

平成 21 年度 卒業論文

Geant4 を用いた ALICE 実験における
PHOS 検出器のシミュレーションとその手法

広島大学理学部物理科学科

クォーク物理学研究室

B066948 岡崎悠介

主査 三好隆博 助教

副査 深澤泰司 教授

指導教員 杉立徹 教授

2010 年 2 月

概要

2008年9月10日から、CERN（欧州合同原子核機構）の有するLHC（大型ハドロン衝突型加速器）が稼動した。そして2009年11月23日、初めての陽子陽子衝突の実験が行われた。このLHC内の実験プロジェクトの1つにALICE（A Large Ion Collider Experiment）実験がある。このプロジェクトで使用される検出器の1つ、PHOS（PHOton Spectrometer）検出器は、本研究グループが国際協力のもと、研究・開発し、粒子衝突によって発生する光子の直接観測を目的とした検出器である。

本論文ではGeant4と呼ばれる、CERNが研究・開発し、加速器実験の分野を中心に他分野においても使用され、日本国内においても研究者によってさらなる研究がなされているシミュレーションソフトウェアを用いてALICE実験内におけるPHOS検出器を再現し、シミュレーションを行い、これを通してGeant4の初歩的なシミュレーションの手法を紹介し、Geant4のさらなる可能性を考察する。

目次

第 1 章 導入	5
1.1 加速器	5
1.2 LHC 加速器	5
1.3 クォーク・グルーオン・プラズマ	6
1.4 ALICE 実験	8
1.5 シンチレーション	9
1.6 PW04	9
1.7 PHOS 検出器	9
1.8 Geant4	11
1.9 Geant4 ユーザ	13
1.10 研究目的	14
第 2 章 研究背景	15
2.1 Geant4 の開発・利用	15
2.2 Geant4 のオブジェクト指向	16
2.3 Geant4 の動作	16
第 3 章 シミュレーション	17
3.1 Geant4 を使う準備	17
3.2 CLHEP のインストール	18
3.3 Geant4 のインストール	19
3.4 Geant4 のチュートリアル	22
3.5 exercise-3	22
第 4 章 プログラム書き換えの手法	23
4.1 構成物質の設定	25
4.2 大きさ・形・物質・位置の設定	26
4.3 入射粒子の設定	28
4.6 シミュレーションの結果	
第 5 章 結論	34
第 6 章 謝辞	35

図目次

図 1-1LHC 概略図	6
図 1-2LHC イメージ図	6
図 1-3ALICE の内部構造	7
図 1-4ALICE 内部構造写真	8
図 1-5PWO クリスタル	10
図 1-6PHOS 検出器 5 モジュール	10
図 1-7Geant4 でシミュレートされた磁場中の粒子の振る舞い	12
図 2-1 医療分野で応用された Geant4	15
図 3-1KEK で公開されている exercise-3 のシミュレーション	22
図 3-2 Exercise-3 のシミュレーション実行	24
図 3-3 Geant4 内での daughterVolume と motherVolume の概念図	27
図 3-4Geant3 で 50GeV の光子を入射したときのヒストグラム	31
図 3-5Geant4 で 50GeV の光子を入射したときのヒストグラム	31
図 3-6 Geant3 で 100GeV の光子を入射したときのヒストグラム	32
図 3-7 Geant4 で 100GeV の光子を入射したときのヒストグラム	32
図 3-8Geant4 で再現された PHOS 検出器 1 モジュールに光子 1 個を入射したとき	26

第 1 章 導入

1.1 加速器

一体「モノ」は何から構成されているのか。紀元前の学者は「モノ」は全て水、土、火、空気から構成されていると考えた。そして中世ヨーロッパでは錬金術が行われ、そして現在までさまざまな物理実験が行われてきた。

20 世紀になり、物質の研究は加速器の開発により大きく進歩した。静電加速器、線形加速器、そして円形加速器が開発された。扱える粒子のエネルギーも格段に大きくなった。

1.2 LHC 加速器

2008 年 9 月 10 日から CERN (欧州合同原子核機構) の有する LHC (大型ハドロン衝突型加速器) が稼動した。LHC は複数の実験施設 (ATLAS, ALICE, CMS, LHCb, TOTEM, LHCf) から成っており、今までで最高のエネルギーを以って粒子を加速させることのできる加速器である。周長およそ 27km で、地下 100m のところでトンネル型で円状になっており、この中で粒子を逆方向に加速させ、衝突させる。それはちょうど正面衝突の事故のように、莫大なエネルギーで衝突しあい、p+p 衝突で重心系エネルギー 14TeV まで、Pb+Pb 衝突で核子当たり重心系エネルギー 5.5TeV まで高めることができると期待されている。以下の図 1-1 と図 1-2 は LHC の各実験プロジェクトの位置と地下に収められている様子をおおまかに描写したものである。

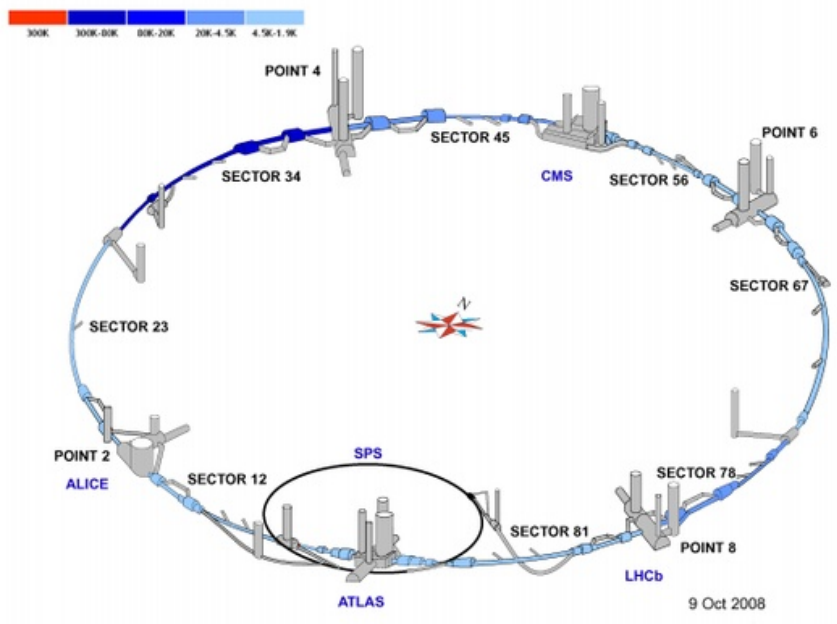


図 1-1 LHC 概略図[*1]

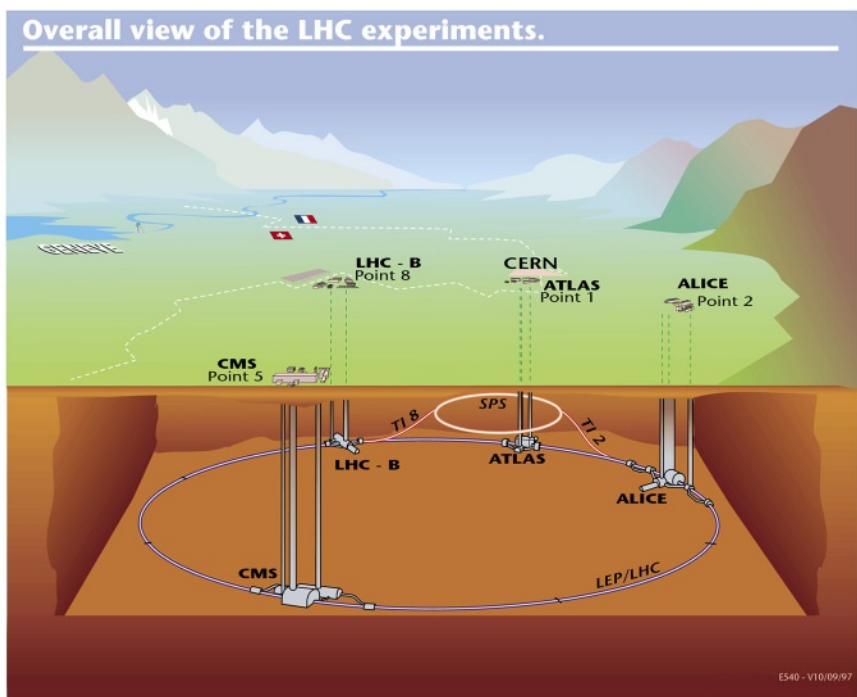


図 1-2 LHC イメージ図[*2]

1.3 クォーク・グルーオン・プラズマ (QGP)

自然界には重力、電磁力、強い力、弱い力の4つの力がある。重力は太陽系、銀河、星などができるとき、支配的となる。強い力はクォーク間で働き、原子核を構成する。弱い力は原子核のベータ崩壊を引き起こす。重力、電磁力は質量が0であるが故、遠距離でもはたらく。弱い力の媒介をするウィークボソンは質量が0ではない。よって近距離で働く力となるが、強い力はこれとは別の理由で近距離力となる。これを説明するのが QCD (quantum chromo dynamics) である。すなわち強い力を媒介するグルーオン自身が強い力で相互作用するため、クォーク間の距離が非常に近くなると強い力が減衰する。これを漸近的自由性という。これによりクォークの密度が高くなり、つまりクォーク間の距離が小さくなり、強い力が十分弱くなったとき、クォークが核子の閉じ込めから開放され、自由に飛び回ることがある。この状態を QGP (quark gluon plasma) という。

1.4 ALICE 実験

LHC 加速器の中の実験プロジェクトの中のひとつで、高エネルギー原子核衝突に特化した唯一の実験である。このプロジェクトの目的は宇宙がビッグバン直後のクォークとグルーオン、核子の閉じ込めから開放された状態で存在する QGP とよばれる状態を衝突によって再現し、その様子を探求することである。これを達成するために QGP からの熱輻射によって発生する光子を測定する。この粒子を効率よく検出できる PHOS 検出器を始め、様々な検出器が置かれている。以下、2枚の図に ALICE の内部構造とその様子を示す。

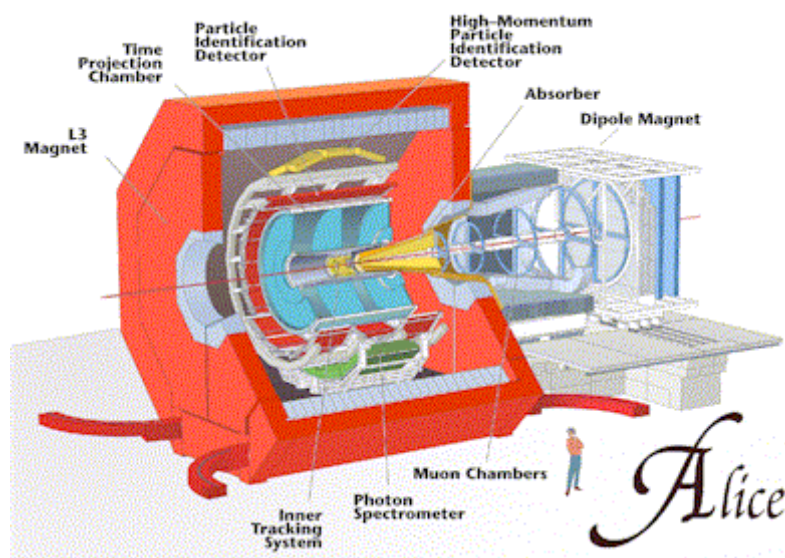


図 1-3 ALICE の内部構造[*3]

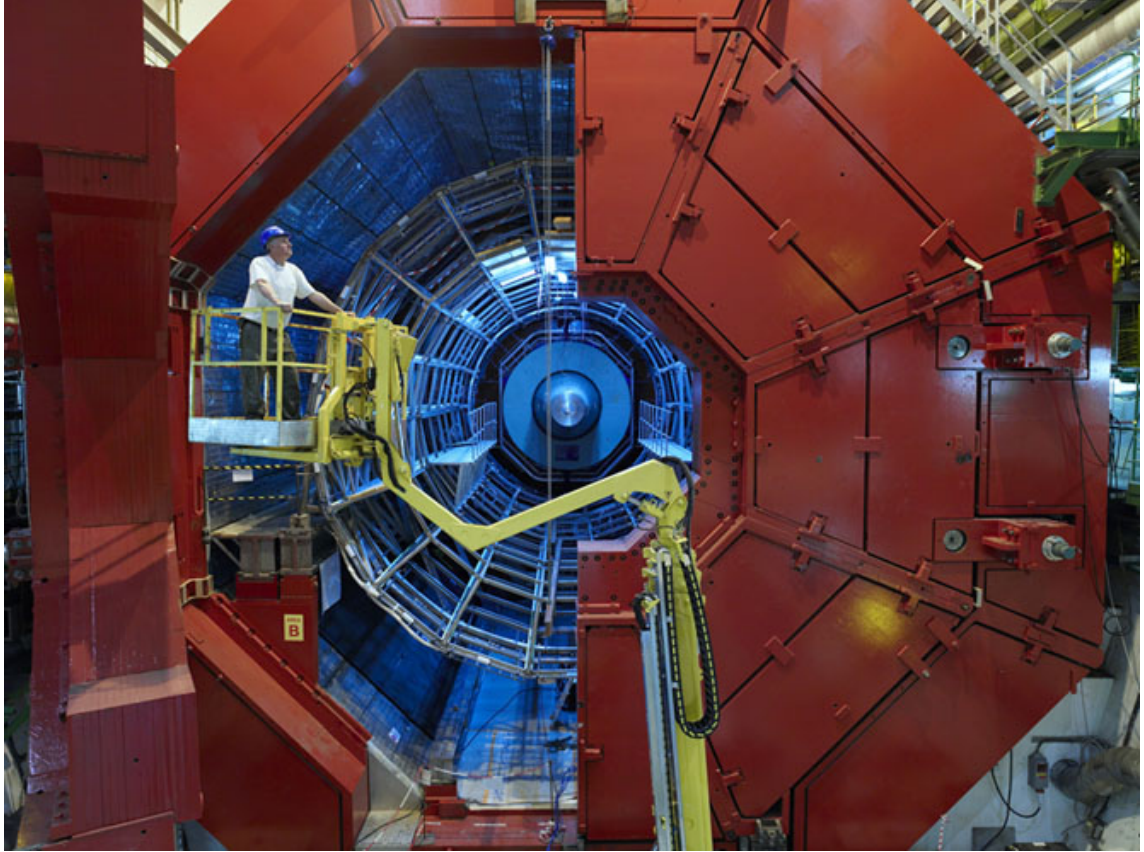


図 1-4 ALICE この中を陽子などが走る[*3]

1.5 シンチレーション

粒子が物質中を通過するとき、相互作用を起こしてエネルギーを失う。その際、シンチレーションと呼ばれる、シンチレータ（PHOS の場合、PWO クリスタル）と粒子が衝突したときに発光する現象がみられる。そしてその光のことをシンチレーション光と呼ばれる。飛んできた粒子のエネルギーがシンチレーション光に変わり、それが検出される。

発光するのはシンチレータを構成する原子内で電子が励起され、その電子が1つ外の軌道へ移動する。しかし、すぐに安定な状態に戻ろうとするので、元の軌道に帰ってくる。この際に差分のエネルギーを光子として放出する。これがシンチレーション光である。

1.6 PWO4

上記のようにシンチレーション光を測定することによって粒子発生の起源を探るのだが、PHOS のシンチレータは PWO4（鉛タングステン酸結晶）で構成されており、その発光量は決して大きくない。しかし、PWO 結晶を-25℃まで冷やすと、発光量が増える。しかし PWO 結晶の密度は他のシンチレータより大きく、放射長が短いので、検出器の小型化が可能である。さらに、モリエール半径が小さいので電磁シャワーの横方向の広がりを抑えることが出来、二粒子分離分解能が高い。

1.7 PHOS 検出器

ALICE 実験で用いられている PHOS 検出器は荷電粒子が入射すると光を発生するクリスタルの部分と、その光を感知する検出素子部分で構成されて、電磁力ロリメータとなっている。PHOS 検出器のエネルギー分解能と位置分解能は高い性能を誇っており、高精度光子検出器として期待されている。PHOS 検出器は以上の特徴を持ったシンチレータが 56×64 個並べて、1 モジュールとし、衝突点から 4.6m の位置、ALICE 検出器の底に 5 モジュール配置され、方位角方向 100° ビーム軸方向は擬ラピディティ領域 $-0.12 < \eta < +0.12$ を占めている。

擬ラピディティは天頂角 θ を使って

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right)$$

と表される。

PWO クリスタル1つと、ALICE に収められている PHOS 検出器 5 モジュールを以下に示す。

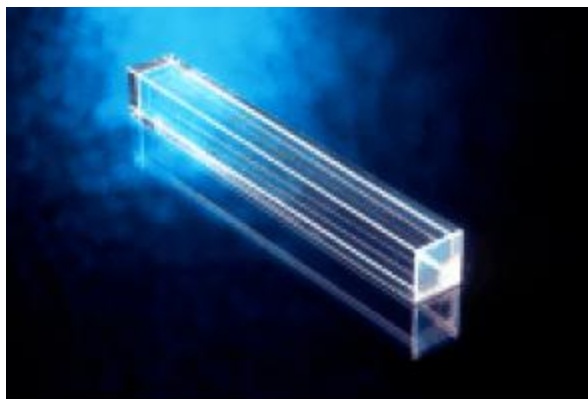


図 1-5 PWO 結晶[*4]

高いエネルギー分解能と高い位置分解能を備えており
高感度光子検出器として期待されている
これが 56×64 個並んで 1 モジュールを構成する

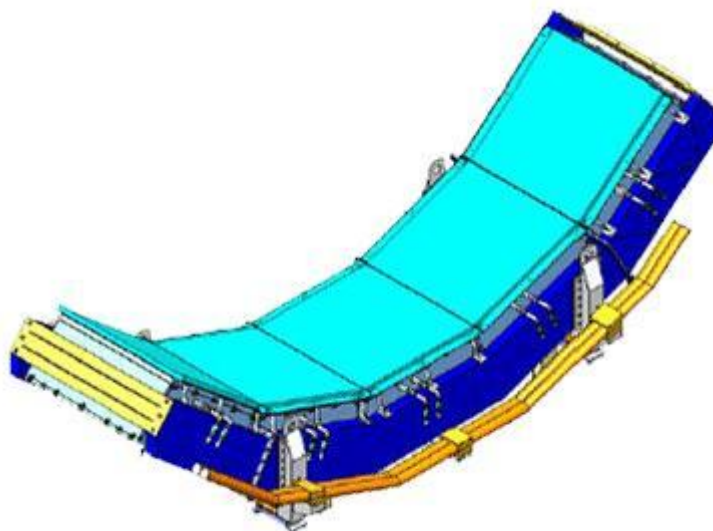


図 1-6 PHOS 検出器 5 モジュール[*5]

以下の表は PWO 結晶とその他検出器の特徴である。

表 1-1 PWO の特徴とその他の物質との比較

物質	密度(g/cm ³)	放射長(cm)	モリエール半径(cm)	減衰時間(ns)	光量 (NaI を 1 としたとき)
NaI	3.67	2.59	4.13	230	1
PWO(fast component)	8.3	0.89	2	10	0.006
PWO(slow component)				50	0.001
BGO	7.13	1.12	2.23	300	0.9
CsI(fast component)	4.51	1.86	3.57	6	2.3
CsI(slow component)				35	5.6

1.8 Geant4

Geant4 とは粒子と物質の相互作用をシミュレーションするソフトウェアである。LHC を有する CERN が Geant3 の開発を終え、その後続のシミュレーションソフトウェアとして開発が開始した。Geant3 が FORTRAN 言語で記述されたのに対し、Geant4 は C++言語で記述されている。開発段階では高エネルギー分野でのシミュレーションを第一の目的とし、同様に他分野での応用も考えられて開発が進められた。現在では宇宙、さらには、医療の分野でも Geant4 が用いられている。以下に Geant4 で荷電粒子の磁場中の振る舞いをシミュレートしたものを示す。

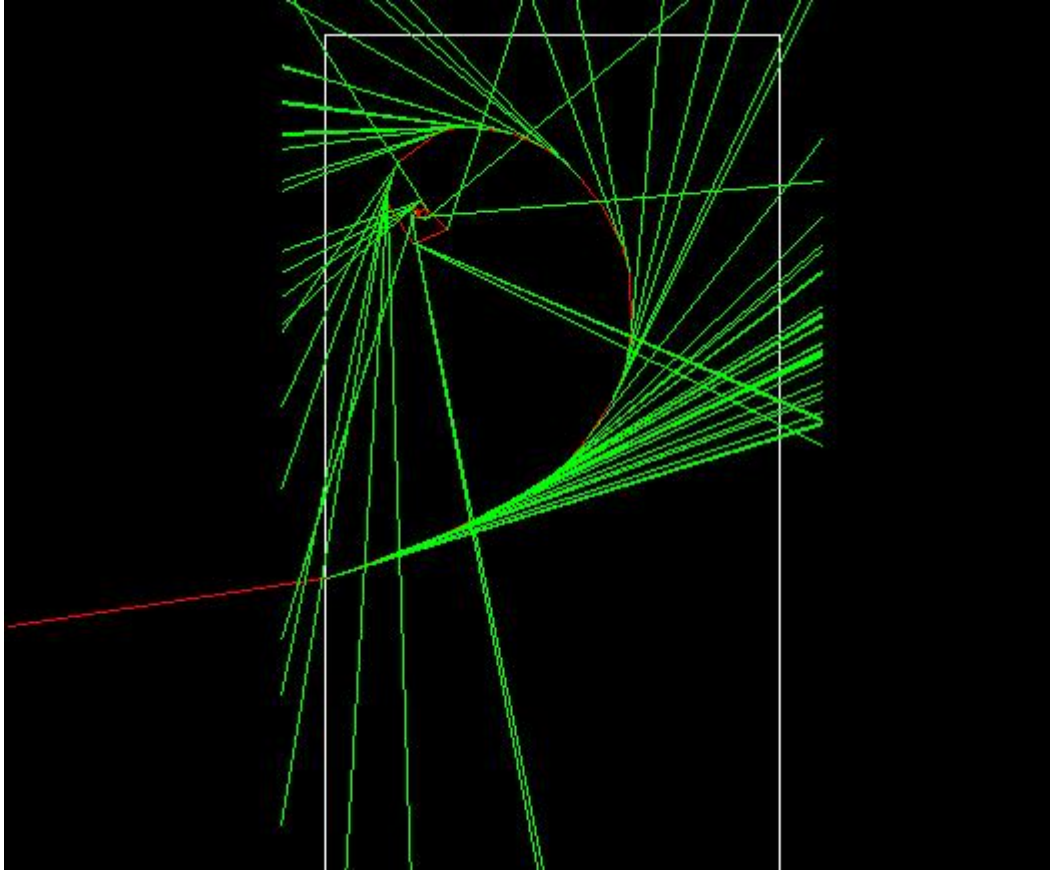


図 1-7 Geant4 での簡単なシミュレート例
磁場中の荷電粒子（赤）の動きと
シンクロトロン輻射によって発生したガンマ線（緑）[*6]

1.9 Geant4 ユーザ

Geant4 を扱う者は以下の 3 つのユーザカテゴリに分類される。

User (レベル 1 ユーザ)

このユーザは用意されたプログラムを元に、入射させる粒子、起こす物理イベント、など基本的な情報をユーザ自身が設定、シミュレーションを行うユーザのことである。よって自ら C++ 言語を用いてプログラムを書くことはない。

User (レベル 2 ユーザ)

このユーザはプロバイダ (後述するレベル 3 ユーザを指す) から提供されたプログラムを用いて自身が必要としている結果を求めるために必要な、一部のプログラムの書き換えを C++ 言語を用いて行う。このユーザは全てのプログラムについて知っている必要はない。

Priver (レベル 3 ユーザ)

このユーザは自身でプログラムを作成するこのできるユーザのことで、オブジェクト指向に基づいて C++ 言語を扱うことの出来る者のことを指す。また、ソフトウェアの全体像について精通している者である。

Geant4 のユーザは以上の 3 つに分類され、自身の必要なように Geant4 に関わる事が出来る。

1.10 研究目的

Geant4 は今では高エネルギー分野はもちろん医療分野や半導体、宇宙分野などにも応用されている。研究者のそれぞれがそれぞれの研究環境の必要に応じて”オリジナルの Geant4 ”を開発している。高エネルギー分野では宇宙誕生の瞬間を探り、医療分野においても人体と放射線の相互作用をシミュレートするために用いられている。そのほかにも半導体、宇宙開発などに利用される、この自由度の高い Geant4 を扱うと実にさまざまな、粒子と物質の相互作用をシミュレートできる。宇宙の誕生の瞬間の探求、放射線医療の効率化、人工衛星の測定器の最適化など、実に幅広いものである。

本研究では、ユーザーレベル 2 となり、Geant4 入門として、すでに PHOS 検出器を再現されている Geant3 と Geant4 のシミュレーション結果を比較し、Geant4 を正しく作動させたことを確認した後、その手法を一般的に解説する。

第2章 研究背景

2.1 Geant4の開発・利用

Geant4とは、粒子と物質の相互作用をシミュレーションするソフトウェアである。Geant4は1970年代から使われていたGeant3の後継のソフトウェアとしてCERNが研究開発を行った。Geant3も同様の目的で開発されていたが、実験装置の巨大化、複雑化などに伴い、新しい機能を盛り込むためのGeant3のアップデートが難しくなった。そこでCERNを始め、日本の高エネルギー加速器研究所（KEK）などの高エネルギー研究機関らがGeant4の開発に乗り出した。Geant4はこれまでGeant3がFORTRAN言語で書かれていた事に対し、Geant4はC++言語で書かれ、オブジェクト指向となった。Ver0.0は1998年に発表され、1998年末にver1.0の世界公開プロジェクトを終了したが、そこから有志が集まり、維持開発のための世界共同グループがバージョンアップが繰り返され、今では高エネルギー分野の他にも宇宙、半導体、医療分野にまで応用されるにいたった。現在でもKEK主催で勉強会などが開かれ、開発も行われている。以下に医療分野で応用されたGeant4の例を示す。

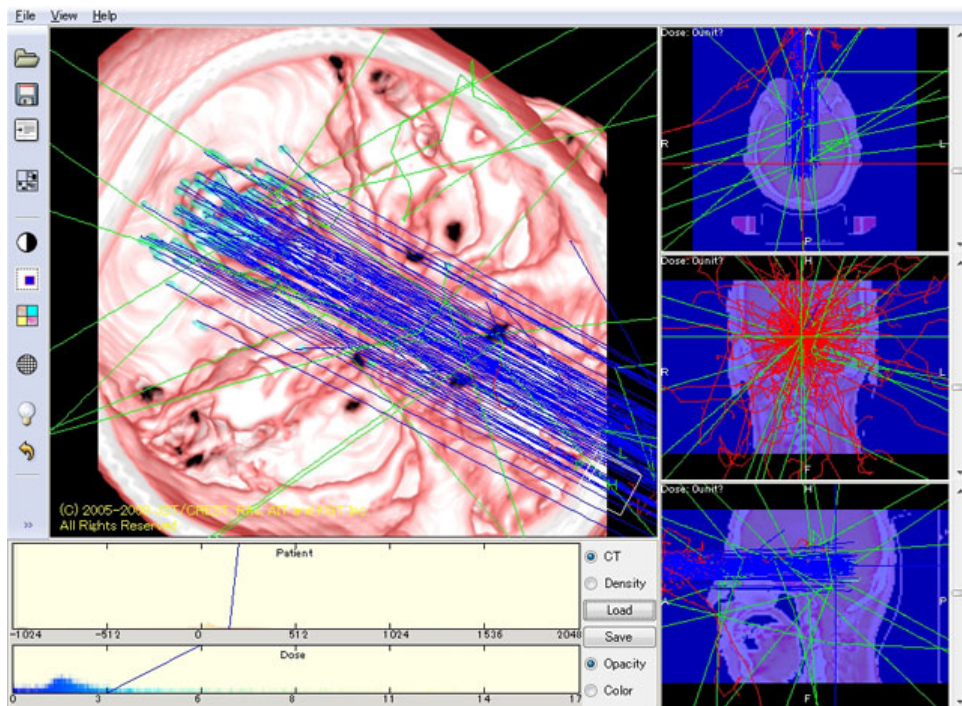


図 2-1 医療分野で応用された Geant4 のシミュレートの様子
放射線治療を的確にかう効率的に行うためシミュレートされている
これは腫瘍の治療のためのシミュレートである[*6]

2.2 Geant4 のオブジェクト指向

Geant4 はオブジェクト指向であると先述したがこれは、ユーザーは Geant4 のプログラムを全て把握する必要はなく、ユーザーは主に入射させる粒子、検出器の素材、形、位置、起こす事象を入力すれば、欲しい結果を入手できる。

2.3 Geant4 の動作

Geant4 は粒子が発射されることからシミュレーションの動作が始まる。粒子が物質に入射、物質と相互作用し 2 次粒子が生成され、別の粒子が様々な方向に飛散していく。飛跡は可視化することが出来、全ての粒子が、止まる（運動量 0 になる）あるいはユーザーが設定した空間の外に達した瞬間シミュレーションが停止する。この間にユーザーが設定した有感な物体（シンチレータ）に粒子が与えた情報を記録する。

第3章 シミュレーション

本研究では実際に ALICE 実験内で使われている PHOS 検出器の 1 モジュールを再現し、ガンマ線を入射し、落としたエネルギーのスペクトルを得る。その結果をまったく同じ条件でシミュレーションを行った Geant3 との結果と比較し、Geant4 が正確に動作していることを確認し、ユーザーが記述すべき Geant4 のプログラムについて解説を行う。

3.1 Geant4 を使う準備

まずは Geant4 を動かすためのパソコンであるが、これは個人用のデスクトップパソコンを使うことにした。用いたのは DELL 製のもので、スペックは以下の通りである。また、コンパイラなども以下に載せておく。

本体 Dell Dimension8200

CPU Intel(R) Pentium(R) 4 1.70GHz

使用可能メモリ 374M

OS SLC4

gcc コンパイラ 3.4.6

Linux を利用したので以下の説明は全て Linux で行われたことということに注意してもらいたい。

Geant4 をインストールするために参考にしたページは”Geant4 Installation Guides”(<http://geant4.slac.stanford.edu/installation/>) である。このページには OS が Linux、Macintosh OSX、Windows でのインストール方法を紹介している。

現在(2009年11月)で、公開されている Geant4 の最新バージョンは ver9.2.p02 であった。これを Geant4 のオフィシャルページ (<http://www.geant4.org/geant4/>) からダウンロードする。そしてさらにこれでは Geant4 は動かない。今回用いた Geant4 を動かすためのソフト CLHEP とそのバージョンは ver2.0.4.2 である。

CLHEP とは、粒子物理学において長らく用いられている、ライブラリである。つまり、用いるマトリクスの操作を行うもので、4次元空間と粒子物理の特性が書かれている。また注意したいのが、Geant4 と CLHEP は別々のソフトウェアで、Geant4 のために CLHEP が開発されたわけでも、CLHEP のために Geant4 が開発されたわけでもない。今後 Geant4 を動かすのに CLHEP がバージョンアップされれば互換性があることは保障されていない。よって CLHEP の付属文書を注意

して読む必要がある。

さて、CLHEP のダウンロードは ” Geant4 Installation Guides ” より、用いる Geant4 のバージョンに合わせてダウンロードを行える。ダウンロードする場所は、

`~home/okazaki` とした。

3.2 CLHEP のインストール

ディレクトリ `~` 内に新しいディレクトリ `CLHEP` というディレクトリを作成する。ここにダウンロードしてきたファイルを移動させて、

```
tar -zxvf clhep-2.0.4.2.tgz
```

で展開する。すると `~/CLHEP` 内に `2.0.4.2` というディレクトリが作成される。この `2.0.4.2` 内にさらに `CLHEP` というディレクトリが作成される。ここに移動し、`~/CLHEP/2.0.4.2/CLHEP` で

```
./configure --prefix /u/ey/perl/CLHEP
```

を実行する。すると、レスポンスにエラーがなければ展開が成功したということである。そして、

```
make
```

を実行し、コンパイルする。さらに

```
make install
```

を実行して、インストールを行う。すると `~/CLHEP/include` と `~/CLHEP/lib` というディレクトリができているはずである。できていれば、CLHEP のインストールは成功している。

3.3 Geant4 のインストール

Geant4 に関する参考にするページは CLHEP に引き続き同じである。ディレクトリ~/内に geant4 というディレクトリを作成。この中に geant4.9.2.p02.tgz をダウンロードし、

```
tar -zxvf geant4.9.2.p02.tgz
```

で展開する。すると~/geant4 内に

```
geant4.9.3.p02
```

というディレクトリが作成される。次に Data File をダウンロードする。先ほどできた geant4.9.2.p02 というディレクトリの中に

Data

というディレクトリを作成し、そこに

<http://geant4.web.cern.ch/geant4/support/source/G4EMLOW.6.2.tar.gz>
から Data file をダウンロードし

```
tar -zxvf G4EMLOW.6.2.tar.gz
```

を行い、展開する。Data file に関してはこれで以上である。次に引き続き Geant4 の設定を行う。以下の作業は慎重に、時間をかけて行う。(参考 URL には急がないでコーヒーかワインでショートブレイクを摂ってくださいと記されている。)では作業行う。

Print | grep G4

を行い、何か変数記号が出たらここまですべて何かを失敗している。展開中のメッセージをログに記録して目を通したりする必要がある。何もエラーがなければディレクトリ~/geant4/geant4.9.2.p02 に移動して

```
./Configure -built
```

を行う。以後いくつかの質問が出るが、ほぼデフォルトのまま答えてよい。(Enter キーを押せばよい。)ただし、以下のメッセージが表示されたときには設定の変更が必要である。

Could not find CLHEP installed on this system!

Please, specify where CLHEP is installed:

つまり Geant4 が CLHEP のインストール先を知らないのでここでその場所を記入しておく必要があるということである。これは /u/ey/perl/CLHEP にインストールされているので

/u/ey/perl/CLHEP

とタイプする。次は

G4VIS_BUILD_OPENGLX_DRIVER

G4VIS_USE_OPENGLX

について、インストールするか聞かれる。これはつまりビジュアライザとして OPENGLX を使うかどうかということである。これは

y

とタイプする。次は

G4VIS_BUILD_RAYTRACK_DRIVER

G4VIS_USE_RAYTRACERX

についてインストールするか聞かれる。これはつまり RayTracerX Visualization driver を使うかどうかということである。これは

y

とタイプする。

残りは全てデフォルトの設定にしておく。あとは表示される指示に従う。先ほど設定したことが行われるので設定が完了するまで待ち、それが終われば

./Configure

で、Configure が実行される。これでそれぞれの動作環境での設定を行う。
ディレクトリ `~/geant4/geant4.9.2.p02` 内に `env.sh` と `env.csh` が作成されているはずである。

次に環境変数の設定を行う。これは Shell が C Shell or TC Shell と Korn Shell の環境でコマンドが異なる。

まずシミュレーションを行うためのディレクトリが必要で、このディレクトリは Geant4 をインストールした場所とは別の場所に作る必要がある。よって `~/`以下に作成した。ディレクトリの名前は `g4work` とした。この work directory に環境変数を path させ、CLHEP の環境変数を設定するために、

C Shell TC Shell のユーザーは、

```
setenv G4WORKDIR ~/g4work
setenv LD_LIBRARY_PATH ~/CLHEP/lib/:$LD_LIBRARY_PATH
source ~/geant4/geant4.9.2.p02/env.csh
```

Korn Shell のユーザーは、

```
export G4WORKDIR ~/g4work
export LD_LIBRARY_PATH ~/CLHEP/lib/:$LD_LIBRARY_PATH
source ~/geant4/geant4.9.2.p02/env.sh
```

を実行する。なお、新しいターミナルを開いたときは毎回この環境変数を設定しなければいけない。

これで Geant4 を動作させる準備が整った。

3.4 Geant4 のチュートリアル

本研究はあらかじめ存在するプログラムを書き換えによる、各々の行いたいシミュレーションを行うことを目的としている。本研究で用いたプログラムは KEK がホームページ上で公開しているチュートリアルの exercise-3 である。なお、この KEK の公開しているチュートリアルは 1 ~ 3 まで公開されているが、全て行ってからプログラムの書き換えをしている。exercise-3 はサンドイッチ型カロリメータと呼ばれる、薄い板状のシンチレータとアブソーバーが交互に重なっているものである。カロリメータとは粒子と物質の相互作用を通して粒子のエネルギーを測定するものである。アブソーバーとは吸収層とも呼ばれ、入射された粒子はこの吸収層で電磁シャワーを起こし、発生した粒子がシンチレータで蛍光しその光を検出する。以下に KEK が公開しているチュートリアル、exercise-3 のジオメトリと e^+ を入射したときのビームの軌跡をシミュレートしたものを示す。

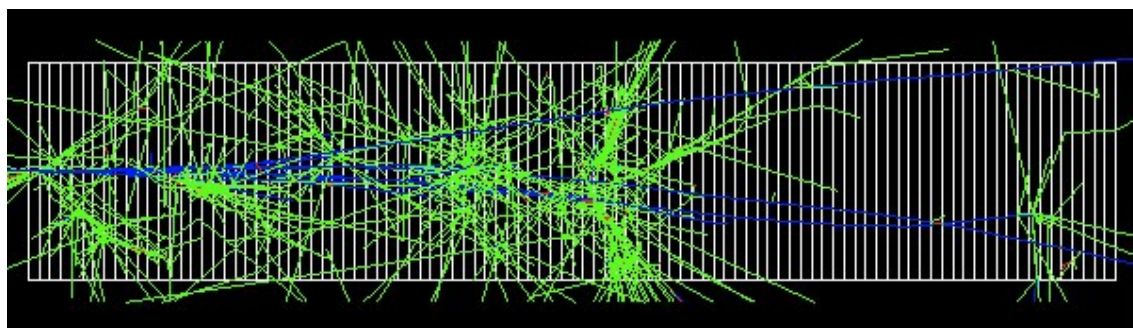


図 3-1 exercise-3 のシミュレートの様子

3.5 exercise-3

このチュートリアルでは先述の通り、Absorber (吸収層) と Gap (シンチレータ) から成っている。この Gap にはすでにシンチレーション光を検出し、ヒストグラムに書き出すまでのプログラムが作成されている。よってこの Gap の形とその素材をプログラムの書き換えによって PWO 結晶にし、配列する。これによって PHOS 検出器 1 モジュールを再現する。

第4章 プログラム書き換えの手法

第3章で Geant4 が正しく動いていることを確認できたので、この章ではプログラム書き換えの手法を一般的に述べていく。

ここでまず、exercise-3 のプログラムの構造を紹介する。

クラスは

ExN03DetectorConstruction

ExN03PhysicsList

ExN03PrimaryGenerationAction

ExN03DetectorMessenger

ExN03EventActionMessenger

ExN03PrimaryGeneratorMessenger

ExN03EventAction

ExN03RunAction

が存在する。

Geant4 を用いたシミュレーションを行うためには、

DetectorConstruction

PhysicsList

PrimaryGenerationAction

の3つのクラスが必要である。

それぞれの内容は

DetectorConstruction

ジオメトリ

PhysicsList

粒子の種類

物理プロセス

PrimaryGenerationAction

発生させるイベントの初期状態

が記述されている。

今回主に書き換えを行ったのは
ExN03DetectorConstruction
である。

このクラスでは検出器の大きさ・形・配置を決定する。以下に、検出器の形を決めて、その単純な箱でしかない内部を物質（水色のもの）で満たし、最後に位置と回転を与えてやることで検出器のジオメトリが決まることを

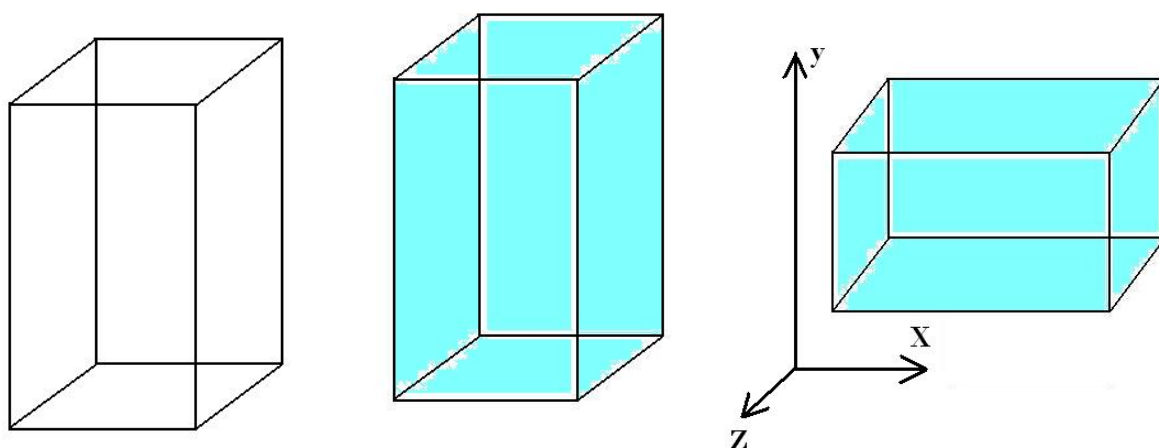


図 3-2 検出器の形がまず決定（左）
次に満たされる物質が決定（中央）
最後に位置と回転が与えられる（右）

形を決めるために Geant4 では 3 つの立体表現を用意している。

Constructed Solid Geometry (CSG)

基本的な立体要素を組み合わせる立体表現で輸送計算が高速であるが、変則的な形状を記述するには不向き。

Boolean Operation

複数の CSG を用いて立体の足し算引き算を行い、立体を記述する。輸送計算には時間がかかる。

Boundary Representation (BREPS)

最も自由度が高く、複雑な形状を面の関数を用いて表現する。しかしその分、

輸送計算には時間がかかる。

本研究では立方体の並んだものを記述するので CSG を使用した。

次に定義した形状を物質で満たすために物質情報を追加する。そしてそれを配置してジオメトリは完成する。

次に使用する粒子だが、Geant4 は粒子を 6 種類のグループに分け、定義されている。それは、

Lepton

Boson

Meson

ShortlivedParticle

Baryon

Ion

である。

実際に入射したとき、粒子と物体は様々な相互作用を起こす。これも Geant4 で実際に起こすかどうかを設定できる。内容は

Electromagnetic process

Hadronic process

Decay process

Others(Transport process etc...)

である。今回は起こりえる相互作用は全て起こすように設定されている。

4.1 構成物質の設定

前章で述べたように Gap にはシンチレーション光を検出するプログラムがすでに書き込まれている。よってこの Gap を PW04 で満たすことにする。

そのため、まず ExN03DetectorConstruction.cc 内で PW04 を定義する。

Geant4 における物質の定義は元素を定義し、その元素を化合物 (ex CO₂) として合成したり、混合物(ex 空気)として合成する。そして物質の定義は主に 3 つのクラスから構成されている。同位体が G4Isotope、元素が G4Element、分子、化合物、混合物が G4Material で定義される。

そこでまず PW04 を構成している、Pb(鉛),W(タングステン),O(酸素)を定義する。すでにチュートリアルのマクロは Pb と O は定義されているので W を定義する。

元素が定義できたのでこれらを用いて PW04 を定義する。

G4Material の項目が G4Element の下にあるので新しく Pb,W,O を用いる。

Ncomponents は使用する元素の種類の数を示すものである。この場合は Pb と W と O なので ncomponents=3

4.2 大きさ・形・物質・位置の設定

書き換えるファイルは先ほど同様に ExN03DetectorConstruction.cc である。

そのために用意されているのは

G4VSolid 形状、寸法を与える

形は先ほどの通り、CSD、

G4LogicalVolume 満たす物質、その daughterVolume と motherVolume を与える

G4VPhysicalVolume 位置、回転を与える

daughterVolume とは、今このチュートリアルにおいて Gap と Absorber が 1 つセットになって Layer と呼ばれる Volume になっており、その Layer が 50 個並んだものが Calorimeter と呼ばれる Volume になっており、その Calorimeter が World と呼ばれている Volume 内に入っている。以下に示すのは exercise-3 のジオメトリと、Absorber、Gap、Layer、Calorimeter、World の motherVolume、daughterVolume としての関係を表したものである。

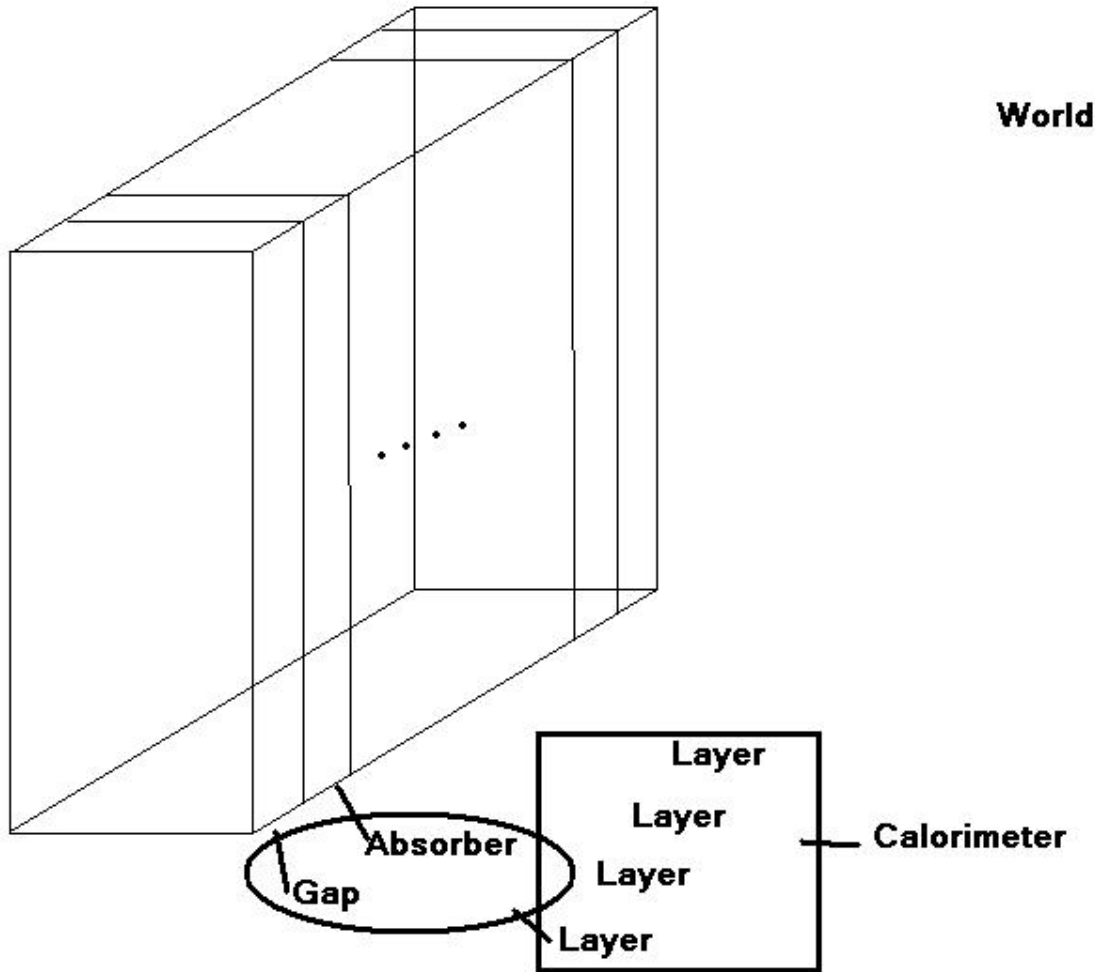


図 3-3 exercise-3 の daughterVolume と motherVolume の関係

Gap, Absorber の motherVolume が Layer
 Layer の daughterVolume が Gap, Absorber, Layer の motherVolume が Calorimeter
 Calorimeter の daughterVolume が Layer, Calorimeter の motherVolume が World
 World の daughterVolume が Calorimeter
 となっている。

よって Gap を solidGap で $22\text{mm} \times 22\text{mm} \times 180\text{mm}$ の大きさにし、logicGap で solidGap 内で定義したクリスタルを PWO で満たし、physiGap で配置する。

クリスタルとクリスタルの隙間は 22.55mm あるので Absorber も先ほどと同様に大きさを $22.55\text{mm} \times 22.55\text{mm} \times 180\text{mm}$ とし、真空で満たし、PWO クリスタルとなった Gap と Absorber を 2 つセットにする。

次に Layer の構造を決定する。これはクリスタルと真空を 1 つずつセットにしたものなので solidLayer で Layer の厚さを 0mm に設定。

logicLayer で Layer 内を満たす物質を真空に設定。

physiLayer で 56×64 個配置し、motherVolume を World に設定する。

4.3 入射粒子の設定

これは PrimaryGeneratorAction に記述されるものだが、この exercise ではすでに gun.mac で見やすいように入射する粒子の種類、エネルギー、方向が示されている。よって必要に応じて書き直せばよい。今回は光子を入射するので gamma と表記する。

以上で今回必要なプログラムの書き換えが終了した。

このシミュレーションを実行するため、まずは make を行う。

```
make
```

で、make ファイルを更新し、実行する。

```
./bin/Linux-g++/ex03
```

で実行。その後、

```
/control/execute vis.mac
```

ジオメトリを読み込み、

```
/control/execute gun.mac
```

入射粒子の情報を読み込み、

```
/runBeamOn n
```

n 個の粒子を発射する。

すると ROOT ファイルが作成されるので、

```
exit
```

で、実行画面から出て、

```
a=TBrowser()
```

で ROOT フォルダを開く。するとヒストグラムを見ることができる。

3.6 シミュレーションの結果

シミュレーションの条件は Geant3, Geant4 共に、1) 運動量 50GeV で光子を 1000 個発生させ、PHOS 検出器に入射させる事象、2) 運動量 100GeV で光子を 1000 個発生させ、PHOS 検出器に入射させる事象の 2 通りシミュレートした。落としたエネルギーを横軸、クラスターの数縦軸に表しヒストグラムにした。そのヒストグラム、4 つを以下に示す。

1) 光子のエネルギーが 50GeV のとき

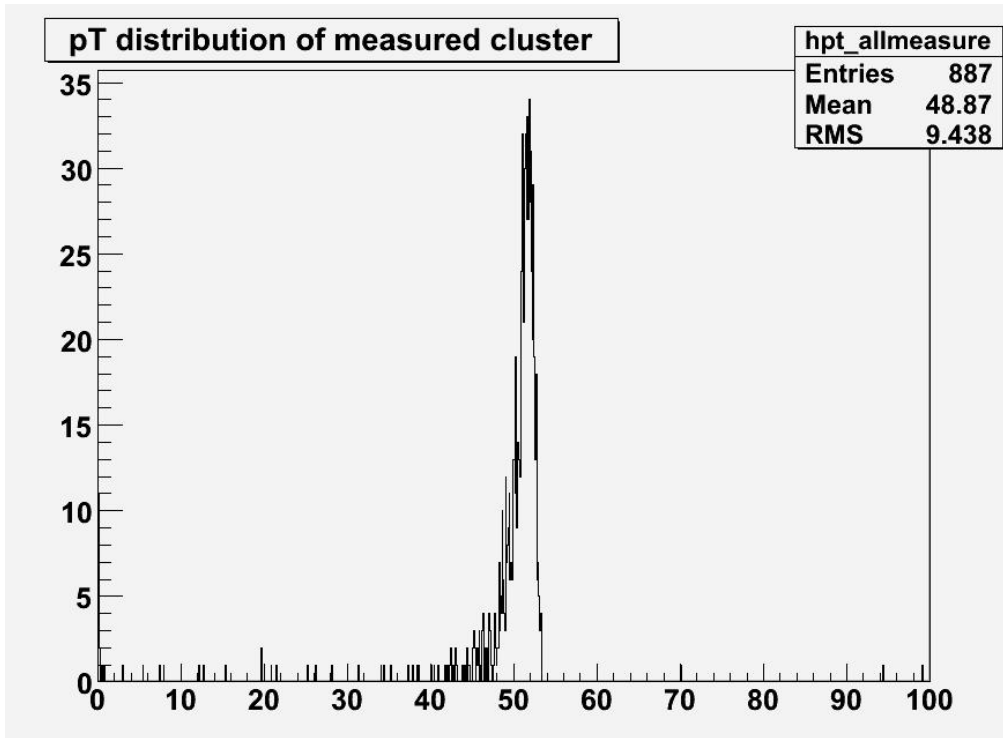


図 3-4 Geant3 50GeV の光子を入射したときのヒストグラム

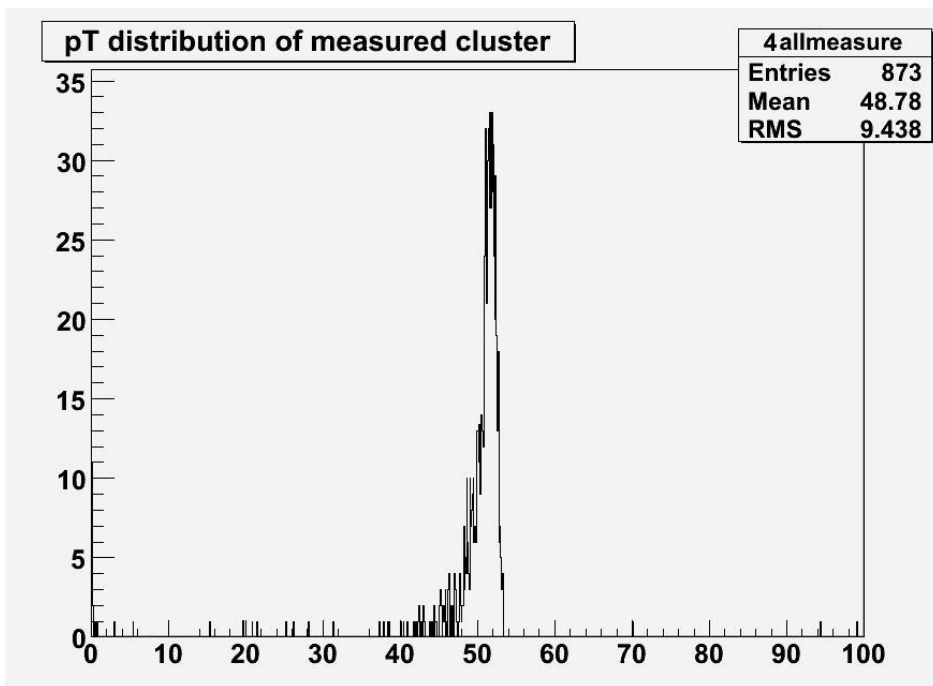


図 3-5 Geant4 50GeV の光子を入射したときのヒストグラム