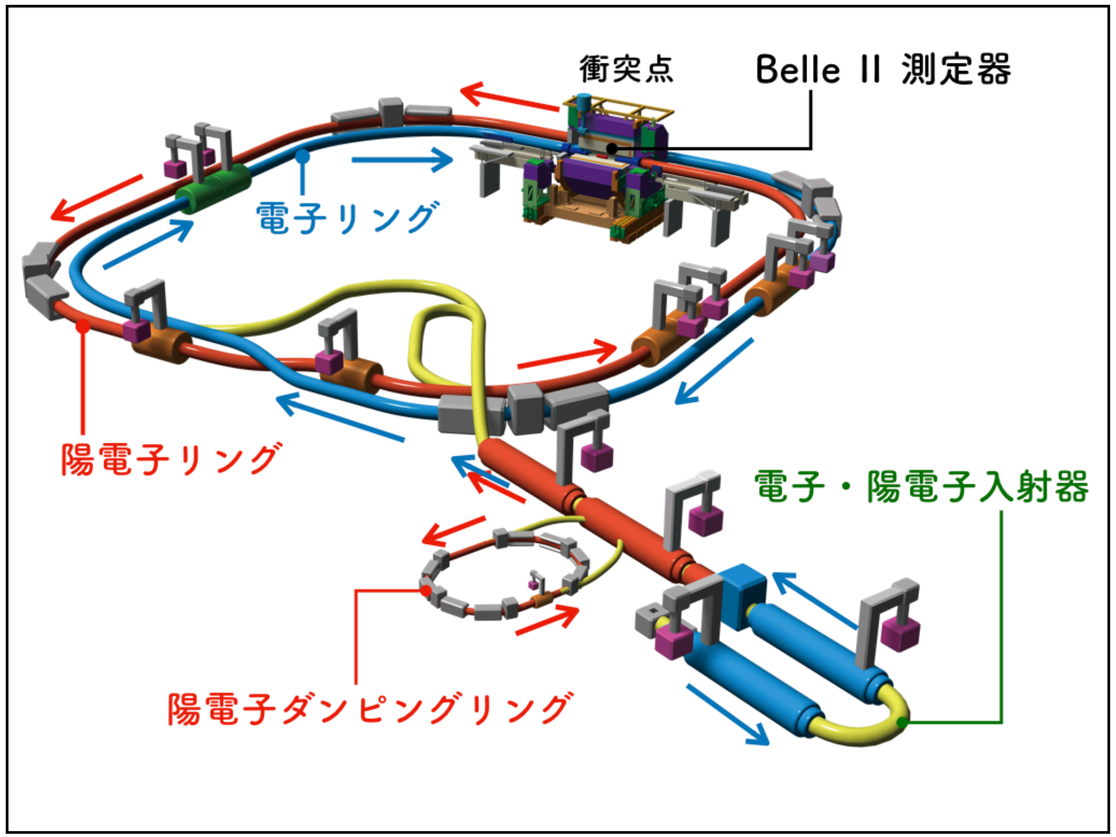
シリコンピクセルを用いた
プリシャワー検出器の
Geant4シミュレーションによる検討

2021/3/1

奈良女子大学高エネルギー研究室　卒業研究発表会
五屋　郁美

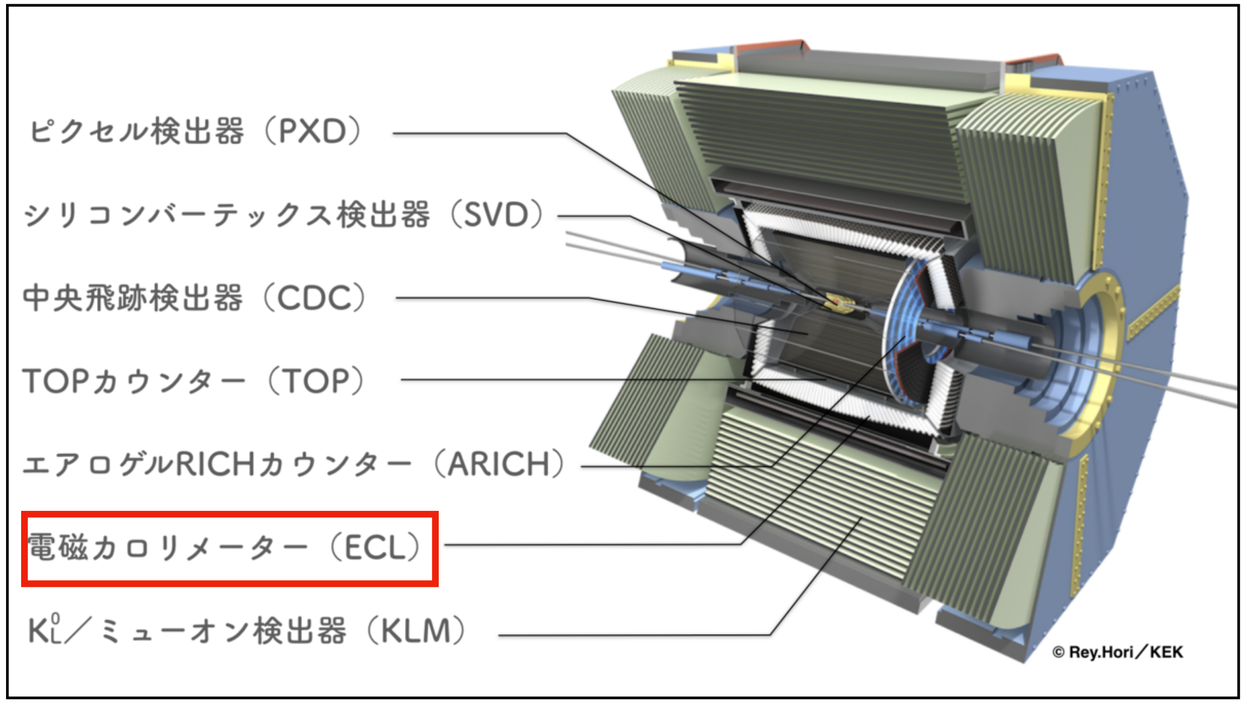
目次

Belle II 実験



- 世界最高ルミノシティを保持し、電子(7GeV)・陽電子(4GeV)を衝突させる加速器実験
- 現在はルミノシティをさらに上げながら運転を継続している
- 大量生産されたBメソンの崩壊現象に関する研究などを行い、新しい物理法則を探索している

Belle II 測定器

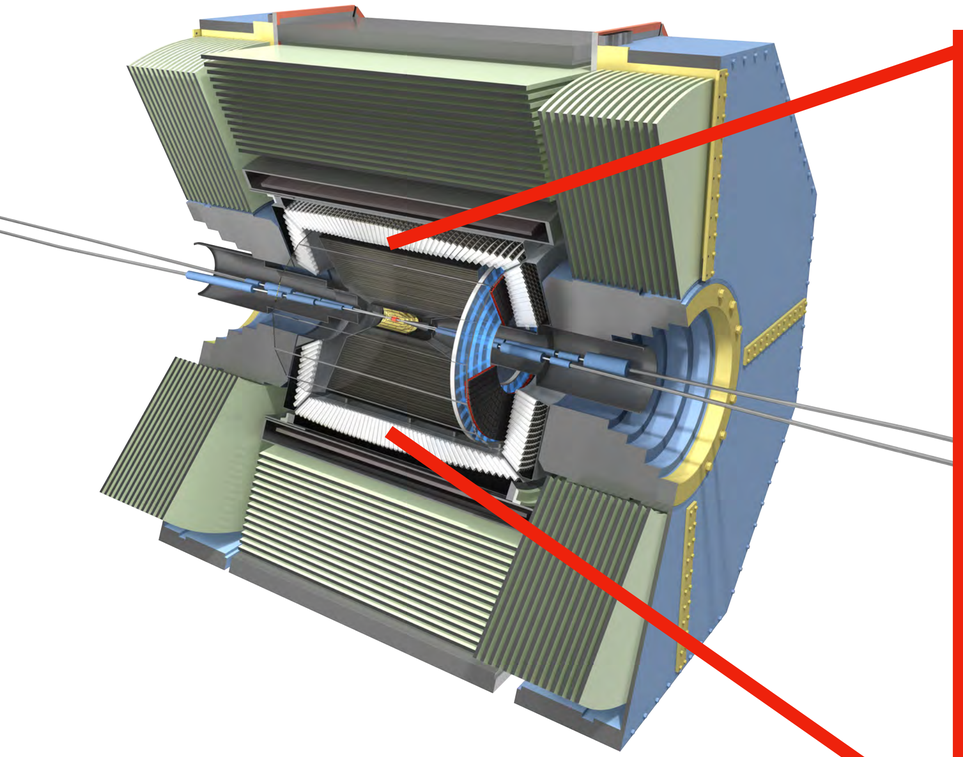


KEKB加速器からSuperKEKB加速器へとアップグレードされたことに伴い、Belle測定器もデータ収集システムやデータ処理の効率化を図り高度化された。

電磁カロリメーター

電磁カロリメーターとは…

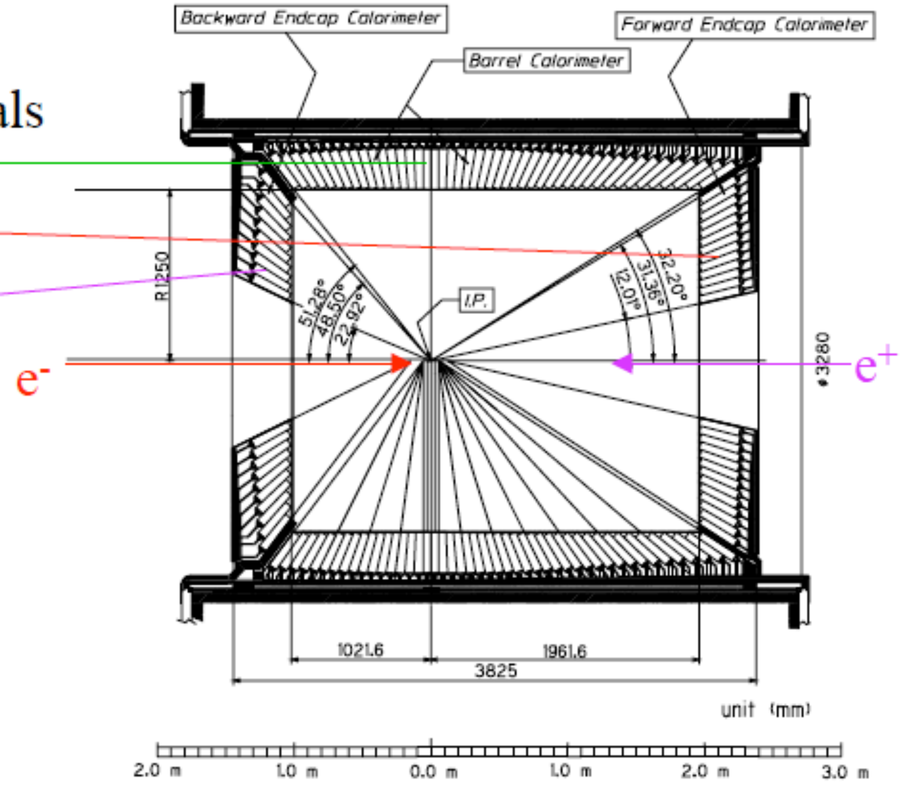
電磁シャワーという現象を用いて光子や電子のエネルギーを測定する



In total, 8736 CsI(T*l*) crystals
(6624 in Barrel,
1152 in Fwd. Endcap
and 960 in Bwd. Endcap)

Covering 12<θ<155° in
Lab. frame.

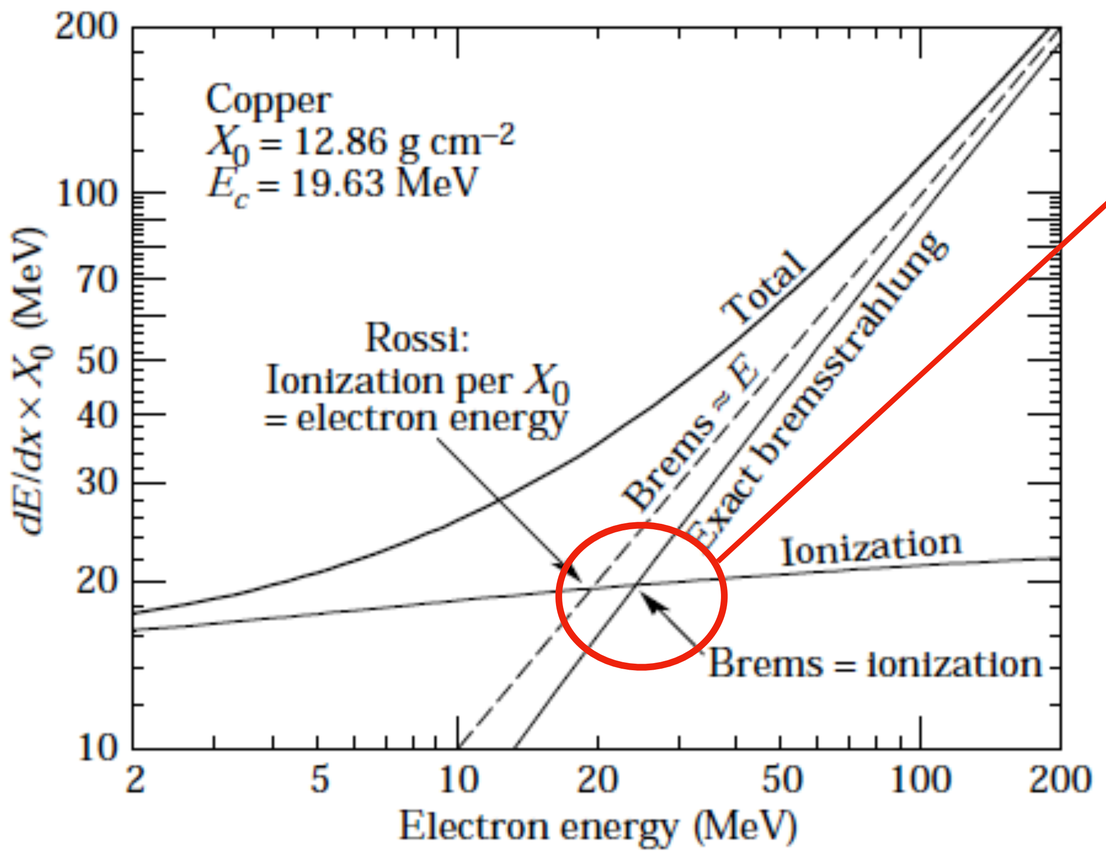
Inner radius = 1250mm.



左図：@High Energy Accelerator Research Organization,
Institute of Particle and Nuclear Studie
参照：https://www.belle2.org/archives/
右図：先生のスライドより

電子と物質の相互作用

電子の入射エネルギーとエネルギー損失の関係



約20~30MeV以上では電離損失による
エネルギー損失を制動放射によるエネルギー損失が
上回る

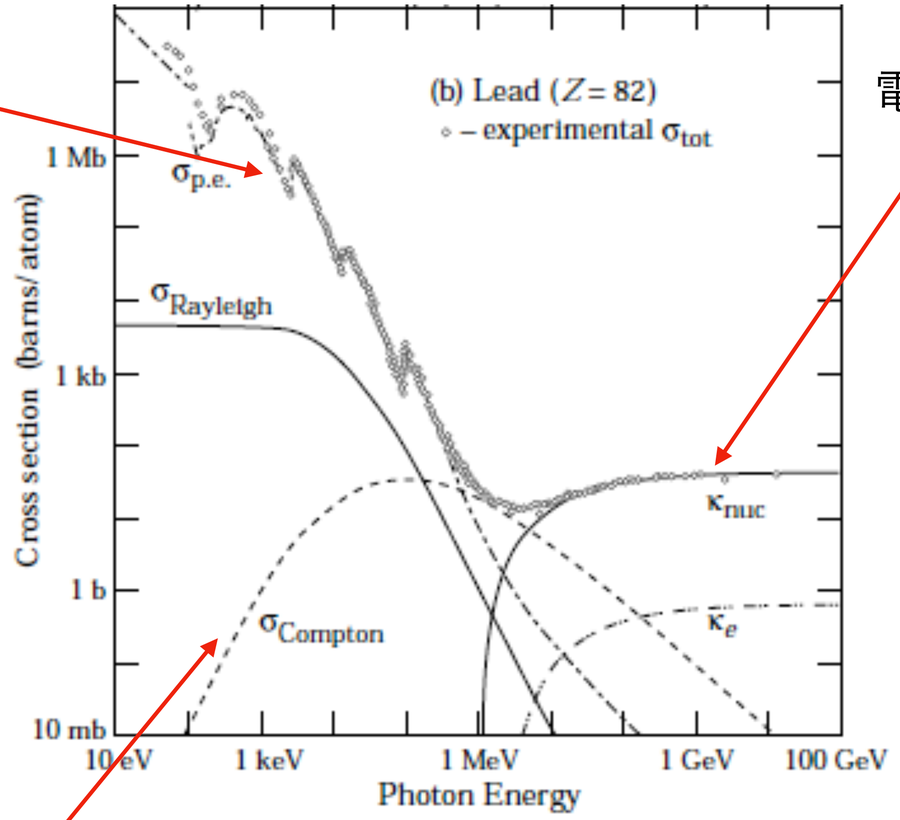
Belle II 実験で検出すべき電子のエネルギーは数十MeV以上であるから、**電磁カロリメーターで検出する**
電子や陽電子のエネルギー損失は制動放射によるものが支配的である

制動放射
電子・陽電子が物質中のクーロン場で減速し、
そのエネルギーをγ線として放出する。

光子と物質の相互作用

光電効果

電子・陽電子対生成



コンプトン散乱

Belle II 実験で検出すべき光子のエネルギーは数十MeV以上であるから、**電磁カロリメーターで検出する光子のエネルギー損失は電子対生成によるものが支配的である**

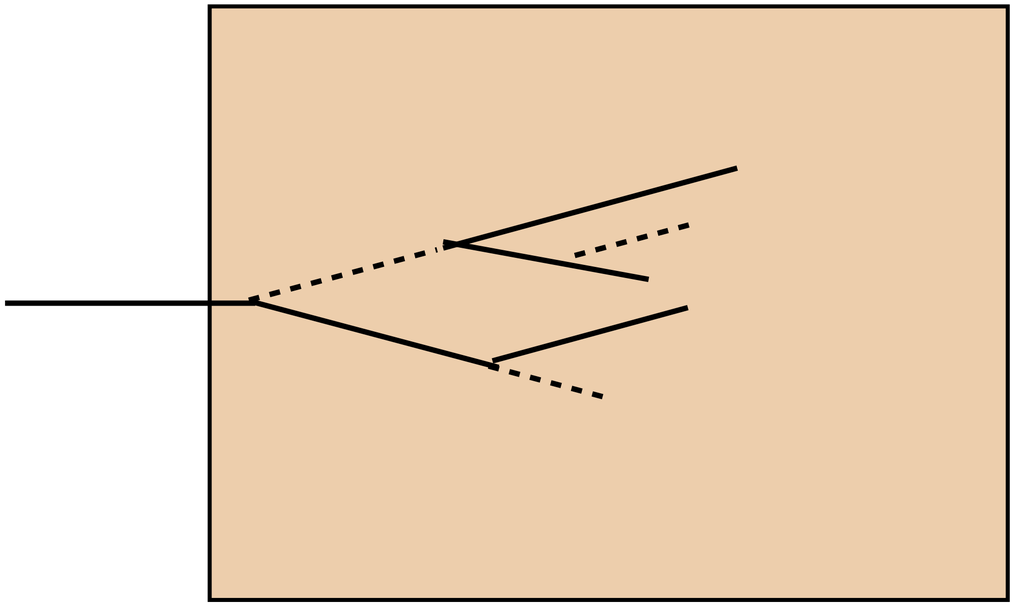
電子・陽電子対生成

1.022MeV以上のエネルギーを持つγ線が物質中に入射した時に原子核近傍で電子・陽電子対を生成する。

電子(陽電子)の静止質量：0.511MeV

0.511×2=1.022MeV

電磁シャワー



実線：電子もしくは陽電子

破線：光子

電磁シャワーとは…

制動放射と電子・陽電子対生成が連鎖して発生すること

電磁カロリメーターでは…

発生した光を電気信号に変換しエネルギー損失を求める

シンチレーターが十分な重さや大きさを持つ場合には

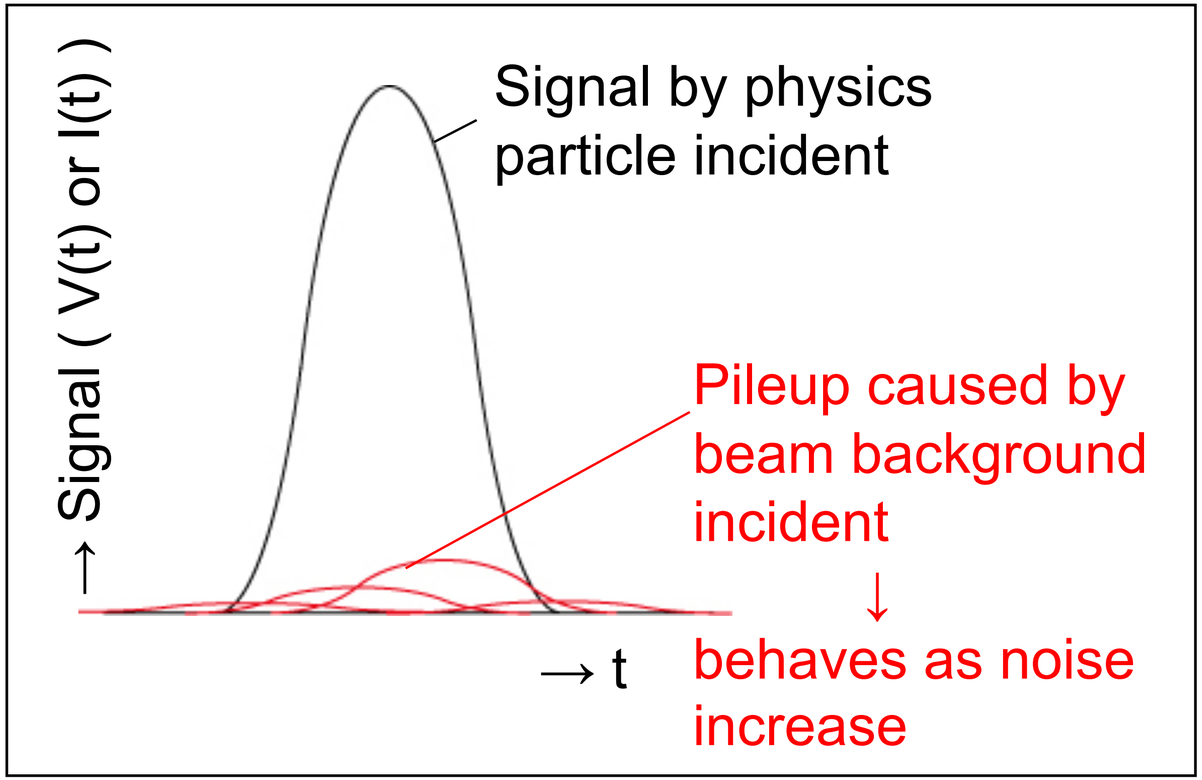
全てのエネルギーを電磁カロリメーター内で失う

Belle II 測定器に用いられている電磁カロリメーターは

もともとエネルギー分解能は良いが、高輝度の環境下では性能に限界が出る可能性がある

目次

動機



CsI(Tl)は発光量は大きいが

減衰時間が約1 μsと長い

⬇

バックグラウンドで発生したγ線によって

パイルアップが発生し

エネルギー分解能が下がる

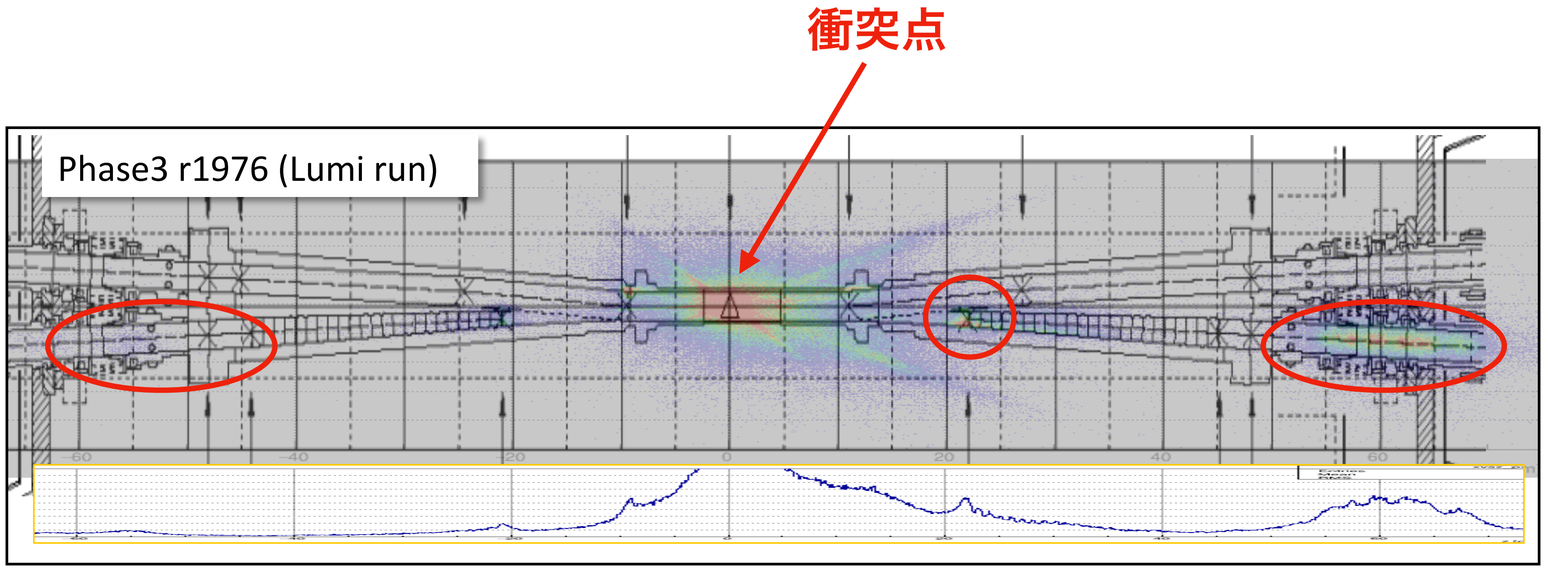
⬇

既存のカロリメーターの前に検出器を置き

バックグラウンドのγ線の影響を小さくしたい

更に機能を持たせたい

バックグラウンド



…ホットスポット(バックグラウンドの発生源)

プリシャワー検出器に

γ線の到来方向がわかる機能を持たせ、γ線を識別できるようにしたい

⬇

プリシャワー検出器の特性等を検討する

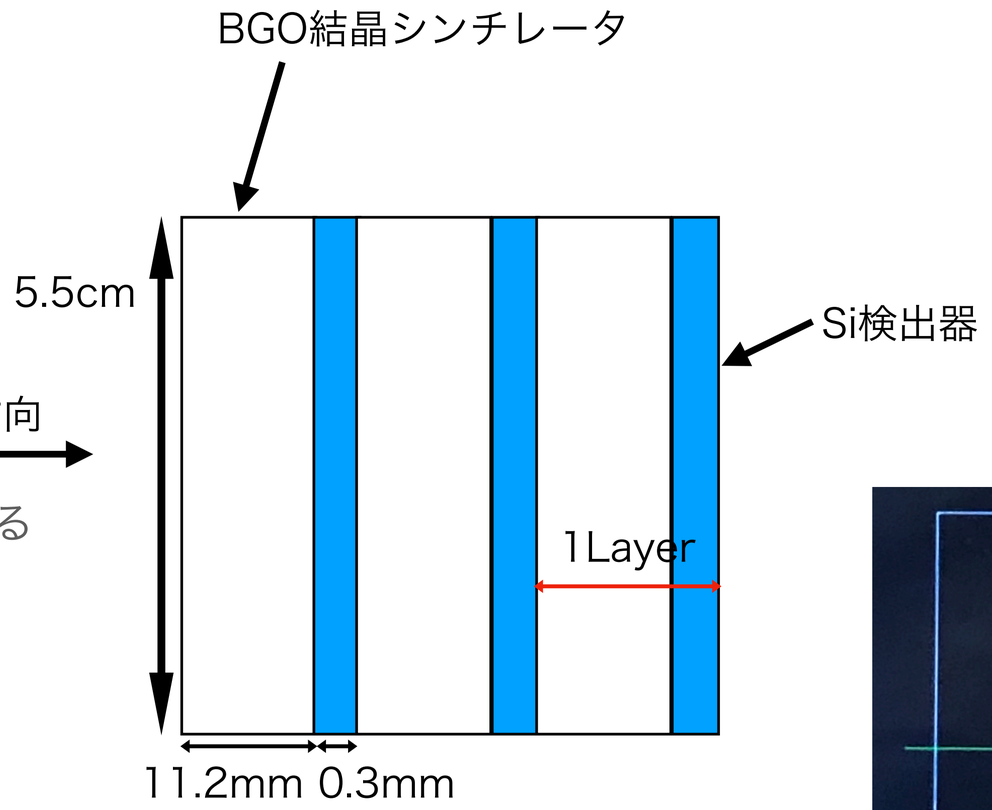
目次

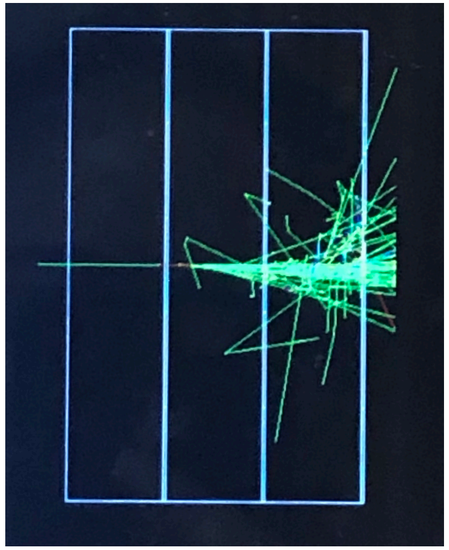
1. 背景
2. 目的
3. 方法
4. 結果
5. まとめ

プリシャワー検出器の構成

- **BGO結晶シンチレータ**
  電磁シャワーの生成
  エネルギー損失の測定
  1放射長＝1.12cm
  →将来的には減衰時間がBGOより短く、
  放射長も近いLYSOを使用することを検討している

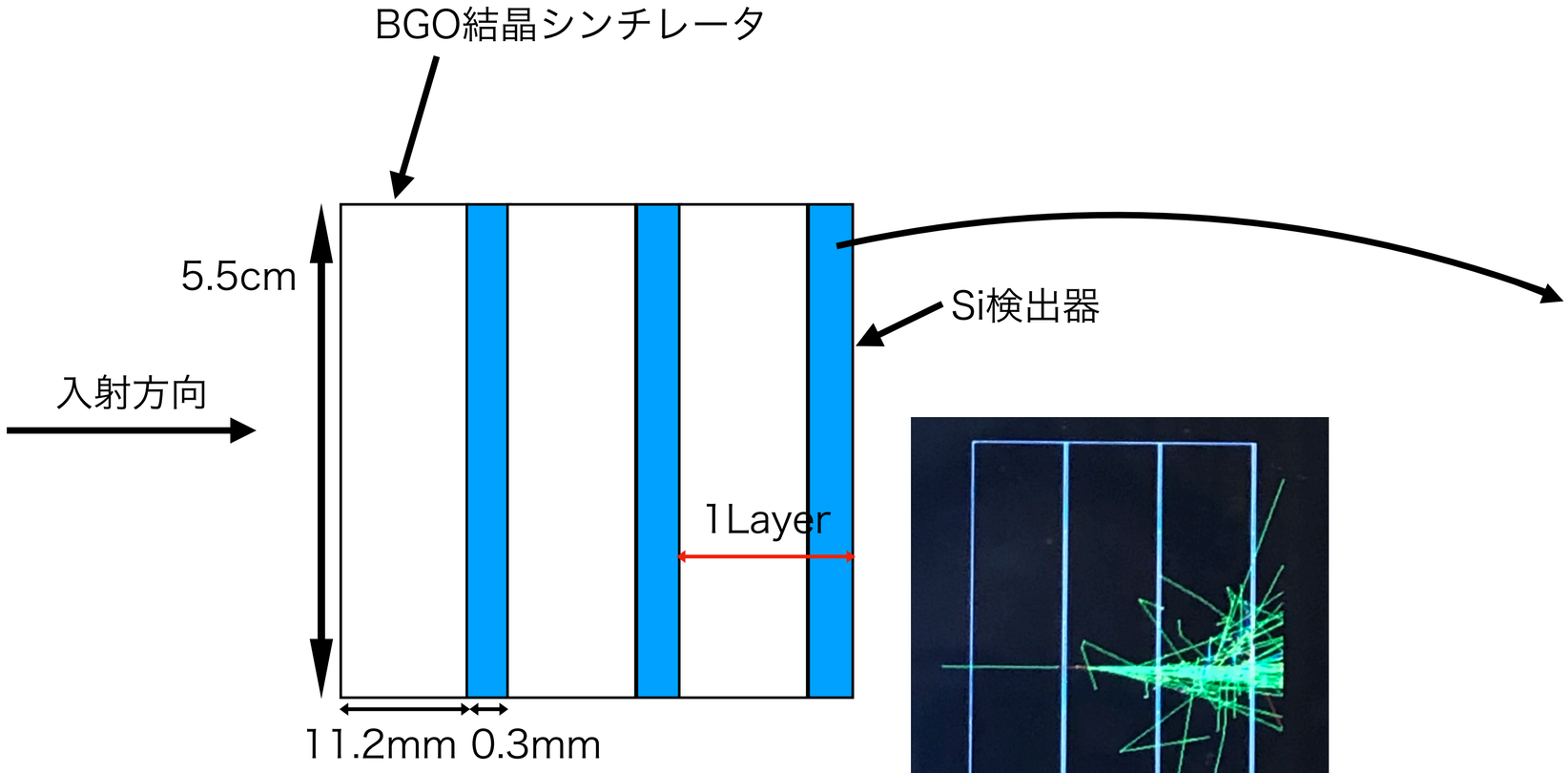
- **Si検出器**
  電子・陽電子の通過位置の検出
  →ピクセル構造を持つものに変更





を組み合わせて1層とし、３層にする

プリシャワー検出器の構成



<入射方向に垂直な面>

5×5 (1.1cm×1.1cm角)
にSi検出器を区切った場合
以下のようにpixel番号を振った

4	9	14	19	24
3	8	13	18	23
2	7	12	17	22
1	6	11	16	21
0	5	10	15	20

5.5cm

5.5cm

更に細かく分割した場合も同様

Geant4について

シミュレーションソフトはGeant4を使用する。

- **Geant4**

粒子が物質中を通過するときの相互作用や過程をシミュレーションするソフトウェア

複数の例題プログラムを持っており、その中からサンプリングカロリメータの例題であるexampleB4aを改変して使用する

ver10.6を使用

昨年までの研究で…

３層構造のプリシャワー検出器が作られている

本年は…

①Si検出器をピクセル構造を持つものに変更する

②各ピクセルに番号を振り、識別できるようにする

③各ピクセルのエネルギー損失を得られるようにする

<変更したファイル>

- B4DetectorConstruction.cc
- B4RunAction.cc
- B4aEventAction.hh
- B4DetectorConstruction.hh
- B4aEventAction.cc
- B4aSteppingAction.cc
- Run2.mac

B4DetectorConstruction.cc

→装置の構成を設定するファイル

```
void B4DetectorConstruction::DefineMaterials()
{
// NIST is GEANT4 Material Database. Instance() func. is static.
auto nistManager = G4NistManager::Instance();
// Air, BGO and Si defined using NIST Manager
nistManager->FindOrBuildMaterial("G4_AIR");
nistManager->FindOrBuildMaterial("G4_BGO");
nistManager->FindOrBuildMaterial("G4_Si");
// Liquid argon material (original example).
// G4double a; // mass of a mole;
// G4double z; // z=mean number of protons;
// G4double density;
// new G4Material("liquidArgon", z=18., a=39.95*g/mole,
density=1.390*g/cm3);
// The argon by NIST Manager is a gas with a different density
// Print materials
G4cout << *(G4Material::GetMaterialTable()) << G4endl;
}
// Define Volumes.
G4VPhysicalVolume* B4DetectorConstruction::DefineVolumes()
{
// Geometry parameters
G4int nofLayers = 3; // Presampler.
G4double absoThickness = 11.2*mm; // BGO 1X0.
G4double gapThickness = 0.3*mm; // Si thickness.
//G4double gapThickness = 10.*mm; // Si thickness.
G4double calorSizeXY = 5.5*cm; // Same cross section as Belle
CsI.
auto layerThickness = absoThickness + gapThickness;
:
auto calorThickness = nofLayers * layerThickness;
auto worldSizeXY = 1.2 * calorSizeXY;
auto worldSizeZ = 1.2 * calorThickness;
// Get materials, sould be consistent with DefineMaterials().
auto defaultMaterial = G4Material::GetMaterial("G4_AIR");
auto absorberMaterial = G4Material::GetMaterial("G4_BGO");
auto gapMaterial = G4Material::GetMaterial("G4_Si");
```

使用する物質の設定

nist…物質の性質が登録されているデータベース

今回はnistに登録されている物質を使用したのでこのように書いた

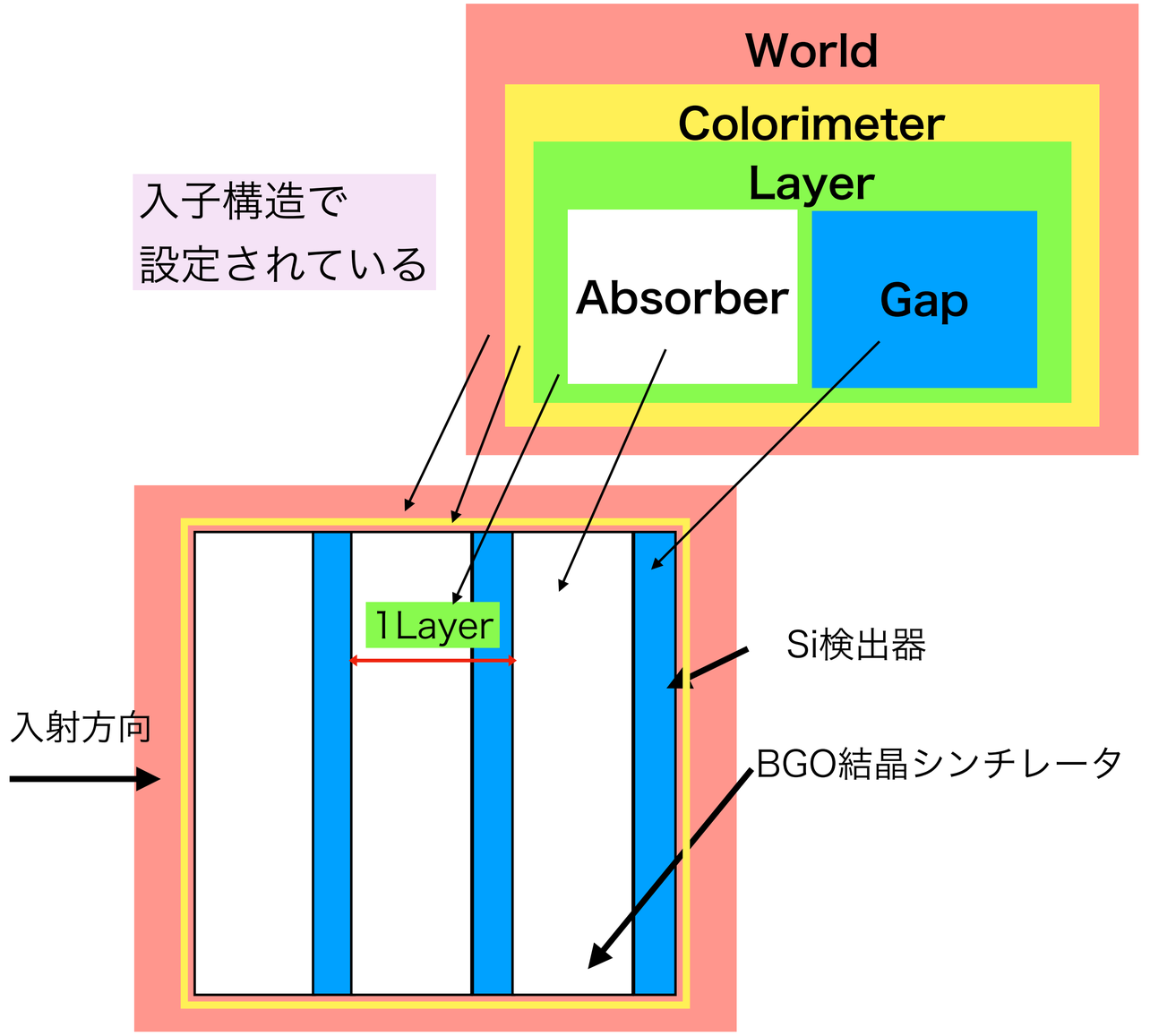
装置の大きさの設定

B4DetectorConstruction.cc

Si検出器の設定

```
//----
// Gap
//----
G4int ndivx = 5;
G4int ndivy = 5;

auto gapS
= new G4Box("Gap", // its name
calorSizeXY/ndivx/2, calorSizeXY/ndivy/2, gapThickness/2);
// its size has been modified.

auto gapLV
= new G4LogicalVolume(gapS, // its solid
                      gapMaterial, // its material
                      "Gap"); // its name

// GI Added 20201210.
for(G4int ix=0; ix<ndivx; ++ix){
  for(G4int iy=0; iy<ndivy; ++iy){
    G4int ipix = ix*ndivx+iy;
    fGapPV[ipix]= new G4PVPlacement(
                        0, // no rotation
                        G4ThreeVector(-calorSizeXY/2+(ix+0.5)*calorSizeXY/ndivx,
                                      -calorSizeXY/2+(iy+0.5)*calorSizeXY/ndivy,
                                     absoThickness/2), // its position
                        gapLV, // its logical volume
                        "Gap", // its name
                        layerLV, // its mother volume
                        false, // no boolean operation
                        ipix, // copy number,should be incremented.
                        fCheckOverlaps); // checking overlaps
  }
}
```

入子構造で
設定されている



B4DetectorConstruction.cc

Si検出器の設定

```
//----
// Gap
//----
G4int ndivx = 5;
G4int ndivy = 5;

auto gapS
= new G4Box("Gap", // its name
calorSizeXY/ndivx/2, calorSizeXY/ndivy/2, gapThickness/2);
// its size has been modified.

auto gapLV
= new G4LogicalVolume(gapS, // its solid
                      gapMaterial, // its material
                      "Gap"); // its name

// GI Added 20201210.
for(G4int ix=0; ix<ndivx; ++ix){
  for(G4int iy=0; iy<ndivy; ++iy){
    G4int ipix = ix*ndivx+iy;
    fGapPV[ipix]= new G4PVPlacement(
                        0, // no rotation
                        G4ThreeVector(-calorSizeXY/2+(ix+0.5)*calorSizeXY/ndivx,
                                      -calorSizeXY/2+(iy+0.5)*calorSizeXY/ndivy,
                                      absoThickness/2), // its position
                        gapLV, // its logical volume
                        "Gap", // its name
                        layerLV, // its mother volume
                        false, // no boolean operation
                        ipix, // copy number,should be incremented.
                        fCheckOverlaps); // checking overlaps
  }
}
```

x軸、y軸方向の分割数を定義
(5×5の1.1cm×1.1cm角に分割する場合)

ピクセル番号

PV…Physical Volume
PVを用いると設定した物質が配置される

Si検出器はGap部分に当たるので、fGapPVを使用

Si検出器をピクセル分割する必要があるので
配列を使用した

B4RunAction.cc

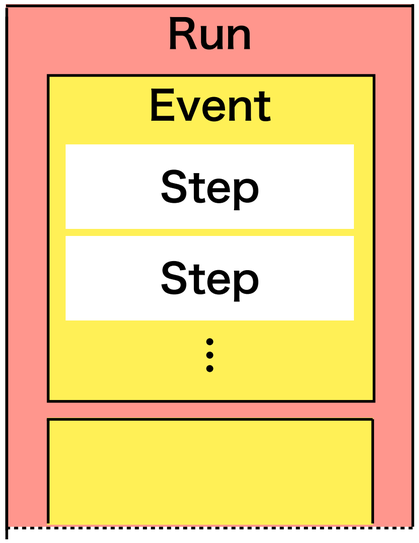
→ヒストグラムの作成やNtupleの作成などを行う


```
// Create directories
//analysisManager->SetHistoDirectoryName("histograms");
//analysisManager->SetNtupleDirectoryName("ntuple");
analysisManager->SetVerboseLevel(1);
analysisManager->SetNtupleMerging(true);
// Note: merging ntuples is available only with Root output
// Book histograms, ntuple
// Creating histograms
analysisManager->CreateH1("Eabs","Edep in absorber", 100, 0.,800*MeV);
analysisManager->CreateH1("Egap","Edep in gap", 100, 0., 100*MeV);
analysisManager->CreateH1("Labs","trackL in absorber", 100, 0.,1*m);
analysisManager->CreateH1("Lgap","trackL in gap", 100, 0., 50*cm);
// Creating ntuple
analysisManager->CreateNtuple("B4", "Edep and TrackL");
analysisManager->CreateNtupleDColumn("Eabs");
analysisManager->CreateNtupleDColumn("Egap");
analysisManager->CreateNtupleDColumn("Labs");
analysisManager->CreateNtupleDColumn("Lgap");
// KM Add 20200216, energy deposit in each layer.
analysisManager->CreateNtupleDColumn("EabsL0");
analysisManager->CreateNtupleDColumn("EgapL0");
analysisManager->CreateNtupleDColumn("EabsL1");
analysisManager->CreateNtupleDColumn("EgapL1");
analysisManager->CreateNtupleDColumn("EabsL2");
analysisManager->CreateNtupleDColumn("EgapL2");
// GI Add 20201026, energy deposit in each pixel
G4int nl = 3;
G4int ndivx = 55;
G4int ndivy = 55;
std::stringstream ss;
for (G4int il=0; il<nl; ++il){
for (G4int ix=0; ix<ndivx; ++ix){
for(G4int iy=0; iy<ndivy; ++iy){
ss.str("");
ss<<"EgapL"<<il<<"P"<<ndivx*ix+iy;
analysisManager->CreateNtupleDColumn(ss.str().c_str());
}
}
}
analysisManager->FinishNtuple();
}
```

ヒストグラムの作成

Ntupleの作成(確保)

シミュレーション１回を１eventといい、eventの集合のことをrunと呼ぶ。更に、1event中の1つの相互作用をstepと呼ぶ。



B4aSteppingAction.cc

→layer番号とpixel番号を得て各関数に引数として渡す

```
if ( volume == fDetConstruction->GetAbsorberPV() ) {
// KM Added 20200216.
G4int lyrid
= step->GetPreStepPoint()->GetTouchableHandle()->GetReplicaNumber(1);
fEventAction->AddAbs(edep,stepLength,lyrid);
// fEventAction->AddAbs(edep,stepLength); // original
}
for(G4int ipix=0; ipix<3025 ; ++ipix){
 if ( volume == fDetConstruction->GetGapPV(ipix) ) {
  if( edep > 0.05 ){ //GI Added 20210202.
  // get layer number (replica Number)
  // KM Added 20200216.
  G4int lyrid
   = step->GetPreStepPoint()->GetTouchableHandle()->GetReplicaNumber(1);
  // GI Added 20201030.
  fEventAction->AddGap(edep,stepLength,lyrid,ipix); // GI Added 20201030.
  // fEventAction->AddGap(edep,stepLength,lyrid); // KM Added 20200216.
  // fEventAction->AddGap(edep,stepLength); // original.
    }
  }
}
```

B4aEventAction.ccでは、
Ntupleに各情報をFillする

<B4aEventAction.hhより>

```
// GI Added 20201207.-------------
inline void B4aEventAction::AddGap(G4double de, G4double dl, G4int  lyr=-1, G4int Pix=-1)
{
 fEnergyGap += de;
 fTrackLGap += dl;
 if(lyr!=-1) fEnergyGapLyr[lyr] +=de;
  if(Pix!=-1) {
   if(lyr==0) fEnergyGapL0Pix[Pix] +=de;
   else if(lyr==1) fEnergyGapL1Pix[Pix] +=de;
   else if(lyr==2) fEnergyGapL2Pix[Pix] +=de;
     }
   }
```

ヘッダーファイル B4aEventAction.hh中でAddGap関数が定義されている。

AddGap関数はGap(Si検出器部分）でのエネルギー損失を
step毎に足し合わせる関数である

run2.mac

→シミュレーションを実行する時に読ませるマクロファイル

```
# Macro file for example B4
#
# To be run preferably in batch, without graphics:
# % exampleB4[a,b,c,d] run2.mac
#
#/run/numberOfWorkers 4
/run/initialize
#
# Default kinemtics:
# electron 50 MeV in direction (0.,0.,1.)
/gun/particle gamma
/gun/energy 1000 MeV
#/gun/position 600 600 0 mm
#/gun/direction 0 0 1
#
# 1000 events
#
/run/printProgress 100
/run/beamOn 1000
```

- 入射粒子：γ線
- 入射粒子のエネルギー：1000MeV

- 何イベント毎に表示するか：100イベント
- 何イベント行うか：1000イベント

シミュレーションが走るとAbsorber、Gapのピクセル毎のエネルギー損失が書き込まれたrootfileが生成される

→解析用のマクロファイル を作成する

解析用マクロ

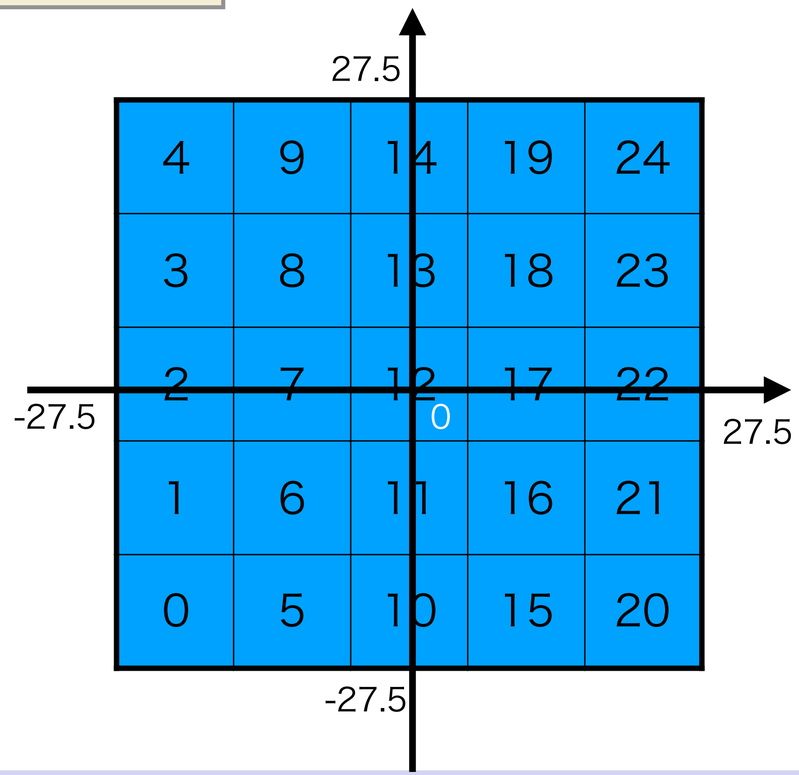
①各ピクセル毎の中心位置の座標を定義する

```
pos_define[ipix][0] = -calorSizeXY/2+(ix+0.5)*calorSizeXY/ndivx; //x座標
pos_define[ipix][1] = -calorSizeXY/2+(iy+0.5)*calorSizeXY/ndivy; //y座標
ipix:ピクセル番号
```

event毎に

②シリコンのlayer毎にシャワーの中心位置を決定する

③②で得た点を結ぶことによりγ線の到来方向を決定する



解析用マクロ

②シリコンのlayer毎にシャワーの中心位置を決定する

ピクセル毎のエネルギー損失で重心を計算する

```
 for(int il=0; il<nl; ++il){
int npix = 0;

for(int ix=0; ix<ndivx; ++ix){

for(int iy=0; iy<ndivy; ++iy){
   int ipix = ndivx*ix+iy;
   int iEpix = ipix + il*ndivx*ndivy;

if(Epix[iEpix]>0.05){
     pos[il*3] = pos[il*3] + pos_define[ipix][0]*Epix[iEpix]/EgapL[il];      //x座標
     pos[il*3+1] = pos[il*3+1] + pos_define[ipix][1]*Epix[iEpix]/EgapL[il];   //y座標
     ++npix;
   }
  }
 }
}
```

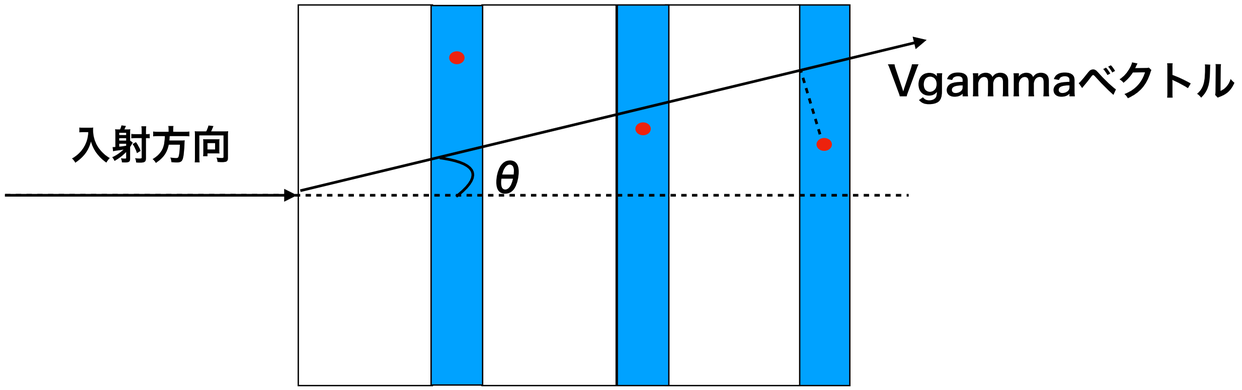
Epix[iEpix]：layer番号il、pixel番号ipixのエネルギー損失[MeV]

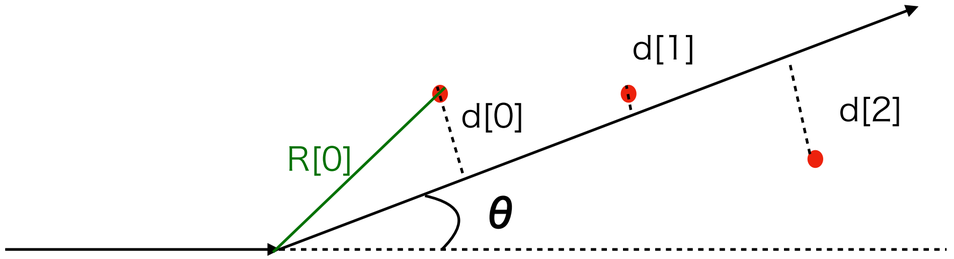
EgapL[il]：layer番号ilのエネルギー損失[MeV]

Si検出器が検出できる電子のエネルギー損失は
50keV以上であるから
0.05MeV以上の条件を設けた

解析用マクロ

③②で得た点を結ぶことによりγ線の到来方向を決定する





・γ線の入射位置は(x,y,z)=(0,0,0)に固定した

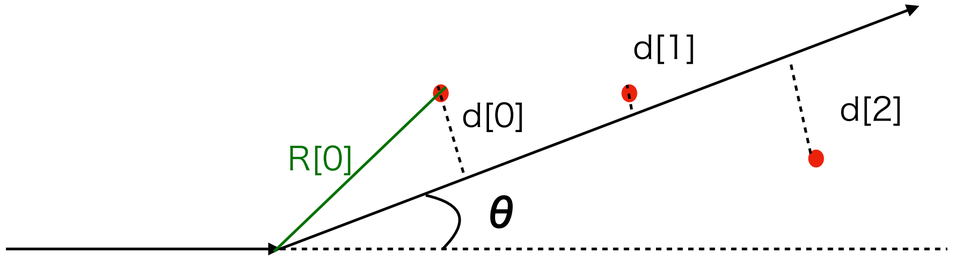
・3layerとも鳴ったシリコンピクセルがあり、シャワーの中心位置が求められたeventに限定した

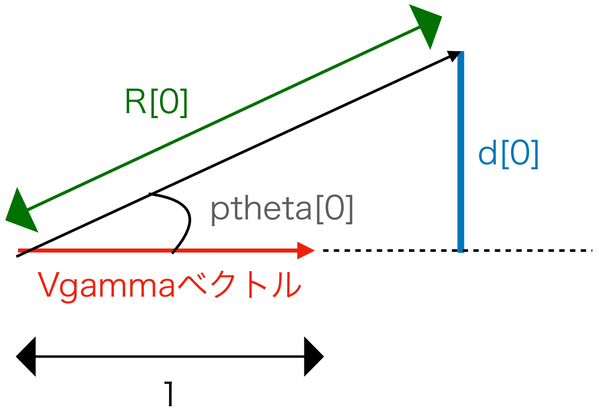
プリシャワー検出器で再構成するγ線の方向を極角θと方位角ϕで表す。
入射位置である(x,y,z)=(0,0,0)からθとϕで決まる方向に直線を伸ばし、
各layerで求めたシャワーの中心位置からの距離の二乗和を最小にするθとϕを求めることとした。

解析用マクロ


```
//analyze function-------------------------------------------------//
std::vector<double> LeastSquares(double pos0,double pos1,double pos3,double pos4,double pos6,double pos7)
{
  double PI = acos(-1.0);

  double calorSizeXY = 55.0 ;
  double absoThickness = 11.2 ;
  double gapThickness = 0.3 ;
  double pos2 = absoThickness+gapThickness/2;
  double pos5 = absoThickness*2+gapThickness*3/2;
  double pos8 = absoThickness*3+gapThickness*5/2;

  double R[3];    //length from incident position
  double d[3];    //length from Vgamma vector
  for(int i=0; i<3; ++i){
    R[i] = 0.;
    d[i] = 0.;
  }
//  double x = 100.;
//  double y = 100.;

  double d2 = 9999.0;
  std::vector<double> angle = {100.,100.};
  std::vector<double> Vgamma = {0.,0.,0.};

//  for(double px=-2.75; px<=2.75; px+=0.01){
//    for(double py=-2.75; py<=2.75; py+=0.01){
     for(double phi=-PI/2; phi<=PI/2; phi+=0.001){
      for(double theta=0.0; theta<PI/2; theta+=0.001){

       Vgamma[0] = sin(theta)*cos(phi);
       Vgamma[1] = sin(theta)*sin(phi);
       Vgamma[2] = cos(theta);

//       double posd0 = pos0 - px;
//       double posd1 = pos1 - py;
//       double posd3 = pos3 - px;
//       double posd4 = pos4 - py;
//       double posd6 = pos6 - px;
//       double posd7 = pos7 - py;
```

①で決定したγ線の通過位置を引数として与える

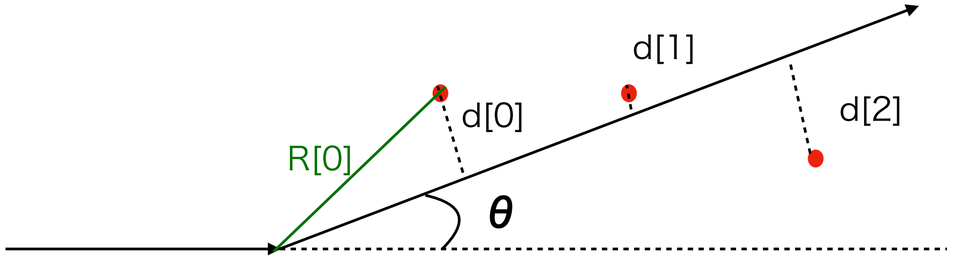


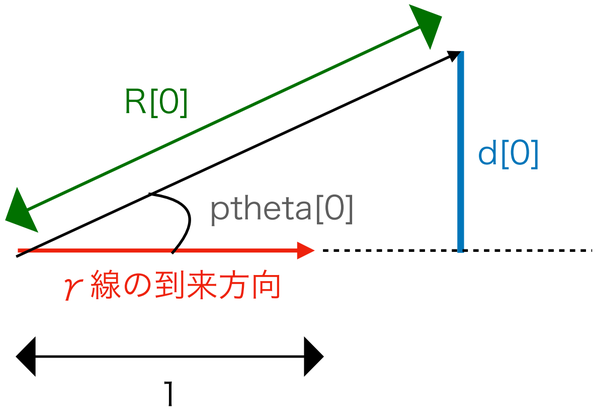


解析マクロ


```
      R[0] = sqrt( pow(pos0,2.0) + pow(pos1,2.0) + pow(pos2,2.0));  //layer0
      R[1] = sqrt( pow(pos3,2.0) + pow(pos4,2.0) + pow(pos5,2.0));  //layer1
      R[2] = sqrt( pow(pos6,2.0) + pow(pos7,2.0) + pow(pos8,2.0));  //layer2

      //R[0] = sqrt( pow(posd0,2.0) + pow(posd1,2.0) + pow(pos2,2.0));  //layer0
      //R[1] = sqrt( pow(posd3,2.0) + pow(posd4,2.0) + pow(pos5,2.0));  //layer1
      //R[2] = sqrt( pow(posd6,2.0) + pow(posd7,2.0) + pow(pos8,2.0));  //layer2


      double ptheta[3];
      ptheta[0] = acos((pos0*Vgamma[0]+pos1*Vgamma[1]+pos2*Vgamma[2])/R[0]);  //layer0
      ptheta[1] = acos((pos3*Vgamma[0]+pos4*Vgamma[1]+pos5*Vgamma[2])/R[1]);  //layer1
      ptheta[2] = acos((pos6*Vgamma[0]+pos7*Vgamma[1]+pos8*Vgamma[2])/R[2]);  //layer2

      //ptheta[0] = acos((posd0*Vgamma[0]+posd1*Vgamma[1]+pos2*Vgamma[2])/R[0]);  //layer0
      //ptheta[1] = acos((posd3*Vgamma[0]+posd4*Vgamma[1]+pos5*Vgamma[2])/R[1]);  //layer1
      //ptheta[2] = acos((posd6*Vgamma[0]+posd7*Vgamma[1]+pos8*Vgamma[2])/R[2]);  //layer2

      double pd2 = 0.;
      for(int i=0; i<3; ++i){
                  d[i] = R[i]*sin(ptheta[i]);
                  pd2 += d[i]*d[i];
      }

      if(pd2 < d2){
                  d2 = d2;
                  angle[0] = theta; // theta
                  angle[1] = phi;   // phi
                  //x = px;
                  //y = py;
       }

         }
      }
//    }
//  }

  return angle;
}
```

Θ、ϕの値の入ったvectorを返す





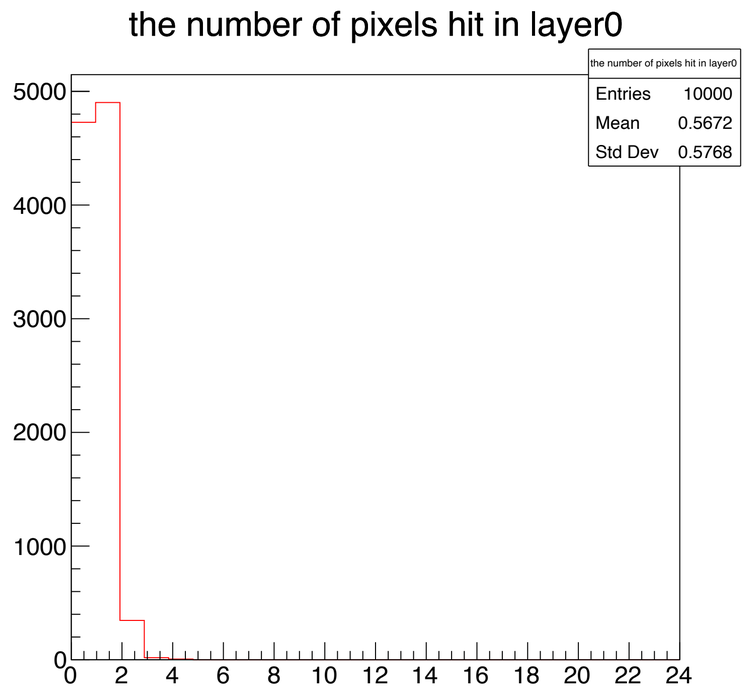
目次

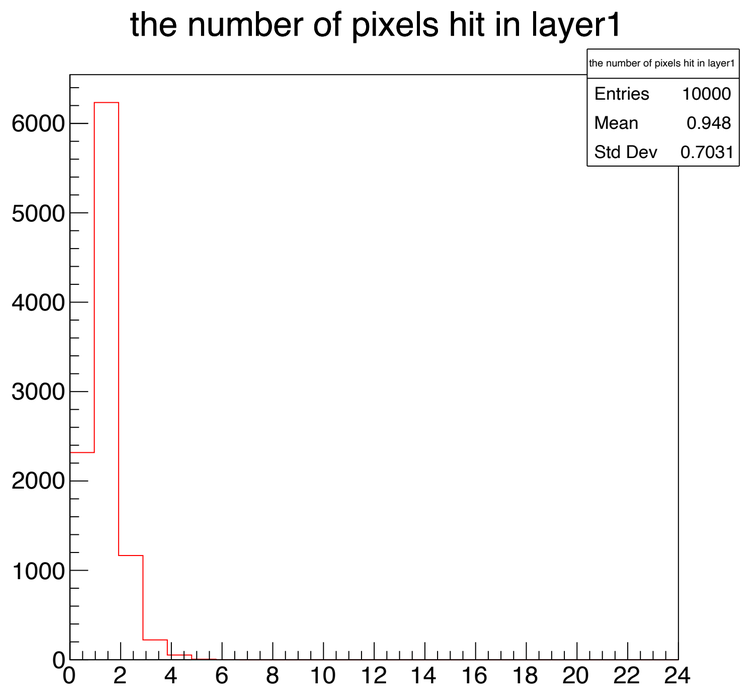
1. 背景
2. 目的
3. 方法
4. 結果
5. まとめ

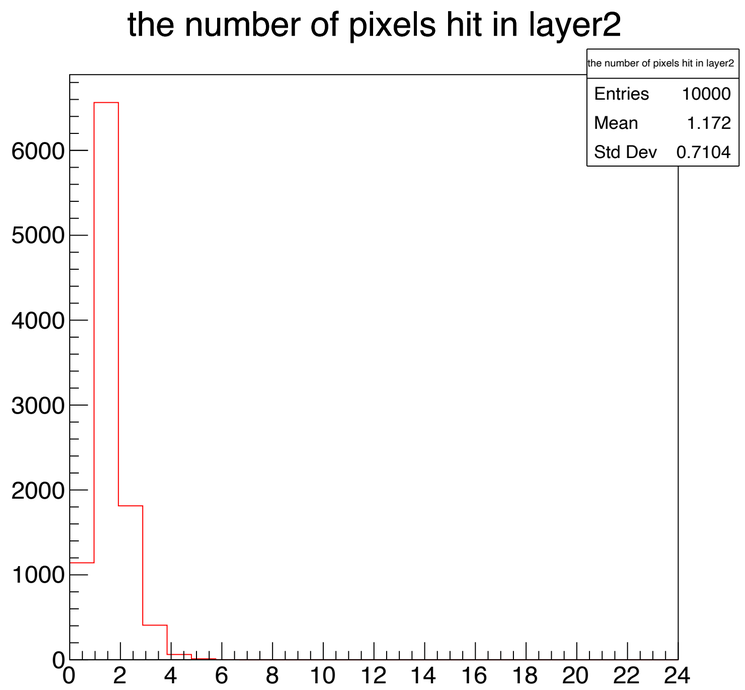
鳴ったピクセルの数

3layer　11mm×11mm（25枚）に分割した場合

1GeVのγ線を打ち込んだ場合に鳴ったピクセルの数（縦軸：イベント数　横軸：なったピクセル数）





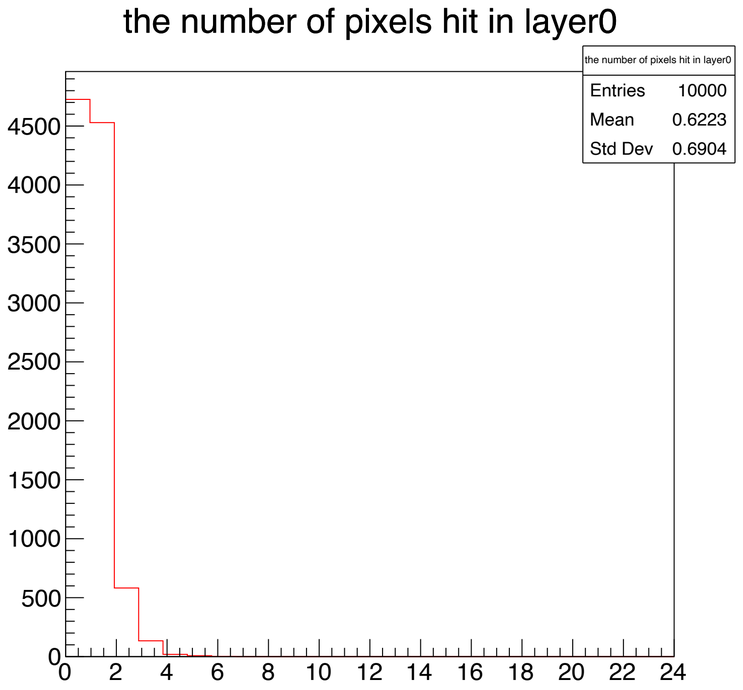


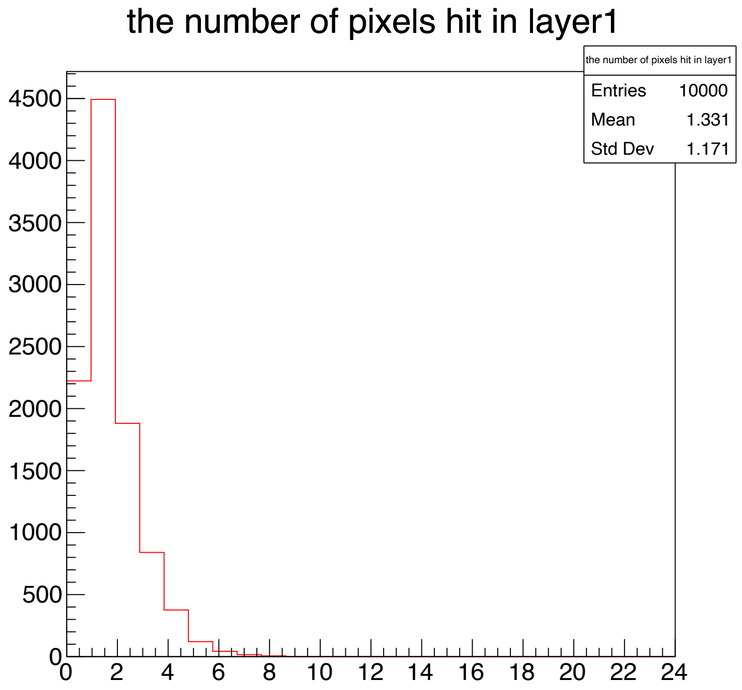
- 鳴っていないピクセルの割合は、layer0:約48%　layer1:約23%　layer2:約11%
- 鳴ったピクセルが1枚の割合は、layer0:約49%　layer1:約62%　layer2:約65%
  →ほとんどのlayerでピクセルは1枚しか鳴っていないので、ピクセルの大きさが大きすぎる

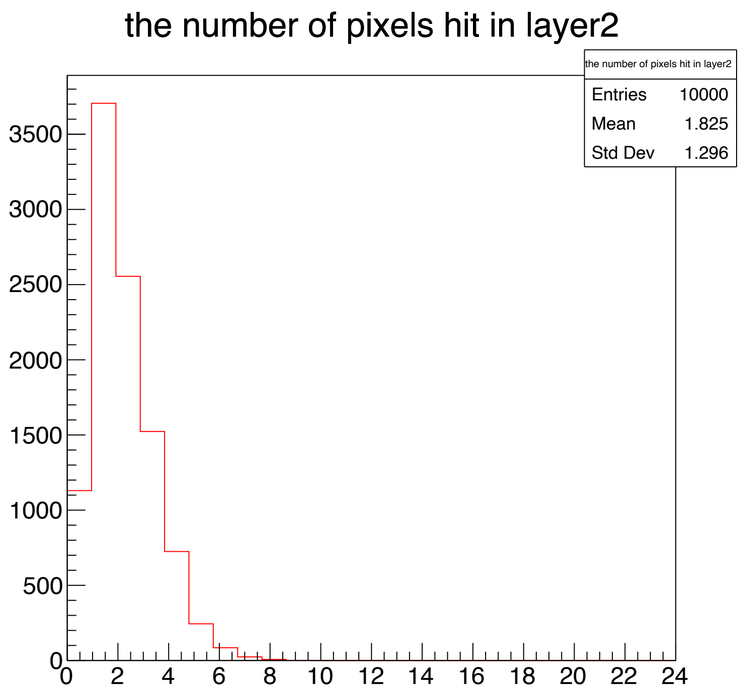
鳴ったピクセルの数

3layer　5mm×5mm（121枚）に分割した場合

1GeVのγ線を打ち込んだ場合に鳴ったピクセルの数（縦軸：イベント数　横軸：なったピクセル数）





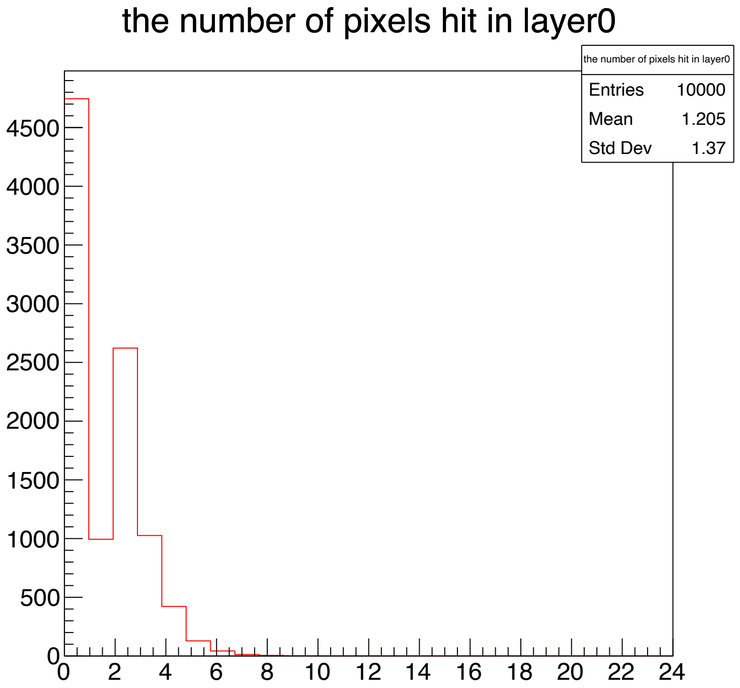


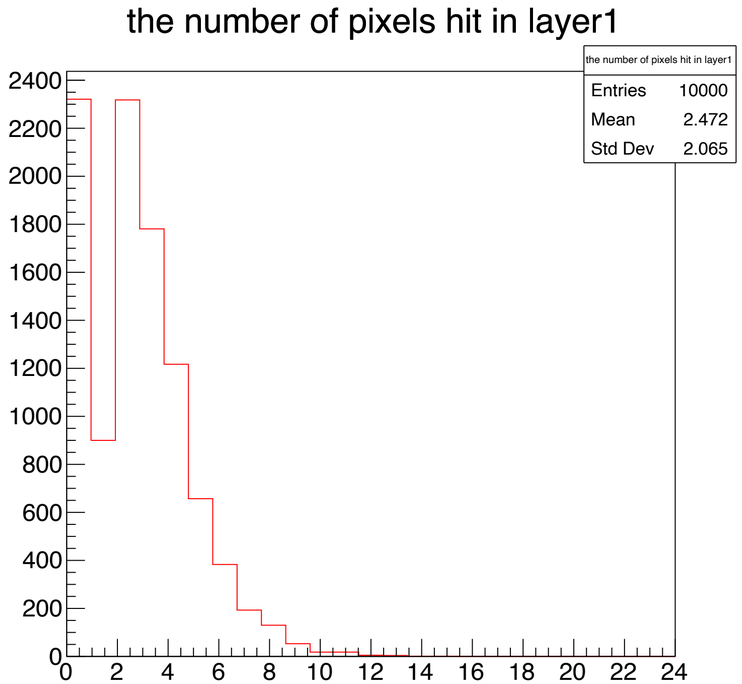
- 鳴っていないピクセルの割合は、layer0:約47%　layer1:約22%　layer2:約11%
- 鳴ったピクセルが1枚の割合は、layer0:約45%　layer1:約45%　layer2:約37%
  →25枚分割の時よりも鳴るピクセル数は増えたが、1枚しか鳴っていないイベントが多いのでまだピクセルが大きい

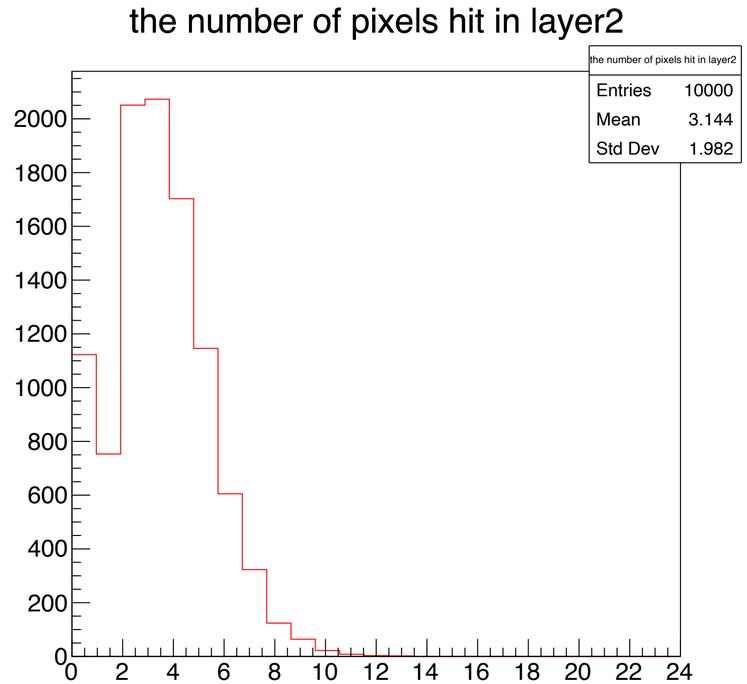
鳴ったピクセルの数

3layer　2.5mm×2.5mm（484枚）に分割した場合

1GeVのγ線を打ち込んだ場合に鳴ったピクセルの数（縦軸：イベント数　横軸：なったピクセル数）





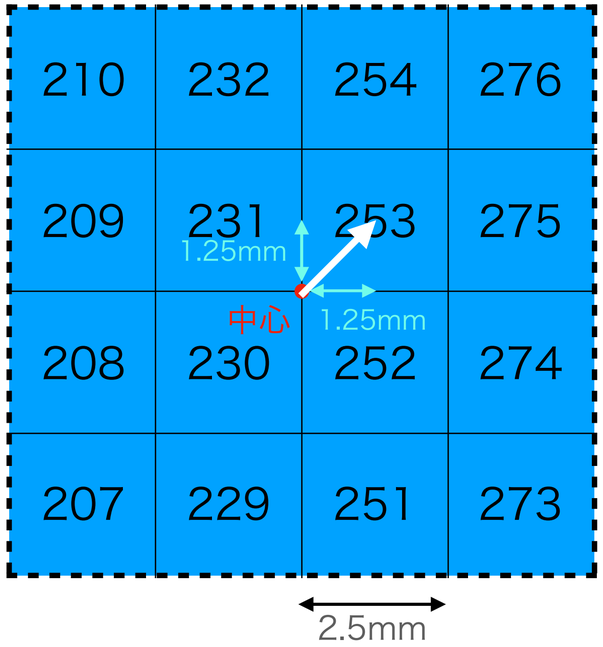


- 鳴っていないピクセルの割合は、layer0:約47%　layer1:約23%　layer2:約11%
- 鳴ったピクセルが1枚の割合は、layer0:約10%　layer1:約9%　layer2:約8%
  →2枚以上のピクセルが鳴ったイベント数が増えた。ピクセルが1枚だけ鳴ったイベントの数が他のイベントに比べて明らかに少ない

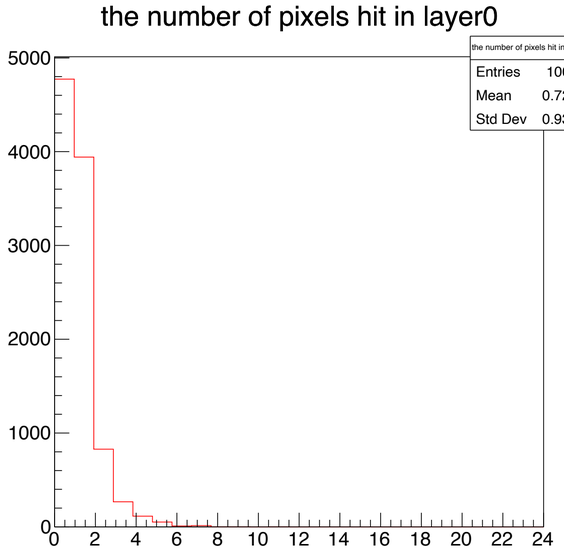
鳴ったピクセルの数

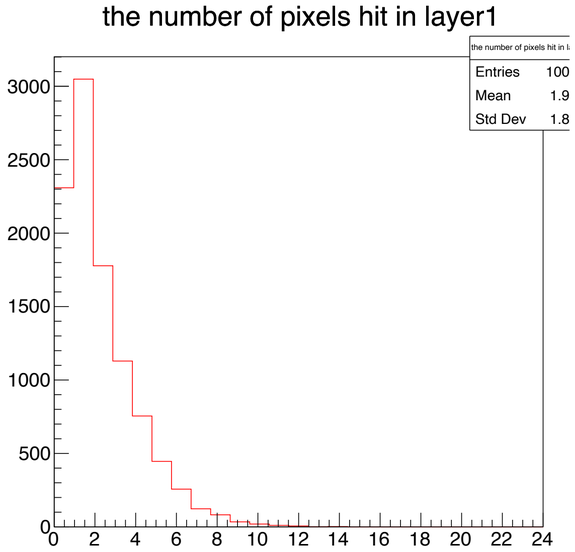
鳴らなかったイベント数はおよそ同じ割合だが、１枚だけ鳴ったイベント数は484枚にカットしている時の方が少ない
→分割数を増やした方が鳴るピクセル数は増加するはずである

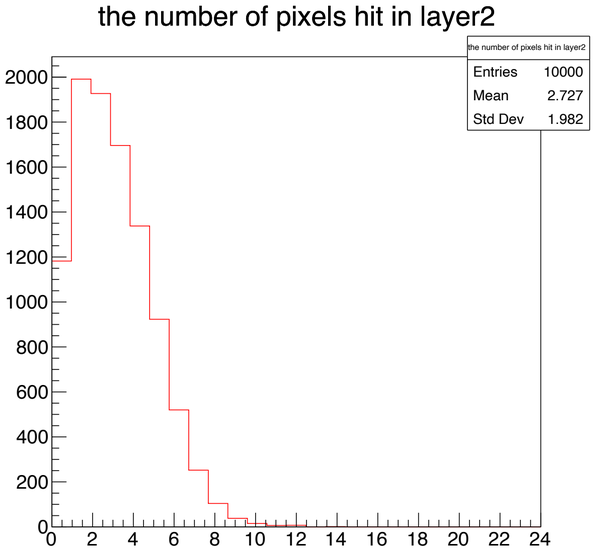
484枚に分割した場合のみ偶数枚にカットしていることが原因ではないか
→ピクセルの中央に入射するように入射位置をずらした



1GeVのγ線を打ち込んだ場合に鳴ったピクセルの数
（縦軸：イベント数　横軸：なったピクセル数）





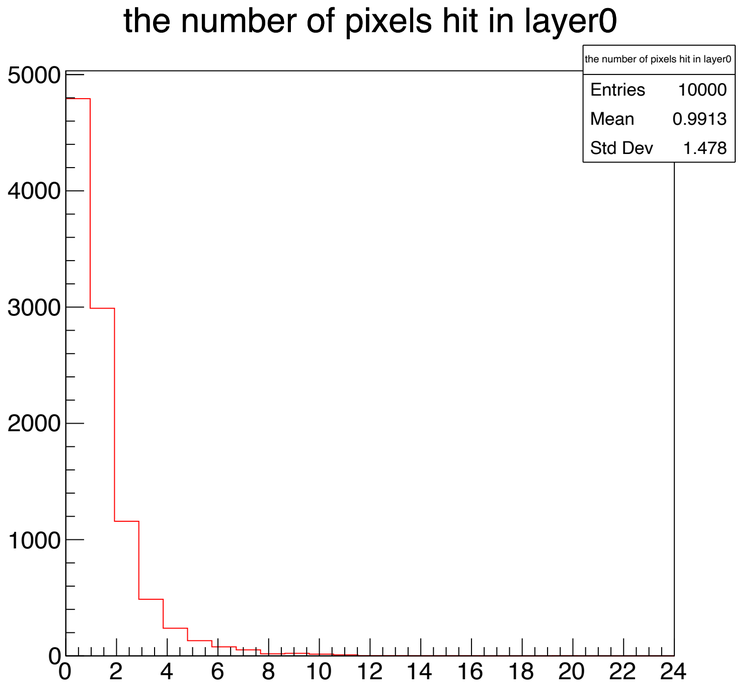


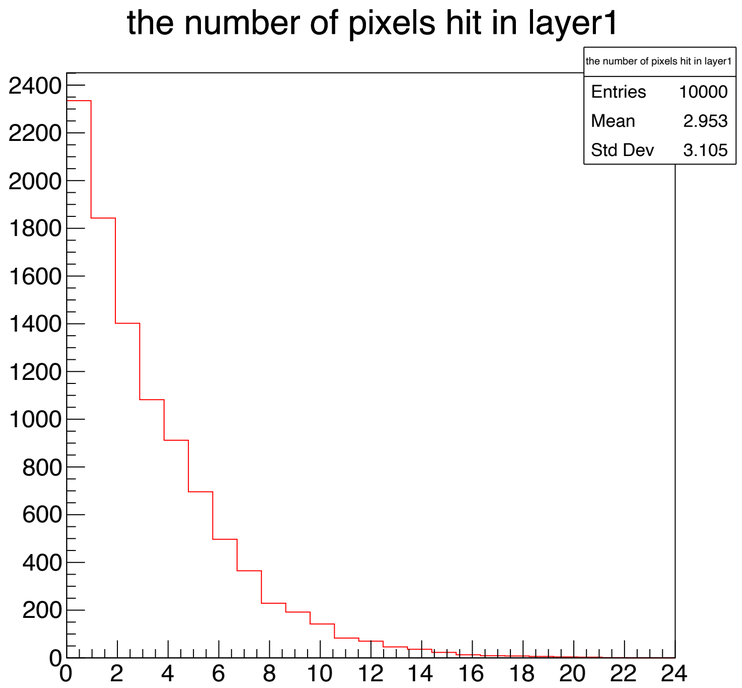
1枚ピクセルが鳴ったイベント数が増加し、奇数枚に分割した場合と似た分布形になった
→偶数枚に分割する場合には入射位置がピクセルの境界部分になることに注意しなければならない
→入射位置を変更してシミュレーションを行う場合にも入射位置に気をつける必要がある

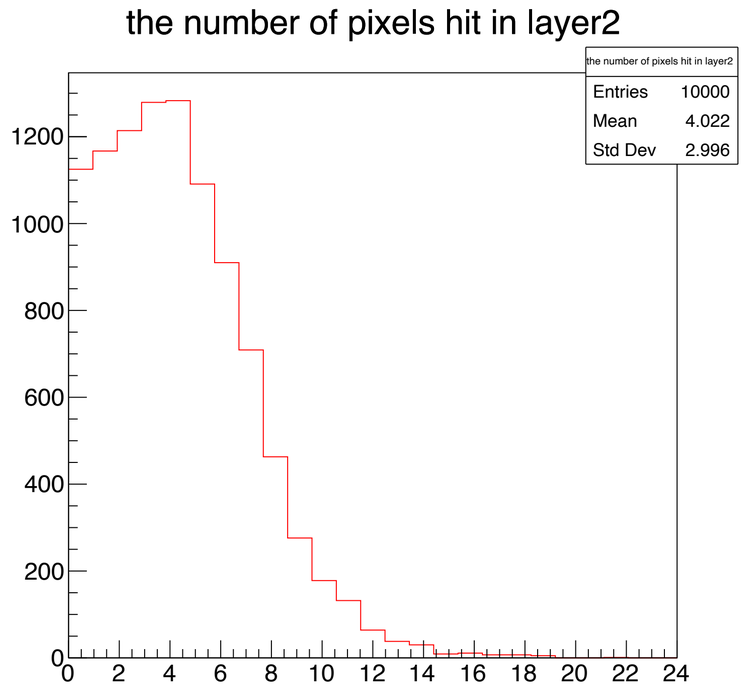
鳴ったピクセルの数

3layer　1mm×1mm（3025枚）に分割した場合

1GeVのγ線を打ち込んだ場合に鳴ったピクセルの数（縦軸：イベント数　横軸：なったピクセル数）







- 鳴っていないピクセルの割合は、layer0:約48%　layer1:約23%　layer2:約11%
- 鳴ったピクセルが1枚の割合は、layer0:約30%　layer1:約18%　layer2:約12%

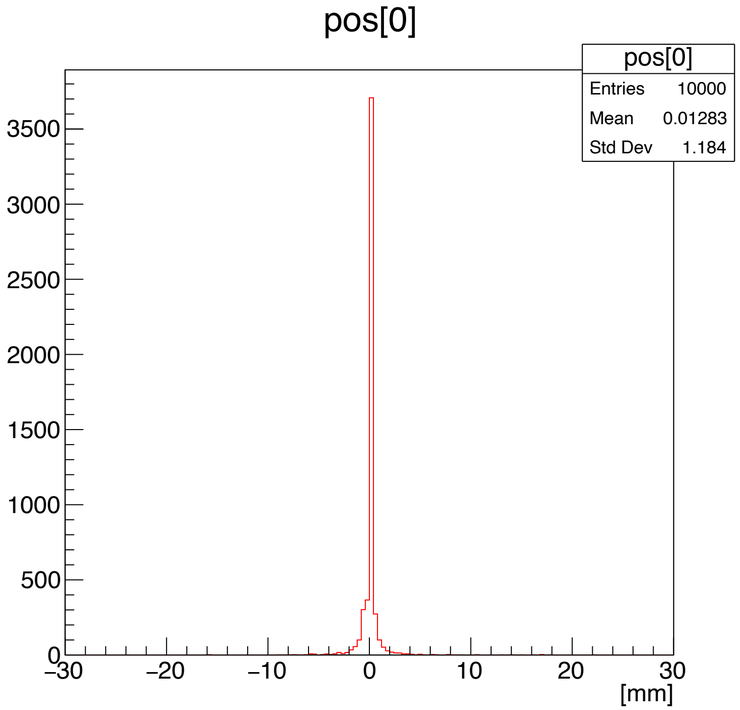
→複数枚鳴るイベントが多数見られる

電磁シャワーの中心位置

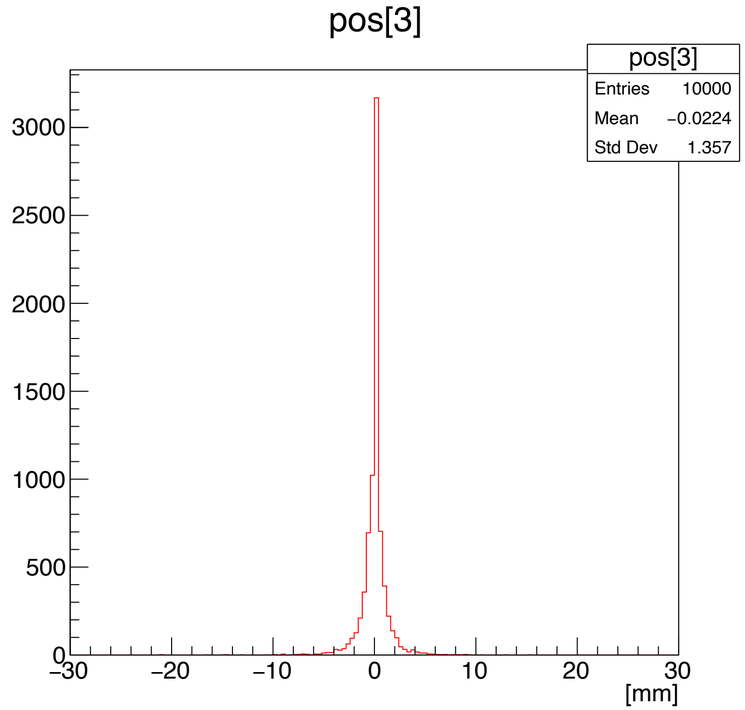
3layer　1mm×1mm（3025枚）に分割した場合

1GeVのγ線を打ち込んだ場合の電磁シャワーの中心位置のx座標（縦軸：イベント数　横軸：座標[mm]）

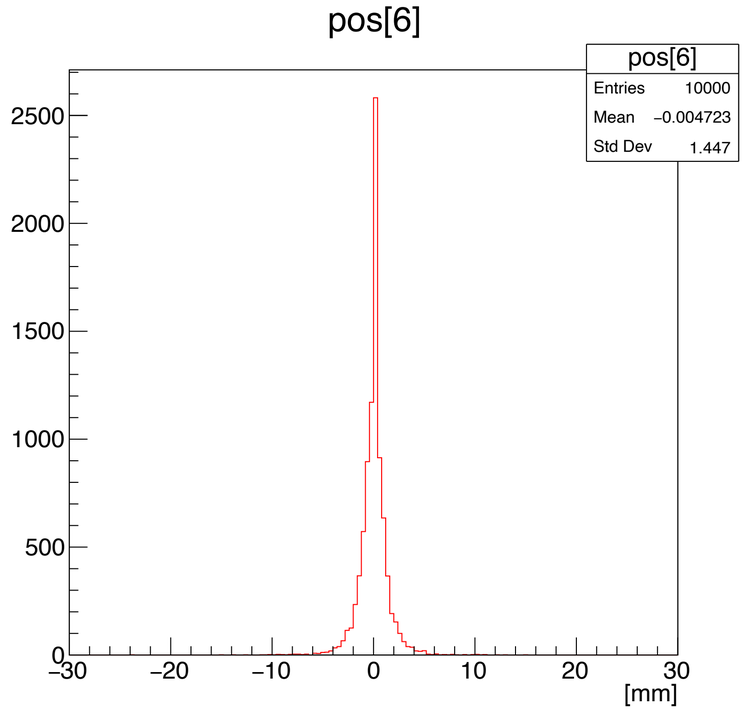
Layer0



Layer1



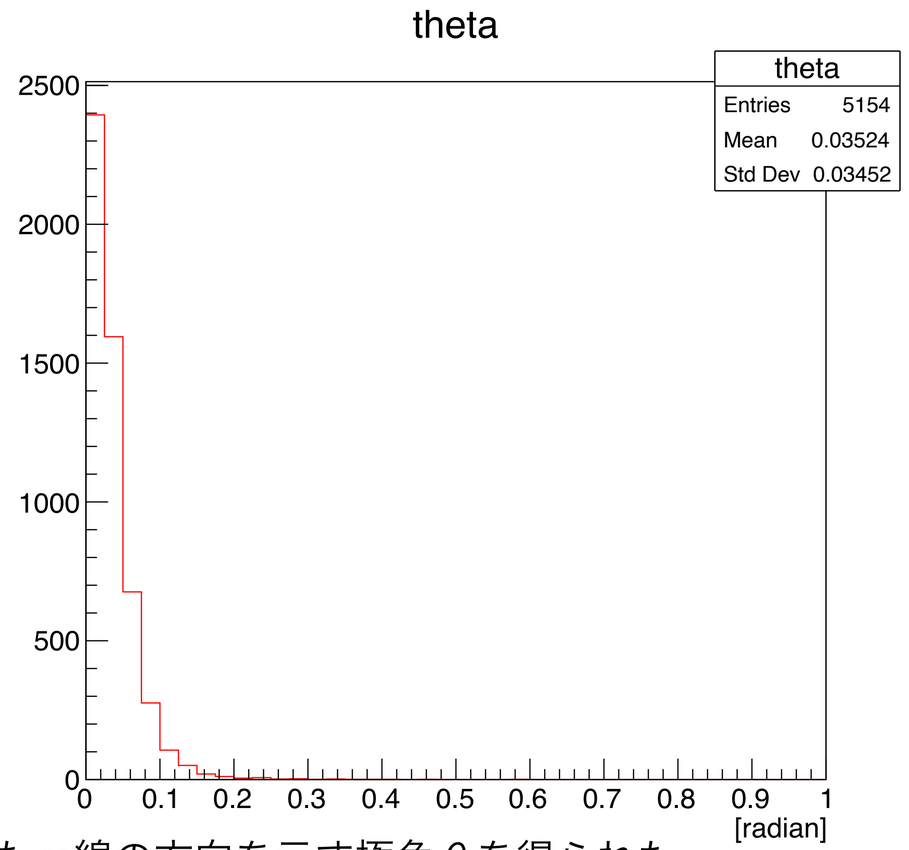
Layer2



- 約1.5mmの分解能で電磁シャワーの中心位置を得られた

3layer　1mm×1mm（3025枚）に分割した場合

1GeVのγ線を打ち込んだ場合に再構成したγ線の方向を示す極角θ（縦軸：イベント数　横軸：theta[rad]）



- 約0.03radiationの分解能で再構成したγ線の方向を示す極角θを得られた

目次

1. 背景
2. 目的
3. 方法
4. 結果
5. まとめ

まとめ

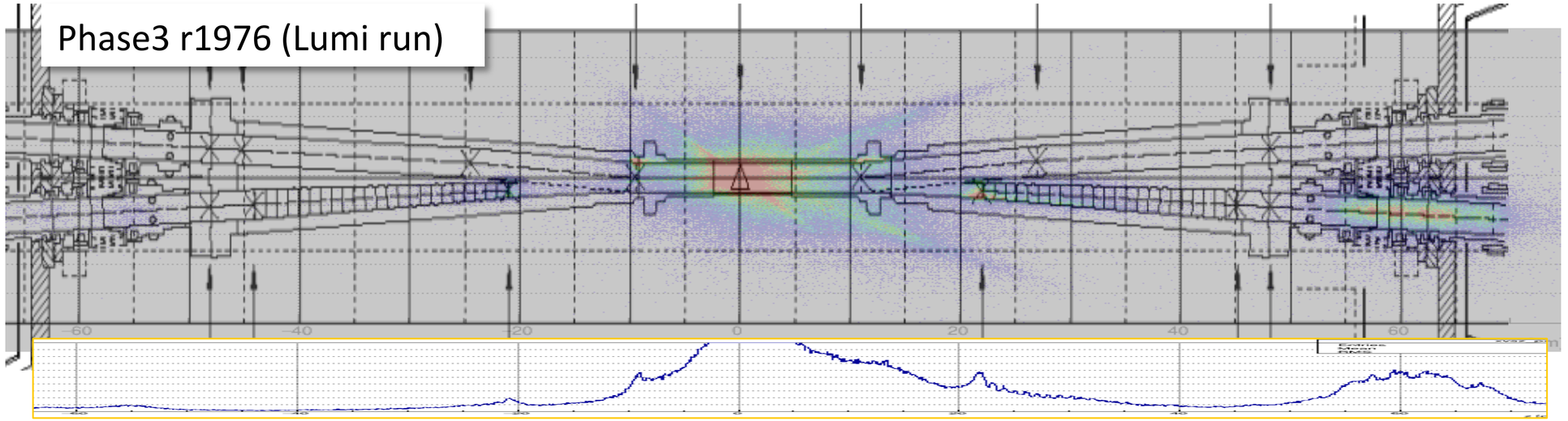
- プリシャワー検出器を導入することを検討するにあたり、
  Si検出器をピクセル構造を持つものに変更し、シミュレーションをすることができた
- 1.1cm角のピクセルでは大きすぎたので、1mm角のピクセルまでシミュレーションを行った
- 1mm角のピクセルにしたとき、シャワーの中心位置の分解能は約1.5mmであった。
- 再構成したγ線の方向を示す角度θの分解能は約0.03radiationと得た（ただし、入射位置はシミュレーションで既知の(x,y,z)=(0,0,0)に固定）。
- 今後、本研究を高度化する取り組みとして、入射方向が変化した場合、入射位置が変化した場合、到来方向に加えて入射位置もフリーパラメーターにした再構成を試みる、といった項目があげられる。

ご静聴
ありがとうございました！

めも

Heavy Metal Shield around Bellows Pipes

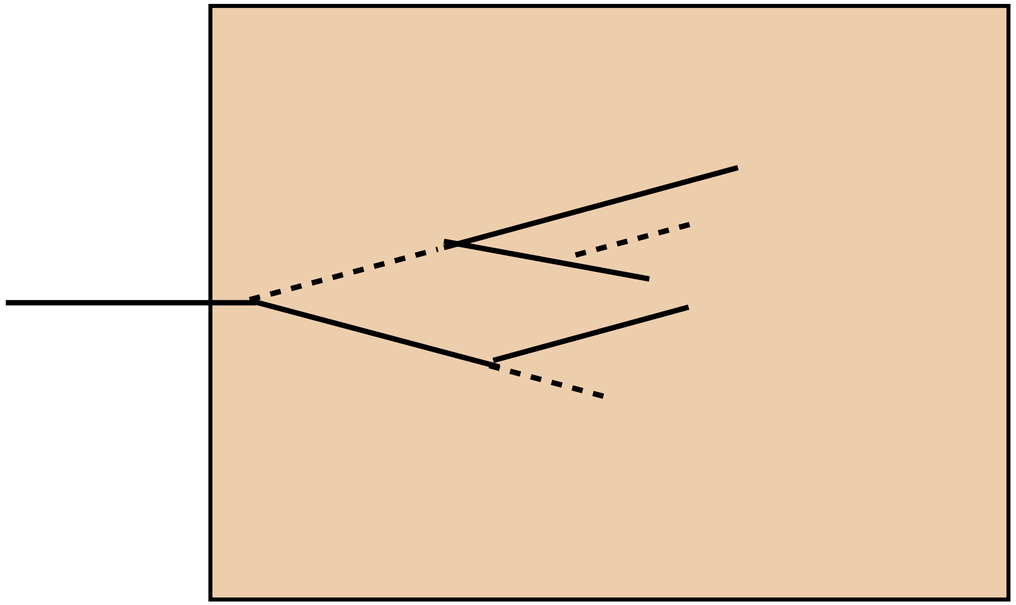
V0 distribution



- **We know the large amount of the beam BG is coming from the FWD bellows region.**
  - Must be a large contribution to the beam BG (at least CDC)
- **The reduction of this component is essential to achieve higher luminosity.**

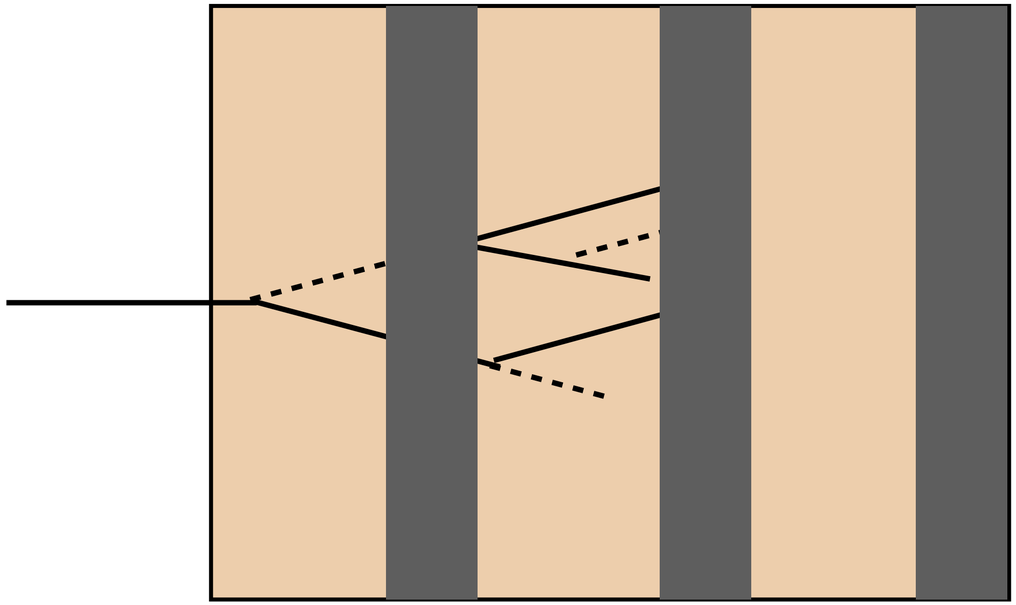
カロリメーター

全吸収型カロリメーター



・結晶シンチレータなどカロリメーターを構成する
全物質が検出器であるもの
・エネルギー分解能に優れる

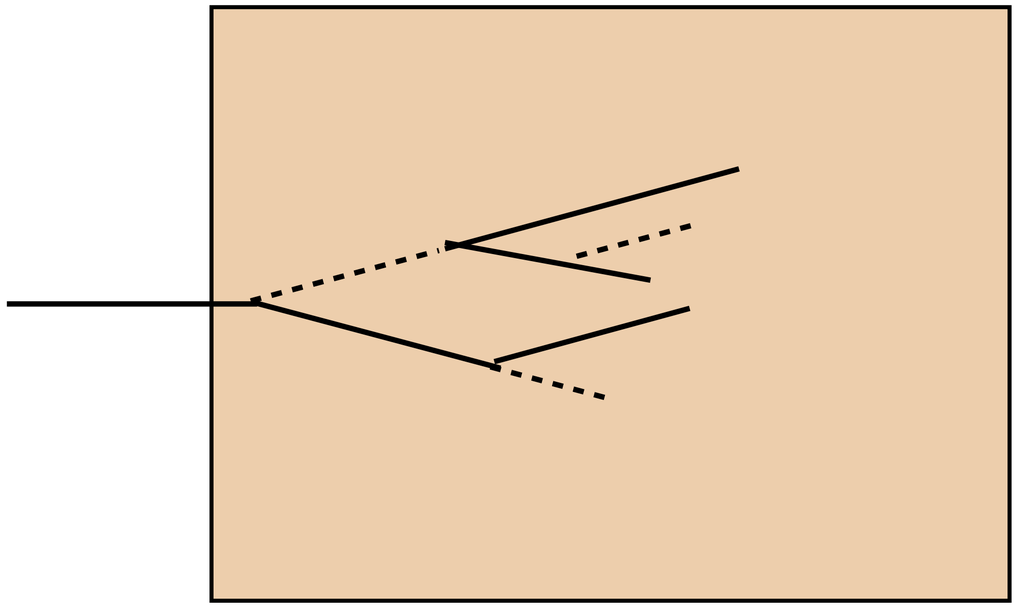
サンプリングカロリメーター



・プラスチックシンチレータなど鉄やタングステンと
検出器を交互に設置しているもの
・エネルギー分解能はある程度劣るが、
奥方向への位置分解能が得られる

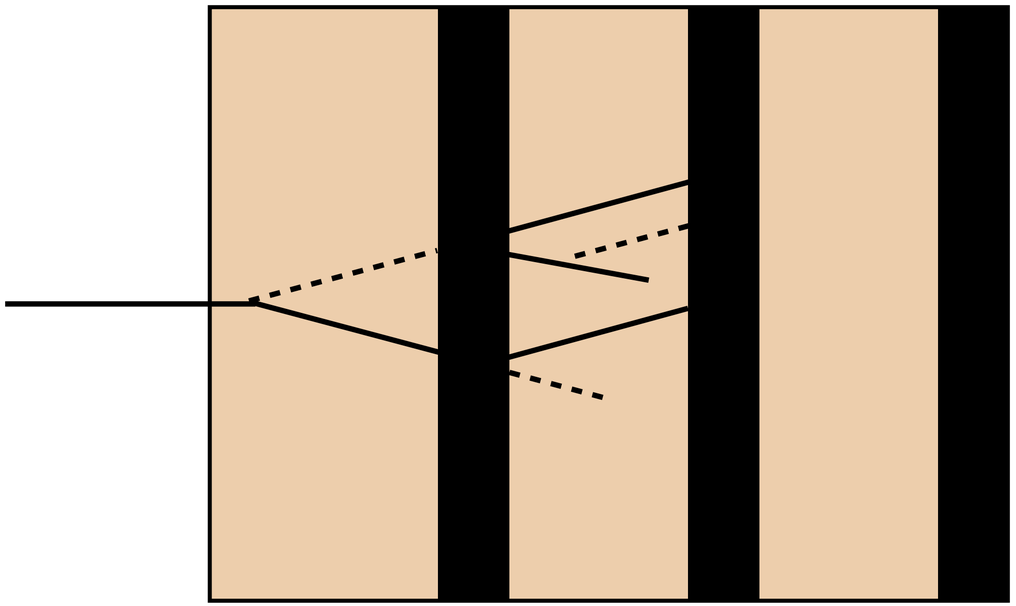
カロリメーター

全吸収型カロリメーター



・物質量の大きな検出層を持つ
・エネルギー分解能に優れている

サンプリングカロリメーター

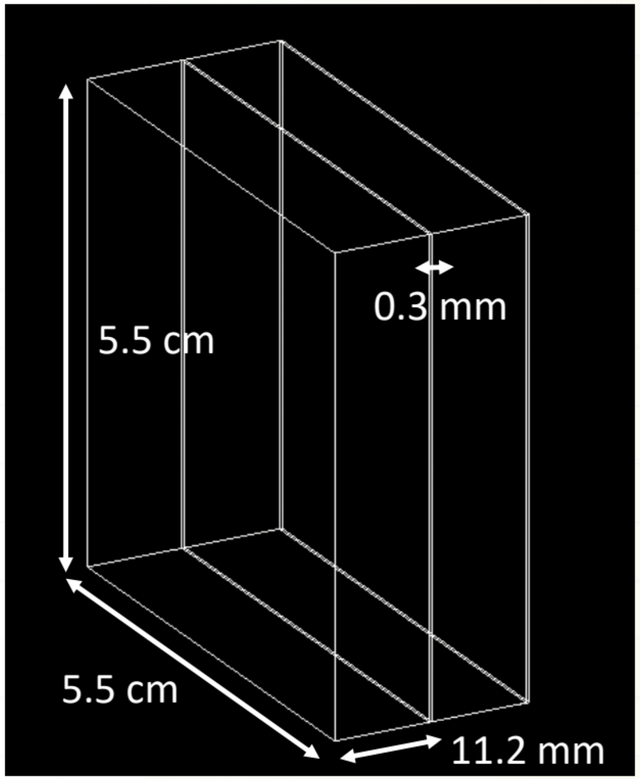


・物質量の大きな吸収層と
物質量の小さな検出層を持つ
・全吸収型に比べてエネルギー分解能は劣るが、
検出器を細分化すると近接したシャワーの分離
が可能になる

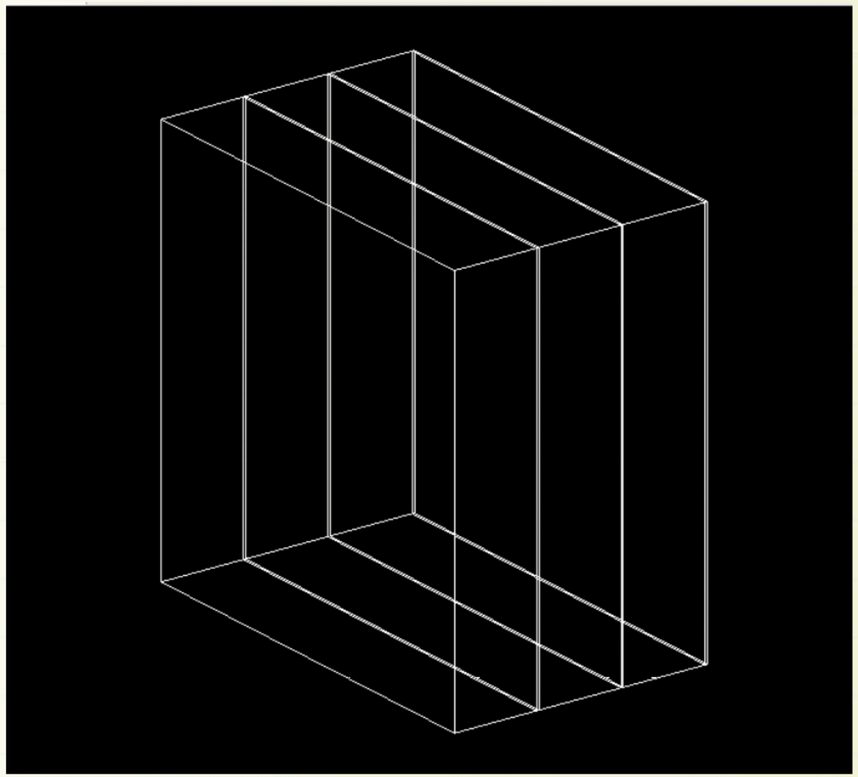
今野先輩の卒論スライドより

装置

- 2Layers



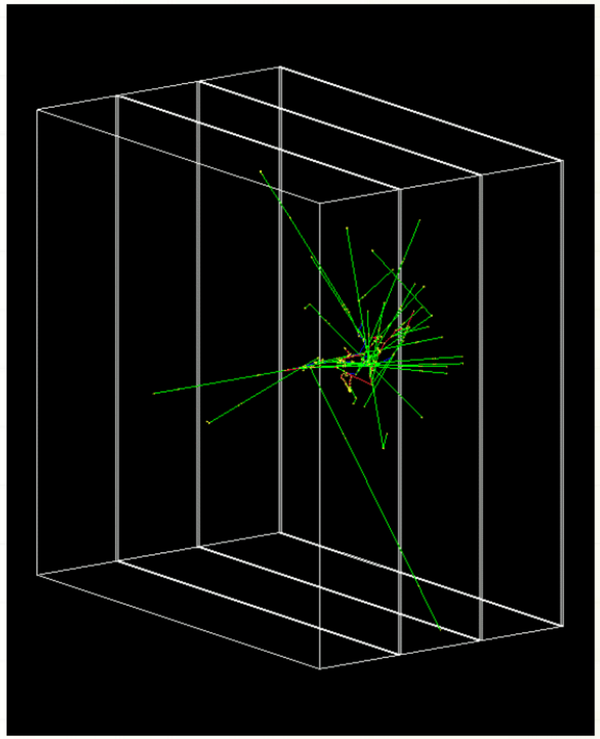
- 3Layers

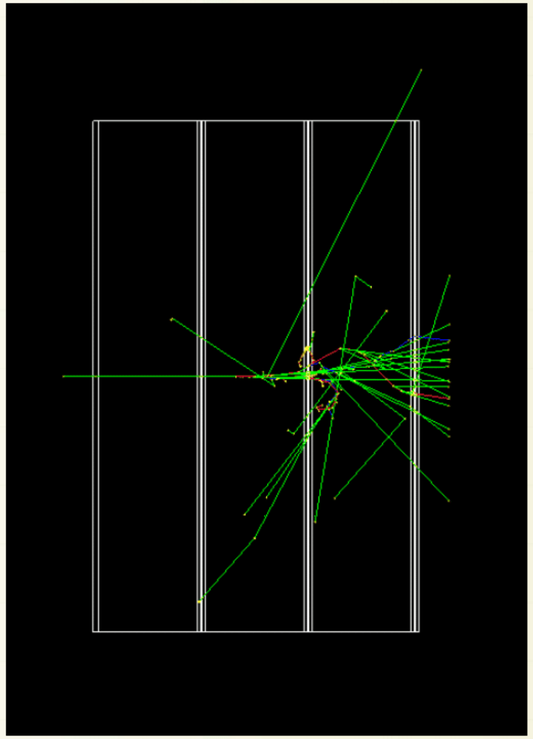


今野先輩の卒論スライドより

装置

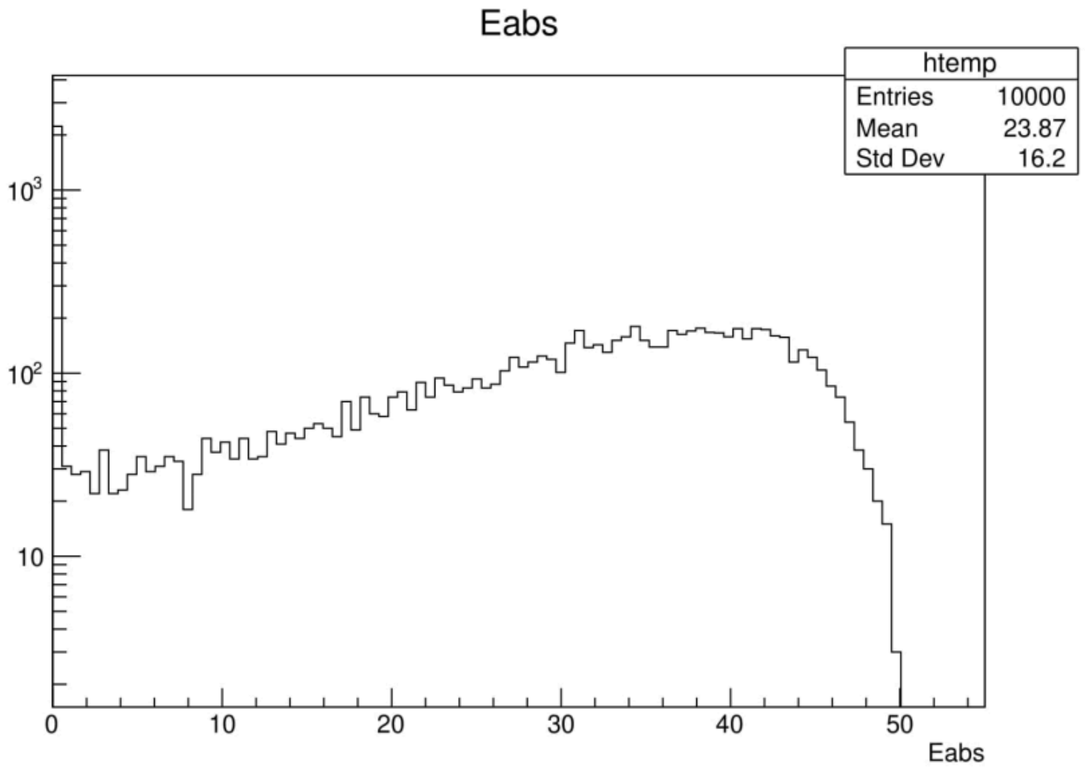
- 1beam入れたときの様子　(200 MeV 3Layers)

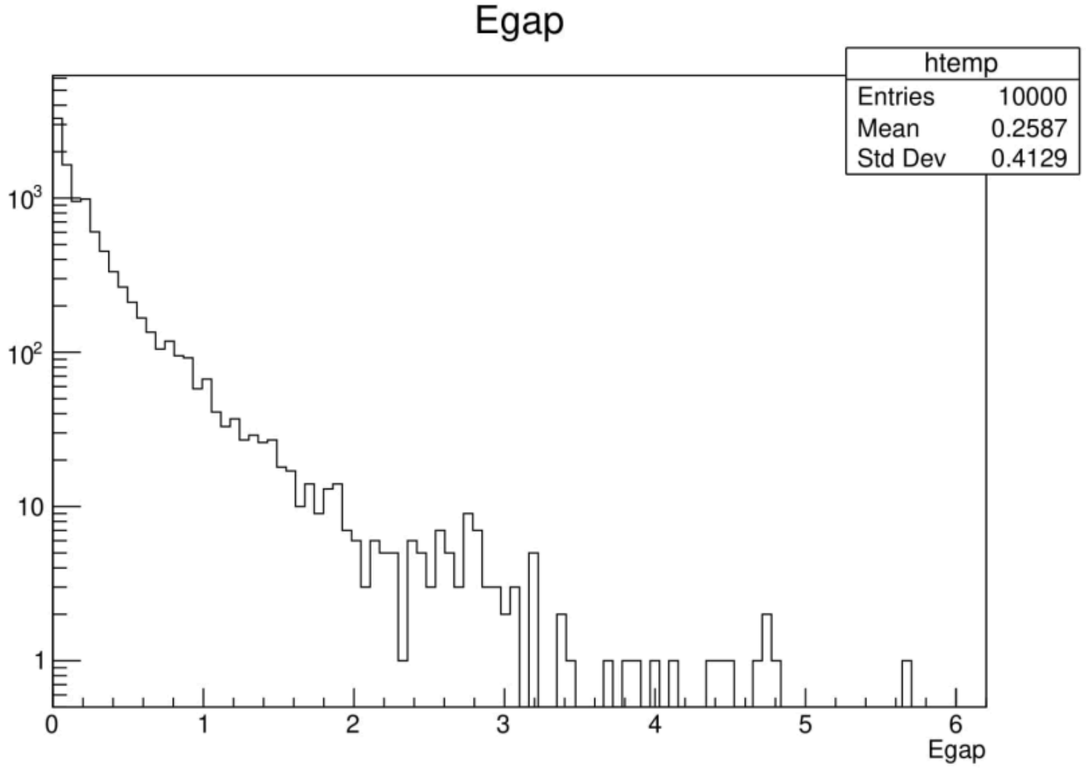




今野先輩の卒論スライドより

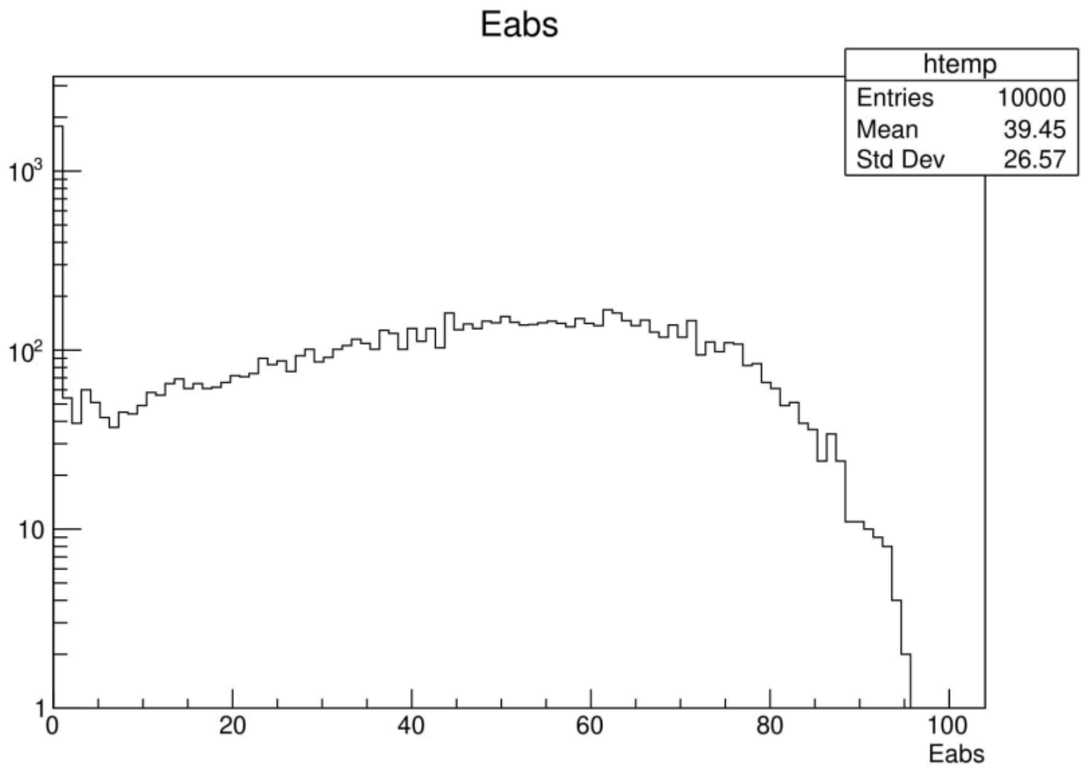
50MeV 3Layer

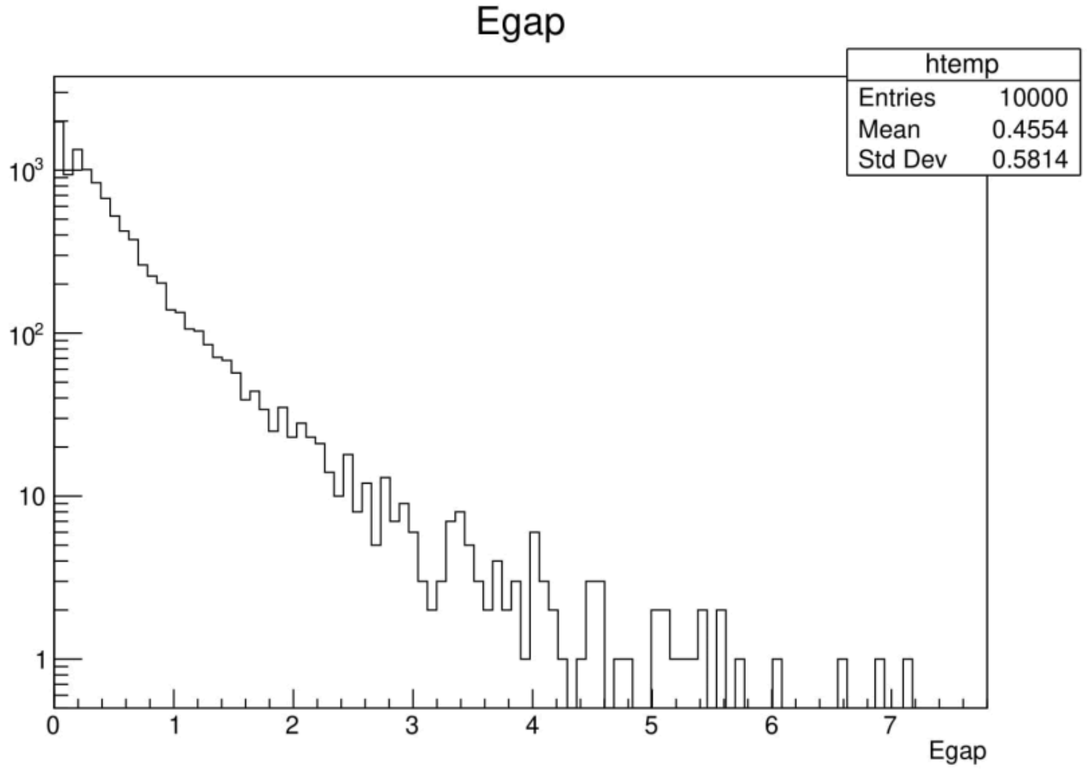




今野先輩の卒論スライドより

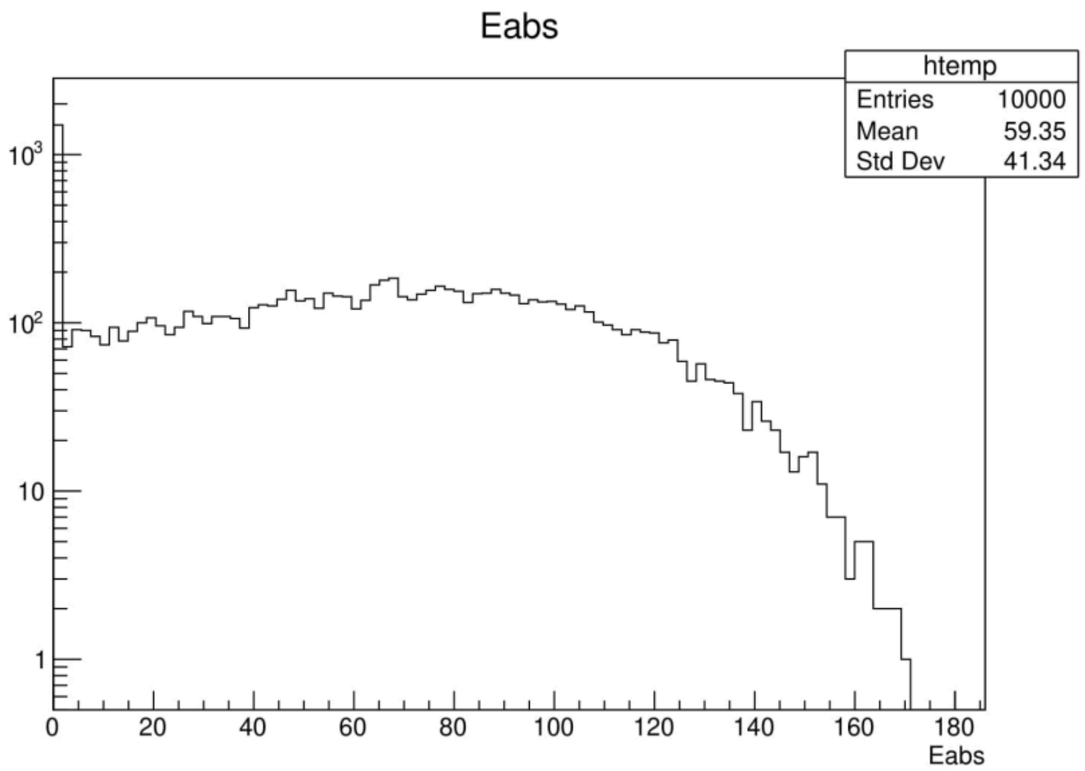
100MeV 3Layer

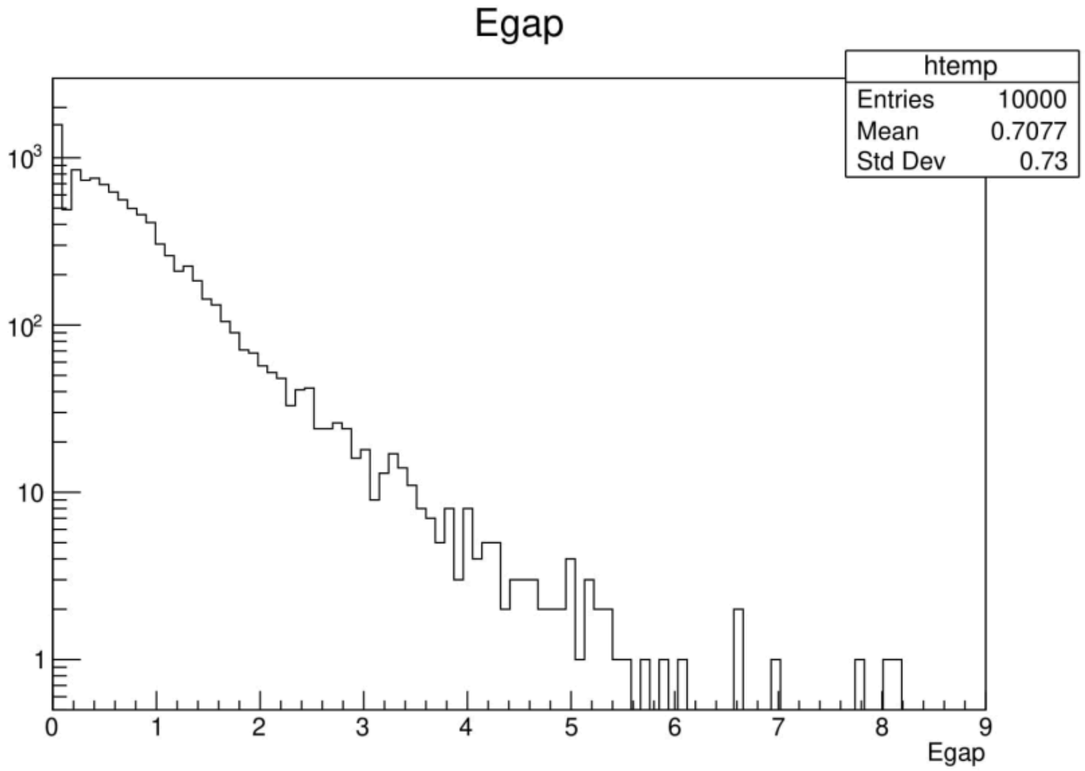




今野先輩の卒論スライドより

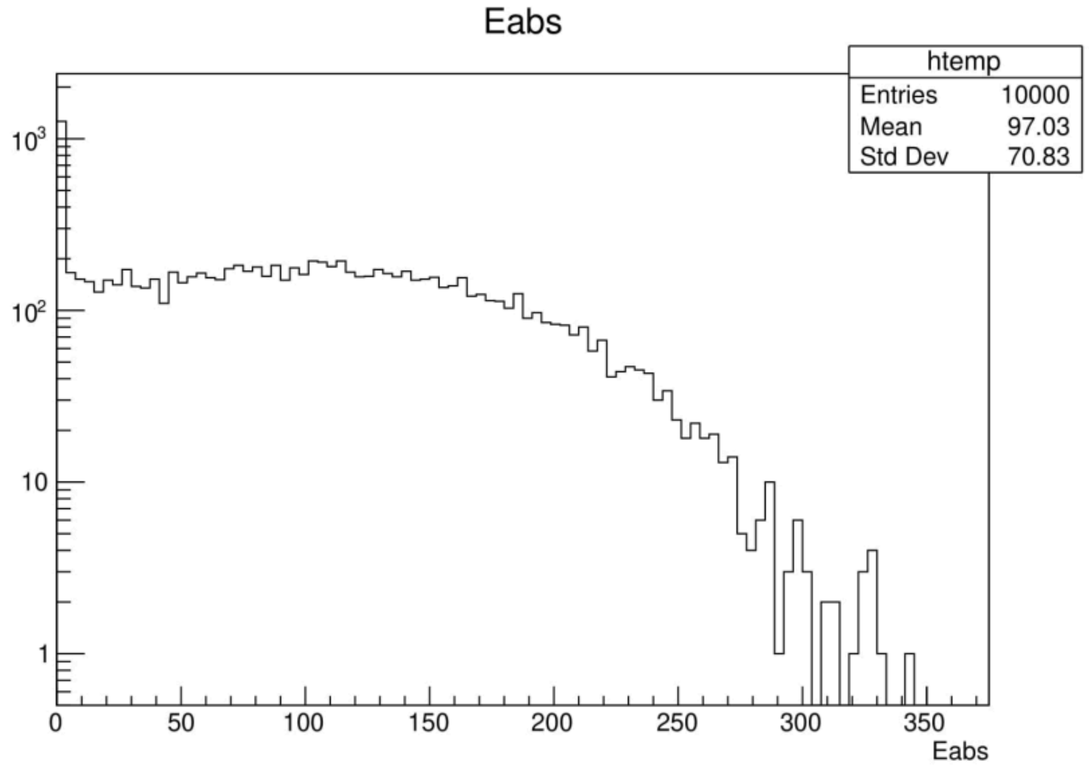
200MeV 3Layer

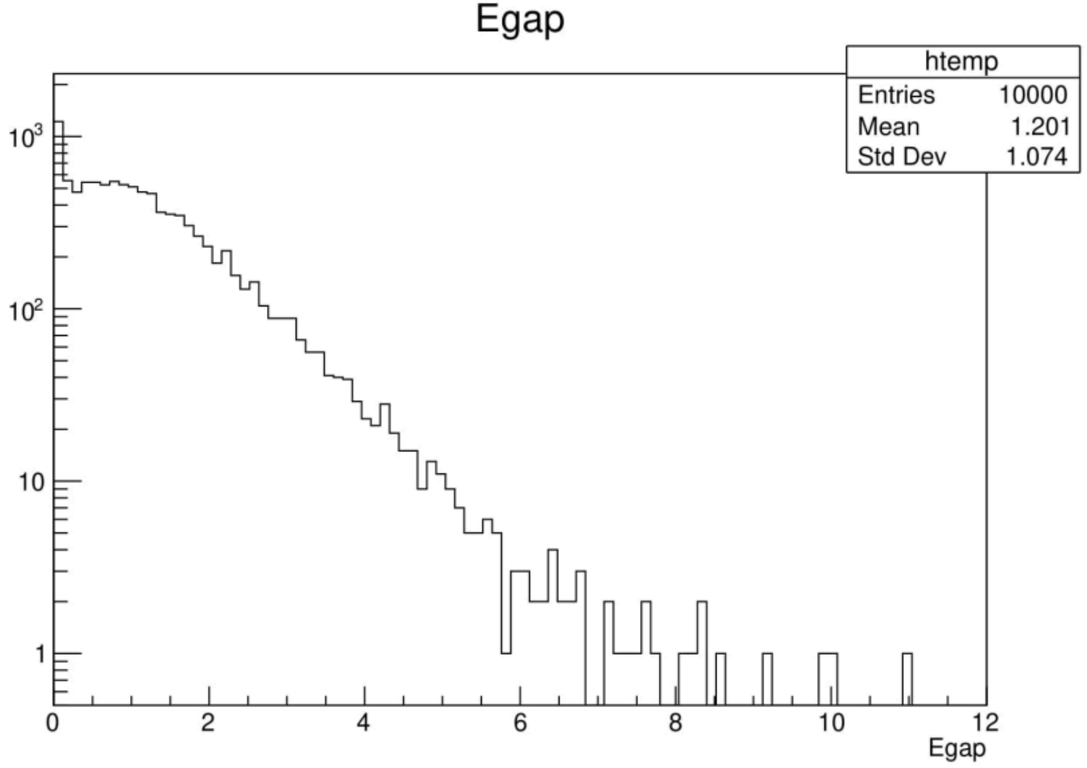




今野先輩の卒論スライドより

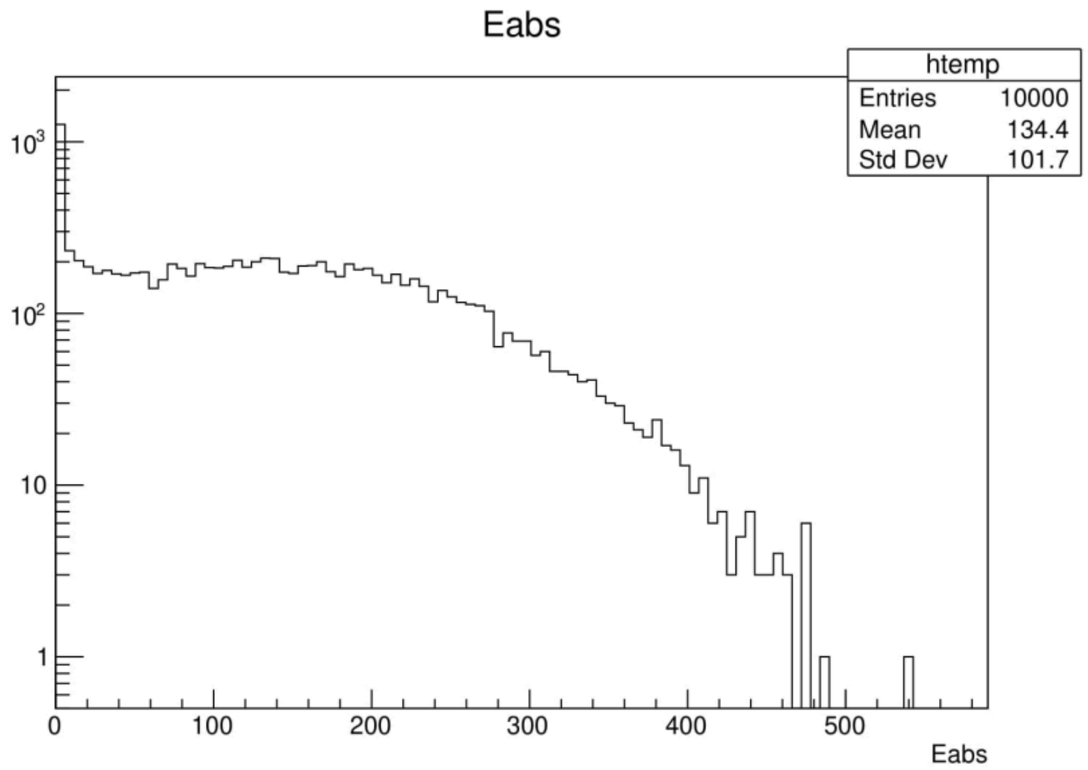
500MeV 3Layer

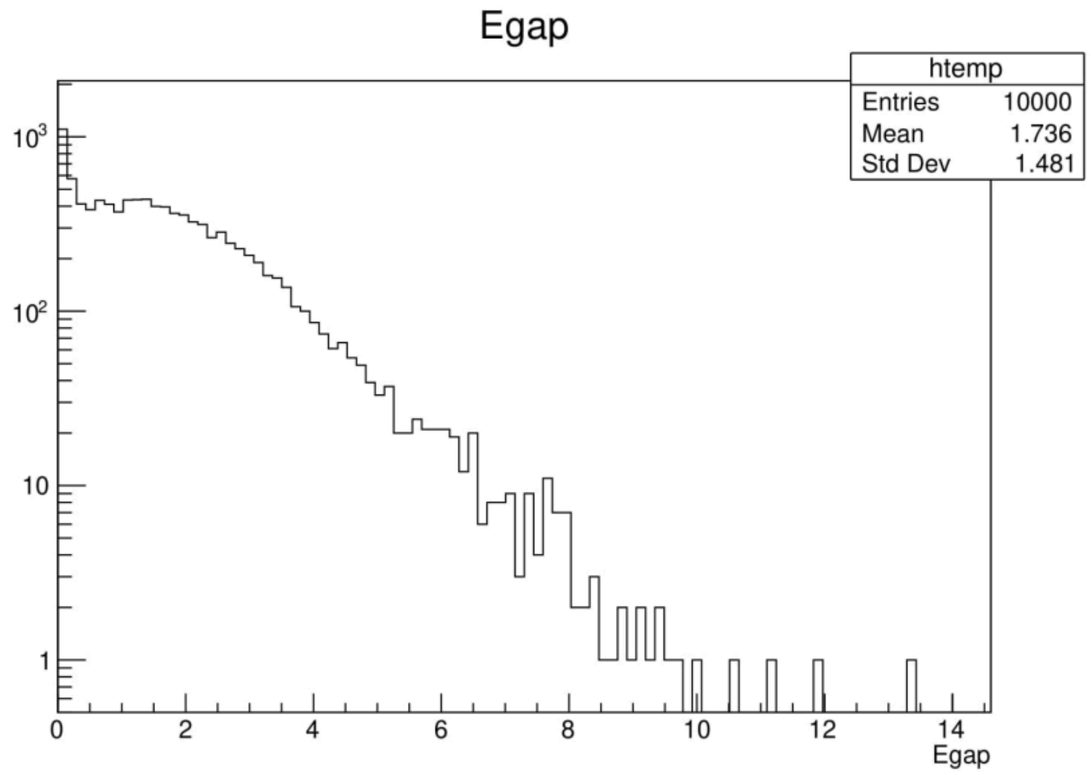




今野先輩の卒論スライドより

1000MeV 3Layer





プリシャワー検出器の物質

吸収層：BGOシンチレータ

・酸化物の一つであり、無色透明の物質

・γ線などの高エネルギー粒子に照射されると、ピーク波長480nmの緑色蛍光を発光する

・蛍光減衰時間：300ns

→ Belle Ⅱ 検出器の電磁カロリメーターに使用されているものと比較すると短い

・密度　7.13 g/cm^3

(吸収層：LYSOシンチレータ)

・放射長：11.4mm

・密度：7.4g/cm^3

・蛍光減衰時間:：約40ns

検出層：シリコン

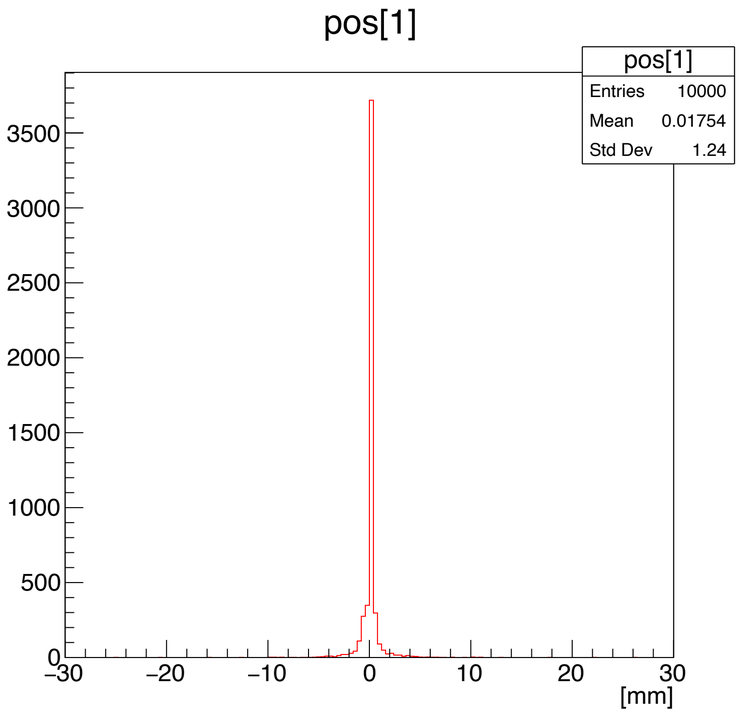
・密度2.33 g/cm^3

電磁シャワーの中心位置

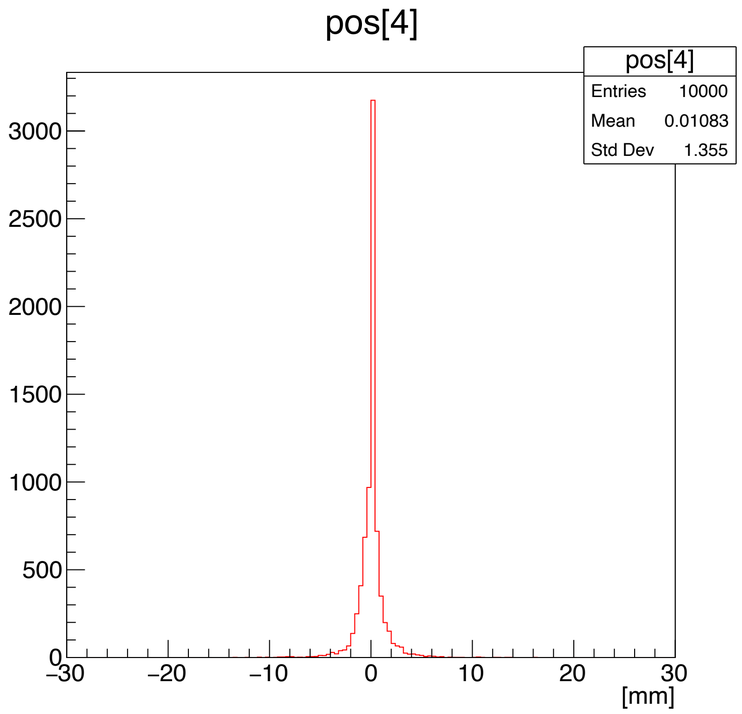
3layer　1mm×1mm（3025枚）に分割した場合

1GeVのγ線を打ち込んだ場合の電磁シャワーの中心位置のy座標（縦軸：イベント数　横軸：座標[mm]）

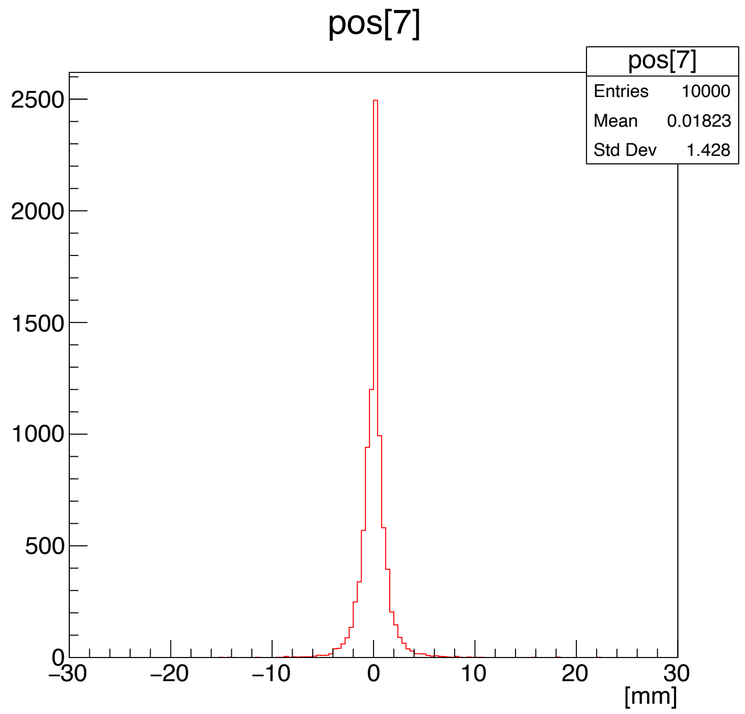
Layer0



Layer1



Layer2



- 約1.5mmの分解能で電磁シャワーの中心位置を得られた