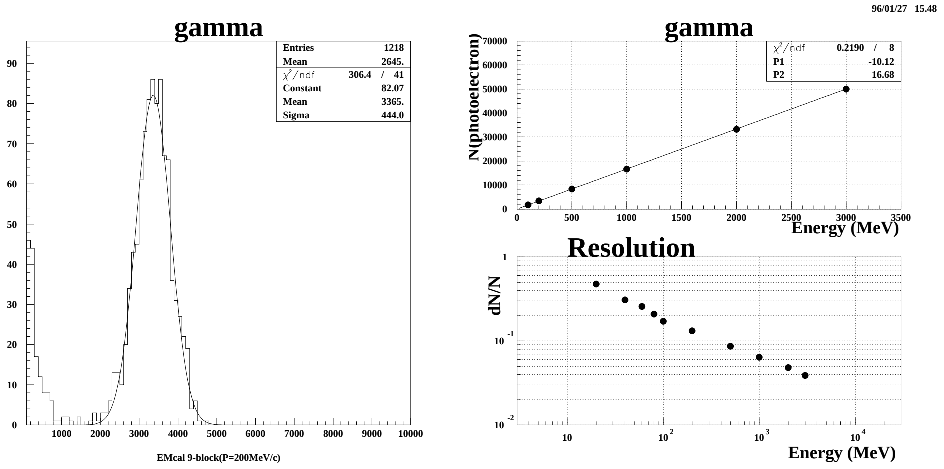


図 3.7: γ 1 個発生時の測定

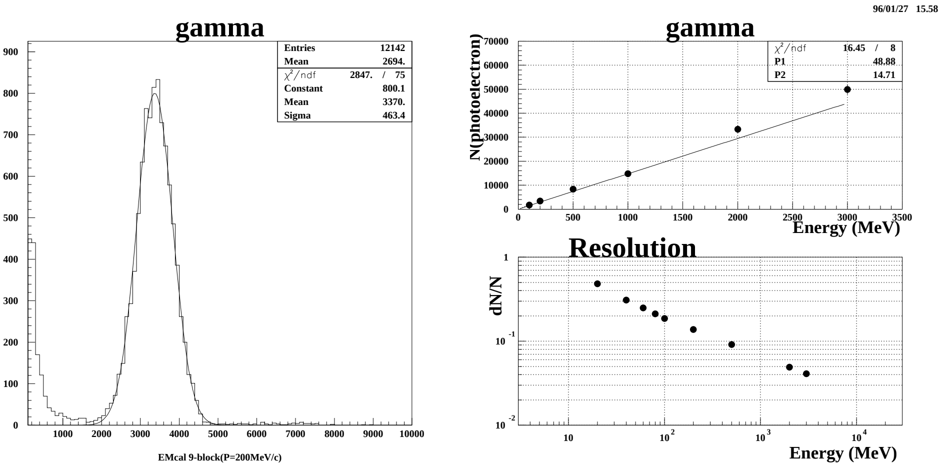


図 3.8: γ 100 個発生時の測定

3.3.3 γ 線における変換式の決定

クラスタリング 9 個のもとで測定する光電子数とエネルギーの変換式を決定する。図 3.9は γ 線の入射エネルギーと光電子数の関係である。 これより

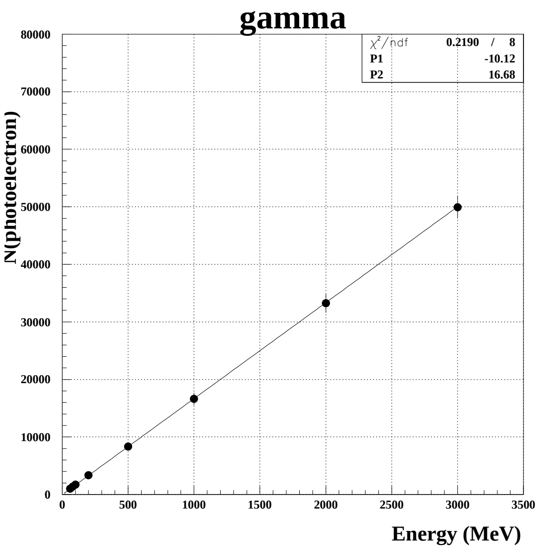


図 3.9: γ 線の測定

$$\mathbf{E} = 5.995 \times 10^{-2} N_{p.e.} \tag{3.5}$$

と決めた。またこの時のエネルギー分解能は

$$\frac{dE}{E} = \frac{6.5\%}{\sqrt{E}} \tag{3.6}$$

である。

第 4 章

シミュレーションの結果

ハドロンを測定する飛行時間測定器への電磁カロリメーターの影響を調べるためにπ^- 中間子入射による測定、及び γ 線のエネルギーを測定する電磁カロリメーターへの飛行時間測定器の影響を調べるために γ 線入射による測定をそれぞれシュミレーションして評価した。そして PHENIX 実験で予想される粒子発生分布の一つ、π^0 中間子を発生させて電磁カロリメーターの動作を評価した。

4.1 π^- 中間子入射による測定

飛行時間測定器は主に電荷を持ったハドロンを識別する検出器である。そこで、π^- 中間子を入射したとき、核反応により発生する二次粒子が飛行時間測定器にどのような影響を及ぼすか電磁カロリメーターがあるときと無いときの比較を行った。

4.1.1 シミュレーション

π^- 中間子を次の条件で入射した。

- 入射粒子は π^- 中間子。
- 1 イベント当たり 1 粒子、方向は検出器に一様に当たるようにランダムにする。
- 運動量 $250MeV/c, 500MeV/c, 1000MeV/c, 1500MeV/c$ についてそれぞれ 10000 イベントおこなう。
- 粒子の崩壊 (IDCAY) を off にする。

π^- 中間子は平均寿命 $2.6 \times 10^{-8} sec$ で μ 粒子とニュートリノ ν_μ に崩壊する。実際は崩壊した μ 粒子を捕らえることも多いが[*] 、今回は核反応からの影響を見るため粒子の崩壊は余計な情報を増やしてしまう。そこで粒子の崩壊を off にして測定した。

---

[*] 飛行時間測定器は途中で崩壊した粒子は正確に識別できない。実験では IDCAY の on/off でシミュレーションを行い、測定値と比較する

同等な条件のもとで電磁カロリメーターがある場合と無い場合をそれぞれ比較した。測定によって得られたシンチレーターにおけるエネルギー損失の分布について入射運動量が $1000MeV$ の場合を例にあげる。(図 4.1　左は飛行時間測定器の後ろに電磁カロリメーターを置かない場合、右は置いた場合)

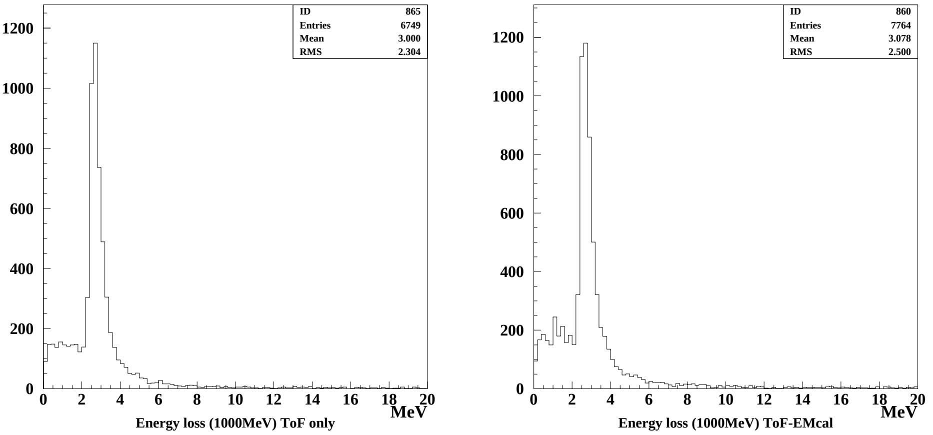


図 4.1: エネルギー損失 (入射運動量 $1000MeV/c$) 左:ToF のみ、右:ToF と EMcal

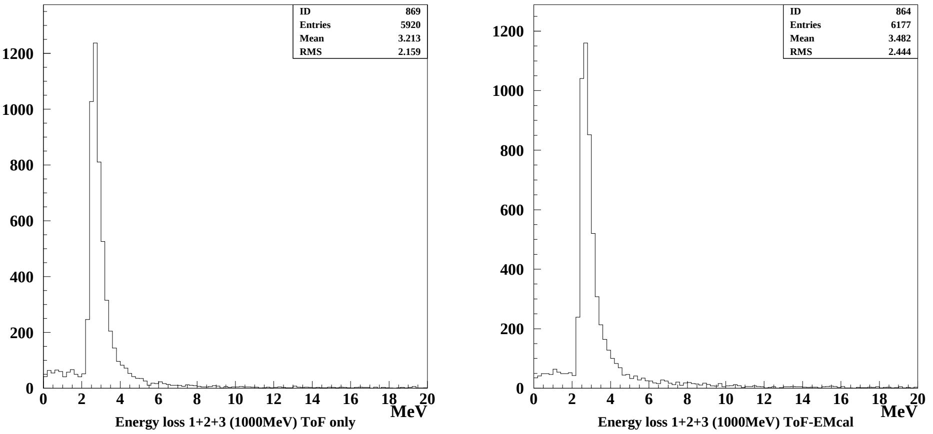


図 4.2: 1 イベントあたりのエネルギー損失 (入射運動量 $1000MeV/c$) 左:ToF のみ、右:ToF と EMcal

1 イベントで多数の粒子が発生した場合を考えているのでシンチレーター 1 本ごとにエネルギー損失を計算している。そのため 1 粒子が 2 本のシンチレーターを通る可能性もある。 $0 \sim 2MeV$ の領域は 1 粒子が 2 本のシンチレーターを通ったときのエネルギー損失であると考えられる。

1 イベントごとにシンチレーター 1 本のエネルギー損失を大きなものから 3 つ filling したので、これを足し合わせた値はほぼ 1 粒子の全エネルギー損失に対応する。（図 4.2 ）これは図 3.4の分布に等しい。

図 4.1,4.2の左右のグラフを比べると、電磁カロリメーターの存在する場合 $1MeV$ のあたりに小さなピークが見られるほか、両者にほとんど差が見られない。

4.1.2 飛行時間測定器の評価

GEANT では実験値として与えられるデータ以外に、崩壊、散乱などの現象を追うことができる。そこで飛行時間測定器のシンチレーターに入った粒子を詳細に調べた。粒子の崩壊を off にしているので、発生する二次粒子は飛行時間測定器自身の物質（カーボン、PMT など）や電磁カロリメーターの鉛ガラスと π^- 中間子の核反応によるものと考えられる。

まず、全イベントに対してシンチレーターに入った粒子が 2 個以上であるイベント数を比べると、飛行時間測定器のみの場合 $20 \sim 24\%$ 、電磁カロリメーターがある場合 $50 \sim 60\%$ と電磁カロリメーターの存在によって大きく増えていることが確認された。さらにその粒子の内訳をみると π^- 中間子の数を 100% として、下の表のようになる。

入射運動量	全粒子数	π^- の数	$\pi^-(\%)$	$\gamma(\%)$	$e(\%)$	$\pi^0(\%)$	$n(\%)$	$p(\%)$
$250MeV/c$	9018	7652	100.	4.4	0.2	0.0	9.5	3.6
$500MeV/c$	7830	6820	100.	3.2	0.1	1.0	8.0	2.1
$1000MeV/c$	7959	6962	100.	2.2	0.1	0.9	7.6	3.3
$1500MeV/c$	7858	6982	100.	2.1	0.1	1.0	5.6	3.6

表：シンチレーターを通過した粒子（ToF のみ）

入射運動量	全粒子数	π^- の数	$\pi^-(\%)$	$\gamma(\%)$	$e(\%)$	$\pi^0(\%)$	$n(\%)$	$p(\%)$
$250MeV/c$	14000	8121	100.	31.0	1.2	0.0	35.2	4.8
$500MeV/c$	16102	7906	100.	27.6	1.6	16.8	53.0	3.3
$1000MeV/c$	13951	7316	100.	25.2	1.0	10.9	47.5	4.3
$1500MeV/c$	13407	7070	100.	23.0	1.0	7.9	51.5	4.2

表：シンチレーターを通過した粒子（ToF と EMcal）

表をみると、電磁カロリメーターの存在により γ 線、電子などの二次粒子の生成量が増えていることがわかる。特に γ や中性子などの中性粒子が多く発生している。また、入射運動量の低い側では π^- 中間子自身の量も増えており、 π^- が反射した場合と考えられる。

π^- 中間子と原子核の核反応には

$$\begin{aligned} \pi^- + p &\longrightarrow \gamma + n \\ \pi^- + p &\longrightarrow \pi^0 + n \end{aligned}$$

等の反応があり、これら二次粒子の多くは鉛ガラスの原子核との核反応により発生していると考えられる。

以上より、電磁カロリメーターが飛行時間測定器に与える影響として正規の粒子(ここでは π^- 中間子)の 8 割ほどの数にあたる中性粒子が鉛ガラスで発生しシンチレーターを通過していることが分かる。しかし、シンチレーターは主に荷電粒子のエネルギー損失に反応するものであるので、この発生した中性粒子の多くは反応せず検出器としての飛行時間測定器に与える影響は少ないと思われる。そこで飛行時間測定器に反応を示した粒子について詳細に調べた。

下の表はシンチレーターでエネルギー損失を起こした粒子について π^- 中間子の数を 100% として示した。

入射運動量	全粒子数	π^- の数	$\pi^-(\%)$	$\gamma(\%)$	$e(\%)$	$\pi^0(\%)$	$n(\%)$	$p(\%)$
$250MeV/c$	7882	7533	100.	1.6	0.8	0.0	1.1	1.0
$500MeV/c$	6702	6408	100.	1.1	0.4	0.9	0.9	1.0
$1000MeV/c$	6811	6418	100.	1.6	0.4	0.5	1.3	1.5
$1500MeV/c$	6779	6437	100.	1.2	0.4	0.7	0.8	1.9

表 : シンチレーターでエネルギーを落とした粒子 (ToF のみ)

入射運動量	全粒子数	π^- の数	$\pi^-(\%)$	$\gamma(\%)$	$e(\%)$	$\pi^0(\%)$	$n(\%)$	$p(\%)$
$250MeV/c$	9449	8471	100.	4.8	3.7	0.0	1.5	1.2
$500MeV/c$	9636	7924	100.	5.8	4.8	6.5	2.1	1.5
$1000MeV/c$	7781	6362	100.	5.8	4.1	5.2	2.6	3.1
$1500MeV/c$	7440	6243	100.	5.3	4.3	2.8	2.6	3.0

表 : シンチレーターでエネルギーを落とした粒子 (ToF と EMcal)

表をみると、飛行時間測定器のみの場合は約 5% 、電磁カロリメーターがある場合は約 18% の数の二次粒子がシンチレーターでエネルギーを落としている。つまり、電磁カロリメーターからの二次粒子の数は比にして 13% あるといえる。

これらの二次粒子によるエネルギー損失がどれくらいか求める必要がある。図 4.3 は電磁カロリメーターがあるときの飛行時間測定器のエネルギー損失の分布であり、斜線部は電磁カロリメーターの存在により発生した二次粒子のエネルギー損失である。図 4.3によると二次粒子による影響は $0 \sim 4MeV$ あたりの領域に見られる。

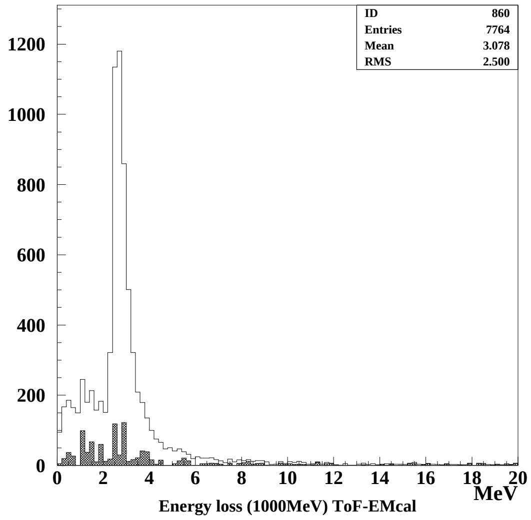


図 4.3: エネルギー損失:二次粒子の影響を重ねてプロット

4.1.3 結論

飛行時間測定器の測定はあるスレッシュホールドを越えたときにカウントするので、ピークより低い領域での二次粒子による損失は飛行時間測定に影響を与えない。飛行時間測定のスレッシュホールドを $2MeV$ として、これより高い領域のみで評価すると二次粒子によるエネルギー損失の割合は 9% である。よって電磁カロリメーターがあることにより飛行時間測定器の粒子識別能力に 9% の影響を及ぼすといえる。

4.2 γ 線入射による測定

電磁カロリメーターは主に γ 線のエネルギーを測定する検出器である。そこで、γ 線を入射したとき前にある飛行時間測定器によりエネルギー測定にどのような影響を及ぼすか比較を行った。

4.2.1 シミュレーション

γ 線を次の条件で入射した。

- 入射粒子は γ 。
- 1 イベント当たり 1 粒子、方向は検出器に一様に当たるようにランダムにする。
- 運動量 $250MeV/c, 500MeV/c, 1000MeV/c, 1500MeV/c$ についてそれぞれ 50000 イベントおこなう。

同等な条件のもと飛行時間測定器がある場合と無い場合で測定し電磁カロリメーターのエネルギー測定を比べた。(図 4.4) 図 4.4 をみると飛行時間測定器があるとき、無いときでほとんど差が見られない。これについて理由を考える。

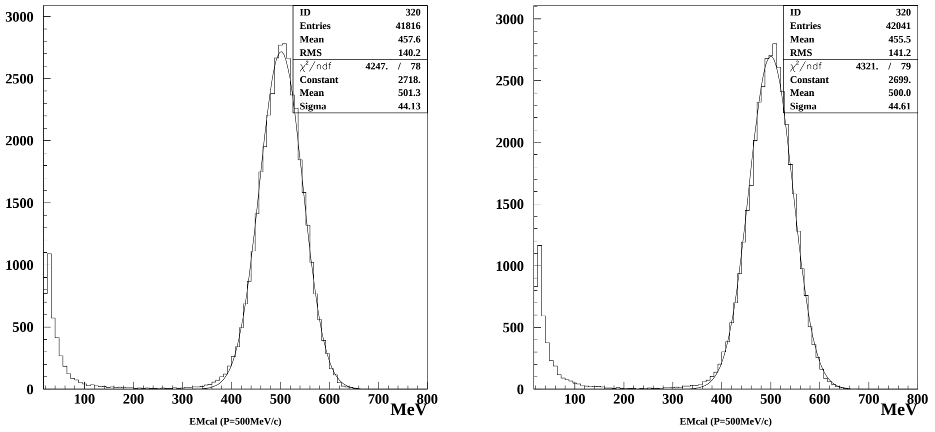


図 4.4: エネルギー測定 (左 ToF なし 右 ToF あり)

4.2.2 電磁カロリメーターの評価

飛行時間測定器の無い場合、粒子の発生点から電磁カロリメーターまでは真空でなにも障害物はないので、ほとんど何も反応を起こさずに 1 粒子が電磁カロリメーターに入射する。この時のエネルギー測定をみると、第 3.3.2 節のクラスタリングにより 9 モジュールで 1 粒子分の測定としているが、シャワーの量や入射の方向によりこの領域から外れて検出してしまう場合がある。それが図 4.4 の低いエネルギー側の統計であると考えられる。そのため 1 粒子に対し 2、3 個のクラスターが反応し 2 粒子以上と識別してしまう場合の対処を考える。

1 イベントに対し 1 番エネルギー測定の大きいクラスターのみを比べた場合、図 4.5,4.6 のようにきれいなガウス分布で表せる。この分布をもとにエネルギー分解能を求めると

$$\frac{dE}{E} = \frac{6.5\%}{\sqrt{E}} \tag{4.1}$$

となる。

図 4.5,4.6 に 2 つめ 3 つめのクラスターの測定量を重ねて表示すると図 4.7,4.8 のようになる。図 4.7,4.8 をみるとそれぞれのエネルギーに対し十分低いエネルギーしか示さない。よって測定したいエネルギー領域により適度なカットをかけることにより十分な精度で識別することが可能である。

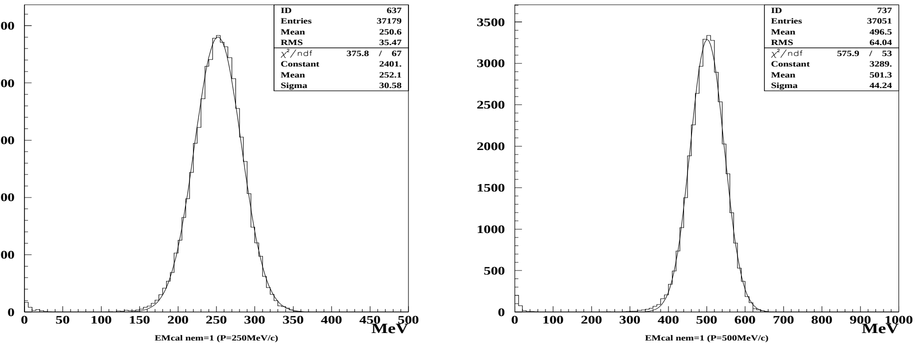


図 4.5: 1 つのクラスターによるエネルギー測定 (左 $250MeV$ 右 $500MeV$)

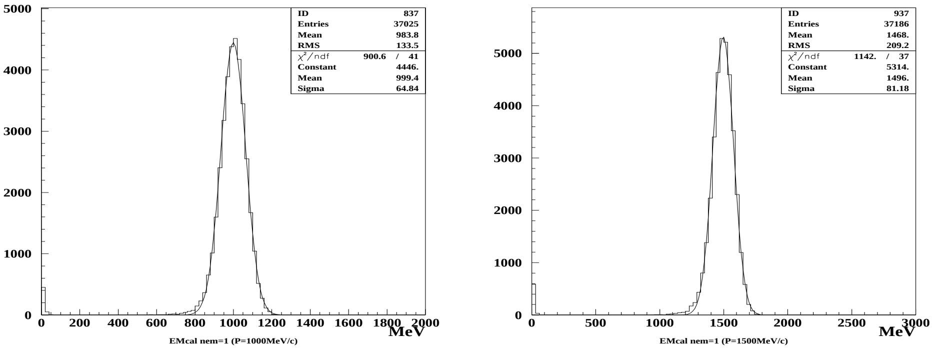


図 4.6: 1 つのクラスターによるエネルギー測定 (左 $1000MeV$ 右 $1500MeV$)

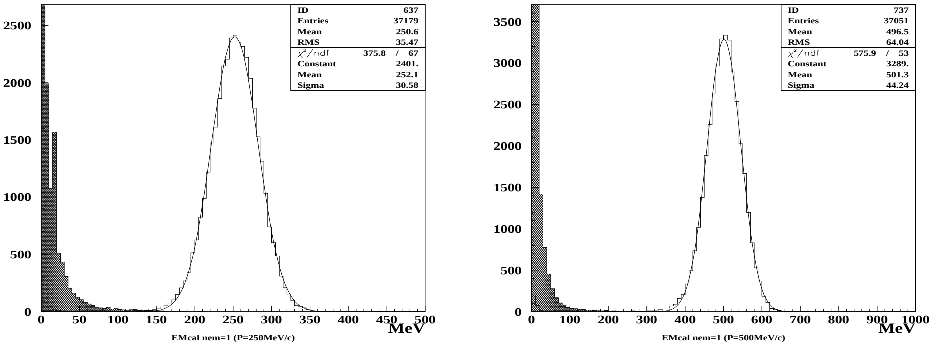


図 4.7: 全てのクラスターによるエネルギー測定 (左 $250MeV$ 右 $500MeV$)

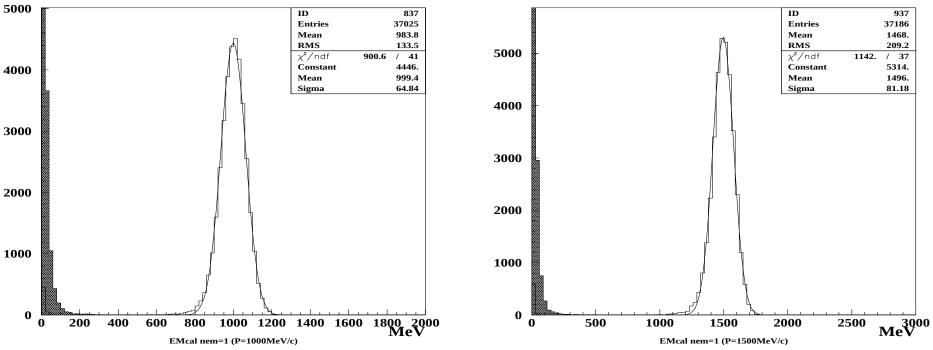


図 4.8: 全てのクラスターによるエネルギー測定 (左 $1000MeV$ 右 $1500MeV$)

4.2.3 飛行時間測定器による影響

3.2 節より飛行時間測定器には $0.1 \sim 0.3$[放射長] ほどの厚さの素材があるので、γ 線の通過に際して吸収、またはコンプトン散乱や対生成による散乱などの相互作用が起こることが考えられる。そこで電磁カロリメーターの表面における粒子の多重度、種類、エネルギーを測りエネルギー測定への影響を調べた。

電磁カロリメーターに達した粒子についてイベントごとの粒子の多重度を図 4.9 と下の表に示す。図 4.9 は横軸に粒子数、縦軸にイベント数を示している。粒子数が 0 個で入射 γ 線が吸収される可能性について、飛行時間測定器があるときと無い時を比べて差が見られないこと、入射粒子のエネルギーが高いので全部吸収される確率は非常に低いことより、γ 線の吸収による多重度 0 の確率はゼロに近いとして評価にはいれなかった。また、粒子数が 1 個の統計は飛行時間測定器で散乱などの反応が起こらなかったイベントとみなせる。粒子数が 2 個以上のイベントが飛行時間測定器により影響を受けたイベントであると見なせる。

そこで粒子数が 2 個以上になるイベントの確率、粒子の種類についてまとめると下の表のようになる。

全イベントのうち $12 \sim 14\%$ の割合で、飛行時間測定器の存在により反応を起こしている。さらに入射エネルギーが高いほど多重度は大きくなっている。

入射運動量	$n \geq 2(\%)$	$n = 2(\%)$	$n = 3(\%)$	$n = 4(\%)$	$n = 5(\%)$	$n = **$
$250MeV/c$	12.0	5.0	3.3	1.9	1.0	··
$500MeV/c$	13.0	4.6	3.4	2.2	1.3	··
$1000MeV/c$	13.7	4.5	3.3	2.3	1.5	··
$1500MeV/c$	13.9	4.3	3.4	2.3	1.6	··

n : 電磁カロリメーターに入射した粒子数 (多重度)

表 : 電磁カロリメーターにおける $n \geq 2(\%)$ のイベント数の割合

入射運動量	全粒子数	$\gamma(\%)$	$e^-(\%)$	$e^+(\%)$
$250MeV/c$	50602	79.7	10.5	9.7
$500MeV/c$	51702	78.1	11.2	10.7
$1000MeV/c$	52548	77.0	11.8	11.2
$1500MeV/c$	52698	77.0	11.7	11.3

表 : 電磁カロリメーターに入射した粒子の種類とその割合

電磁カロリメーターに到達した粒子の全エネルギーと入射エネルギーとの差より、飛行時間測定器によるエネルギー損失を示すと図 4.10 のようになる。ただし 90% 以上のイベントがエネルギー損失ゼロであるため、エネルギー損失をおこしたイベントのみの統計をとっている。エネルギー損失が $10MeV$ 以上になる確率は約 5% 、$20MeV$ 以上になる確率は約 2% ほどである。電磁カロリメーターのエネルギー分解能 $\frac{6.5\%}{\sqrt{E}}$ より、 $250MeV$ のエネルギーに対して標準偏差は $32MeV$ であるので、エネルギー損失の影響は小さいと思われる。

粒子の多重度は多いが図 4.4 をみるとエネルギー測定にはさほどの影響が見当たらない。その理由をみつけるため散乱した粒子の位置関係を調べた。2 つ以上に散乱した粒子について、1 番エネルギーの高い粒子を中心としたときの 2 番目、3 番目の粒子の入射したカロリメーターのモジュールの位置をもとめ統計をとると図 4.11 のようになる。

図 4.11にて (0,0) 座標の統計は 2 つ、3 つに散乱した粒子が同じモジュールに入射していることを意味している。1 つのモジュールに入射している確率は 80% 以上、クラスターアルゴリズムより 9 つのモジュールに入射している確率は 95% 以上とほとんど 1 つのクラスターに収まる範囲であることがわかる。さらにこれらの散乱した粒子は γ 線や電子、陽電子であるためカロリメーターで電磁シャワーを起こすためには差し支えがない。また、エネルギーが高いほど横への広がりが小さく 1 つのクラスターに収まる確率はさらに高くなる。

図 4.12に散乱した粒子の多重度と測定量の関係を示す。横軸は飛行時間測定器によって散乱した粒子数を 5 つまで追い、縦軸にその測定量を入射エネルギーで割った値を示している。これと粒子の多重度（図 4.9）より、入射エネルギーが $250MeV$ の場合は 7% の確率でエネルギーを 5% 小さく測定してしまう。入射エネルギーが $1000MeV$ の場合は 4% の確率でエネルギーを 2% 小さく測定してしまう。

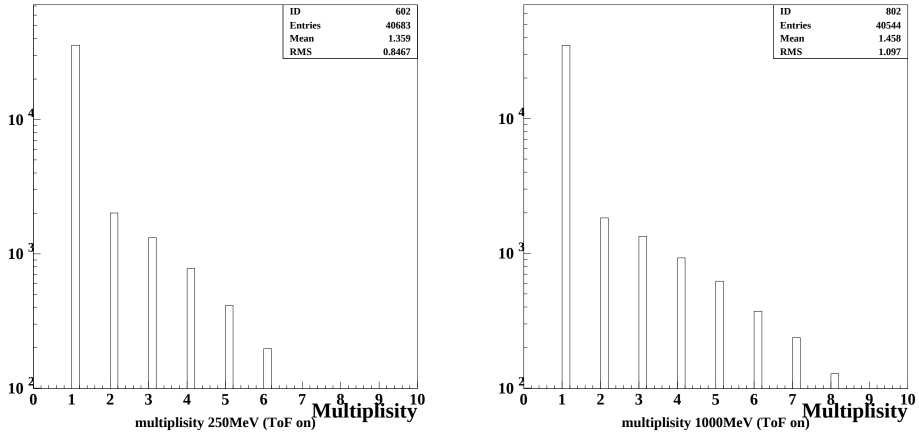


図 4.9: 飛行時間測定器の影響による粒子の多重度（左 $250MeV$ 右 $1000MeV$）

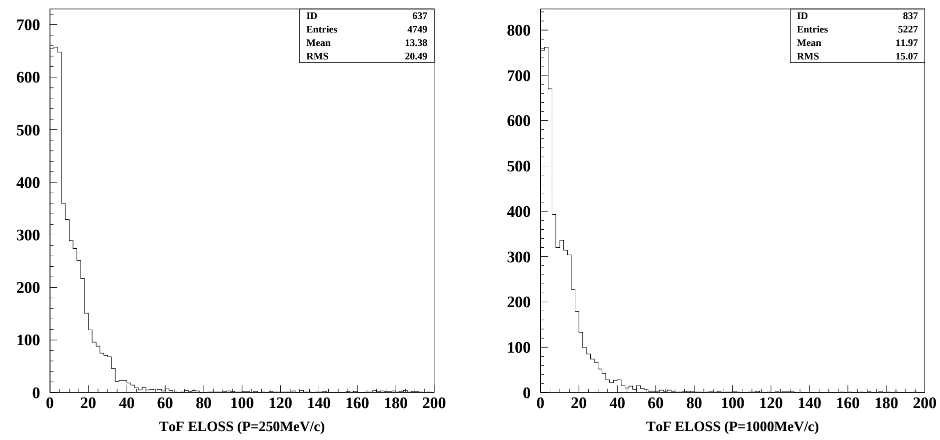


図 4.10: 飛行時間測定器で落としたエネルギー (左 250MeV 右 1000MeV)

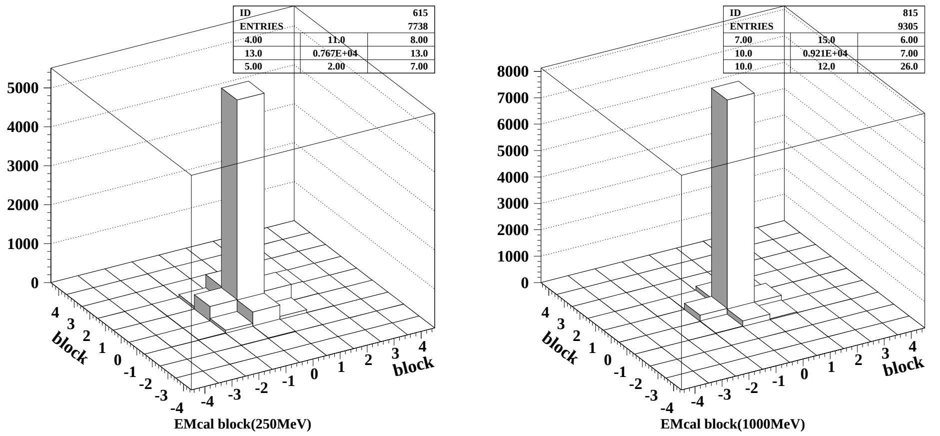


図 4.11: 散乱した粒子の位置関係 (左 250MeV 右 1000MeV)

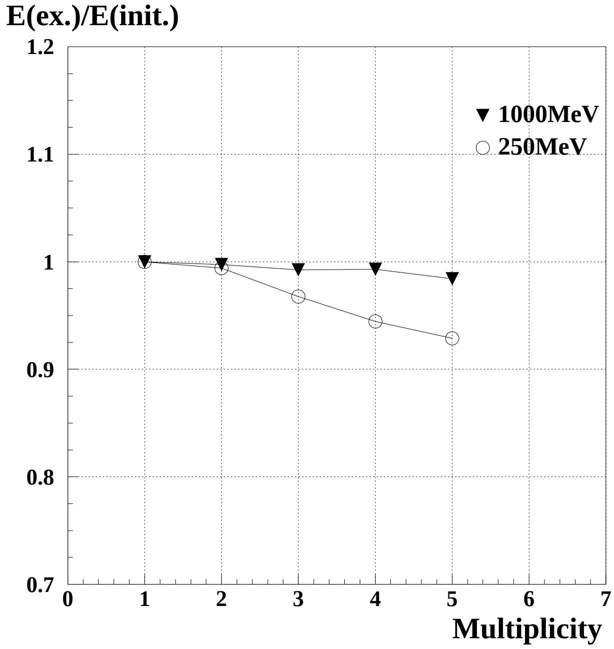


図 4.12: 散乱した粒子とエネルギー測定量

4.2.4 結論

電磁カロリメーターの測定はクラスタリングによる識別の不確さより、本来の粒子のエネルギー測定以外にそのクラスターから外れた場所で新たなクラスタリングを行いエネルギー測定をしてしまう。しかしこれによって測定される量は正規の粒子のエネルギーに比べて十分低く、ピークがはっきりわかれているので、検出したいエネルギー領域に対してカットをかけることにより回避することができる。

飛行時間測定器の存在による影響として γ 線は十数 % の割合でコンプトン散乱、対生成等の反応を起こす。入射粒子のエネルギーが高いものほどそれが大きくなる。しかし電磁カロリメーター上の広がりが小さいことよりその影響は少なく、エネルギー測定に対して $4 \sim 7\%$ の確率でエネルギーを $2 \sim 5\%$ 小さく測定してしまう程度である。また、入射エネルギーが大きいほど飛行時間測定器の影響は少ない。

4.3 π^0 中間子測定 (PHENIX)

PHENIX 実験において電磁カロリメーターは γ 線の測定の一つとして平均寿命 $8.4\times10^{-17}sec$ で 2γ に崩壊するπ^0 中間子を測定する。

そこで、ここでは電磁カロリメーターの位置情報も必要になる測定としてπ^0 中間子を実際の PHENIX 実験で予想される確立分布に従って発生させ、検出器を評価した。

4.3.1 π^0 発生メカニズム

π^0 の発生メカニズムは CDR [1, Chapter3] によると CERN の UA1 という実験[†] のデータより、以下のような発生分布になる。

P_t 分布で （P_t はビーム軸と垂直な方向の運動量）

$$\frac{dN_{\pi^0}}{dP_t} = P_t \exp(-6P_t) \tag{4.2}$$

また、ラピディティ y は $-3 < y < 3$ の範囲で一様になる。

ラピディティは式 4.3 で示されるので

$$y = \frac{1}{2}\ln\left[\frac{E - P_{//}c}{E + P_{//}c}\right] \tag{4.3}$$

ビーム軸方向の運動量 $P_{//}$ は 式 4.4 で表せる。

$$P_{//} = \sqrt{{m_{\pi^0}}^2 + {P_t}^2}\sinh y \tag{4.4}$$

また一回のイベントで発生する π^0 中間子の数は

$$\frac{dN_{\pi^0}}{y} = 375 \tag{4.5}$$

である。

PHENIX のシミュレーショングループはこの値を元に粒子発生のプログラムをつくっている。本研究の π^0 発生メカニズムもこれを元につくった。

4.3.2 π^0 測定メカニズム

π^0 中間子の測定は、実際には崩壊したあとの 2γ の測定による。保存量である不変質量 m_{π^0} の分布をを求める。

エネルギー、運動量の保存より

$$\begin{aligned} E_{\pi^0} &= E_{\gamma1} + E_{\gamma2} \\ P_{\pi^0} &= P_{\gamma1} + P_{\gamma2} \end{aligned}$$

[†] $\sqrt{s} = 200GeV$ の $p\bar{p}$衝突実験

γ の質量は $m_\gamma = 0$ であるので $E_\gamma = |P_\gamma|$ である。

π^0 中間子の不変質量は

$$\begin{aligned} m_{\pi^0} &= \sqrt{{E_{\pi^0}}^2 - {P_{\pi^0}}^2} \\ &= 2\sqrt{E_{\gamma 1} E_{\gamma 2}} \sin \frac{\phi}{2} \end{aligned} \tag{4.6}$$

$$\phi \quad : \quad 2\gamma \text{ の角度}$$

と 2γ のエネルギーと角度より求める。

π^0 中間子の寿命は短いので、検出器の位置と比較して衝突点で 2γ に崩壊していると考えて差し支えない。これと電磁カロリメーターにおける位置情報より角度 ϕ が求まり、電磁カロリメーターで測ったエネルギーより π^0 中間子の不変質量分布を求めることができる。

4.3.3 PHENIX におけるシミュレーション

電磁カロリメーターでの 2γ の測定は、全てのペアについて m_{π^0} を求めその分布を表す。発生する π^0 中間子が 1 イベントあたり数個ならば識別は容易だが、PHENIX 実験のように 1 イベントあたり数千もの π^0 中間子が発生する場合、偽の 2γ を多く測定するためバックグランドが大きくなる。

π^0 中間子の P_t と 2γ の角度 ϕ には式 4.7 のような関係がある

$$P_t = \frac{m_{\pi^0}}{\tan \frac{\phi}{2}} \tag{4.7}$$

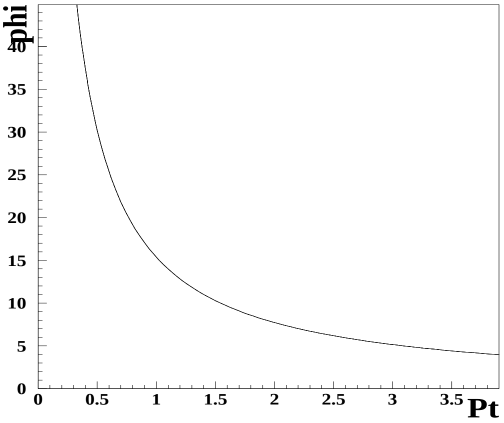


図 4.13: $P_t(GeV/c)$ と ϕ の関係

P_t が大きいほど ϕ が小さくなり、 π^0 からの本物の 2γ が検出器にはいる確率が高くなるので m_{π^0} 分布で π^0 中間子の質量のピークの割合が大きくなる。

図 4.14,4.15 はPHENIXの π^0 中間子のシミュレーション結果である [1, Chapter3] 。4.3.1節で述べた通りにπ^0 中間子を発生させ、それを 2.56×10^6 イベント測定している。γ 線のスレッシュホールドは $250MeV$ で、2γ のproper cut として $\alpha \equiv \frac{|E_{\gamma1}+E_{\gamma2}|}{E_{\gamma1}+E_{\gamma2}} < 0.4$ のカットをかけている。

4.3.4 シミュレーション

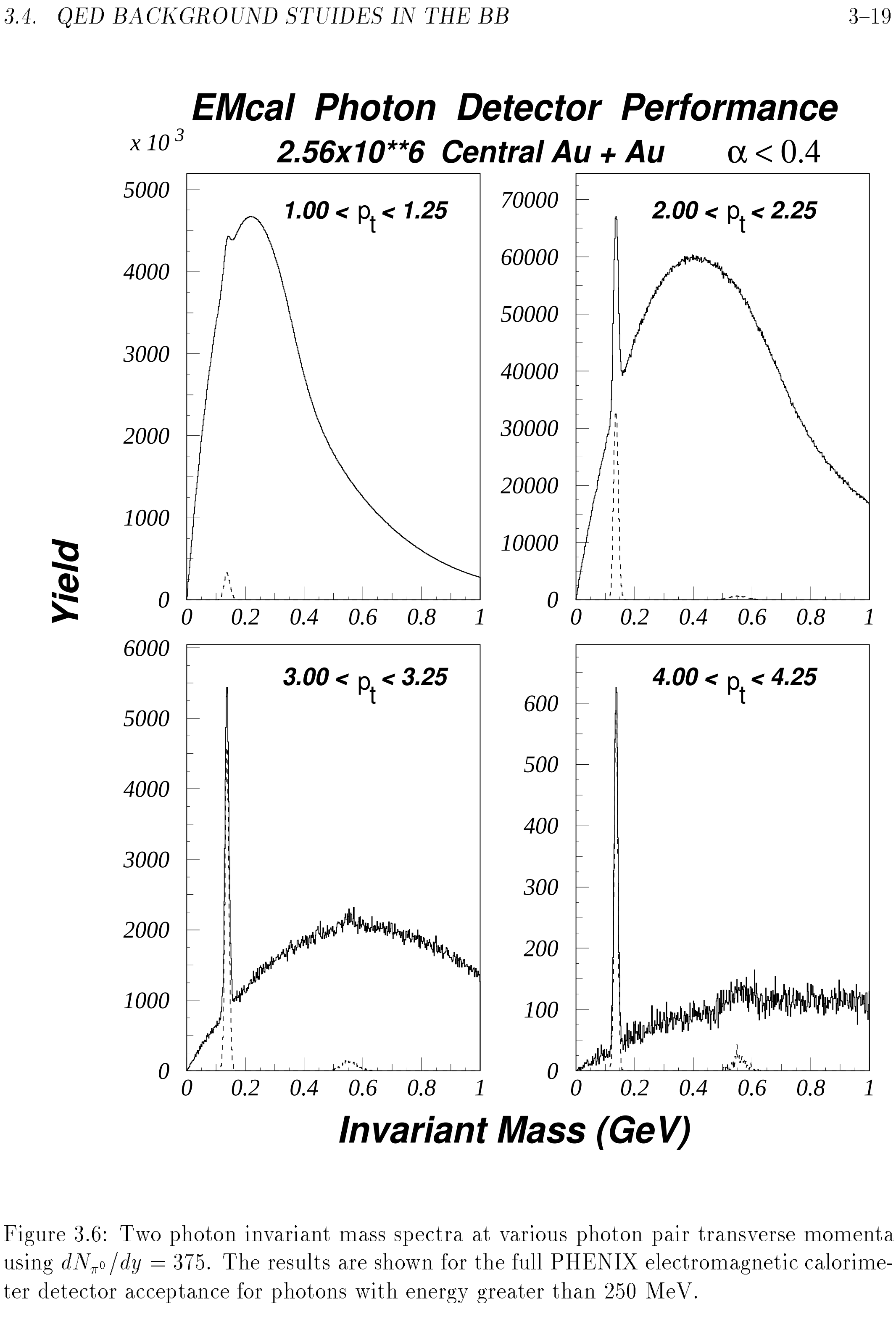
飛行時間測定器がある場合と無い場合とで電磁カロリメーターに影響をおよぼすか確認するため、以上のアルゴリズムにしたがってシミュレーションを行った。測定数は 1.0×10^5 イベント行った。

電磁カロリメーターは、本来 PHENIX 実験では 8 ヶ所ありビーム軸の周り 180° を覆っている。しかし、今回のシミュレーションでは飛行時間測定器のある位置のものしか設定していないので約 30° しか覆っていない。そのため P_t が低く 2γ の角度 ϕ が大きい領域はほとんど測定できない。またバックグラウンドも大きくなってしまい、PHENIX シミュレーショングループの計算した m_{π^0} の分布と比べるとπ^0 の収量は小さくなる。

図 4.16は $1.00 < P_t < 1.25(GeV/c)$ の測定、図 4.17は $2.00 < P_t < 2.25(GeV/c)$ の測定でそれぞれ飛行時間測定器がある場合と無い場合を比べた。また π^0 からの本物の 2γ による測定量を重ねてプロットしてある。

電磁カロリメーターの立体角が小さいこと、計算したイベント数が少ないことより P_t の高い領域の統計はあまりとれておらず、統計的なゆらぎが大きいと考えられる。 $P_t > 2.5GeV/c$ の領域はほとんど統計がとれておらず従って図 4.14のような $P_t \sim 3GeV/c, 4GeV/c$ の測定は全くとれていない。

この測定より π^0 中間子の測定量 (ピークの量) と全測定量の比 (peak-to-total ratio) $Q_{\pi^0} = N_{\pi^0}/(N_{\pi^0} + N_{b.g.})$ を各 P_t cut のもとで求めると図 4.18 のようになる。



3.4. QED BACKGROUND STUIDES IN THE BB 3–19

Figure 3.6: Two photon invariant mass spectra at various photon pair transverse momenta using $dN_{\pi^0}/dy = 375$. The results are shown for the full PHENIX electromagnetic calorimeter detector acceptance for photons with energy greater than 250 MeV.

図 4.14: m_{π^0} の分布 (CDR より抜粋)

3–20 *CHAPTER 3. PHYSICS CAPABILITIES*

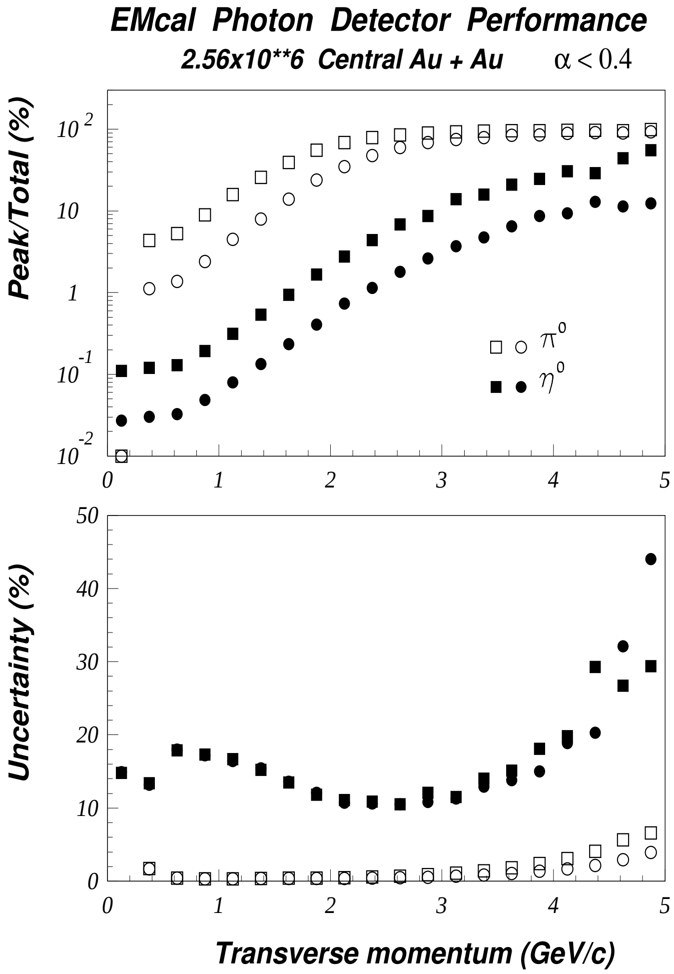


Figure 3.7: PHENIX electromagnetic calorimeter detector performance for $dN_{\pi^0}/dy = 95$ (squares) and $dN_{\pi^0}/dy = 375$ (circles), the multiplicity anticipated at RHIC. Top: π^0 (open points) and η (solid points) measurement showing the transverse momentum dependence of the peak-to-total ratio (Q). An asymmetry cut of $\alpha \leq 0.4$ is applied. Bottom: The statistical uncertainties in the meson yield extraction. Solid points: π^0, open points: η. Systematic errors are not taken into account.

図 4.15: peak-to-total ratio (CDR より抜粋)

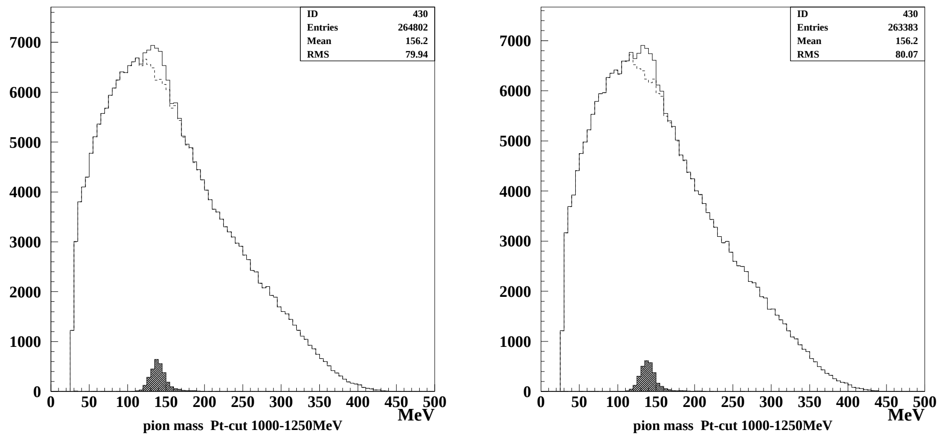


図 4.16: m_{π^0} の分布 P_t cut $1.00 \sim 1.25(GeV/c)$ (左 ToF なし 右 ToF あり)

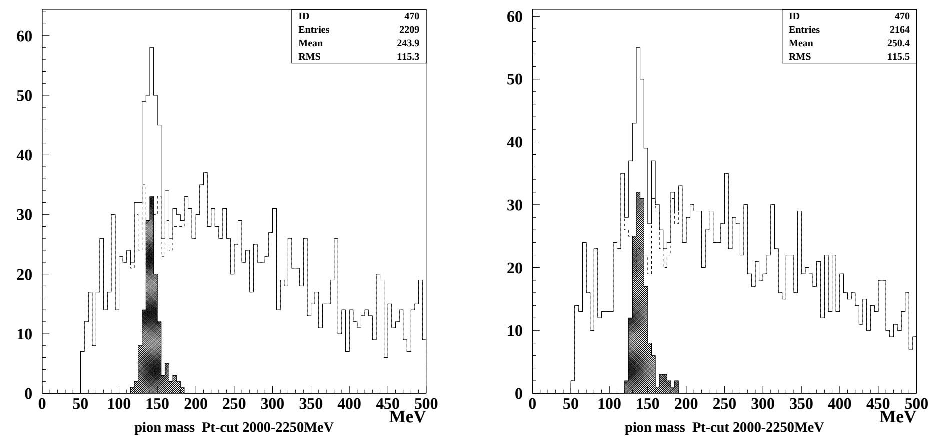


図 4.17: m_{π^0} の分布 P_t cut $2.00 \sim 2.25(GeV/c)$ (左 ToF なし 右 ToF あり)

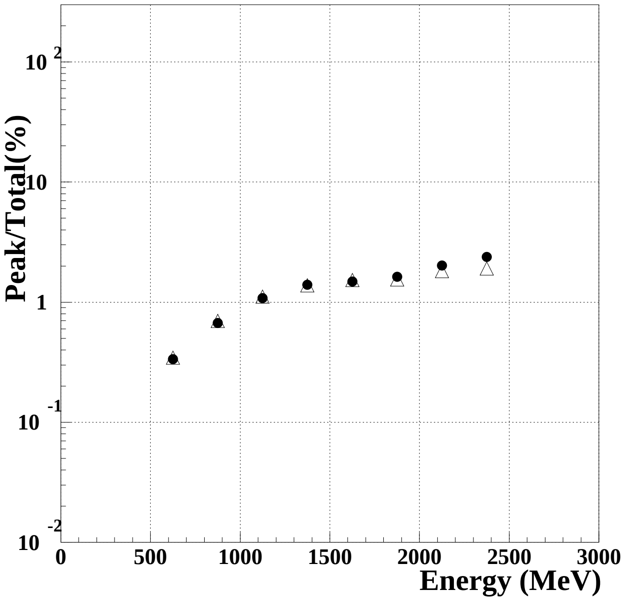


図 4.18: peak-to-total ratio ●：ToF なし、△：ToF あり

4.3.5 結論

π^0 中間子不変質量の測定にて、π^0 の質量 $135MeV$ にピークがあり正しく測定されていることが確認された。またピークの量と全統計量の比（図 4.18）は立体角が小さいなどの理由から CDR の図 4.17 よりは低いが P_t が大きくなるほど比が高くなるという傾向は合っている。このシミュレーションにおける電磁カロリメーターのエネルギー測定、位置測定が十分正しく、実験の再現性が高いことが示されたといえる。

飛行時間測定器の存在による影響についてそれがある場合と無い場合を比べると π^0 中間子の識別に対し $P_t \sim 1GeV/c$ のときの分解能は、飛行時間測定器が無いときで $9.5MeV$、あるときで $9.6MeV$ である。m_{π^0} の分布をみる限り影響は見当たらない。図 4.18 より $P_t < 2GeV/c$ の領域ではピークの量と全統計量の比は一致している。$P_t > 2GeV/c$ の領域では両者に 0.4% ほどの差が見られる。この領域は統計量が比較的少なくゆらぎの効果が大きいので飛行時間測定器の存在による影響が表れているとはいえない。

よって 4.2節で述べられた γ 線に対する影響はπ^0 中間子測定では無視できる量だとみなせる。

第 5 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、PHENIX 実験における飛行時間測定器と電磁カロリメーターの動作環境を計算機上に実現した。そしてこれら検出器間で与える影響について評価した。

電磁カロリメーターの存在による飛行時間測定器の測定への影響をπ^- 中間子入射の測定により調べた。電磁カロリメーターにて核反応を起こして発生する二次粒子は、多くが中性粒子であり、シンチレーターに反応するのは僅かである。さらにしきい値を設定することにより、飛行時間測定に対して電磁カロリメーターによる影響は数 % 程度である。

飛行時間測定器の存在による電磁カロリメーターのエネルギー測定への影響をγ 線入射の測定により調べた。飛行時間測定器によって γ 線は $12 \sim 14\%$ の確率で散乱を起こす。しかし、散乱した粒子のうち 95% 以上が 1 つのクラスターに入射するため、エネルギー測定に対する影響は小さい。$4 \sim 7\%$ の確率でエネルギーを $2 \sim 5\%$ 小さく測定してしまう程度であることを示した。また、入射エネルギーが大きいほど飛行時間測定器の影響は少ない。

さらに、電磁カロリメーターの測定する粒子の代表例として π^0 中間子についての測定を行った結果、飛行時間測定器による影響は見られず、無視できる量であることを示した。

謝辞

本卒業研究を行うにあたり指導教官である八木浩輔教授、三明康郎助教授には多大なご指導を下さいましてここに深く感謝の意を表します。特に三明助教授にはとても魅力のある研究テーマを与えて下さるとともに本研究においてさまざまな面で指導、助言をいただきました。

栗田和好講師、西村俊二助手、研究室の先輩である佐藤進氏、倉田美月氏、叶建治氏には特に本研究テーマであるシミュレーションについて多くのアドバイスをいただきました。

池田時浩氏、絵野沢和彦氏、中條達也氏、横田幸郎氏、大熊靖夫氏、樋口理子氏、宮本祐子氏、芳賀美紀子氏、林祥子氏、加藤純夫氏には物理や計算機など様々な方面で相談に乗ってもらい、また数々のアドバイスをしていただきました。

この場を借りてこれらの皆様に感謝の意を表します。

参考文献

[1] PHENIX Conceptual Design Report (CDR)
Brookhaven National Laboratory

[2] A Primer for the PHENIX Simulation Codes PISA and PISORP
Brookhaven National Laboratory

[3] PHISICAL REVIEW D
The American Phisical Society

[4] GEANT – Detector Destription and Simulation Tool
CERN Program Library

[5] PAW – Physics Analysis Workstation
CERN Program Library

[6] HBOOK – Statisical Analysis and Histogramming
CERN Program Library

[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 1971)

[8] 加藤貞幸　：放射線計測
新物理学シリーズ26　(培風館 1994)

[9] Glenn F.Knoll　：放射線計測ハンドブック　(第二版)
木村逸郎、阪井英次 訳　(日刊工業新聞社 1991)

付録 A

GEANT

GEANT とは、粒子の物質通過に対する相互作用を評価するモンテカルロ・シミュレーション・コードである。高エネルギー物理学実験の分野では検出器の評価や測定の解析のためによく使われている。

利用者依存部のサブルーチン

UGINIT	:	**GEANT** の開始
UGLAST	:	**GEANT** の終了
UGEOM	:	セットアップの入力
GUKINE	:	発生させる粒子の入力
GUSTEP	:	各ステップにおける計算内容
GUOUT	:	1 イベントの終了

GEANT Program Flow Chart

```
MAIN *user routine

     GZEBRA *initialisation of ZEBRA system,dynamic core allocation
     UGINIT *user routine.
         GINIT *initialise the GEANT common blocks
         GFFGO *read free format data records
         GZINIT *initialise the memory manager
         GPART/GSPART *fill the data structure JPART
         GMATE/GSMATE *fill the data structure JMATE
         <user code> *it is the responsibility of the user to:
           - define the geometry of the different components of the setup
             ([GEOM]),stored in the JROTM and JVOLUM data structures
           - define tracking medium parameters([CONS],[TRAK]),stored
             in the JTMED datastructure.
           - specity which elements of the geometrical setup should be
             considered as sensitivedetectors,giving a response when hit
             by a particle([HITS]).
           - usually all done in a user routine called UGEOM.
         GPHYSI *compute energy loss and cross-section tables and store them.

     GRUN *loop over events.
         GTRIGI *initialise event processing and create the event header
                bank JHEAD.
         GTRIG *process one event.
              GUKINE *(user)generates or reads the kinematics of the event and
                    stores it in the data structures JVERTX and JKINE([KINE]).
              GUTREV *calling GTREVE which performs the following operations
                  GTREVE *loop over tracks,including any secondaries generated.
                       GUTRAK *(user)
                           GTRACK *control tracking of current track.
                                GMEDIA *find current volume/tracking medium.
                                GTVOL *loop over successive media seen by the
                                      particle.
                                    GTGAMA/GTELEC/... *tracking of particle
                                                      according to type.
                                    GUSTEP *(user)recording of hits in data
                                           structure JHITS and of space points
                                           in data structure JXYZ.
                  GUDIGI *simulate the detector responses for the event,
                         using the information recorded in the data structure
                         JHITS during particle transport,and store the results
                         in the data structure JDIGI([HITS]).
                  GUOUT *perform all the processing at the end of the event
                        and output the relevant data structures([IOPA]).
         GTRIGC *clearing of memory for next event.
     UGLAST *(user)
         GLAST *standard GEANT termination.
```

付録 B

プログラムリスト

今回の卒業研究で作ったプログラムのリストを一部のせる。

gugeom.f　検出器の配置についてのプログラム

gxuser.f : ToF　第4.1節の測定のためのプログラム

gxuser.f : PHENIX　第4.3節の測定のためのプログラム

B.1　gugeom.f

```
***
*     gugeom.f
***
*
      SUBROUTINE UGEOM
*
************************************************************************
*                                                                      *
*             User routine to define geometry of detector              *
*                                                                      *
************************************************************************
*
C      implicit none
*
*KEEP,GCFLAG.
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEEP,GCTRAK.
      PARAMETER (MAXMEC=30)
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(MAXMEC)
     +     ,NAMEC(MAXMEC),NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG
     +     ,STEP  ,SNEXT ,SFIELD,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL
     +     ,ISTOP ,IGAUTO,IEKBIN, ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN
     +     ,NLVSAV,ISTORY
      CHARACTER*4 NAMEC
      PARAMETER (MAXME1=30)
```

```
      COMMON/GCTPOL/POLAR(3), NAMEC1(MAXME1)
C
*KEEP,GCVOLU.
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     +LVOLUM(15),LINDEX(15),INFROM,NLEVMX,NLDEV(15),LINMX(15),
     +GTRAN(3,15),GRMAT(10,15),GONLY(15),GLX(3)
      CHARACTER*4 NAMES
C
*KEND.
C
      INTEGER IVOLU,IFIELD
C
C ---> Vectors
      real VMOTH(3)/4000.,2000.,4000./
      real pVERT(3)/0.,50.,50./         ! vector of VERTEX (TUBE)[psudo volume]
      real pTFPN(3)/24.5,100.,16./
      real pEMCL(3)/196.8,98.4,45./
      real VSCTS(3)/0.75,21.7,0.75/     ! vector of TOF-short (BOX)
      real VSCTL(3)/0.75,31.9,0.75/     ! vector of TOF-long  (BOX)
      real VEMCL(3)/196.8,98.4,25./     ! vector of EMcal (BOX)
      real VHCMB(3)/24.,95.,1.0/        ! vector of TOF-frame (BOX)
      real VCRBN(3)/24.,95.,0.15/       ! vector of TOF-frame-Carbon (BOX)
      real VALM1(3)/24.3,0.65,2.05/     ! vector of TOF-frame-Aluminium1 (BOX)
      real VALM2(3)/23.5,0.8,3.1/       ! vector of TOF-frame-Aluminium2 (BOX)
      real VALM3(3)/0.8,95.,0.8/        ! vector of TOF-frame-Aluminium3 (BOX)
      real VPMTS(3)/0.,0.95,3.3/        ! vector of PMT (TUBE)
      real VMMTL(3)/1.04,1.09,2.1/      ! vector of u-metal (TUBE)
      real VCABL(3)/24.,90.,0.4/       ! vector of cable (BOX)
C
      REAL FIELDM,TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,UBUF(3)
C
C ---> Integer number of material
      integer tof_mate, lucite_mate, glass_mate, mumetal_mate
      integer cable_mate, pbglass_mate, gas_mate
C
C ---> Plastic (lucite) for cover plates (C5H8O2) ---
      real AP(3) /12.,1.,16./
      real ZP(3) /6.,1.,8./
      real WP(3) /5.,8.,2./
      real DP /1.18/
C
C ---> Borosilicate glass (PMT glass) [Pyrex]
      real AG(4)/16.0,28.086,10.81,22.99/ ! O,Si,B,Na
      real ZG(4)/8.,14.,5.,11./
      real WG(4)/2.01,0.8,.24,.1/  ! 80% SiO2 12% B2O3 5% Na2O
      real DG/2.23/
C
C ---> Cable
      real AC(4)/63.54,12.01,1.01,35.45/  ! Cu,C,H,Cl
      real ZC(4)/29.00,6.00,1.00,17.00/
      real WC(4)/0.393,1.214,2.117,0.311/ ! 39.3% Cu 31.1% C2H3Cl 29.6% C2H4
      real DC/1.936/
C
C ---> Integer number of medium
      integer gas_med, tof_med, lucite_med, glass_med, mumetal_med
      integer cable_med, carbon_med, alumi_med, pbglass_med
C
C ---> Parameter of position
      real YMMTL(6)/92.20,86.40,50.10,41.45,26.80,18.15/  ! Y-axis of metal
      real DXMMT(6)/0.,1.5,0.,0.,1.5,1.5/                 ! DX of metal
C
      integer IXTF, ITOF, JTOF, IXPM, IYPM, KOSU
      real XTOF, YTOF1,YTOF2,YTOF3, ZTOF
      real XMET, YMET, ZMET
      real XPMT, YPMT, ZPMT
```

```
*     ----------------------------------------------------------
***
*
*             Define material
*
csn.. Stores the constants for the material IMATE in the data
csn.. structire JMATE
csn..
csn..    GSMATE(IMATE,CHNAMA,A,Z,DENS,RADL,ABSL,UBUF,NWBUF)
csn..
csn.. IMATE: material number
csn.. CHNAMA: material name
csn.. A: atomic weight
csn.. Z: atomic number
csn.. DENS: density in g/cm*3
csn.. RADL: radiation length in cm
csn.. ABSL: absorption length in cm (ignored)
csn.. UBUF: array of NWBUF additional user parameters
csn.. NWBUF: number of user words in UBUF
csn..
cak..
cak.. Defines mixture or compound IMATE as composed by NMATE
cak.. materials defined via the arrays A,Z and WMAT.
cak.. Mixtures of compounds can also be defined.
cak..
cak..    GSMIXT(IMATE,NAMATE,A,Z,DENS,NLMAT,WMAT*)
cak..
cak.. IMATE: material number
cak.. NAMATE: mixture name
cak.. A: array of atomic weights
cak.. Z: array of atomic numbers
cak.. DENS: density in g/cm*3
cak.. NLMAT: number of elements in the mixture
cak..        >0 WMAT: the proportion by weights
cak..        <0 WMAT: the proportion by number of atoms
cak..
*
C
C ---> define material for Air
C
      gas_mate=600
      CALL GSMATE(601,'He(gas) $',4.00,2.0 ,1.785E-4,
     +            5.284E+5,3.221E+5,ubuf,nwbuf)
C
C ---> define material for TOF
C
      tof_mate=700
      call GSMATE(tof_mate,'BC404 $',6.221,3.373,1.032,
     +            0.424E02,0.820E02,ubuf,nwbuf)
C
c      lucite_mate=tof_mate+1
c      CALL GSMIXT(lucite_mate,'Lucite    $',AP,ZP,DP,-3,WP)
C
      glass_mate=tof_mate+2
      CALL GSMIXT(glass_mate,'Borosilicate$',AG,ZG,DG,-4,WG)
C
      mumetal_mate=tof_mate+3
      CALL GSMATE(mumetal_mate,'Fe    $',55.85,26., 7.87,
     +            1.76,17.1,ubuf,nwbuf)
C
      cable_mate=tof_mate+4
      CALL GSMIXT(cable_mate,'Cable$',AC,ZC,DC,-4,WC)
C
C ---> define material for EMcal
C
```

```
      pbglass_mate=800
      CALL GSMATE(pbglass_mate,'S pb-glass',116.162,47.045,4.010,
     +            0.252E01,0.384E02,ubuf,nwbuf)

***
*
*             Define tracking medium
*
csn.. assoiates a set of tracking parameters to a material, defining a
csn.. so-called tracking medium. The routine stores the parameters of the
csn.. tracking medium ITMED in the data structure JTMED.
csn..
csn..    GSTMED(ITMED,NATMED,NMAT,ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX,
csn..           DEEMAX,EPSIL,STMIN,UBUF,NWBUF)
csn..
csn.. ITMED: tracking medium number
csn.. NATMED: tracking medium name
csn.. NMAT: material number corresponding to ITMED
csn.. ISVOL: sensitivity flag (<=0 not a sensitive volume)
csn..        define 0 or 1
csn.. IFIELD: magnetic field flag
csn.. 0=no magnetic, 1=strongly inhomogeneous,2=inhomogeneous, 3=uniform)
csn.. FIELDM: maximum fiedl value in Kilogauss
csn.. TMAXFD: max angu deviation due to the mag. field permitted in one step
csn.. STEMAX: maximum step permitted (cm)
csn.. DEEMAX: maximum fractional energy loss in one step (0<DEFMAX<=1)
csn.. EPSIL: boundary crossing precision (cm)
csn.. STMIN: minimum value for the maximum step imposed by energy loss,etc (cm)
csn.. UBUF: array of NWBUF additional user parameters
csn.. NWBUF: number of additional user parameters
cak..
cak.. Automatic calculation takes plase in case the input value is negative
cak..
cak.. COMMON/GCTRAK/
cak.. IGAUTO : Automatic computation of DEEMAX,STMIN,TMAXFD,STEMAX
cak..          Flag set to 0 tracking parameters are given by the user
cak..                      1 tracking parameters are calculated by GEANT
*
      IGAUTO = 0
*
      IFIELD=0
      FIELDM=0.
      TMAXFD=20.0
      STEMAX=0.10
      DEEMAX=0.05
      EPSIL =0.01
      STMIN =1.0
*
      STEMAX_cal=1.0
      DEEMAX_cal=0.05
      STMIN_cal =1.0
C
      CALL GSTMED(600,'Vaccume $',16,0,IFIELD,FIELDM,TMAXFD,
     +            -1.,-1.,EPSIL,-1.,UBUF,3)
      CALL GSTMED(601,'Air $',15,0,IFIELD,FIELDM,TMAXFD,
     +            -1.,-1.,EPSIL,-1.,UBUF,3)
      CALL GSTMED(602,'He(gas) $',601,0,IFIELD,FIELDM,TMAXFD,
     +            -1.,-1.,EPSIL,-1.,UBUF,3)
C
      tof_med=700
      CALL GSTMED(tof_med,'Scintillator $',tof_mate,0,IFIELD,FIELDM,
     +            TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,UBUF,3)
C
c      lucite_med=tof_med+1
c      CALL GSTMED(lucite_med,'Light Guide $',mumetal_mate,0,IFIELD,FIELDM,
```

```
c     +            TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,UBUF,3)
C
      glass_med=tof_med+2
      CALL GSTMED(glass_med,'PMT glass $',glass_mate,0,IFIELD,FIELDM,
     +            TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,UBUF,3)
C
      mumetal_med=tof_med+3
      CALL GSTMED(mumetal_med,'Mu-metal $',mumetal_mate,0,IFIELD,FIELDM,
     +            TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,UBUF,3)
C
      cable_med=tof_med+4
      CALL GSTMED(cable_med,'Cable $',cable_mate,0,IFIELD,FIELDM,
     +            TMAXFD,STEMAX,DEEMAX,EPSIL,STMIN,UBUF,3)
C
      carbon_med=tof_med+5
      CALL GSTMED(carbon_med,'Carbon $',6,0,IFIELD,FIELDM,TMAXFD,
     +            STEMAX,DEEMAX,EPSIL,STMIN,UBUF,3)
C
      alumi_med=tof_med+6
      CALL GSTMED(alumi_med,'Aluminium $',9,0,IFIELD,FIELDM,TMAXFD,
     +            STEMAX,DEEMAX,EPSIL,STMIN,UBUF,3)
C
      pbglass_med=800
      CALL GSTMED(pbglass_med,'Pb-glass $',pbglass_mate,0,IFIELD,FIELDM,
     +            TMAXFD,STEMAX_cal,DEEMAX_cal,EPSIL,STMIN_cal,UBUF,3)
C
***
*
*             Define relation between axis
*
csn.. Stores rotaion matrix IROT in the data structure JROTM.
csn..
csn..    CSROTM(IROT,THETA1,PHI1,THETA2,PHI2,THETA3,PHI3)
csn..
csn.. IROT: number of the rotation matrix
csn.. THETA1(2/3): polar angle for the exis x(y/z)
csn.. PHI1(2/3): azimuthal angle for axis x(y/z)
*
      CALL GSROTM(1,90.,0.,90.,90.,0.,0.)
      CALL GSROTM(2,180.,0.,90.,90.,90.,0.)
      CALL GSROTM(3,90.,0.,180.,90.,90.,90.)
**
*
*             Define volumes
*
csn.. Defines a volume with a given name/shape/tracking medium num/shape param
csn..
csn..    GSVOLU(CHNAME,CHSHAP,NMED,PAR,NPAR,IVOLU*)
csn..
csn.. CHNAME: char*4 volume name,  it must be unique
csn.. CHSHAP: char*4 name of one of GEANT shape
csn.. NMED: int tracking medium number for the volume.
csn.. PAR: array containing the shape parameters
csn.. NPAR: number of parameters.
csn.. IVOLU: internal volume number.
*
      if(ISWIT(6) .eq. 1 )then
         gas_med = 601
      elseif( ISWIT(6) .eq. 2 )then
         gas_med = 602
      else
         gas_med = 600
      endif
C
      CALL GSVOLU('MOTH','BOX ',gas_med,VMOTH,3,IVOLU)
```

```
C
      CALL GSVOLU('PSU2','TUBE',gas_med,pVERT,3,IVOLU)
      CALL GSATT('PSU2','SEEN',0)
      CALL GSVOLU('PSU3','TUBE',gas_med,pVERT,3,IVOLU)
      CALL GSATT('PSU3','SEEN',0)
      CALL GSVOLU('PSU4','TUBE',gas_med,pVERT,3,IVOLU)
      CALL GSATT('PSU4','SEEN',0)
      CALL GSVOLU('PSU5','TUBE',gas_med,pVERT,3,IVOLU)
      CALL GSATT('PSU5','SEEN',0)
C
      CALL GSVOLU('VTF1','BOX ',gas_med,pTFPN,3,IVOLU)
      CALL GSATT('VTF1','SEEN',0)
      CALL GSVOLU('VTF2','BOX ',gas_med,pTFPN,3,IVOLU)
      CALL GSATT('VTF2','SEEN',0)
      CALL GSVOLU('VCAL','BOX ',gas_med,pEMCL,3,IVOLU)
      CALL GSATT('VCAL','SEEN',0)
C
      CALL GSVOLU('SCTS','BOX ',tof_med,VSCTS,3,IVOLU)
      CALL GSATT('SCTS','SEEN',1)
      CALL GSATT('SCTS','COLO',2)
      CALL GSVOLU('SCTL','BOX ',tof_med,VSCTL,3,IVOLU)
      CALL GSATT('SCTL','SEEN',1)
      CALL GSATT('SCTL','COLO',2)
C
      CALL GSVOLU('HCMB','BOX ',gas_med,VHCMB,3,IVOLU)
      CALL GSATT('HCMB','SEEN',1)
      CALL GSATT('HCMB','COLO',1)
      CALL GSVOLU('CRBN','BOX ',carbon_med,VCRBN,3,IVOLU)
      CALL GSATT('CRBN','SEEN',1)
      CALL GSATT('CRBN','COLO',1)
      CALL GSVOLU('ALM1','BOX ',alumi_med,VALM1,3,IVOLU)
      CALL GSATT('ALM1','SEEN',1)
      CALL GSATT('ALM1','COLO',6)
      CALL GSVOLU('ALM2','BOX ',alumi_med,VALM2,3,IVOLU)
      CALL GSATT('ALM2','SEEN',1)
      CALL GSATT('ALM2','COLO',6)
      CALL GSVOLU('ALM3','BOX ',alumi_med,VALM3,3,IVOLU)
      CALL GSATT('ALM3','SEEN',1)
      CALL GSATT('ALM3','COLO',6)
C
      CALL GSVOLU('PMT ','TUBE',glass_med,VPMTS,3,IVOLU)
      CALL GSATT('PMT ','SEEN',1)
      CALL GSATT('PMT ','COLO',4)
      CALL GSVOLU('MMTL','TUBE',mumetal_med,VMMTL,3,IVOLU)
      CALL GSATT('MMTL','SEEN',1)
      CALL GSATT('MMTL','COLO',4)
      CALL GSVOLU('CABL','BOX ',cable_med,VCABL,3,IVOLU)
      CALL GSATT('CABL','SEEN',1)
      CALL GSATT('CABL','COLO',5)
C
      CALL GSVOLU('EMCL','BOX ',pbglass_med,VEMCL,3,IVOLU)
      CALL GSATT('EMCL','SEEN',1)
      CALL GSATT('EMCL','COLO',3)
**
*
*            Position volumes
*
csn.. Places a copy of a volume previously defined by a call to GSVOLU
csn.. inside its mother volume CHMOTH.
csn..
csn..    GSPOS(CHNAME,NR,CHMOTH,X,Y,Z,IROT,CHONLY)
csn..
csn.. CHNAME: char*4 name of the volume being positioned
csn.. NR: copy number of the volume CHNAME being positioned
csn.. X/Y/Z: x/y/z position of the volume int the mother reference system
```

```
csn.. IROT: rotation matrix number describing the orientation of the vol
csn..       relative to the coordinate system of the mother
csn.. CHONLY: char*4 flag to indicate whether a point found to be in this
csn..         volume may also be in other volumes which are... (ONLY/MANY)
*
      CALL GSPOS('PSU2',1,'MOTH',0.,0.,0.,1,'ONLY')
      CALL GSPOS('PSU3',1,'PSU2',0.,0.,0.,1,'ONLY')
      CALL GSPOS('PSU4',1,'PSU3',0.,0.,0.,1,'ONLY')
      CALL GSPOS('PSU5',1,'PSU4',0.,0.,0.,1,'ONLY')
*
      if(ISWIT(7) .eq. 3)then
         CALL GSPOS('VCAL',1,'MOTH',75.,0.,0.,2,'ONLY')
      else
         CALL GSPOS('VCAL',1,'MOTH',568.,0.,0.,2,'ONLY')
      endif
*
      CALL GSPOS('EMCL',1,'VCAL',0.,0.,-20.,1,'ONLY')
      CALL GSDVT('PbGx','EMCL',4.1,1,pbglass_med,100)
      CALL GSDVT('PbGy','PbGx',4.1,2,pbglass_med,100)
*
      if((ISWIT(7) .eq. 0).or.(ISWIT(7) .eq. 3))then
C              ! nothing
      elseif(ISWIT(7) .eq. 1)then
         CALL GSPOS('VTF1',1,'MOTH',504.,0.,0.,2,'MANY')
         CALL GSPOS('VTF2',1,'MOTH',504.,0.,0.,2,'MANY')
      else
         CALL GSPOS('VTF1',1,'MOTH',504.,0.,-171.5,2,'MANY')
         CALL GSPOS('VTF2',1,'MOTH',504.,0.,-171.5,2,'MANY')
         CALL GSPOS('VTF1',2,'MOTH',504.,0.,-122.5,2,'MANY')
         CALL GSPOS('VTF2',2,'MOTH',504.,0.,-122.5,2,'MANY')
         CALL GSPOS('VTF1',3,'MOTH',504.,0., -73.5,2,'MANY')
         CALL GSPOS('VTF2',3,'MOTH',504.,0., -73.5,2,'MANY')
         CALL GSPOS('VTF1',4,'MOTH',504.,0., -24.5,2,'MANY')
         CALL GSPOS('VTF2',4,'MOTH',504.,0., -24.5,2,'MANY')
         CALL GSPOS('VTF1',5,'MOTH',504.,0.,  24.5,2,'MANY')
         CALL GSPOS('VTF2',5,'MOTH',504.,0.,  24.5,2,'MANY')
         CALL GSPOS('VTF1',6,'MOTH',504.,0.,  73.5,2,'MANY')
         CALL GSPOS('VTF2',6,'MOTH',504.,0.,  73.5,2,'MANY')
         CALL GSPOS('VTF1',7,'MOTH',504.,0., 122.5,2,'MANY')
         CALL GSPOS('VTF2',7,'MOTH',504.,0., 122.5,2,'MANY')
         CALL GSPOS('VTF1',8,'MOTH',504.,0., 171.5,2,'MANY')
         CALL GSPOS('VTF2',8,'MOTH',504.,0., 171.5,2,'MANY')
      endif
*
      CALL GSPOS('HCMB',1,'VTF1',0.,0., 0.,1,'ONLY')
      CALL GSPOS('CRBN',1,'VTF1',0.,0., 1.15,1,'ONLY')
      CALL GSPOS('CRBN',2,'VTF1',0.,0.,-1.15,1,'ONLY')
      CALL GSPOS('CABL',1,'VTF2',1.5,0,13.,1,'ONLY')
      CALL GSPOS('ALM1',1,'VTF2',0., 95.65,0.75,1,'ONLY')
      CALL GSPOS('ALM1',2,'VTF2',0.,-95.65,0.75,1,'ONLY')
      CALL GSPOS('ALM2',1,'VTF2',1.5, 97.7,11.6,1,'ONLY')
      CALL GSPOS('ALM2',2,'VTF2',1.5,-97.7,11.6,1,'ONLY')
      CALL GSPOS('ALM3',1,'VTF2', 25.,0.,11.6,1,'ONLY')
      CALL GSPOS('ALM3',2,'VTF2',-22.,0.,11.6,1,'ONLY')
*
      do IXTF = 1,16
         XTOF = (IXTF-1)*3.0-(24.0-0.75)+DXMMT(1)
         YTOF1= 69.7
         YTOF2= 11.65
         YTOF3= -56.6
         ZTOF = 2.1
         ITOF = IXTF
         JTOF = IXTF + 32
         CALL GSPOS('SCTS',ITOF,'VTF1',XTOF,YTOF1,ZTOF,1,'ONLY')
         CALL GSPOS('SCTL',ITOF,'VTF1',XTOF,YTOF2,ZTOF,1,'ONLY')
```

```
         CALL GSPOS('SCTL',JTOF,'VTF1',XTOF,YTOF3,ZTOF,1,'ONLY')
         XTOF = 0.- XTOF
         YTOF1= 0.- YTOF1
         YTOF2= 0.- YTOF2
         YTOF3= 0.- YTOF3
         ZTOF = 2.1
         ITOF = IXTF + 16
         JTOF = IXTF + 48
         CALL GSPOS('SCTS',ITOF,'VTF1',XTOF,YTOF1,ZTOF,1,'ONLY')
         CALL GSPOS('SCTL',ITOF,'VTF1',XTOF,YTOF2,ZTOF,1,'ONLY')
         CALL GSPOS('SCTL',JTOF,'VTF1',XTOF,YTOF3,ZTOF,1,'ONLY')
      enddo
*
      do IXPM = 1, 16
         XMET = (IXPM-1)*3.0-(24.0-0.75)+DXMMT(1)
         YMET = YMMTL(1)
         ZMET = 6.6
         XPMT = XMET
         YPMT = YMET
         ZPMT = ZMET + 1.2
         KOSU = IXPM
         CALL GSPOS('MMTL',KOSU,'VTF1',XMET,YMET,ZMET,1,'ONLY')
         CALL GSPOS('PMT ',KOSU,'VTF1',XPMT,YPMT,ZPMT,1,'ONLY')
         XMET = 0.- XMET
         YMET = 0.- YMET
         ZMET = ZMET
         XPMT = XMET
         YPMT = YMET
         ZPMT = ZMET + 1.2
         KOSU = IXPM + 16
         CALL GSPOS('MMTL',KOSU,'VTF1',XMET,YMET,ZMET,1,'ONLY')
         CALL GSPOS('PMT ',KOSU,'VTF1',XPMT,YPMT,ZPMT,1,'ONLY')
      enddo
*
      do 100 IYPM = 2,6
         do 10 IXPM = 1,16
            XMET = (IXPM-1)*3.0-(24.0-0.75)+DXMMT(IYPM)
            YMET = YMMTL(IYPM)
            ZMET = 4.2
            XPMT = XMET
            YPMT = YMET - 1.2*((-1.)**IYPM)
            ZPMT = ZMET
            KOSU = IXPM + 32*(IYPM-1)
            CALL GSPOS('MMTL',KOSU,'VTF1',XMET,YMET,ZMET,3,'ONLY')
            CALL GSPOS('PMT ',KOSU,'VTF1',XPMT,YPMT,ZPMT,3,'ONLY')
            XMET = 0.- XMET
            YMET = 0.- YMET
            ZMET = ZMET
            XPMT = 0.- XPMT
            YPMT = 0.- YPMT
            ZPMT = ZPMT
            KOSU = IXPM + 16 + 32*(IYPM-1)
            CALL GSPOS('MMTL',KOSU,'VTF1',XMET,YMET,ZMET,3,'ONLY')
            CALL GSPOS('PMT ',KOSU,'VTF1',XPMT,YPMT,ZPMT,3,'ONLY')
 10      continue
 100  continue
      CALL GGCLOS
      RETURN
      END
*
*
```

B.2 gxuser.f : ToF

```
***
*     gxuser.f
***
*
      SUBROUTINE UGINIT
*
************************************************************************
*                                                                      *
*             To initialise GEANT3.21 program and read data cards      *
*                                                                      *
************************************************************************
*
      PARAMETER (NWGEAN=9000000,NWPAW=2000000,NWKUIP=30000)
      COMMON/GCBANK/GEANT(NWGEAN)
      COMMON/PAWC/PAW(NWPAW)
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +      ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
C
      COMMON/GCUNIT/LIN,LOUT,NUNITS,LUNITS(5)
      INTEGER LIN,LOUT,NUNITS,LUNITS
      COMMON/GCMAIL/CHMAIL
      CHARACTER*132 CHMAIL
C
*KEEP,GCCUTS.
      COMMON/GCCUTS/CUTGAM,CUTELE,CUTNEU,CUTHAD,CUTMUO,BCUTE,BCUTM
     +             ,DCUTE ,DCUTM ,PPCUTM,TOFMAX,GCUTS(5)
C
*KEEP,GCPHYS.
      COMMON/GCPHYS/IPAIR,SPAIR,SLPAIR,ZINTPA,STEPPA
     +             ,ICOMP,SCOMP,SLCOMP,ZINTCO,STEPCO
     +             ,IPHOT,SPHOT,SLPHOT,ZINTPH,STEPPH
     +             ,IPFIS,SPFIS,SLPFIS,ZINTPF,STEPPF
     +             ,IDRAY,SDRAY,SLDRAY,ZINTDR,STEPDR
     +             ,IANNI,SANNI,SLANNI,ZINTAN,STEPAN
     +             ,IBREM,SBREM,SLBREM,ZINTBR,STEPBR
     +             ,IHADR,SHADR,SLHADR,ZINTHA,STEPHA
     +             ,IMUNU,SMUNU,SLMUNU,ZINTMU,STEPMU
     +             ,IDCAY,SDCAY,SLIFE ,SUMLIF,DPHYS1
     +             ,ILOSS,SLOSS,SOLOSS,STLOSS,DPHYS2
     +             ,IMULS,SMULS,SOMULS,STMULS,DPHYS3
     +             ,IRAYL,SRAYL,SLRAYL,ZINTRA,STEPRA
      COMMON/GCPHLT/ILABS,SLABS,SLLABS,ZINTLA,STEPLA
     +             ,ISYNC
     +             ,ISTRA
C
*KEEP,GCFLAG.
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEND.
cn..          Define of HBOOK word
      integer   istat
      character emtags(6)*5
      data      emtags/'I','E','x','y','r','sita'/
      character caltags(40)*6
      data caltags/'I','N','E0','Eless','Etot',
     +     'E1','IP1','x1','y1','i1','j1','E2','IP2','x2','y2','i2',
     +     'j2','E3','IP3','x3','y3','i3','j3','Nem','Eem1','iem1',
```

```
     +     'jem1','Ntof','Etof1','IPtof1','itof1','jtof1',
     +     'Etof2','IPtof2','itof2','jtof2',
     +     'Etof3','IPtof3','itof3','jtof3'/
C
***   -----------------------------------------------------------------
*
*
*             Initialize GEANT
*
*
      CALL GINIT           ! Initialize GEANT system
*
*
*             Input tracking thresholds
*
*
      CUTGAM=0.001 ! threshold for gammas transport
      CUTELE=0.001 ! threshold for electrons transport
      CUTHAD=0.01  ! threshold for hadrons   transport
      CUTNEU=0.01  ! threshold for neutral hadrons transport
      CUTMUO=0.01  ! threshold for muons transport
      BCUTE=CUTGAM ! threshold for photons produced by electron Brems.
      BCUTM=CUTGAM ! threshold for photons produced by muon Brems.
      DCUTE=CUTELE ! threshold for electron produced by electron&delta-rays
      DCUTM=CUTELE ! threshold for electron produced by muon delta-rays
      PPCUTM=0.002 ! threshold for e+e- pair production by muons
C      TOFMAX=1010  ! threshold on time of flight counted
C                                         from primary interaction time
*
*
*             Input physics processes
*
*
      IPAIR=1 ! pair production with generation of electron/positron
      ICOMP=1 ! compton scattering with generation of electron
      IPHOT=1 ! photo-electric effect with generation of electron
      IPFIS=0 ! photo-fission effect with generation of secondaries
      IDRAY=1 ! delta ray production with generation of electron
      IANNI=1 ! positron annihilation with generation of photons
      IBREM=1 ! bremsstrahlung with generation of gamma
      IHADR=1 ! hadronic interactions with generation of secondaries
      IMUNU=1 ! muon nuclear interaction with generation of secondaries
      IDCAY=1 ! decay in flight with generation of secondaries
      ILOSS=2 ! continuous energy loss without generation of delta ray
C                     and Landau-Vaviliov-Gauss fluctuations
      IMULS=1 ! multiple scattering according to Moli'ere theory
      IRAYL=0 ! Rayleigh scattering
      ILABS=1 ! absorbtion of Cerenkov photons with posssible detection
      ISYNC=0 ! To control synchrotron radiation
      ISTRA=0 ! To control energy loss fluctuation model
**
      write(6,*)' '
      write(6,*)'      Type in !'
      write(6,*)' '
*
      write(6,*)' '
      write(6,*)'Select of GAS '
      write(6,*)'      Vaccume ===>0'
      write(6,*)'      Air     ===>1'
      write(6,*)'      He(gas) ===>2'
      read(5,*) ISWIT(6)
*
      write(6,*)' '
      write(6,*)'Select of ToF status '
      write(6,*)'    non-panel ===>0'
```

```
      write(6,*)'      1-panel ===>1'
      write(6,*)'      8-panel ===>2'
      write(6,*)'    non-panel &'
      write(6,*)'    EMcal 50cm===>3'
      read(5,*) ISWIT(7)
*
*
*             Initialize GEANT/ZBOOK data structures
*
*
      CALL GZINIT          ! Initialize mother banks
*
*
*             Initialize graphics package
*
*
      CALL GDINIT          ! Initialize drawing packag
*
      CALL GPART           ! Store the standard particle constants
      CALL GMATE           ! Initialize material constants
*
      CALL UGEOM           ! Define user geometry
      CALL GPHYSI          ! Initialize GEANT physics processes
*
      write(6,*)' ---------- USER Selection ----------'
      if(ISWIT(6) .eq. 1)then
         write(6,*)' Air'
      elseif(ISWIT(6) .eq. 2)then
         write(6,*)' Helium gas'
      else
         write(6,*) ' Vaccume'
      endif
      if(ISWIT(7) .eq. 0)then
         write(6,*)' This geom. is EMcal only'
      elseif(ISWIT(7) .eq. 1)then
         write(6,*)' This geom. is ToF 1-panel and EMcal small'
      elseif(ISWIT(7) .eq. 3)then
         write(6,*)' This geom. is EMcal only (near)'
      else
         write(6,*)' This geom. is ToF 8-panel and EMcal '
         write(6,*)'      Full Scale !!'
      endif
      write(6,*)' '
      write(6,*)' ------------------------------------'
      write(6,*)' '
*
**
*
*             Open HBOOK
*
      CALL HROPEN(1,'PHENIX','phenix.nt','n',1024,istat)
**
*
*             Booking Histogram & Ntuple file
*
      CALL HBOOKN(30,'Energy of gamma',6,'PHENIX',5000,emtags)
C
      CALL HBOOKN(60,'Energy(Pt= 250MeV)',40,'PHENIX',5000,caltags)
      CALL HBOOKN(70,'Energy(Pt= 500MeV)',40,'PHENIX',5000,caltags)
      CALL HBOOKN(80,'Energy(Pt=1000MeV)',40,'PHENIX',5000,caltags)
      CALL HBOOKN(90,'Energy(Pt=1500MeV)',40,'PHENIX',5000,caltags)
C
      if((ISWIT(7) .ne. 0) .and. (ISWIT(7) .ne. 3))then
         CALL HBOOK1(200,'(ToF) Energy loss(MeV)',100,0.,20.,0.)
         CALL HBOOK2(250,'ToF Pair pro. and Compton scat.(P=250MeV/c)',
```

```
     +        100,-200.,200.,50,-100.,100.,0.)
         CALL HBOOK2(260,'ToF Pair pro. and Compton scat.(P=500MeV/c)',
     +        100,-200.,200.,50,-100.,100.,0.)
         CALL HBOOK2(270,'ToF Pair pro. and Compton scat.(P=1000MeV/c)',
     +        100,-200.,200.,50,-100.,100.,0.)
         CALL HBOOK2(280,'ToF Pair pro. and Compton scat.(P=1500MeV/c)',
     +        100,-200.,200.,50,-100.,100.,0.)
      endif
C
      CALL HBOOK1(300,'EMcal (P=250MeV/c)',100,0.,400,0.)
      CALL HBOOK1(310,'EMcal (P=250MeV/c)',200,0.,400,0.)
      CALL HBOOK1(320,'EMcal (P=500MeV/c)',100,0.,800,0.)
      CALL HBOOK1(330,'EMcal (P=500MeV/c)',200,0.,800,0.)
      CALL HBOOK1(340,'EMcal (P=1000MeV/c)',100,0.,1600,0.)
      CALL HBOOK1(350,'EMcal (P=1000MeV/c)',200,0.,1600,0.)
      CALL HBOOK1(360,'EMcal (P=1500MeV/c)',100,0.,2400,0.)
      CALL HBOOK1(370,'EMcal (P=1500MeV/c)',200,0.,2400,0.)
*
      END
*
***
*
      SUBROUTINE UGLAST
*
************************************************************************
*                                                                      *
*            Termination routine to print histograms and statistics   *
*                                                                      *
************************************************************************
*
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEND.
cn..        Define of HBOOK word
      integer icycle
C
*     ---------------------------------------------------------------------
*
      IF(JULAST.NE.0)THEN
         CALL CSJCAL(JULAST,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
      CALL GLAST
**
*
*
*           Close HIGZ
*
*
      CALL IGEND
**
*
*           Close HBOOK
*
      CALL HROUT(0,icycle,' ')
      CALL HREND('PHENIX')
**
*
      RETURN
 99   END
*
***
*
      SUBROUTINE GUKINE
```

```
*
************************************************************************
*                                                                      *
*             Generates Kinematics for primary tracks                  *
*                                                                      *
************************************************************************
*
*KEEP,GCFLAG.
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +     ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +     ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
C
*KEEP,GCONST.
      COMMON/GCONST/PI,TWOPI ,PIBY2,DEGRAD,RADDEG,CLIGHT ,BIG,EMASS
      COMMON/GCONSX/EMMU,PMASS,AVO
C
*KEEP,GCSCAN.
      PARAMETER (MSLIST=32,MAXMDT=3)
      COMMON/GCSCAN/SCANFL,NPHI,PHIMIN,PHIMAX,NTETA,TETMIN,TETMAX,
     +              MODTET,IPHIMI,IPHIMA,IPHI1,IPHIL,NSLMAX,
     +              NSLIST,ISLIST(MSLIST),VSCAN(3),FACTX0,FACTL,
     +              FACTR,IPHI,ITETA,ISCUR,SX0,SABS,TETMID(MAXMDT),
     +              TETMAD(MAXMDT)
      LOGICAL SCANFL
      COMMON/GCSCAC/SFIN,SFOUT
      CHARACTER*80 SFIN,SFOUT
C
*KEEP,GCOMIS.
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48)
      common/uENERGY/Etot,ENEcal(10),xcal(10),ycal(10),ical(10),
     +     jcal(10),Ncal,Ntof,ENEtof(10),itof(10),jtof(10),IP(10)
C
*KEND.
C
      REAL      UBUF,PLAB(3),VERT(3)
      INTEGER   NVTX,NT,IPART
C
      real      pt, p4(4), Init_pt, y_min, y_max, y
      integer   Iseed, Init_PART,In_part,Iptswit, IDtof, IDcal
      real      Nckov,Nck(96,48)
      integer   inum,ick, jck
      character eventname*20
      real      theta, phi, p_theta, p_phi
C
*     -----------------------------------------------------------------
*
      IF(JUKINE.NE.0)THEN
         CALL CSJCAL(JUKINE,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
*
*             Define type, momentum and vertex of particle
*
*
      save Init_PART,Init_num,Init_pt,Iptswit
```

```
*
      IF(IEVENT.EQ.1)THEN
         WRITE(6,*)' '
         WRITE(6,*)'      Type in !'
*
         WRITE(6,*)' '
         WRITE(6,*)'Initial particle number ..integer'
         WRITE(6,*)' <example>  1,Gamma   3,Electron   6,Muon-'
         WRITE(6,*)'            8,Pion+   9,Pion-     11,Kaon+'
         WRITE(6,*)'           12,Kaon-  14,Proton     etc.   '
         READ(5,*)Init_PART
         In_part = Init_PART
*
         WRITE(6,*)' '
         WRITE(6,*)'Number of paeticles  ..integer'
         READ(5,*)Init_num
      ENDIF
*
      VERT(1) = 0.
      VERT(2) = 0.
      VERT(3) = 0.
      CALL GSVERT(VERT,0,0,UBUF,1,NVTX)
*
      if(IEVENT.EQ.1)then
         Init_pt = 0.25
      endif
      if(IEVENT.EQ.10001)then
         Init_pt = 0.50
      endif
      if(IEVENT.EQ.20001)then
         Init_pt = 1.0
      endif
      if(IEVENT.EQ.30001)then
         Init_pt = 1.5
      endif
*
      Iseed = NRNDM(2)+1812
      do inum = 1,Init_num
         thita = 0.400          !  = arctan(200/500)
         p_theta = 3.1415926/2 - thita + 2.*thita*ran(Iseed)
         phi = 0.200            !  = arctan(100/500)
         p_phi = -phi + 2.*phi*ran(Iseed)
*
         pt = Init_pt
         PLAB(1) = pt*sin(p_theta)*cos(p_phi)
         PLAB(2) = pt*sin(p_theta)*sin(p_phi)
         PLAB(3) = pt*cos(p_theta)
         CALL GSKINE(PLAB,Init_PART,NVTX,UBUF,1,NT)
      enddo
*
      ELOSStof = 0.
      Nckov = 0.
      do ick = 1,96
         do jck = 1,48
            Nck(ick,jck) = 0.
            if(jck .le. 8)then
               ELOSS(jck,ick) = 0.
            endif
         enddo
      enddo
C
      Etot = 0.
      Ncal = 0
      Ntof = 0
      do i = 1,10
```

```
         ENEcal(i) = 0.
         xcal(i) = 0.
         ycal(i) = 0.
         ical(i) = 0
         jcal(i) = 0
         ENEtof(i) = 0.
         itof(i) = 0
         jtof(i) = 0
         IP(i) = 0
      enddo
*
*
*
*             Kinematic debug (controled by ISWIT(1))
*
*
      IF((IDEBUG.EQ.1).AND.(ISWIT(1).EQ.1))THEN
         CALL GPRINT(’VERT’,0)
         CALL GPRINT(’KINE’,0)
      ENDIF
*
*
 99   END
*
***
*
      SUBROUTINE GUTREV
C.
C.    ******************************************************************
C.    *                                                                *
C.    *       User routine to control tracking of one event            *
C.    *       Called by GRUN                                           *
C.    *                                                                *
C.    ******************************************************************
C.
C.
      CALL GTREVE
      RETURN
      END
*
***
*
      SUBROUTINE GUTRAK
C.
C.    ******************************************************************
C.    *                                                                *
C.    *       User routine to control tracking of one track            *
C.    *                                                                *
C.    *    ==>Called by : GTREVE                                       *
C.    *                                                                *
C.    ******************************************************************
C.
*KEEP,GCFLAG
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +     ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +     ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
      CHARACTER*4 NAPART
C
*KEEP,GCTRAK.
      PARAMETER (MAXMEC=30)
```

```
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(MAXMEC)
     +     ,NAMEC(MAXMEC),NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG
     +     ,STEP  ,SNEXT ,SFIELD,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL
     +     ,ISTOP ,IGAUTO,IEKBIN, ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN
     +     ,NLVSAV,ISTORY
      CHARACTER*4 NAMEC,NAMEC1
      PARAMETER (MAXME1=30)
      COMMON/GCTPOL/POLAR(3), NAMEC1(MAXME1)
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48)
C
      real Init_ENERGY
C.
C.    ------------------------------------------------------------------
C.
      if((VECT(1) .le. 0.1).and.(VECT(2) .le. 0.1).and.
     +     (VECT(3) .le. 0.1).and.(IPART .eq. In_part))then
         Init_ENERGY = GETOT*1.0E+3   ! GETOT:(GeV)--> Init_ENERGY:(MeV)
      endif
*
      CALL GTRACK
      RETURN
      END
*
***
*
      SUBROUTINE GUFLD(X,F)
C.
C.    ******************************************************************
C.    *                                                                *
C.    *       User routine to compute the magnetic field F             *
C.    *       at space point X                                         *
C.    *       Called by GHELIX,GRKUTA                                  *
C.    *                                                                *
C.    ******************************************************************
C.
*KEEP,GCVOLU.
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     +    LVOLUM(15),LINDEX(15),INFROM,NLEVMX,NLDEV(15),LINMX(15),
     +    GTRAN(3,15),GRMAT(10,15),GONLY(15),GLX(3)
      CHARACTER*4 NAMES
C
      DIMENSION X(3),F(3)
C.
C.    ------------------------------------------------------------------
*
      F(1)=0.
      F(2)=0.
      F(3)=0.
C
      RETURN
      END
*
***
*
      SUBROUTINE GUSTEP
*
************************************************************************
*                                                                      *
*            User routine called at the end of each tracking step     *
*            MEC   is the mechanism origin of the step                *
*            INWVOL is different from 0 when the track has reached    *
*                   a volume boundary                                 *
*            ISTOP is different from 0 if the track has stopped       *
```

```
*                                                                        *
*             Called by GTRACK                                           *
*                                                                        *
*************************************************************************
*
*KEEP,GCFLAG
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +     ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +     ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
      CHARACTER*4 NAPART
C
*KEEP,GCKING.
      INTEGER MXGKIN
      PARAMETER (MXGKIN=100)
      COMMON/GCKING/KCASE,NGKINE,GKIN(5,MXGKIN),
     +                           TOFD(MXGKIN),IFLGK(MXGKIN)
      INTEGER       KCASE,NGKINE ,IFLGK,MXPHOT,NGPHOT
      REAL          GKIN,TOFD,XPHOT
C
      PARAMETER (MXPHOT=800)
      COMMON/GCKIN2/NGPHOT,XPHOT(11,MXPHOT)
C
      COMMON/GCKIN3/GPOS(3,MXGKIN)
      REAL          GPOS
C
*KEEP,GCMATE.
      COMMON/GCMATE/NMAT,NAMATE(5),A,Z,DENS,RADL,ABSL
C
*KEEP,GCTMED.
      COMMON/GCTMED/NUMED,NATMED(5),ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX
     +     ,DEEMAX,EPSIL,STMIN,CFIELD,CMULS,IUPD,ISTPAR,NUMOLD
      COMMON/GCTLIT/THRIND,PMIN,DP,DNDL,JMIN,ITCKOV,IMCKOV,NPCKOV
C
*KEEP,GCTRAK.
      PARAMETER (MAXMEC=30)
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(MAXMEC)
     +     ,NAMEC(MAXMEC),NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG
     +     ,STEP  ,SNEXT ,SFIELD,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL
     +     ,ISTOP ,IGAUTO,IEKBIN, ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN
     +     ,NLVSAV,ISTORY
      CHARACTER*4 NAMEC
      PARAMETER (MAXME1=30)
      COMMON/GCTPOL/POLAR(3), NAMEC1(MAXME1)
C
*KEEP,GCVOLU.
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     +     LVOLUM(15),LINDEX(15),INFROM,NLEVMX,NLDEV(15),LINMX(15),
     +     GTRAN(3,15),GRMAT(10,15),GONLY(15),GLX(3)
      CHARACTER*4 NAMES
C
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEEP,GCONST.
      COMMON/GCONST/PI,TWOPI ,PIBY2,DEGRAD,RADDEG,CLIGHT ,BIG,EMASS
      COMMON/GCONSX/EMMU,PMASS,AVO
C
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48)
      common/uENERGY/Etot,ENEcal(10),xcal(10),ycal(10),ical(10),
```

```
     +     jcal(10),Ncal,Ntof,ENEtof(10),itof(10),jtof(10),IP(10)
C
*KEND.
C
      real      TOFobs, ELOSSobs, Npe, beta
      real      ELOSStof, ELOSS(8,96), Nckov, Nck(96,48)
      integer   ick, jck, itf, jtf, Ncal/0/
      real      Etot/0./,ENEcal(10),xcal(10),ycal(10)
      integer   ical(10),jcal(10),IP(10)
*     -----------------------------------------------------------------
*
      IF(JUSTEP.NE.0)THEN
         CALL CSJCAL(JUSTEP,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
*
*             Print the tracking and physics
*
*
      IF(ISWIT(1).EQ.1)THEN
         CALL GPCXYZ
      ENDIF
*
*
*             Save in the stack particle generated during the current step
*
*
      IF(NGKINE.NE.0)THEN
         CALL GSKING(0)
      ENDIF
*
*
*             Stores current space point into JXYZ
*
*
      CALL GSXYZ
**
*
*             Filling Physics parameter
*
      if((ISWIT(7) .ne. 0) .and. (ISWIT(7) .ne. 3))then
         if((NAMES(3) .eq. 'SCTS').or.(NAMES(3) .eq. 'SCTL'))then
            TOFobs = TOFG * 1.0E+9 ! TOFG:(sec)  --> TOFobs:(nsec)
            ELOSSobs = DESTEP * 1.0E+3 ! DESTEP:(GeV)--> ELOSSobs:(MeV)
            ELOSStof = ELOSStof + ELOSSobs
            if(NAMES(3) .eq. 'SCTS')then
               itf = NUMBER(2)
               if((NUMBER(3) .ge. 1).and.(NUMBER(3) .le. 16))then
                  jtf = NUMBER(3)
               else
                  jtf = 113 - NUMBER(3)
               endif
            elseif(NAMES(3) .eq. 'SCTL')then
               itf = NUMBER(2)
               if((NUMBER(3).ge. 1).and.(NUMBER(3).le.16))then
                  jtf = 32 + NUMBER(3)
               elseif((NUMBER(3).ge.17).and.(NUMBER(3).le.32))then
                  jtf = 81 - NUMBER(3)
               elseif((NUMBER(3).ge.33).and.(NUMBER(3).le.48))then
                  jtf = 32 + NUMBER(3)
               elseif((NUMBER(3).ge.49).and.(NUMBER(3).le.64))then
                  jtf = 81 - NUMBER(3)
               endif
            endif
```

```
            ELOSS(itf,jtf) = ELOSS(itf,jtf) + ELOSSobs
         endif
         if((NAMES(3) .eq. 'SCTS').or.(NAMES(3) .eq. 'SCTL').or.
     +        (NAMES(3) .eq. 'PMT ').or.(NAMES(3) .eq. 'MMTL').or.
     +        (NAMES(3) .eq. 'CRBN').or.(NAMES(3) .eq. 'HCMB').or.
     +        (NAMES(3) .eq. 'ALM1').or.(NAMES(3) .eq. 'ALM2').or.
     +        (NAMES(3) .eq. 'ALM3').or.(NAMES(3) .eq. 'CABL'))then
            if((LMEC(NMEC) .ge. 6).and.(LMEC(NMEC) .le. 8))then
               if((pt .ge. 0.240) .and. (pt .lt. 0.260))then
                  CALL HFILL(250,VECT(3),VECT(2),1.)
               endif
               if((pt .ge. 0.490) .and. (pt .lt. 0.510))then
                  CALL HFILL(260,VECT(3),VECT(2),1.)
               endif
               if((pt .ge. 0.990) .and. (pt .lt. 1.010))then
                  CALL HFILL(270,VECT(3),VECT(2),1.)
               endif
               if((pt .ge. 1.490) .and. (pt .lt. 1.510))then
                  CALL HFILL(280,VECT(3),VECT(2),1.)
               endif
            endif
         endif
      endif
      if(((NAMES(3).eq.'SCTS').or.(NAMES(3).eq.'SCTL')).and.
     +     (Ncal.lt.10).and.(INWVOL.eq.1).and.
     +     (itf.lt.10).and.(jtf.lt.100))then
         Etot = Etot + GETOT*1.0E3
         Ncal = Ncal + 1
         ENEcal(Ncal) = GETOT*1.0E3
         IP(Ncal)   = IPART
         xcal(Ncal) = VECT(3)
         ycal(Ncal) = VECT(2)
         ical(Ncal) = itf
         jcal(Ncal) = jtf
      endif
      if((NAMES(5).eq.'PbGy').and.(CHARGE.ne.0).and.(VECT(7).ne.0.))then
         beta = VECT(7)/GETOT   ! VECT(7):(GeV/c) GETOT:(GeV)
         Npe = STEP*370.*1.0*(1. - 1./(1.5*beta)**2) !1.5=kusseturitu
         if(1./(beta*1.5) .lt. 1.)then
            Nckov = Nckov + Npe
            ick = NUMBER(4)
            jck = NUMBER(5)
            Nck(ick,jck)=Nck(ick,jck) + Npe
         endif
      endif
C
      if(INWVOL .eq. 2)then
         NAMES(3)='    '
         NAMES(4)='    '
         NAMES(5)='    '
      endif
*
  99  CONTINUE
      END
*
*
***
*
      SUBROUTINE GUOUT
*
************************************************************************
*                                                                      *
*             User routine called at the end of each event.            *
*                                                                      *
************************************************************************
```

```
*
*KEEP,GCFLAG.
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEEP,GCVOLU.
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     +     LVOLUM(15),LINDEX(15),INFROM,NLEVMX,NLDEV(15),LINMX(15),
     +     GTRAN(3,15),GRMAT(10,15),GONLY(15),GLX(3)
      CHARACTER*4 NAMES
C
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48)
      common/uENERGY/Etot,ENEcal(10),xcal(10),ycal(10),ical(10),
     +     jcal(10),Ncal,Ntof,ENEtof(10),itof(10),jtof(10),IP(10)
C
*KEND.
C
      real      range_EM,event_em(6),event_cal(40)
      integer   level_cal(10)/1,2,3,4,5,6,7,8,9,10/
      integer   level_em(10)/1,2,3,4,5,6,7,8,9,10/
      integer   level_tof(10)/1,2,3,4,5,6,7,8,9,10/
      real      ENEtof(10)
      integer   Ntof,itof(10),jtof(10),IPtof(10),temp
      real      pt,Init_ENERGY,ELOSStof,ELOSS(8,96),Nckov,Nck(96,48),Npi
      integer   icycle, itf, jtf, ick, jck, i, j, maxN, NN
      parameter (maxN = 1000)
      real      Nem(maxN),E_em(maxN),Rem(maxN),sita_em(maxN)
      integer   xem(maxN),yem(maxN)
      real      E_pi,Pt_pi,phi_pi,M_pi
*     ------------------------------------------------------------------
*
      IF(JUOUT.NE.0)THEN
         CALL CSJCAL(JUOUT,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
      CALL ICLRWK(0,0)
**
*
*             Filling Histogram file
*
C
      do i = 1,10
         IPtof(i) = 0
      enddo
      do itf = 1,8
         do jtf = 1,96
            if(ELOSS(itf,jtf) .ge. 0.1)then
               Ntof = Ntof + 1
               ENEtof(Ntof) = ELOSS(itf,jtf)
               itof(Ntof) = itf
               jtof(Ntof) = jtf
               do i = 1,10
                  if((itf .eq. ical(i)).and.(jtf .eq. jcal(i)))then
                     IPtof(Ntof) = IP(i)
                  endif
               enddo
               CALL HFILL(200,ELOSS(itf,jtf),0.,1.)
            endif
```

```
         enddo
      enddo
C
      if(ISWIT(7) .eq. 3)then
         range_EM = 60.
      else
         range_EM = 530.
      endif
C
      NN = 0
      do ick = 2,95
         do jck = 2,47
            if((Nck(ick,jck) .ne. 0.).and.
     +           (Nck(ick,jck) .gt. Nck(ick-1,jck-1)).and.
     +           (Nck(ick,jck) .gt. Nck(ick-1,jck)).and.
     +           (Nck(ick,jck) .gt. Nck(ick-1,jck+1)).and.
     +           (Nck(ick,jck) .gt. Nck(ick,jck-1)).and.
     +           (Nck(ick,jck) .gt. Nck(ick,jck+1)).and.
     +           (Nck(ick,jck) .gt. Nck(ick+1,jck-1)).and.
     +           (Nck(ick,jck) .gt. Nck(ick+1,jck)).and.
     +           (Nck(ick,jck) .gt. Nck(ick+1,jck+1)))then
               NN = NN + 1
               xem(NN) = ick
               yem(NN) = jck
               Rem(NN) = 4.1*sqrt((ick-48.5)**2 + (jck-24.5)**2)
               sita_em(NN) = atan(Rem(NN)/range_EM)*180./3.1415926
            endif
         enddo
      enddo
C
      if(NN .ge. 1)then
         do i = 1,NN
            Nem(i) = Nck(xem(i)-1,yem(i)-1)+Nck(xem(i)-1,yem(i))
     +           +Nck(xem(i)-1,yem(i)+1)+Nck(xem(i),yem(i)-1)
     +           +Nck(xem(i),yem(i))+Nck(xem(i),yem(i)+1)
     +           +Nck(xem(i)+1,yem(i)-1)+Nck(xem(i)+1,yem(i))
     +           +Nck(xem(i)+1,yem(i)+1)
C
            E_em(i) = 5.995E-2*Nem(i) + 0.606   ! MeV : cut= 1MeV
C
            event_em(1) = IEVENT
            event_em(2) = E_em(i)
            event_em(3) = xem(i)
            event_em(4) = yem(i)
            event_em(5) = Rem(i)
            event_em(6) = sita_em(i)
            CALL HFN(30,event_em)
C
            if(E_em(i) .gt. 20.)then
               if((pt .ge. 0.240) .and. (pt .lt. 0.260))then
                  CALL HFILL(300,E_em(i),0.,1.)
                  CALL HFILL(310,E_em(i),0.,1.)
               endif
               if((pt .ge. 0.490) .and. (pt .lt. 0.510))then
                  CALL HFILL(320,E_em(i),0.,1.)
                  CALL HFILL(330,E_em(i),0.,1.)
               endif
               if((pt .ge. 0.990) .and. (pt .lt. 1.010))then
                  CALL HFILL(340,E_em(i),0.,1.)
                  CALL HFILL(350,E_em(i),0.,1.)
               endif
               if((pt .ge. 1.490) .and. (pt .lt. 1.510))then
                  CALL HFILL(360,E_em(i),0.,1.)
                  CALL HFILL(370,E_em(i),0.,1.)
               endif
```

```
               CALL HFILL(300,E_em(i),0.,1.)
               CALL HFILL(310,sita_em(i),0.,1.)
            endif
         enddo
      endif
      do i = NN+1,10
         E_em(i) = 0.
         xem(i) = 0.
         yem(i) = 0.
      enddo
C
      do i = 1,9
         do j = i+1,10
            if(ENEcal(level_cal(i)) .lt. ENEcal(level_cal(j)))then
               temp = level_cal(i)
               level_cal(i) = level_cal(j)
               level_cal(j) = temp
            endif
            if(E_em(level_em(i)) .lt. E_em(level_em(j)))then
               temp = level_em(i)
               level_em(i) = level_em(j)
               level_em(j) = temp
            endif
            if(ENEtof(level_tof(i)) .lt. ENEtof(level_tof(j)))then
               temp = level_tof(i)
               level_tof(i) = level_tof(j)
               level_tof(j) = temp
            endif
         enddo
      enddo
C
      write(6,*) IEVENT,Init_ENERGY,'(MeV) N_ToF ',Ntof,' N_EMcal',Ncal
C
      event_cal(1) = IEVENT
      event_cal(2) = Ncal
      event_cal(3) = Init_ENERGY
      if((Ncal .eq. 0.).or.(Init_ENERGY - Etot .lt. 0.))then
         event_cal(4) = 0.
      else
         event_cal(4) = Init_ENERGY - Etot
      endif
      event_cal(5) = Etot
      event_cal(6) = ENEcal(level_cal(1))
      event_cal(7) = IP(level_cal(1))
      event_cal(8) = xcal(level_cal(1))
      event_cal(9) = ycal(level_cal(1))
      event_cal(10) = ical(level_cal(1))
      event_cal(11) = jcal(level_cal(1))
      event_cal(12) = ENEcal(level_cal(2))
      event_cal(13) = IP(level_cal(2))
      event_cal(14) = xcal(level_cal(2))
      event_cal(15) = ycal(level_cal(2))
      event_cal(16) = ical(level_cal(2))
      event_cal(17) = jcal(level_cal(2))
      event_cal(18) = ENEcal(level_cal(3))
      event_cal(19) = IP(level_cal(3))
      event_cal(20) = xcal(level_cal(3))
      event_cal(21) = ycal(level_cal(3))
      event_cal(22) = ical(level_cal(3))
      event_cal(23) = jcal(level_cal(3))
      event_cal(24) = NN
      event_cal(25) = E_em(level_em(1))
      event_cal(26) = xem(level_em(1))
      event_cal(27) = yem(level_em(1))
      event_cal(28) = Ntof
```

```
      event_cal(29) = ENEtof(level_tof(1))
      event_cal(30) = IPtof(level_tof(1))
      event_cal(31) = itof(level_tof(1))
      event_cal(32) = jtof(level_tof(1))
      event_cal(33) = ENEtof(level_tof(2))
      event_cal(34) = IPtof(level_tof(2))
      event_cal(35) = itof(level_tof(2))
      event_cal(36) = jtof(level_tof(2))
      event_cal(37) = ENEtof(level_tof(3))
      event_cal(38) = IPtof(level_tof(3))
      event_cal(39) = itof(level_tof(3))
      event_cal(40) = jtof(level_tof(3))

C
      if((pt .ge. 0.240) .and. (pt .lt. 0.260))then
         CALL HFN(60,event_cal)
      elseif((pt .ge. 0.490) .and. (pt .lt. 0.510))then
         CALL HFN(70,event_cal)
      elseif((pt .ge. 0.990) .and. (pt .lt. 1.010))then
         CALL HFN(80,event_cal)
      elseif((pt .ge. 1.490) .and. (pt .lt. 1.510))then
         CALL HFN(90,event_cal)
      endif
**
*
*             Draw track of particle and detector
*
*
      if( ISWIT(5) .eq. 0 )then
         if( ISWIT(2) .eq. 2 )then
            CALL GDRAW('MOTH',0.,0.,0.,0.5,10.,0.030,0.030)
         elseif( ISWIT(2) .eq. 3)then
            CALL GDRAW('MOTH',90.,90.,270.,0.5,10.,0.030,0.030)
         elseif( ISWIT(2) .eq. 4)then
            CALL GDRAW('MOTH',90.,0.,0.,10.,10.,0.030,0.030)
         elseif( ISWIT(2) .eq. 5)then
            CALL GDRAW('MOTH',90.,0.,0.,10.,10.,0.050,0.050)
         else
            CALL GDRAW('MOTH',60.,120.,0.,3.,5.,0.030,0.030)
            CALL GDMAN(18.,4.0)
         endif
*
         CALL GDHEAD(101,'TOF and EM-cal$',.5)
         CALL GDXYZ(0)
         CALL GDAXIS(0.,0.,0.,100.)
         CALL GDPART(0,11,0.25)
         CALL GDSCAL(10.,2.)
      endif
*
      RETURN
 99      END
*
*
```

B.3 gxuser.f : PHENIX

```
***
*     gxuser.f
***
*
      SUBROUTINE UGINIT
*
************************************************************************
*                                                                      *
*             To initialise GEANT3.21 program and read data cards      *
*                                                                      *
************************************************************************
*
      PARAMETER (NWGEAN=9000000,NWPAW=2000000,NWKUIP=30000)
      COMMON/GCBANK/GEANT(NWGEAN)
      COMMON/PAWC/PAW(NWPAW)
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +      ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
C
      COMMON/GCUNIT/LIN,LOUT,NUNITS,LUNITS(5)
      INTEGER LIN,LOUT,NUNITS,LUNITS
      COMMON/GCMAIL/CHMAIL
      CHARACTER*132 CHMAIL
C
*KEEP,GCCUTS.
      COMMON/GCCUTS/CUTGAM,CUTELE,CUTNEU,CUTHAD,CUTMUO,BCUTE,BCUTM
     +             ,DCUTE ,DCUTM ,PPCUTM,TOFMAX,GCUTS(5)
C
*KEEP,GCPHYS.
      COMMON/GCPHYS/IPAIR,SPAIR,SLPAIR,ZINTPA,STEPPA
     +             ,ICOMP,SCOMP,SLCOMP,ZINTCO,STEPCO
     +             ,IPHOT,SPHOT,SLPHOT,ZINTPH,STEPPH
     +             ,IPFIS,SPFIS,SLPFIS,ZINTPF,STEPPF
     +             ,IDRAY,SDRAY,SLDRAY,ZINTDR,STEPDR
     +             ,IANNI,SANNI,SLANNI,ZINTAN,STEPAN
     +             ,IBREM,SBREM,SLBREM,ZINTBR,STEPBR
     +             ,IHADR,SHADR,SLHADR,ZINTHA,STEPHA
     +             ,IMUNU,SMUNU,SLMUNU,ZINTMU,STEPMU
     +             ,IDCAY,SDCAY,SLIFE ,SUMLIF,DPHYS1
     +             ,ILOSS,SLOSS,SOLOSS,STLOSS,DPHYS2
     +             ,IMULS,SMULS,SOMULS,STMULS,DPHYS3
     +             ,IRAYL,SRAYL,SLRAYL,ZINTRA,STEPRA
      COMMON/GCPHLT/ILABS,SLABS,SLLABS,ZINTLA,STEPLA
     +             ,ISYNC
     +             ,ISTRA
C
*KEEP,GCFLAG.
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEND.
cn..          Define of HBOOK word
      integer   istat
      character chtags(3)*5
      data      chtags/'E','ToF','N'/
      character ckovtags(4)*5
      data      ckovtags/'I','N','x','y'/
      character emtags(6)*5
      data      emtags/'I','E','x','y','r','sita'/
```

```
      character pitags(6)*5
      data      pitags/'I','E1','E2','Pt','phi','M'/
C
***   -----------------------------------------------------------------
*
*
*             Initialize GEANT
*
*
      CALL GINIT           ! Initialize GEANT system
*
*
*             Input tracking thresholds
*
*
      CUTGAM=0.001 ! threshold for gammas transport
      CUTELE=0.001 ! threshold for electrons transport
      CUTHAD=0.01  ! threshold for hadrons   transport
      CUTNEU=0.01  ! threshold for neutral hadrons transport
      CUTMUO=0.01  ! threshold for muons transport
      BCUTE=CUTGAM ! threshold for photons produced by electron Brems.
      BCUTM=CUTGAM ! threshold for photons produced by muon Brems.
      DCUTE=CUTELE ! threshold for electron produced by electron&delta-rays
      DCUTM=CUTELE ! threshold for electron produced by muon delta-rays
      PPCUTM=0.002 ! threshold for e+e- pair production by muons
C      TOFMAX=1010  ! threshold on time of flight counted
C                                         from primary interaction time
*
*
*             Input physics processes
*
*
      IPAIR=1 ! pair production with generation of electron/positron
      ICOMP=1 ! compton scattering with generation of electron
      IPHOT=1 ! photo-electric effect with generation of electron
      IPFIT=0 ! photo-fission effect with generation of secondaries
      IDRAY=1 ! delta ray production with generation of electron
      IANNI=1 ! positron annihilation with generation of photons
      IBREM=1 ! bremsstrahlung with generation of gamma
      IHADR=1 ! hadronic interactions with generation of secondaries
      IMUNU=1 ! muon nuclear interaction with generation of secondaries
      IDCAY=1 ! decay in flight with generation of secondaries
      ILOSS=2 ! continuous energy loss without generation of delta ray
C                 and Landau-Vaviliov-Gauss fluctuations
      IMULS=1 ! multiple scattering according to Moli'ere theory
      IRAYL=0 ! Rayleigh scattering
      ILABS=1 ! absorbtion of Cerenkov photons with posssible detection
      ISYNC=0 ! To control synchrotron radiation
      ISTRA=0 ! To control energy loss fluctuation model
**
      write(6,*)' '
      write(6,*)'      Type in !'
      write(6,*)' '
*
      write(6,*)' '
      write(6,*)'Select of GAS '
      write(6,*)'      Vaccume ===>0'
      write(6,*)'      Air     ===>1'
      write(6,*)'      He(gas) ===>2'
      read(5,*) ISWIT(6)
*
      write(6,*)' '
      write(6,*)'Select of ToF status '
      write(6,*)'    non-panel ===>0'
      write(6,*)'      1-panel ===>1'
```

```
      write(6,*)'      8-panel ===>2'
      write(6,*)'    non-panel &'
      write(6,*)'    EMcal 50cm===>3'
      read(5,*) ISWIT(7)
*
*
*             Initialize GEANT/ZBOOK data structures
*
*
      CALL GZINIT          ! Initialize mother banks
*
*
*             Initialize graphics package
*
*
      CALL GDINIT          ! Initialize drawing packag
*
      CALL GPART           ! Store the standard particle constants
      CALL GMATE           ! Initialize material constants
*
      CALL UGEOM           ! Define user geometry
      CALL GPHYSI          ! Initialize GEANT physics processes
*
      write(6,*)' ---------- USER Selection ----------'
      if(ISWIT(6) .eq. 1)then
         write(6,*)' Air'
      elseif(ISWIT(6) .eq. 2)then
         write(6,*)' Helium gas'
      else
         write(6,*) ' Vaccume'
      endif
      if(ISWIT(7) .eq. 0)then
         write(6,*)' This geom. is EMcal only'
      elseif(ISWIT(7) .eq. 1)then
         write(6,*)' This geom. is ToF 1-panel and EMcal small'
      elseif(ISWIT(7) .eq. 3)then
         write(6,*)' This geom. is EMcal only (near)'
      else
         write(6,*)' This geom. is ToF 8-panel and EMcal '
         write(6,*)'      Full Scale !!'
      endif
      write(6,*)' '
      write(6,*)' ------------------------------------'
      write(6,*)' '
*
**
*
*             Open HBOOK
*
      CALL HROPEN(1,'PHENIX','phenix.nt','n',1024,istat)
**
*
*             Booking Histogram & Ntuple file
*
      CALL HBOOKN(10,'Ntuple',3,'PHENIX',5000,chtags)
      CALL HBOOKN(20,'Cherenkov Number',4,'PHENIX',5000,ckovtags)
      CALL HBOOKN(30,'Energy of gamma',6,'PHENIX',5000,emtags)
      CALL HBOOKN(40,'Energy of pion0',6,'PHENIX',5000,pitags)
      CALL HBOOKN(50,'Energy of pion0 (trur)',6,'PHENIX',5000,pitags)
      CALL HBOOKN(60,'Energy of pion0 (B.G.)',6,'PHENIX',5000,pitags)
C
      CALL HBOOK1(100,'Pt-distribution(MeV/c)',100,0.,2000.,0.)
      CALL HBOOK1(110,'dN/(Pt*dPt)(MeV/c)',100,0.,2000.,0.)
      CALL HBOOK1(120,'rapidity-distribution',100,-5.,5.,0.)
C
```

```
      if((ISWIT(7) .ne. 0) .and. (ISWIT(7) .ne. 3))then
         CALL HBOOK1(200,'(ToF) Energy loss(MeV)',100,0.,20.,0.)
         CALL HBOOK2(250,'ToF Pair pro. and Compton scat.',
     +        100,-200.,200.,50,-100.,100.,0.)
      endif
C
      CALL HBOOK1(300,'(EMcal) GAMMA (MeV)',200,0.,2000.,0.)
      CALL HBOOK1(310,' GAMMA sita-distribution ',120,0.,40.,0.)
      CALL HBOOK1(320,' pion --> 2-GAMMA (MeV)',100,0.,2000.,0.)
      CALL HBOOK1(330,' 2-GAMMA phi-distribution ',120,0.,60.,0.)
      CALL HBOOK1(340,' mass of pion',200,0.,1000.,0.)
      CALL HBOOK1(350,' mass of pion (true)',200,0.,1000.,0.)
      CALL HBOOK1(360,' mass of pion (B.G.)',200,0.,1000.,0.)
*
      END
*
***
*
      SUBROUTINE UGLAST
*
************************************************************************
*                                                                      *
*             Termination routine to print histograms and statistics   *
*                                                                      *
************************************************************************
*
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEND.
cn..          Define of HBOOK word
      integer icycle
C
*     -----------------------------------------------------------------
*
      IF(JULAST.NE.0)THEN
         CALL CSJCAL(JULAST,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
      CALL GLAST
**
*
*
*             Close HIGZ
*
*
      CALL IGEND
**
*
*             Close HBOOK
*
      CALL HROUT(0,icycle,' ')
      CALL HREND('PHENIX')
**
*
      RETURN
 99   END
*
***
*
      SUBROUTINE GUKINE
*
************************************************************************
*                                                                      *
*             Generates Kinematics for primary tracks                  *
```

```
*                                                                      *
************************************************************************
*
*KEEP,GCFLAG.
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +     ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +     ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
C
*KEEP,GCONST.
      COMMON/GCONST/PI,TWOPI ,PIBY2,DEGRAD,RADDEG,CLIGHT ,BIG,EMASS
      COMMON/GCONSX/EMMU,PMASS,AVO
C
*KEEP,GCSCAN.
      PARAMETER (MSLIST=32,MAXMDT=3)
      COMMON/GCSCAN/SCANFL,NPHI,PHIMIN,PHIMAX,NTETA,TETMIN,TETMAX,
     +              MODTET,IPHIMI,IPHIMA,IPHI1,IPHIL,NSLMAX,
     +              NSLIST,ISLIST(MSLIST),VSCAN(3),FACTX0,FACTL,
     +              FACTR,IPHI,ITETA,ISCUR,SX0,SABS,TETMID(MAXMDT),
     +              TETMAD(MAXMDT)
      LOGICAL SCANFL
      COMMON/GCSCAC/SFIN,SFOUT
      CHARACTER*80 SFIN,SFOUT
C
*KEEP,GCOMIS.
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48),ITRA_pi(96,48)
C
*KEND.
C
      REAL      UBUF,PLAB(3),VERT(3)
      INTEGER   NVTX,NT,IPART
C
      real      pt, p4(4), Init_pt,Init_num, y_min, y_max, y
      integer   Iseed, Init_PART,In_part,Iptswit, IDtof, IDcal
      real      Nckov,Nck(96,48)
      integer   inum, ick, jck, ITRA_pi(96,48)
      character eventname*20
C
*     ------------------------------------------------------------------
*
      IF(JUKINE.NE.0)THEN
         CALL CSJCAL(JUKINE,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
*
*             Define type, momentum and vertex of particle
*
*
      save Init_PART,Init_num,Init_pt,Iptswit
*
      IF(IEVENT.EQ.1)THEN
         WRITE(6,*)' '
         WRITE(6,*)'      Type in !'
*
         WRITE(6,*)' '
         WRITE(6,*)'Initial particle number ..integer'
```

```
         WRITE(6,*)' <example>  1,Gamma   3,Electron   6,Muon-'
         WRITE(6,*)'            8,Pion+   9,Pion-     11,Kaon+'
         WRITE(6,*)'           12,Kaon-  14,Proton     etc.   '
         WRITE(6,*)' '
         WRITE(6,*)' Init_PART = 7 : pion0 '
C         READ(5,*)Init_PART
*
         WRITE(6,*)' '
         WRITE(6,*)'Number of paeticles  ..integer'
         WRITE(6,*)'  dN/dy = 375'
C         READ(5,*)Init_num
      ENDIF
*
      VERT(1) = 0.
      VERT(2) = 0.
      VERT(3) = 0.
      CALL GSVERT(VERT,0,0,UBUF,1,NVTX)
*
      y_min = -3.
      y_max =  3.
      Init_PART = 7
      Init_num = 375*6          ! dN/dy * (y_max - y_min)
C
      Iptswit = 0
      Iseed = NRNDM(2)
      do inum = 1,Init_num
         call pion_gen(y_min,y_max,y,p4,Init_pt,Iptswit,Iseed)
         pt = sqrt(p4(1)**2 + p4(2)**2)
*
         PLAB(1) = p4(1)
         PLAB(2) = p4(2)
         PLAB(3) = p4(3)
         CALL GSKINE(PLAB,Init_PART,NVTX,UBUF,1,NT)
*
         CALL HFILL(100,pt*1.0E+3,0.,1.) ! Fill of Pt-distribution
         CALL HFILL(110,pt*1.0E+3,0.,1./pt) ! Fill of Pt-distribution
         CALL HFILL(120,y,0.,1.) ! Fill of rapidity-distribution
      enddo
*
      ELOSStof = 0.
      Nckov = 0.
      do ick = 1,96
         do jck = 1,48
            Nck(ick,jck) = 0.
            ITRA_pi(ick,jck) = 0
            if(jck .le. 8)then
               ELOSS(jck,ick) = 0.
            endif
         enddo
      enddo
*
*
*
*            Kinematic debug (controled by ISWIT(1))
*
*
      IF((IDEBUG.EQ.1).AND.(ISWIT(1).EQ.1))THEN
         CALL GPRINT('VERT',0)
         CALL GPRINT('KINE',0)
      ENDIF
*
*
 99   END
*
***
```

```
*
      SUBROUTINE GUTREV
C.
C.    ******************************************************************
C.    *                                                                *
C.    *       User routine to control tracking of one event            *
C.    *       Called by GRUN                                           *
C.    *                                                                *
C.    ******************************************************************
C.
C.
      CALL GTREVE
      RETURN
      END
*
***
*
      SUBROUTINE GUTRAK
C.
C.    ******************************************************************
C.    *                                                                *
C.    *       User routine to control tracking of one track            *
C.    *                                                                *
C.    *    ==>Called by : GTREVE                                       *
C.    *                                                                *
C.    ******************************************************************
C.
*KEEP,GCFLAG
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +     ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +     ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
      CHARACTER*4 NAPART
C
*KEEP,GCTRAK.
      PARAMETER (MAXMEC=30)
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(MAXMEC)
     +     ,NAMEC(MAXMEC),NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG
     +     ,STEP  ,SNEXT ,SFIELD,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL
     +     ,ISTOP ,IGAUTO,IEKBIN, ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN
     +     ,NLVSAV,ISTORY
      CHARACTER*4 NAMEC
      PARAMETER (MAXME1=30)
      COMMON/GCTPOL/POLAR(3), NAMEC1(MAXME1)
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48),ITRA_pi(96,48)
C
      real Init_ENERGY
C.
C.    ------------------------------------------------------------------
C.
      if((VECT(1) .le. 0.1).and.(VECT(2) .le. 0.1).and.
     +     (VECT(3) .le. 0.1).and.(IPART .eq. In_part))then
         Init_ENERGY = GETOT*1.0E+3   ! GETOT:(GeV)--> Init_ENERGY:(MeV)
      endif
*
      CALL GTRACK
      RETURN
      END
*
```

```
***
*
      SUBROUTINE GUFLD(X,F)
C.
C.    ******************************************************************
C.    *                                                                *
C.    *       User routine to compute the magnetic field F             *
C.    *       at space point X                                         *
C.    *       Called by GHELIX,GRKUTA                                  *
C.    *                                                                *
C.    ******************************************************************
C.
*KEEP,GCVOLU.
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     +     LVOLUM(15),LINDEX(15),INFROM,NLEVMX,NLDEV(15),LINMX(15),
     +     GTRAN(3,15),GRMAT(10,15),GONLY(15),GLX(3)
      CHARACTER*4 NAMES
C
      DIMENSION X(3),F(3)
C.
C.    ------------------------------------------------------------------
*
      F(1)=0.
      F(2)=0.
      F(3)=0.
C
      RETURN
      END
*
***
*
      SUBROUTINE GUSTEP
*
************************************************************************
*                                                                      *
*            User routine called at the end of each tracking step      *
*            MEC   is the mechanism origin of the step                 *
*            INWVOL is different from 0 when the track has reached     *
*                   a volume boundary                                  *
*            ISTOP is different from 0 if the track has stopped        *
*                                                                      *
*            Called by GTRACK                                          *
*                                                                      *
************************************************************************
*
*KEEP,GCFLAG
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +     ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
*KEEP,GCKINE.
      COMMON/GCKINE/IKINE,PKINE(10),ITRA,ISTAK,IVERT,IPART,ITRTYP
     +     ,NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
      CHARACTER*4 NAPART
C
*KEEP,GCKING.
      INTEGER MXGKIN
      PARAMETER (MXGKIN=100)
      COMMON/GCKING/KCASE,NGKINE,GKIN(5,MXGKIN),
     +                           TOFD(MXGKIN),IFLGK(MXGKIN)
      INTEGER       KCASE,NGKINE ,IFLGK,MXPHOT,NGPHOT
      REAL          GKIN,TOFD,XPHOT
C
      PARAMETER (MXPHOT=800)
```

```
      COMMON/GCKIN2/NGPHOT,XPHOT(11,MXPHOT)
C
      COMMON/GCKIN3/GPOS(3,MXGKIN)
      REAL          GPOS
C
*KEEP,GCMATE.
      COMMON/GCMATE/NMAT,NAMATE(5),A,Z,DENS,RADL,ABSL
C
*KEEP,GCTMED.
      COMMON/GCTMED/NUMED,NATMED(5),ISVOL,IFIELD,FIELDM,TMAXFD,STEMAX
     +     ,DEEMAX,EPSIL,STMIN,CFIELD,CMULS,IUPD,ISTPAR,NUMOLD
      COMMON/GCTLIT/THRIND,PMIN,DP,DNDL,JMIN,ITCKOV,IMCKOV,NPCKOV
C
*KEEP,GCTRAK.
      PARAMETER (MAXMEC=30)
      COMMON/GCTRAK/VECT(7),GETOT,GEKIN,VOUT(7),NMEC,LMEC(MAXMEC)
     +     ,NAMEC(MAXMEC),NSTEP ,MAXNST,DESTEP,DESTEL,SAFETY,SLENG
     +     ,STEP  ,SNEXT ,SFIELD,TOFG  ,GEKRAT,UPWGHT,IGNEXT,INWVOL
     +     ,ISTOP ,IGAUTO,IEKBIN, ILOSL, IMULL,INGOTO,NLDOWN,NLEVIN
     +     ,NLVSAV,ISTORY
      CHARACTER*4 NAMEC
      PARAMETER (MAXME1=30)
      COMMON/GCTPOL/POLAR(3), NAMEC1(MAXME1)
C
*KEEP,GCVOLU.
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     +     LVOLUM(15),LINDEX(15),INFROM,NLEVMX,NLDEV(15),LINMX(15),
     +     GTRAN(3,15),GRMAT(10,15),GONLY(15),GLX(3)
      CHARACTER*4 NAMES
C
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEEP,GCONST.
      COMMON/GCONST/PI,TWOPI ,PIBY2,DEGRAD,RADDEG,CLIGHT ,BIG,EMASS
      COMMON/GCONSX/EMMU,PMASS,AVO
C
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48),ITRA_pi(96,48)
C
*KEND.
C
      real      TOFobs, ELOSSobs, Npe, beta
      real      ELOSStof, ELOSS(8,96), Nckov, Nck(96,48)
      integer   ick, jck
*     -----------------------------------------------------------------
*
      IF(JUSTEP.NE.0)THEN
         CALL CSJCAL(JUSTEP,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
*
*             Print the tracking and physics
*
*
      IF(ISWIT(1).EQ.1)THEN
         CALL GPCXYZ
      ENDIF
*
*
*             Save in the stack particle generated during the current step
*
*
      IF(NGKINE.NE.0)THEN
```

```
        CALL GSKING(0)
      ENDIF
*
*
*             Stores current space point into JXYZ
*
*
      CALL GSXYZ
**
*
*             Filling Physics parameter
*
      if((ISWIT(7) .ne. 0) .and. (ISWIT(7) .ne. 3))then
         if((NAMES(3) .eq. 'SCTS').or.(NAMES(3) .eq. 'SCTL'))then
            TOFobs = TOFG * 1.0E+9 ! TOFG:(sec)  --> TOFobs:(nsec)
            ELOSSobs = DESTEP * 1.0E+3 ! DESTEP:(GeV)--> ELOSSobs:(MeV)
            ELOSStof = ELOSStof + ELOSSobs
            if(ELOSSobs .ne. 0)then
               if(NAMES(3) .eq. 'SCTS')then
                  itf = NUMBER(2)
                  if((NUMBER(3) .ge. 1).and.(NUMBER(3) .le. 16))then
                     jtf = NUMBER(3)
                  else
                     jtf = 113 - NUMBER(3)
                  endif
               elseif(NAMES(3) .eq. 'SCTL')then
                  itf = NUMBER(2)
                  if((NUMBER(3).ge. 1).and.(NUMBER(3).le.16))then
                     jtf = 32 + NUMBER(3)
                  elseif((NUMBER(3).ge.17).and.(NUMBER(3).le.32))then
                     jtf = 81 - NUMBER(3)
                  elseif((NUMBER(3).ge.33).and.(NUMBER(3).le.48))then
                     jtf = 32 + NUMBER(3)
                  elseif((NUMBER(3).ge.49).and.(NUMBER(3).le.64))then
                     jtf = 81 - NUMBER(3)
                  endif
               endif
               ELOSS(itf,jtf) = ELOSS(itf,jtf) + ELOSSobs
            endif
         endif
         if((NAMES(3) .eq. 'SCTS').or.(NAMES(3) .eq. 'SCTL').or.
     +        (NAMES(3) .eq. 'PMT ').or.(NAMES(3) .eq. 'MMTL').or.
     +        (NAMES(3) .eq. 'CRBN').or.(NAMES(3) .eq. 'HCMB').or.
     +        (NAMES(3) .eq. 'ALM1').or.(NAMES(3) .eq. 'ALM2').or.
     +        (NAMES(3) .eq. 'ALM3').or.(NAMES(3) .eq. 'CABL'))then
            if((LMEC(NMEC) .ge. 6).and.(LMEC(NMEC) .le. 8))then
               CALL HFILL(250,VECT(3),VECT(2),1.)
            endif
         endif
      endif
      if((NAMES(5).eq.'PbGy').and.(CHARGE.ne.0).and.(VECT(7).ne.0.))then
         beta = VECT(7)/GETOT   ! VECT(7):(GeV/c) GETOT:(GeV)
         Npe = STEP*370.*1.0*(1. - 1./(1.5*beta)**2) !1.5=kusseturitu
         if(1./(beta*1.5) .lt. 1.)then
            Nckov = Nckov + Npe
            ick = NUMBER(4)
            jck = NUMBER(5)
            Nck(ick,jck)=Nck(ick,jck) + Npe
            if(ITRA_pi(ick,jck) .eq. 0)then
               ITRA_pi(ick,jck) = ITRA
            endif
         endif
      endif
C
      if(INWVOL .eq. 2)then
```

```
         NAMES(3)='    '
         NAMES(4)='    '
         NAMES(5)='    '
      endif
*
  99  CONTINUE
      END
*
*
***
*
      SUBROUTINE GUOUT
*
************************************************************************
*                                                                      *
*             User routine called at the end of each event.            *
*                                                                      *
************************************************************************
*
*KEEP,GCFLAG.
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST,IDRUN,IDEVT,IEORUN
     +        ,IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
      COMMON/GCFLAX/BATCH, NOLOG
      LOGICAL BATCH, NOLOG
C
      COMMON/GCOMIS/ICOMIS,JUINIT,JUGEOM,JUKINE,JUSTEP,JUOUT,JULAST
C
*KEEP,GCVOLU.
      COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
     +     LVOLUM(15),LINDEX(15),INFROM,NLEVMX,NLDEV(15),LINMX(15),
     +     GTRAN(3,15),GRMAT(10,15),GONLY(15),GLX(3)
      CHARACTER*4 NAMES
C
*KEEP,user common block
      common/uELOSS/In_part,IDcal,pt,Init_ENERGY,ELOSStof,ELOSS(8,96),
     +              Nckov,Nck(96,48),ITRA_pi(96,48)
C
*KEND.
C
      real      range_EM,event(3),event_ck(4),event_em(6),event_pi(6)
      real      pt,Init_ENERGY,ELOSStof,ELOSS(8,96),Nckov,Nck(96,48),Npi
      integer   icycle, ick, jck, i, j, maxN, NN
      parameter (maxN = 1000)
      real      Nem(maxN),E_em(maxN),Rem(maxN),sita_em(maxN),Rx,Ry
      integer   xem(maxN),yem(maxN),ITRA_pion(maxN)
      real      E_pi,Pt_pi,phi_pi,M_pi
*     ------------------------------------------------------------------
*
      IF(JUOUT.NE.0)THEN
         CALL CSJCAL(JUOUT,0,X,X,X,X,X,X,X,X,X,X)
         GO TO 99
      ENDIF
*
      CALL ICLRWK(0,0)
**
*
*             Filling Histogram file
*
      write(6,*) IEVENT,Init_ENERGY,'(MeV) ToF ',ELOSStof,' EMcal',Nckov
C
      event(1) = Init_ENERGY
      event(2) = ELOSStof
      event(3) = Nckov
      CALL HFN(10,event)
C
```

```
      do itf = 1,8
         do jtf = 1,96
            if(ELOSS(itf,jtf) .ge. 0.1)then
               CALL HFILL(200,ELOSS(itf,jtf),0.,1.)
            endif
         enddo
      enddo
C
      event_ck(1) = IEVENT
      event_ck(2) = Nckov
      event_ck(3) = 0.
      event_ck(4) = 0.
      CALL HFN(20,event_ck)
C
      if(ISWIT(7) .eq. 3)then
         range_EM = 60.
      else
         range_EM = 530.
      endif
C
      NN = 0
      do ick = 1,96
         do jck = 1,48
            if(Nck(ick,jck) .ne. 0.)then
               event_ck(1) = IEVENT
               event_ck(2) = Nck(ick,jck)
               event_ck(3) = ick
               event_ck(4) = jck
               CALL HFN(20,event_ck)
               if((ick .ne. 1).and.(ick .ne. 96).and.
     +              (jck .ne. 1).and.(jck .ne. 48).and.
     +              (Nck(ick,jck) .gt. Nck(ick-1,jck-1)).and.
     +              (Nck(ick,jck) .gt. Nck(ick-1,jck)).and.
     +              (Nck(ick,jck) .gt. Nck(ick-1,jck+1)).and.
     +              (Nck(ick,jck) .gt. Nck(ick,jck-1)).and.
     +              (Nck(ick,jck) .gt. Nck(ick,jck+1)).and.
     +              (Nck(ick,jck) .gt. Nck(ick+1,jck-1)).and.
     +              (Nck(ick,jck) .gt. Nck(ick+1,jck)).and.
     +              (Nck(ick,jck) .gt. Nck(ick+1,jck+1)))then
                  NN = NN + 1
                  xem(NN) = ick
                  yem(NN) = jck
                  Rem(NN) = 4.1*sqrt((ick-48.5)**2 + (jck-24.5)**2)
                  Rx = 4.1*(ick - 48.5)
                  Ry = sqrt((4.1*(jck-24.5))**2 + range_EM**2)
                  sita_em(NN) = atan(Rx/Ry)*180./3.1415926
               endif
            endif
         enddo
      enddo
C
      if(NN .ge. 1)then
         do i = 1,NN
            Nem(i) = Nck(xem(i)-1,yem(i)-1)+Nck(xem(i)-1,yem(i))
     +           +Nck(xem(i)-1,yem(i)+1)+Nck(xem(i),yem(i)-1)
     +           +Nck(xem(i),yem(i))+Nck(xem(i),yem(i)+1)
     +           +Nck(xem(i)+1,yem(i)-1)+Nck(xem(i)+1,yem(i))
     +           +Nck(xem(i)+1,yem(i)+1)
C
            E_em(i) = 5.995E-2*Nem(i) + 0.606   ! MeV : cut= 1MeV
C
            ITRA_pion(i) = ITRA_pi(xem(i),yem(i))
C
            event_em(1) = IEVENT
            event_em(2) = E_em(i)
```

```
            event_em(3) = xem(i)
            event_em(4) = yem(i)
            event_em(5) = Rem(i)
            event_em(6) = sita_em(i)
            if(E_em(i) .gt. 10.)then
               CALL HFN(30,event_em)
               CALL HFILL(300,E_em(i),0.,1.)
               CALL HFILL(310,sita_em(i),0.,1.)
            endif
         enddo
      endif
      if(NN .ge. 2)then
         do i = 1,NN-1
            do j = i+1,NN
               if((E_em(i) .gt. 250.).and.(E_em(j) .gt. 250.))then
                  Npi = Nem(i) + Nem(j)
                  E_pi = E_em(i) + E_em(j)
                  Pt_pi = E_em(i)*cos(sita_em(i)*3.1415926/180.)
     +                 +E_em(j)*cos(sita_em(j)*3.1415926/180.)
                  phi_pi=acos((Rem(i)**2+Rem(j)**2+2*range_EM**2
     +                 -((xem(i)-xem(j))**2+(yem(i)-yem(j))**2)*4.1**2)
     +                 /(2*sqrt((Rem(i)**2+range_EM**2)
     +                 *(Rem(j)**2 + range_EM**2))))*180./3.1415926
                  M_pi = 2*sqrt(E_em(i)*E_em(j))*
     +                 sin(phi_pi/2*3.1415926/180.)
C
                  event_pi(1) = IEVENT
                  if(E_em(i) .ge. E_em(j))then
                     event_pi(2) = E_em(i)
                     event_pi(3) = E_em(j)
                  else
                     event_pi(2) = E_em(j)
                     event_pi(3) = E_em(i)
                  endif
                  event_pi(4) = Pt_pi
                  event_pi(5) = phi_pi
                  event_pi(6) = M_pi
C
                  if(phi_pi .ge. 3.)then
                     CALL HFN(40,event_pi)
                     CALL HFILL(320,E_pi,0.,1.)
                     CALL HFILL(330,phi_pi,0.,1.)
                     CALL HFILL(340,M_pi,0.,1.)
                     if(ITRA_pion(i) .eq. ITRA_pion(j))then
                        CALL HFN(50,event_pi)
                        CALL HFILL(350,M_pi,0.,1.)
                     else
                        CALL HFN(60,event_pi)
                        CALL HFILL(360,M_pi,0.,1.)
                     endif
                  endif
               endif
            enddo
         enddo
      endif
C
**
*
*             Draw track of particle and detector
*
*
      if( ISWIT(5) .eq. 0 )then
         if( ISWIT(2) .eq. 2 )then
            CALL GDRAW('MOTH',0.,0.,0.,0.5,10.,0.030,0.030)
         elseif( ISWIT(2) .eq. 3)then
```

```
            CALL GDRAW('MOTH',90.,90.,270.,0.5,10.,0.030,0.030)
         elseif( ISWIT(2) .eq. 4)then
            CALL GDRAW('MOTH',90.,0.,0.,10.,10.,0.030,0.030)
         elseif( ISWIT(2) .eq. 5)then
            CALL GDRAW('MOTH',90.,0.,0.,10.,10.,0.050,0.050)
         else
            CALL GDRAW('MOTH',60.,120.,0.,3.,5.,0.030,0.030)
         endif
*
         CALL GDHEAD(101,'TOF and EM-cal$',.5)
         CALL GDXYZ(0)
         CALL GDAXIS(0.,0.,0.,100.)
         CALL GDPART(0,11,0.25)
         CALL GDMAN(18.,4.0)
         CALL GDSCAL(10.,2.)
      endif
*
      RETURN
 99      END
*
*
**
*
      subroutine pion_gen(y_min,y_max,y,p4,Init_pt,Iptswit,Iseed)
*
*************************************************************************
*  this routine generates pions with constant rapidity density
*  and with the slope of 150 MeV/c.  pion0 = 139 MeV
*     y_min  ; INPUT  : Lower limit of pion rapidity
*     y_max  ; INPUT  : Higher limit of pion rapidity
*     y      ; OUTPUT : pion rapidity
*     p4     ; OUTPUT : Four Momentum of the particles
*     Init_pt; INPUT  : Initial momemtum Pt (if Iptswit=1)
*     Iptswit; INPUT  : Pt switch 0 or 1
*     Iseed  ; IN/OUT : INTEGER*4  Seed of random number
*************************************************************************
*
      real y, y_min, y_max, mass, pt, p4(4), slope, sita, arg
C
      y = min(y_min,y_max) + abs(y_max-y_min)*ran(Iseed)  ! rapidity
C
      if(Iptswit .eq. 0)then
         slope = 1./6.          ! 1/6 [GeV/c] ==> 150 [MeV/c]
         pt = ran_pt(slope, Iseed)
      elseif(Iptswit .eq. 1)then
         pt = Init_pt
      endif
C
      arg = 2.*3.1415926*ran(Iseed)
C
      mass  = 0.139
      p4(1) = pt*cos(arg)
      p4(2) = pt*sin(arg)
      p4(3) = sqrt(pt**2 + mass**2)*(exp(y) - exp(-y))/2. ! p_para
      p4(4) = sqrt(mass**2 + p4(1)**2 + p4(2)**2 + p4(3)**2)
C
      return
      end
**
*
      real function ran_pt(Slope,Iseed)
*
*************************************************************************
*  Generate pt when the invariant cross section follows
*  Exponential distribution.
```

```
*       Slope ; INPUT  : Slope of exponential distribution
*       Iseed ; INPUT  : INTEGER*4  Seed of random number
*************************************************************************
*       dN
*      ---- == Pt*exp(-Pt/S)   ;  S = slope
*       dPt
*************************************************************************
*
      common/com_ran_pt/prob, pt_slope
      external pt_func
      real prob, pt_slope
C
      pt_slope = Slope
      prob = ran(Iseed)
C
      CALL RZERO(0., 5., ran_pt, R, 0.01, 100, pt_func)
      return
      end
**
      real function pt_func(x)
      common/com_ran_pt/prob, pt_slope
C
      pt_func = 1.- (1.+ x/pt_slope)*exp(-x/pt_slope) - prob
      return
      end
*
*
```