Geant4 Manual

Ver 0.4

山形大学理工学研究科博士前期課程 1 年
伊藤 祐輔

2001 年 2 月 27 日

目 次

概 要

この冊子では、Geant4 の簡単な使い方を書こうと思います。ただ、書いている本人もまだまだ分からない事だらけなので、分かった部分やもっといい方法が分かった場合には随時書き換えて行くつもりです。
とりあえず今のところ参考としているのは正規の Geant4 マニュアルです。正直に言うと、今書いているこれを読むよりも既存のマニュアルを直接読む方がすごく分かりやすいです。もしこのマニュアルが欲しい方は

http://wwwinfo.cern.ch/asd/geant4/G4UsersDocuments/Overview/html/index.html

まで行けば、手に入ると思います。インストール方法等についての document がありますので、そちらの資料が欲しい方もこちらを参考にして下さい。ただし、置いてある場所が CERN である事かも分かるとは思いますが、とりあえず全部英語ですので英語の苦手な方は翻訳ソフトでも使って下さい。お奨めは、Windows の翻訳サーフィンです。ナイスな翻訳が楽しいのでお奨めです。
さて、このマニュアルで紹介する内容としては、

1. geometry の書き方
2. データの取り出し方
3. その他

を考えています。

1 は主に geometry を書く手順を紹介します。
使用するファイルは Calorimeter.cc や Counter.cc などです。ただし、これは書きたい対象により様々な名前でさまざまな書き方で表されるものであって、一概に決められるものではありません。これについてはその時に述べる事とします。

2 は EventAction.cc に主に書かれる内容です。ここは、正直まだかなり black box の状態なのでこれからの改善を期待するパートだと思って下さい。

3 では compile の仕方、実際の走らせ方などを記して行こうと思います。ここもまだ分からない事だらけなので、逐次更新していきます。

第1章　さまざまな設定

1.1　geometry の設定

Geant4 とは、ある検出器の response を調べる為の simulation tool です。これを実行する為には、大まかに以下のような手順を踏む必要があります。

⋆ solid を作る

⇓

⋆ solid を使って logical volume を作る

⇓

⋆ logical volume を使って physical volume を作る

⇓

⋆ どこから energy を取り出すのかを決める

⇓

⋆ compile,link をする　(失敗)⇒ 作りなおし

(成功)

⇓

⋆ さまざまな値を与えて実行する

以上のような手順で、実際の simulation を行う事が可能となります。これらについては以下で詳しく説明します。

1.1.1 solid を作る

まずはsolid を作る。solid とはある決まった形を持った物質そのものである。solid が持つのは

1. 大まかな形
2. それらを記述する為の大きさの情報

の情報である。これらは以下のように記述する。

```
// define solid
G4Box *baf2Solid= new G4Box("BaF2",dxyzBaF2.x()/2.,
                              dxyzBaF2.y()/2., dxyzBaF2.z()/2.);
```

これは baf2Solid と言う名前の四角い箱を作っています。その事は *G4Box* クラスのオブジェクトとして baf2Solid を作っている事から分かります。それの大きさは x 方向は dxyzBaF2.x()/2.,y 方向は dxyzBaF2.y()/2.,z 方向は dxyzBaF2.z()/2. の大きさを持っている事がこれから分かります。これらの変数は事前に定義しておく必要がありますが直接ここに値を入れても一向に構いません。ただし、他でこの大きさを参照する場合に変数で扱っておいた方が変更も楽に行なえるのでなるべく変数として扱う事をお奨めします。
また、変数それぞれを 2 で割っているのは、Box の各辺はある点を中心としてそこから半分ずつ伸びており、それの両方の和として (一つの辺の長さとして) 変数の値に等しい長さを持つのである。これには注意してほしい。

1.1.2 logical volume を作る

次に logical volume を作ります。これには予め solid を作っておく必要があり、これがないと作ることは出来ない。手順を守って作りましょう。logical volume は以下のようにして作る事が出来ます。

```
// define logical volume
G4Material *baf2= materialMgr-> GetMaterial("BaF2");
G4LogicalVolume* baf2LV= new G4LogicalVolume(baf2Solid, baf2, "BAF2_LV");
```

まず、初めに"baf2"と言う物質を定義しています。これはあとで使う為です。名前等は好きにつけて下さい。ただし、あとで分からなくなったりしないようなものにした方が良いです。。。
このように、GEANT の simulation 内で使う物質は全て使う前に定義しないといけません。ただし実際に化合物等を作るのは"Materials.icc"と言う file 内です。あとは、各々使う時に MaterialManager クラスのオブジェクト (上で言う materialMgr)

から欲しい物質を引っ張り出してきて名前を割り当てるだけでOKです[1]。
これを使ってそれぞれのG4Materialクラスのオブジェクトに代入しましょう。これでlogical volumeを作る為に必要なものはそろいました。よってlogical volumeを作ります。これには

さっき作ったsolidの名前

さっき作った物質名

今から作るlogical volumeの呼び名

と言う情報が入ります。最後の呼び名以外は既に作ってあるものの名前なので間違わないように入れましょう。これでlogical volumeが作られた事になります。

1.1.3 physical volumeを作る

次にphysical volumeを作ります。これにより、自分の作りたい物質を実際に配置する事までが可能となります。これは以下のようになります。

```
// make physical volumes and place them
 G4int copyNo= 0;
     G4PVPlacement* baf2PV=
      new G4PVPlacement(0, G4ThreeVector(0., 0., 0.,),
                        "BaF2_PV", baf2LV,
                        world->GetDownArea(),FALSE, copyNo);
```

これらをそれぞれの部分ごとに説明します。
最初の"copyNo"とは、整数として定義されていて、初期値として0が代入されています。このように変数は初期化しておく事が大事です。忘れないようにしましょう。
それで次は本体を作ります。
まず"G4PVPlacement"クラスのオブジェクトを作り、その名前はbaf2PVとしています。そしてそれには

△ 回転するかしないか(するならその大きさも)

△ その配置位置(x,y,z全部の情報)

△ Physical Volumeの呼び名

△ 使用するlogical volumeの名前

△ 配置するmother volumeのphysical volume名

[1]MaterialManagerクラスのオブジェクト(materialMgr)を実際に作っているのはhead fileですので、詳しくはそちらを見て下さい

△ bool 演算子 (今のところ全くの意味不明)

△ copyNo(この物体の physical volume に割り当てる数)

と言う情報が含まれます。実際に代入している順番とここに書いた説明とは一対一対応するはずなので、それぞれ参考にしてみて下さい。
ここで”配置する mother volume”と書いた場所について説明する。
まず物体を考えた場合に、それを配置する為にはかならず元となる場所が必要となる。ところがそれを繰り返すと一番おおもとのものが必ず必要となります。これを”world volume”と言います。これは”配置する mother volume”を書かず、かつ配置する mother volume の名前のところは”0”だけでよいです。各 physical volume を配置する為にもかならず一つは”world volume”を作りましょう。

```
// make world volume
     G4PVPlacement *world_volume=
      new G4PVPlacement(0, G4ThreeVector(0., 0., 0.,),
                        "worldPV", worldLV,0,false, copyNo);
```

これが world_volume が出来上がります。

以上のようにして、自分が使いたい検出器を配置する事が出来ます。これを繰り返せば好きなものを好きな位置に置く事が可能となります。

1.2 Sensitive Detector

これで検出器の大きさ、位置等の情報は書けるようになりましたので次にそれらが実際に情報を取れるようにします。
これを設定するクラスは Calorimeter に対しては CalorimeterSD,Counter に対しては CounterSD など、名前の後ろに SD を付けたものを使う場合が多いです。この SD とは”<u>S</u>ensitive <u>D</u>etector”の頭文字を使ったものです。これらの記述の例をここで挙げたいと思います。

第2章　さまざまな相互作用過程

Geant4 で使う事の出来る相互作用には、電磁相互作用とハドロン相互作用がある。