

Monte Carlo ㇏ ㇏ 2

K. Miwa

GEANTって何？

- CERNで主に開発された素粒子・原子核実験のシミュレーション用に開発されたツール
- 粒子を発生させ、様々な物理プロセスにしたがって粒子をtransportする。
 - 磁場や電場の中での粒子の運動
 - 電磁相互作用
 - ハドロン相互作用
 - 崩壊
- 検出器を配置して、その中でのエネルギー損失や、粒子の通過した場所などを調べることができる。

でも、よくよく考えてみると

- 粒子を発生させ、様々な物理プロセスにしたがって粒子をtransportする。
 - 磁場や電場の中での粒子の運動
 - 微分方程式を自分で解けば粒子の軌跡は計算できる
 - 電磁相互作用
 - 教科書に載っているプロセスで自分で計算できる
 - ハドロン相互作用
 - これも反応断面積が分かればあとは運動学を解くだけ
 - 崩壊
 - これも角度分布が分かればあとは運動学を解くだけ

GEANTは便利なツールだけれども、それを用いた結果が妥当であるかどうかをチェックできる目を身につけないといけない。
GEANTのツールを組み合わせる我々はすぐにミスしてしまうから。

What is Geant ?

- Monte Carlo simulation tool for nuclear and particle physics
- Set up detectors
 - Sensitive detector which returns useful information such as energy deposit and hit position.
- Take into account physical processes which you register
 - EM process
 - Decay process
 - Hadronic process
- Transport particle according to the equation of motion at the field.
 - Electric field
 - Magnetic field

GEANTの準備

- すでにdhcp2576にはGeant4がインストール済み
 - version 10.05
- 自分のPCや、farmなどの自分のlocalディレクトリにGeant4をインストールしたい人は以下のインストールメモを参考にしてください。

https://1drv.ms/u/s!AmounVSpTEfRgQD2-_v881RZ8_mg

The screenshot shows a OneNote page titled "E40 parameter guide". On the left, there is a table of contents with the following items:

BGO	CMakeのインストール
CFT	CLHEPのインストール
Geant4のインストール	Expatのインストール
Geant4のversion up	Qtのインストール
	Freetypeのインストール
	DAWNのインストール
	Geant4のインストール
	Geant4の環境変数の設定
	Geant4のX がこける問題
	Geant4のユーザープログ...

The main content area shows a section titled "CMakeのインストール" dated "2018年12月19日 水曜日 9:31". The text reads:

Geant4のversionを上げようと決意をする。
Geant4のホームページのインストールガイドを見ているとまず、cmakeの要件が書いてある。

- CMake 3.3 or higher.

On Linux, we recommend that you use CMake as provided through the package management system of your distribution, unless it does not meet the minimum version requirement. In that case, we recommend you install it using the Linux binary installer for the latest version of CMake, available with instructions from [the Kitware download site](#). This installer is highly portable and should work on the vast majority of distributions.

On macOS and Windows, CMake is not installed by default, so we recommend that you install it using the most recent Darwin64 dmg (macOS) or Win32 exe (Windows) installers supplied by [the Kitware download site](#). On macOS, you may also use the [Homebrew](#) or [Macports](#) package managers to install the required version.

For more information on CMake, the [CMake Help and Documentation](#) should be consulted.

とあったので、cmakeをインストールする。

Geant4の準備

- geant4のパッケージは

- /home/sks/prog/geant4.10.05

- にありその中に右の3つのディレクトリがある

```
[sks@dhcp2576 geant4.10.05]$ ls  
build  install  source
```

- build : Geant4のライブラリをコンパイルするために使用したディレクトリ
- install : ユーザーが各自のプログラムを作るときにリンクするライブラリなどが入っている。また環境変数を設定するためのスクリプトも入っている。
- source : Geant4のソースファイルなどが入っている。またGeant4のプログラムの例題なども入っている。

Geant4の準備

- 環境変数の設定
- Geant4を使ったプログラムをコンパイルしたり、実行するときに必要な環境変数を、terminalに設定する必要がある。
- 以下を実行
 - > cd /home/sks/prog/geant4.10.05/install/bin
 - > source geant4.sh
 - このgeant4.shを実行することによって、これを行ったterminalにおいて必要な環境変数の設定がなされた。
- 環境変数の設定を忘れて、geant4のプログラムを実行すると以下のようにエラーとなるので注意。

```
[miwa9@localhost G4_work]$ ./exampleB2a
Available UI session types: [ GAG, tcsh, csh ]

----- EEEE ----- G4Exception-START ----- EEEE -----
*** G4Exception : PART70000
        issued by : G4NuclideTable
G4ENSDFSTATEDATA environment variable must be set
*** Fatal Exception *** core dump ***

*** Break *** segmentation violation
```

Geant4の準備

- Geant4に用意されている例題
 - 以下のディレクトリにGeant4が提供している例題がいくつも入っている。自分で作るプログラムの参考になると思うので、いくつか読んでみることをお勧めします。
 - /home/sks/prog/geant4.10.05/source/examples

```
[sks@dhcp2576 examples]$ ls
advanced  extended  GNUmakefile  README  README.HowToRunMT
basic     CMakeLists.txt  History      README.HowToRun  novice
```


Geant4の準備

- 例題の実行
- 最近(?でもない)は、Geant4のプログラムはソースファイルなどを入れるディレクトリとコンパイルや実行をするディレクトリを明確に分ける
- ソースファイルが入っているディレクトリを**_src, コンパイルするディレクトリを**_workと表示する。
- 以下のようにGeant4のexampleのうちのB1をコピーして作る。
 - > cd usr/**/MonteCalro
 - > mkdir geant4 先週作ったMonteCalroのディレクトリの中にgeant4を作る
 - > cd geant4
 - > mkdir B1 geant4のディレクトリ内にB1というディレクトリを作る
 - > cd B1
 - > cp -r /home/sks/prog/geant4.10.05/source/examples/basic/B1 . Geantの例題のうちB1のディレクトリをコピー
 - > mv B1 B1_src 先ほどコピーしたB1ディレクトリはソースが入っているので、B1_srcと名前を変更
 - > mkdir B1_work B1をコンパイルするために使用するディレクトリを作成
 - B1というディレクトリ内にはB1_src, B1_workという2つのディレクトリが存在している

```
[sks@dhcp2576 B1]$ ls  
B1_src  B1_work  _
```

Geant4の準備

- 例題の実行(続き)
 - > cd B1_work
 - cmakeコマンドを使ってB1_workディレクトリ内にプログラムをコンパイルする環境を作る
 - > cmake -DGeant4_DIR=(Geantのライブラリのディレクトリ) (ユーザープログラムのソースディレクトリ)

```
[sks@dhcp2576 B1_work]$ cmake -DGeant4_DIR=/home/sks/prog/geant4.10.05/install/lib64/Geant4-10.5.0 ../B1_src
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/sks/user/miwa/MonteCarlo/geant4/B1/B1_work
```

Geant4の準備

- 例題の実行(続き)

- cmakeを実行するとB1_workディレクトリ内に以下のファイルが出来上がっている

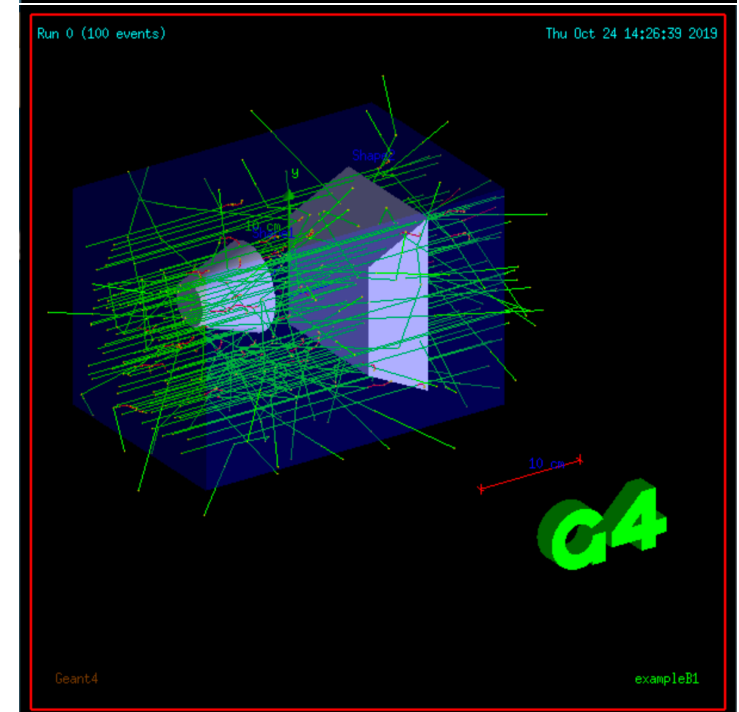
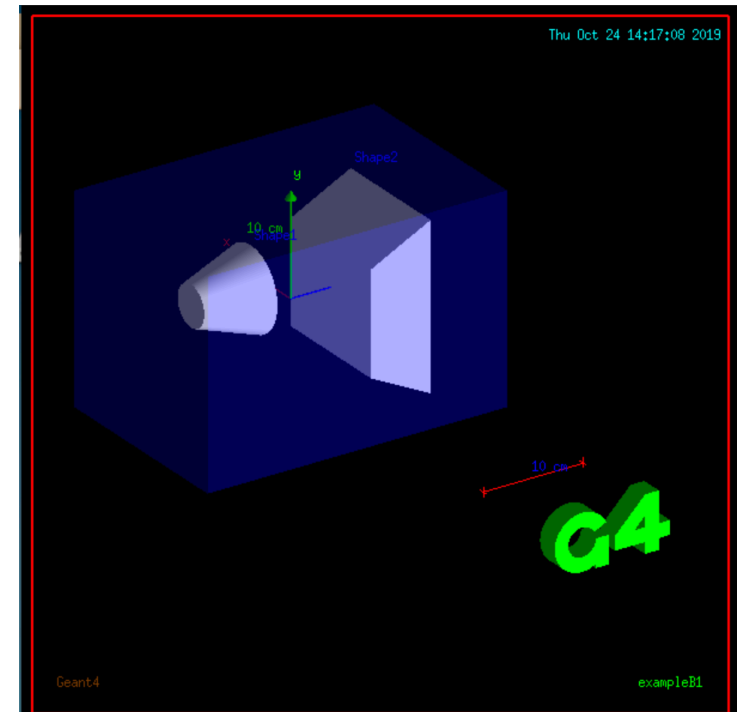
```
[sks@dhcp2576 B1_work]$ ls
CMakeFiles          Makefile           exampleB1.in      init_vis.mac      run2.mac
CMakeCache.txt     cmake_install.cmake  exampleB1.out    run1.mac          vis.mac
```

- cmakeコマンドでMakefileが作成される。またプログラムの実行に使われるmacroファイルも一緒にコピーされる。
 - これらはB1_srcディレクトリのCMakeLists.txtに記述されている(と思われる)
- ここまでエラーなどなければmakeでプログラムをコンパイル
- > make
- 実行ファイル(example01)が作られている

```
[sks@dhcp2576 B1_work]$ make
Scanning dependencies of target exampleB1
[ 12%] Building CXX object CMakeFiles/exampleB1.dir/exampleB1.cc.o
[ 25%] Building CXX object CMakeFiles/exampleB1.dir/src/B1ActionInitialization.c
c.o
[ 37%] Building CXX object CMakeFiles/exampleB1.dir/src/B1DetectorConstruction.c
c.o
[ 50%] Building CXX object CMakeFiles/exampleB1.dir/src/B1EventAction.cc.o
[ 62%] Building CXX object CMakeFiles/exampleB1.dir/src/B1PrimaryGeneratorAction
.cc.o
[ 75%] Building CXX object CMakeFiles/exampleB1.dir/src/B1RunAction.cc.o
[ 87%] Building CXX object CMakeFiles/exampleB1.dir/src/B1SteppingAction.cc.o
[100%] Linking CXX executable exampleB1
[100%] Built target exampleB1
[sks@dhcp2576 B1_work]$ ls
CMakeFiles          cmake_install.cmake  exampleB1.out      run2.mac
CMakeCache.txt     exampleB1            init_vis.mac       vis.mac
Makefile           exampleB1.in         run1.mac
```

Geant4の準備

- 例題B1の実行
 - `> ./exampleB1`
 - `./exampleB1` でプログラムを実行
 - 右のようなディスプレイが出てればOK
 - もしエラーが出たら次のページを参考に
 - geant上のコマンドプロンプトで
 - `IDLE> /run/beamOn 1000`
 - とするとイベントが発生する



Geant4の準備

- ディスプレイ関係のエラー

- 原因は特定できていないが、しばしばOpenGLのディスプレイを表示するときエラーが出る。

```
# Use this open statement to create an OpenGL view:
/vis/open OGL 600x600-0+0
/vis/sceneHandler/create OGL
/vis/viewer/create !! 600x600-0+0
G4OpenGLXViewer::G4OpenGLXViewer failed to allocate a standard colormap.
X Error of failed request: BadMatch (invalid parameter attributes)
Major opcode of failed request: 1 (X_CreateWindow)
Serial number of failed request: 39
Current serial number in output stream: 46
```

vis.mac内でOpenGLを開くところを変更してみる

```
# Use these open statements to open selected visualization
#
# Use this open statement to create an OpenGL view:
#/vis/open OGL 600x600-0+0
#
# Use this open statement instead for OpenGL in immediate mode.
vis/open OGLIX
#
# Use this open statement to create an OpenInventor view:
#/vis/open OI
#
```

LinuxでのエラーはこれでOKだった

MacでXを飛ばすとエラーが出るときには解決しなかった。

先頭に#を入れてコメントアウト

ここを追加で記入

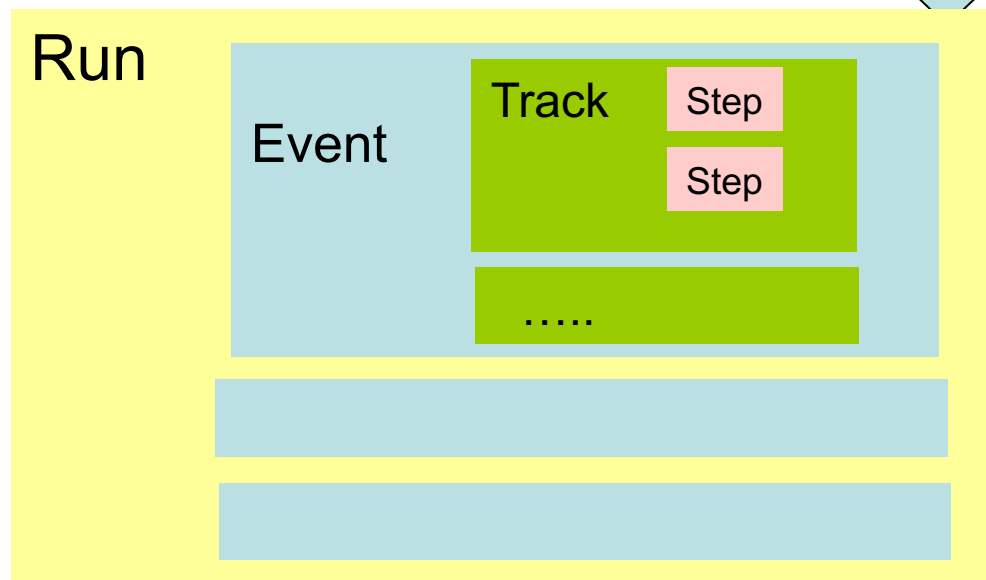
Geant4の概要説明

- 非常に簡単な例題を用意しているので、これをコピーする。
 - > cd user/**/MonteCarlo/geant4
 - > mkdir N00
 - > cd N00
 - > cp -r /home/sks/user/miwa/MonteCalro/geant4/N00/N00_src .
 - > mkdir N00_work
 - > cd N00_work
 - > cmake -DGeant4_DIR=/home/sks/prog/geant4.10.05/install/lib64/Geant4-10.5.0 ../N00_src

```
[sks@dhcp2576 N00]$ cmake -DGeant4_DIR=/home/sks/prog/geant4.10.05/install/lib64/Geant4-10.5.0 ../N00_src
```

- > make

Overview of program



Start simulation
> beamOn #eventNv

- Start simulation
- Detector set up
 - Physics list taken into account
 - One event consists of many tracks
 - One Tracks consists of many steps

Registor of each manager

- Example of ExN00
- exampleN00.cc (main function)
 - Define run manager
 - Resister other managers which define “Detector”, “Physics” , “PrimaryGeneratorAction” and “EventAction”
 - The simulation is executed based on the registered processes.

```
// Construct the default run manager
G4RunManager * runManager = new G4RunManager;

// set mandatory initialization classes
ExN00DetectorConstruction* detector = new ExN00DetectorConstruction;
runManager->SetUserInitialization(detector);
runManager->SetUserInitialization(new ExN00PhysicsList);

// set user action classes
runManager->SetUserAction(new ExN00PrimaryGeneratorAction());
ExN00EventAction* eventaction = new ExN00EventAction;
runManager->SetUserAction(eventaction);
```


Detector setting (Material)

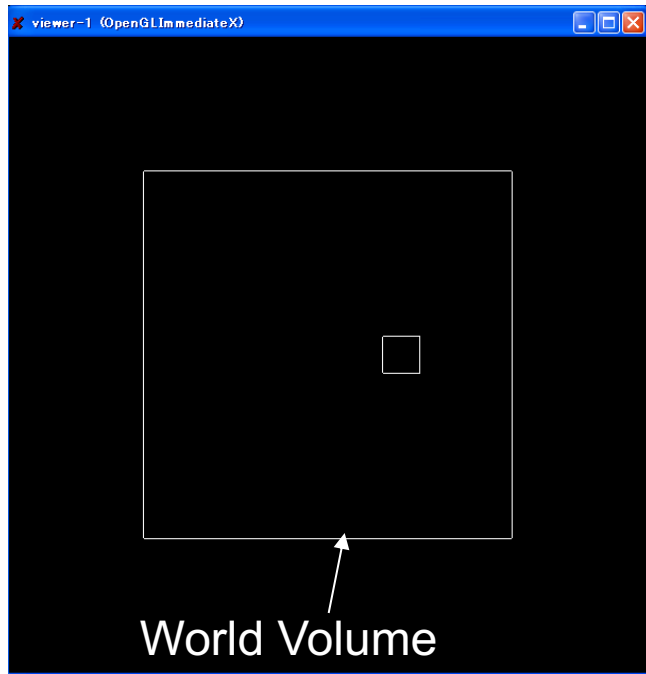
- We have to define a material of detector.
- G4Element (describes the properties of the atoms)
 - Atomic number
 - Atomic mass
- G4Material (describes the macroscopic properties of matter)
 - Density
 - State,
 - temperature

```
G4Element* H = new G4Element("Hydrogen",symbol="H" , z= 1., a= 1.01*g/mole);  
G4Element* C = new G4Element("Carbon" ,symbol="C" , z= 6., a= 12.01*g/mole);;
```

```
new G4Material("Aluminium", z=13., a=26.98*g/mole, density=2.700*g/cm3);  
new G4Material("liquidArgon", z=18., a= 39.95*g/mole, density= 1.390*g/cm3);  
new G4Material("Lead" , z=82., a= 207.19*g/mole, density= 11.35*g/cm3);
```

```
// define a material from elements. case 1: chemical molecule
```

```
G4Material* Sci =  
  new G4Material("Scintillator", density= 1.032*g/cm3, ncomponents=2);  
Sci->AddElement(C, natoms=9);  
Sci->AddElement(H, natoms=10);
```



Detector setting (Detector)

- At first, we define the largest volume, “World Volume”
- The other volumes are created and placed inside previous volumes, including in the World Volume

Setting

- Shape and size of volume (detector)
- Material of detector
- Setting position

SolidVolume

(define the shape and size,
Box, Tube, Polycon)

LogicalVolume

(define nature of the detector)

PhysicalVolume

(define the setting position at
the mother volume)

```
G4double WorldSizeX=1.0*m;
G4double WorldSizeYZ=1.0*m;

solidWorld = new G4Box("World",           //its name
                      WorldSizeX/2,WorldSizeYZ/2,WorldSizeYZ/2); //its size

logicWorld = new G4LogicalVolume(solidWorld, //its solid
                                 Air,         //its material
                                 "World");    //its name

physiWorld = new G4PVPlacement(0,         //no rotation
                              G4ThreeVector(), //at (0,0,0)
                              logicWorld, //its logical volume
                              "World",    //its name
                              0,         //its mother volume
                              false,     //no boolean operation
                              0);       //copy number
```

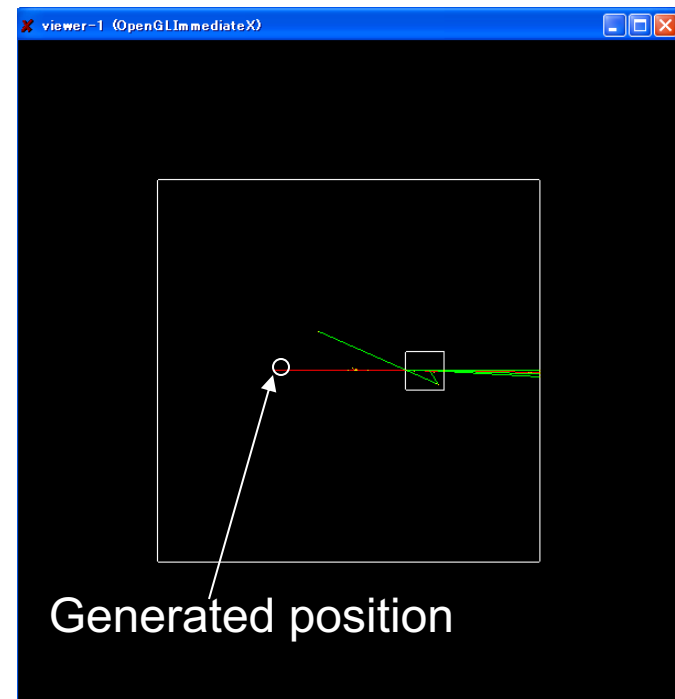
Primary Generation Action

- We have to specify how a primary event should be generated
- Kind of particle, energy, generated position, direction etc..
- G4ParticleGun generates primary particle(s) with a given momentum and position

```
G4int n_particle = 1;
particleGun = new G4ParticleGun(n_particle);

// default particle kinematic

G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
G4String particleName;
G4ParticleDefinition* particle
    = particleTable->FindParticle(particleName="e-");
particleGun->SetParticleDefinition(particle);
particleGun->SetParticleMomentumDirection(G4ThreeVector(1.,0.,0.));
particleGun->SetParticleEnergy(50.*MeV);
particleGun->SetParticlePosition(G4ThreeVector(-20.0*cm,0.*cm,0.*cm));
particleGun->GeneratePrimaryVertex(anEvent);
```



Commands of Geant4

- Start a run
 - Idle> /run/beamOn 1000
 - 1000 events are generated
- Execute macro file
 - Idle> /control/execute vis.mac
- Change view point
 - Idle> /vis/viewer/set/viewpointThetaPhi 45 45
- Finish Geant
 - Idle> exit

Problem

- Execute these example programs
- Modify ExN00
 - Change material of calorimeter
 - Scintillator → C, Fe, H₂O, Pb etc.
 - Change size and shape of calorimeter
 - Box → Tube etc.
 - Change primary particle
 - “e-” → “pi-”, “kaon-”, “proton”
 - Energy