sPHENIX実験INTT検出器のための Event Displayの開発

2023/03/03 高エネルギー物理学研究室卒業研究発表会
奈良女子大学 B4 藤原愛実

目次

1. 研究背景

QGP

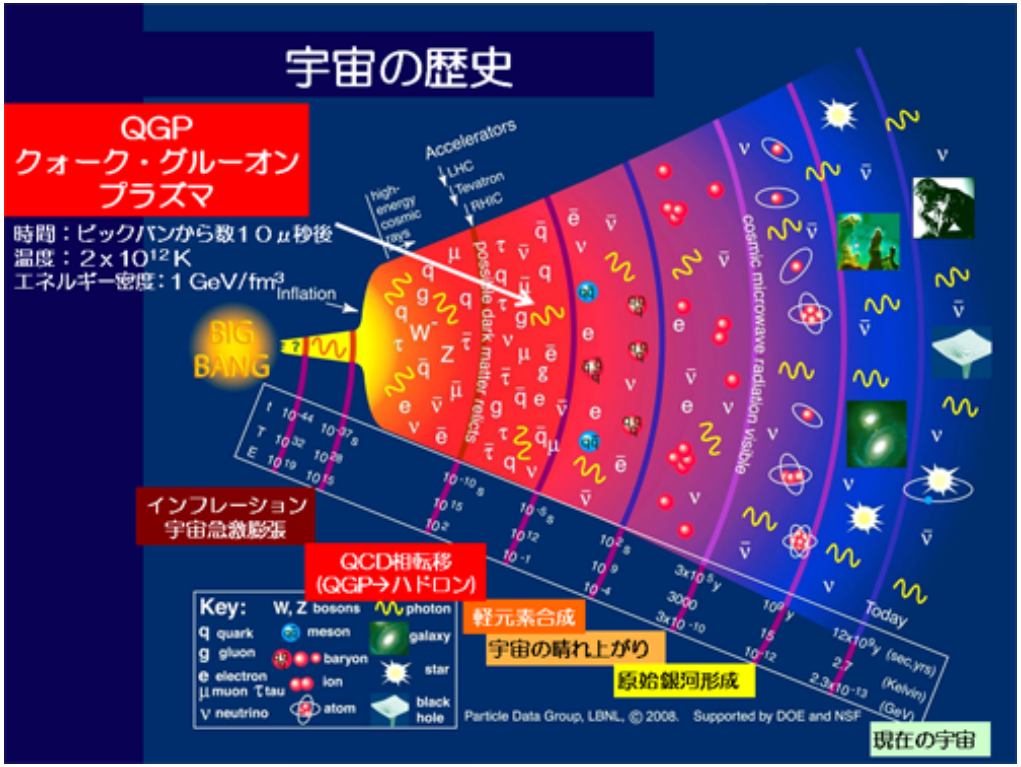
sPHENIX実験

INTT

QGP

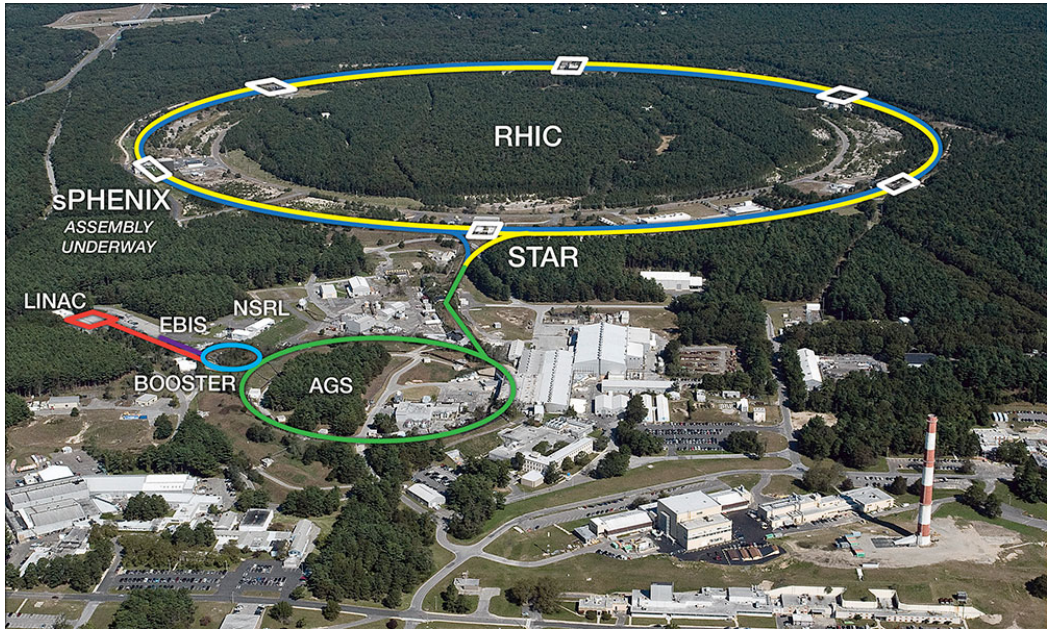
- クォークグルーオンプラズマ(QGP)は、通常では核子内に閉じ込められているクォークやグルーオンが超高温または高密度になることでプラズマ化した状態
- QGP は宇宙誕生後数十μ秒後に実現していた

QGP の性質を調べることで宇宙初期の状態を調べることができる



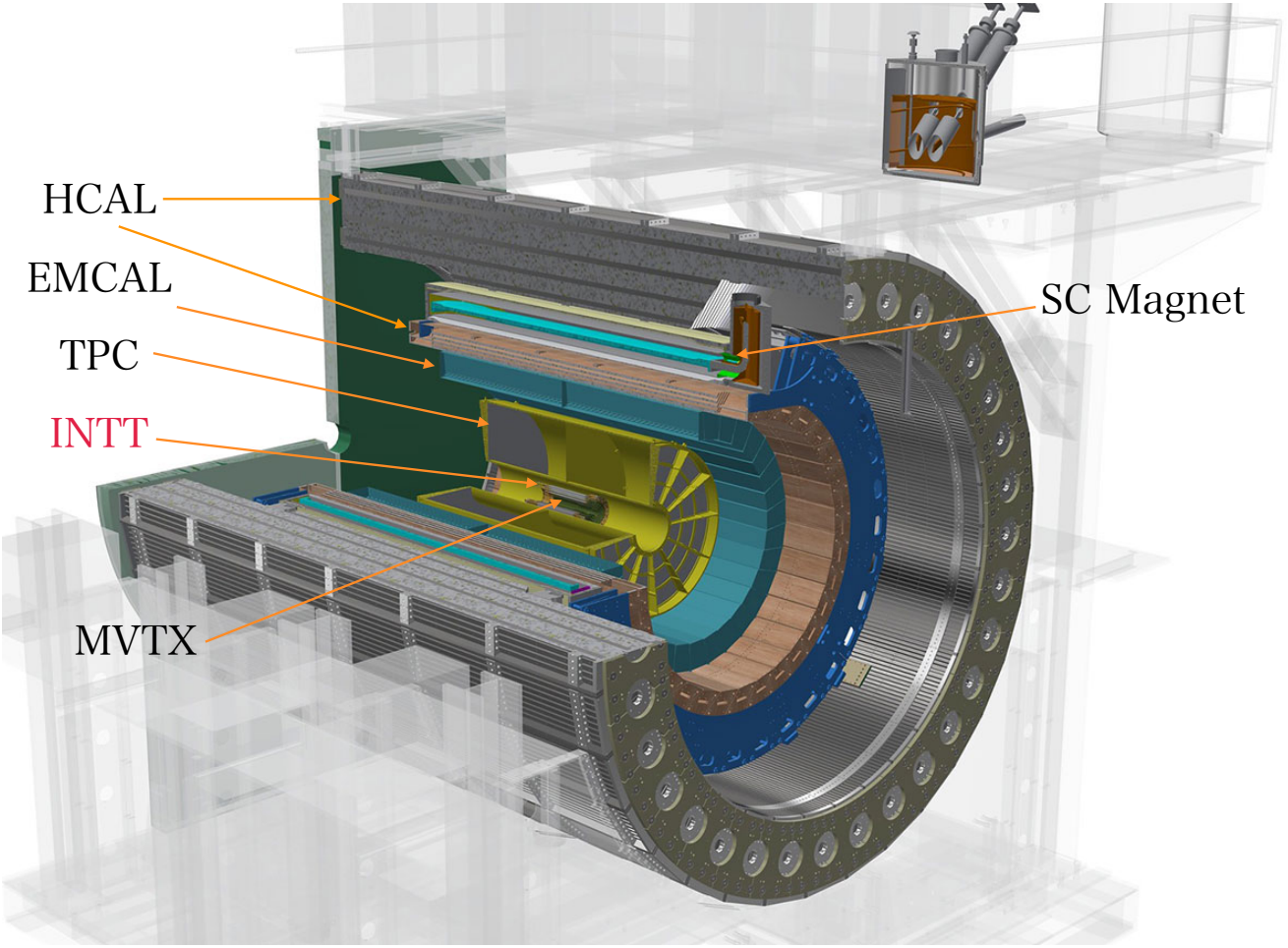
sPHENIX実験

- 米国ブルックヘブン国立研究所(BNL)のRHIC加速器を用いた実験
- 2000-2016年に行われたPHENIX 実験を高度化した実験
- 金原子核対衝突(200GeV)でQGPを実現する
- 2023年から、実験開始予定



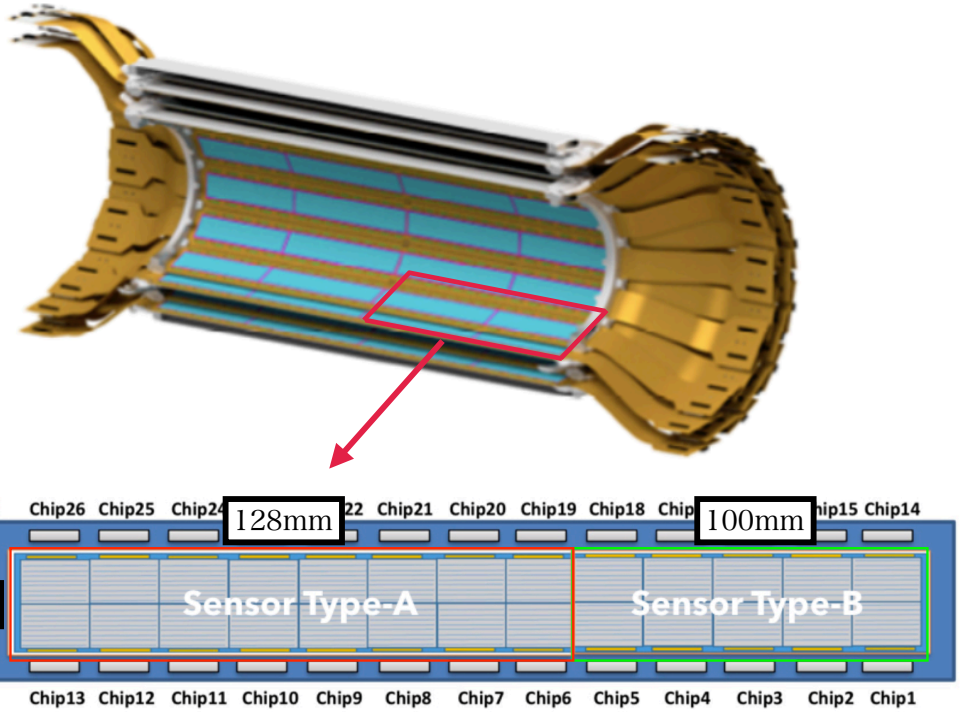
INTermediate Tracker(INTT)

- sPHENIX実験で使用される3つの飛跡検出器のうちの一つで、MVTXとTPCの間に配置される
- 時間分解能が高いことが特徴
- どのビーム交差で起きたイベントかを特定するために大きな役割を果たす



INTT用シリコンセンサー

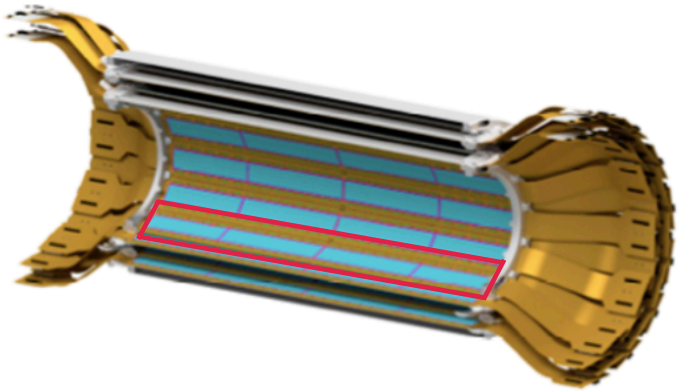
- INTT ではストリップ長の異なるtypeA、typeBという２種類のシリコンセンサーを使用
- 26個のストリップセンサーで1つのシリコンセンサー(ハーフラダー)が構成されている
- 各センサー領域のサイズはtypeAが128mm×19.9mm、typeBが100mm×19.9mm で厚さはどちらも0.32mm

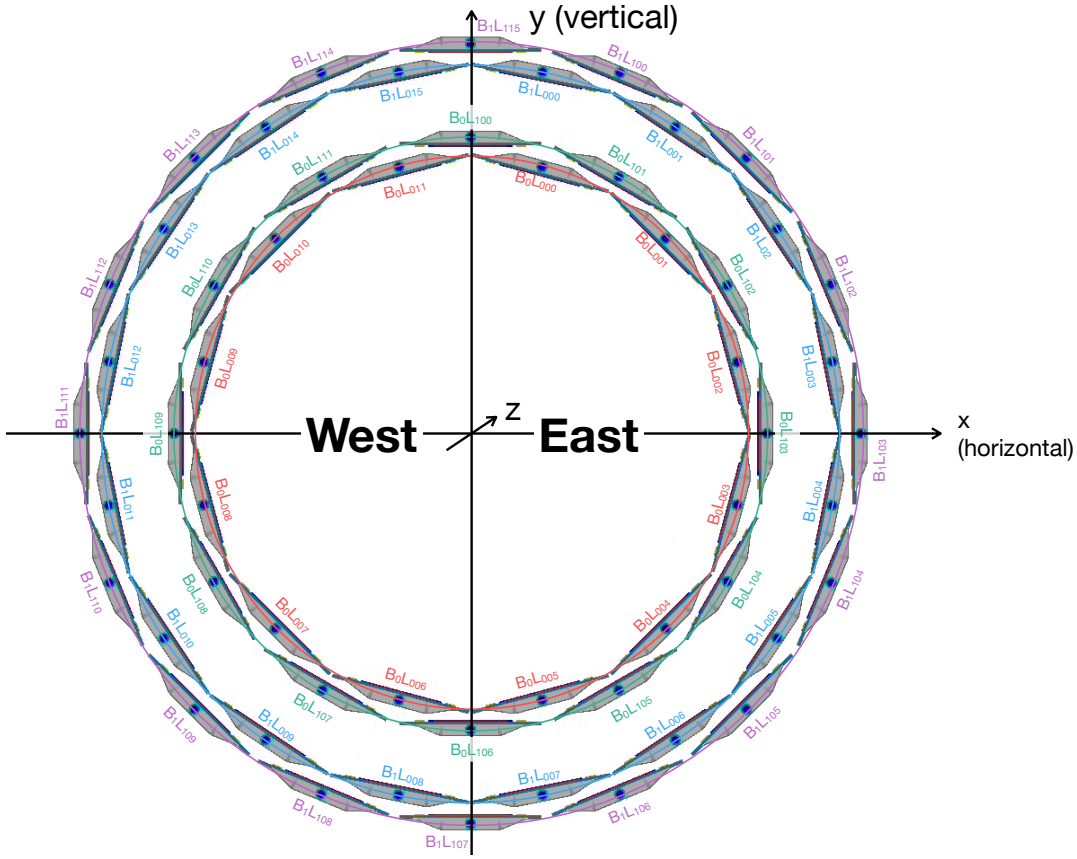


※以降、シリコンセンサーのtypeA またはtype Bのことをセグメントと呼ぶ

INTT用シリコンセンサー

- ハーフラダー2枚でフルラダーを構成している
- INTT はバレル状の2層構造で内側は24枚、外側は32枚のフルラダーで構成されている





2.研究目的

研究目的

Event Display とは

ROOT

研究目的

INTT Event Display の役目

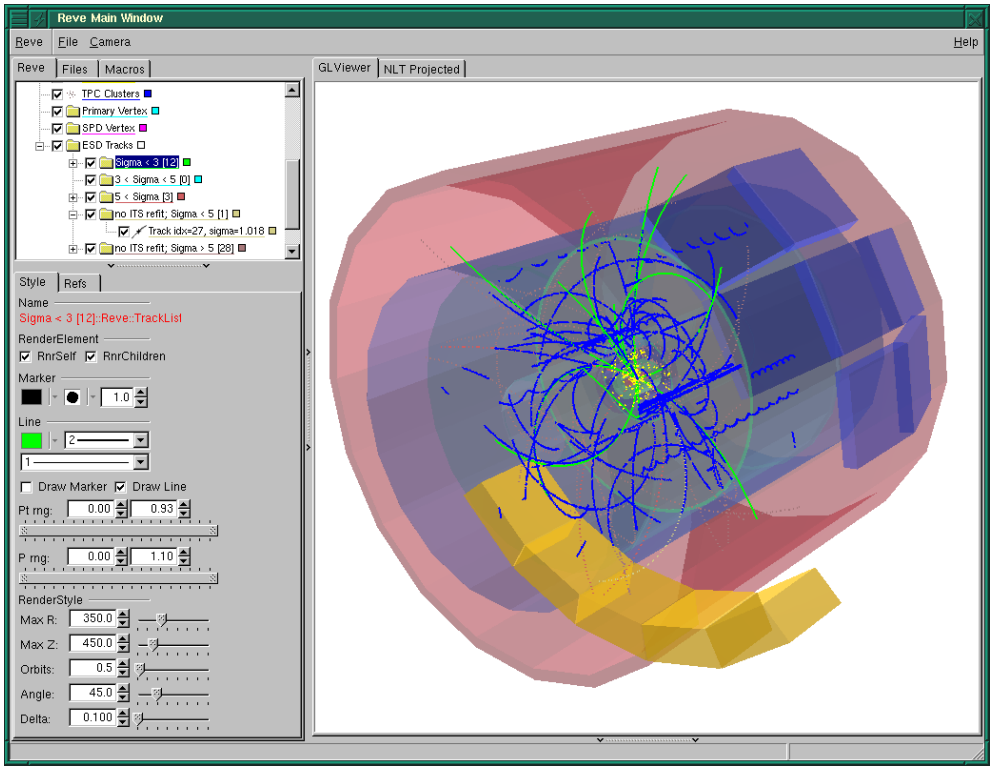
- INTT 上のヒット位置など、INTTに関する情報をイベントごとに一目見て確認する
- ラダーのアライメントを確認する
- INTT が正確に組み立てられていて、正常に稼働しているかを実験時にその場で確認する

EventDisplayとは

- 1イベント毎の粒子のヒット位置を、再構成した飛跡で結び、検出器のGeometry とともに描画したもの
- 一回の衝突を1イベントと数える
- 1イベントごとに複数回のヒットがある

イベント1

Hit 1の座標	(x_1,y_1,z_1)
Hit 2の座標	(x_2,y_2,z_2)
Hit 3の座標	(x_3,y_3,z_3)
⋮	⋮



ROOT

- CERN が開発した高エネルギー物理の解析に広く用いられているソフトウェア
- 基本的なプログラミング言語はC++
- 高エネルギー物理学の解析に特化したライブラリが数多く用意されている
- 本研究で主に使用しているライブラリはgeometry 作成用のTGeoとイベントディスプレイ開発用のTEve

Eve(Event Visualisation Environment)

- ROOTのイベント可視化環境
- イベントディスプレイ作成に必要な機能
  - ワールド(座標空間)作成
  - Hit 座標の保持
  - カメラの設定

  などが実装されている

TGeoManager

- ROOT のGeometry 作成をサポートしているクラス
- 検出器のGeometry 作成に必要な機能
  - ワールド(座標空間)の生成
  - 箱や筒などの簡単な形状の3DCG の作成
  - オブジェクトの階層構造の構築
  - 作成したオブジェクトの回転、移動
  - 作成したオブジェクトの複製

  などが実装されている

TGeoManager

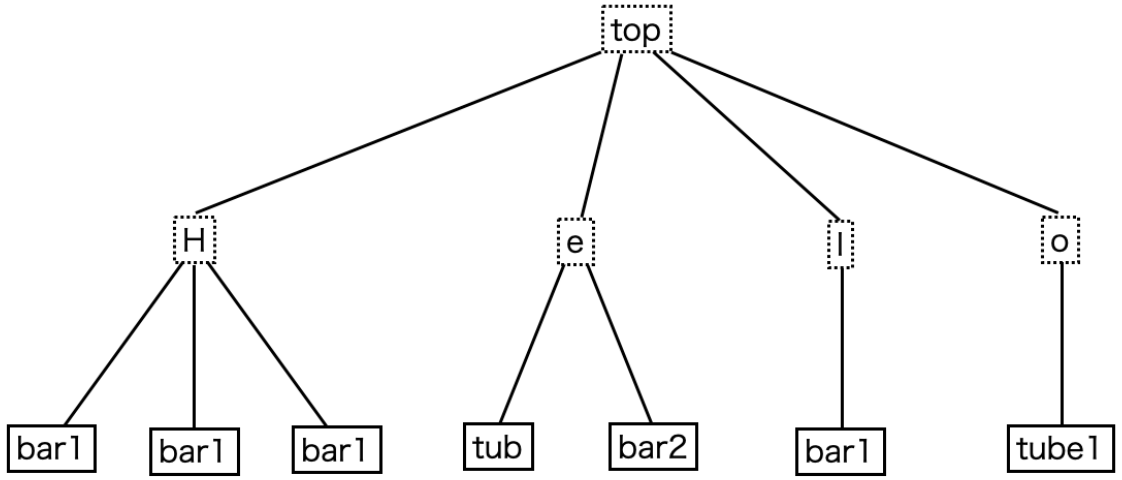
Example

```
1  void Hello(){
2      gSystem->Load("libGeom");
3      TGeoManager *geom = new TGeoManager("hello", "hello");
4
5      //top作成
6      TGeoVolume *top=geom->MakeBox("Top",NULL,100.,100.,100.);
7      geom->SetTopVolume(top);
8
9      //H
10     TGeoVolume * H=geom-> MakeBox("H",NULL,100.,100.,100.);
11     H->SetVisibility(kFALSE);
12     TGeoVolume * bar1=geom->MakeBox("BOX",NULL,0.5,0.5,6.0);
13     TGeoRotation * rot1 =new TGeoRotation("rot1",0,90,0);
14     TGeoCombiTrans * combi1 =new TGeoCombiTrans(0,0,5.5,rot1);
15     H->AddNode(bar1,0,combi1);
16     H->AddNode(bar1,1,0);
17     TGeoCombiTrans * combi2 =new TGeoCombiTrans(0,0,-5.5,rot1);
18     H->AddNode(bar1,2,combi2);
19
20     //l
21     TGeoVolume * l=geom-> MakeBox("l",NULL,100.,100.,100.);
22     l->SetVisibility(kFALSE);
23     TGeoCombiTrans * combi3 =new TGeoCombiTrans(0,0,0,rot1);
24     l->AddNode(bar1,0,combi3);
25
```

```
26     //e
27     TGeoVolume * e=geom-> MakeBox("e",NULL,100.,100.,100.);
28     e->SetVisibility(kFALSE);
29     TGeoVolume *bar2=gGeoManager->MakeBox("BOX",NULL,0.5,0.5,4.0);
30     e->AddNode(bar2,0,0);
31     TGeoVolume *tub=gGeoManager->MakeTubs("TUB",NULL,3.,4.,0.5,90.,400.);
32     TGeoRotation * rot2=new TGeoRotation("rot2",270,90,0);
33     TGeoCombiTrans *combi4=new TGeoCombiTrans(0,0,0,rot2);
34     e->AddNode(tub,1,combi4);
35
36     //o
37     TGeoVolume * o=geom-> MakeBox("o",NULL,100.,100.,100.);
38     TGeoVolume *tube=gGeoManager->MakeTube("TUBE",NULL,3.0,4.0,0.5);
39     TGeoCombiTrans *combi5=new TGeoCombiTrans(0,0,0,rot2);
40     o->AddNode(tube,0,combi4);
41
42     TGeoTranslation * tr1 =new TGeoTranslation(0,0,-17.5);
43     top->AddNode(H,1,tr1);
44     TGeoTranslation * tr2 =new TGeoTranslation(0,-2,-5.5);
45     top->AddNode(e,2,tr2);
46     top->AddNode(l,3,0);
47     TGeoTranslation * tr3 =new TGeoTranslation(0,0,4);
48     top->AddNode(l,4,tr3);
49     TGeoTranslation * tr4 =new TGeoTranslation(0,-2,12);
50     top->AddNode(o,5,tr4);
51
52     geom->CloseGeometry();
53     top->Draw("ogl");
54 }
```

TGeoManager

Example Nodeの階層構造



- 点線で囲まれたオブジェクトは表示されていない
- 階層構造を作れるのはTGeoVolumeオブジェクトのみ
- 親オブジェクトを移動、回転させると子オブジェクトも移動、回転する

3. INTT Event Displayの開発

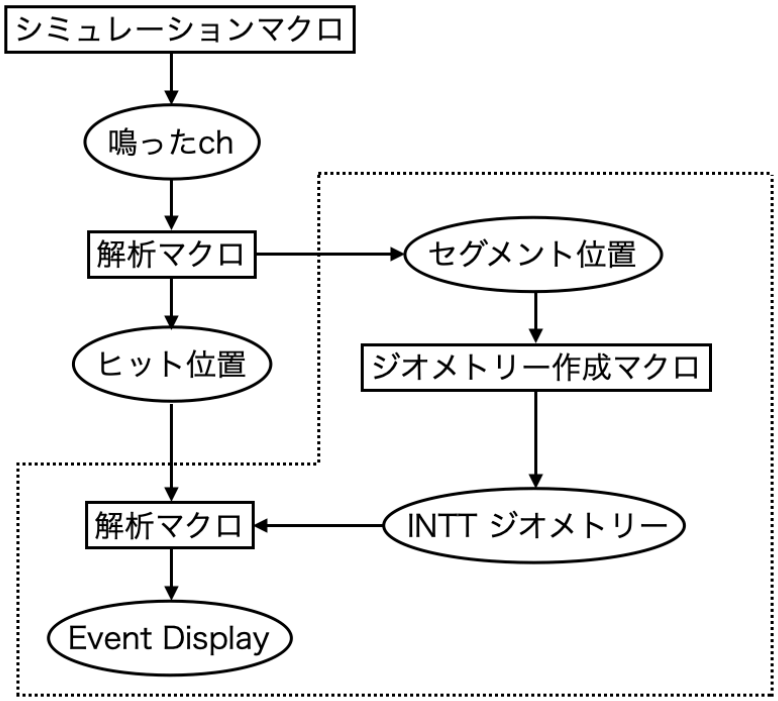
INTT Event Displayに実装されている機能

- ヒット位置とINTT Geometry を同時に描画
- 実験データと同じ形式のシミュレーションデータを読み込み、Event Display を出力する

Viewer機能

- r-ϕ プロジェクション

データフロー



解析マクロでしている処理

- シミュレーションで生成した、鳴ったchのデータをヒット位置に変換
- セグメントの位置を取得、ファイルに保存
- ヒット位置とGeometry を共に描画することでEvent Display 描画

INTT Geometry 作成

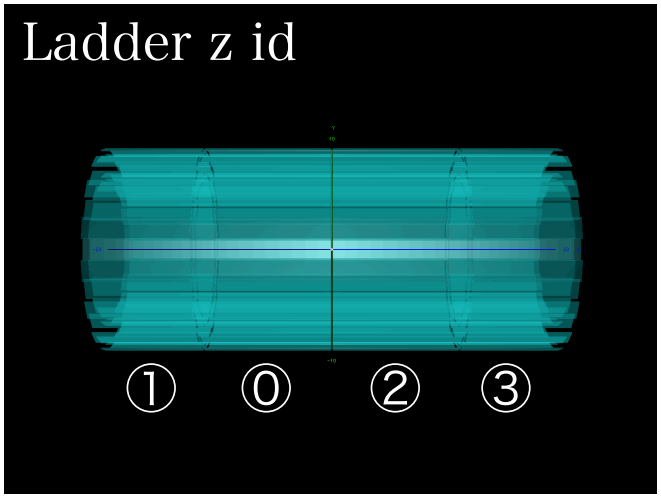
- セグメントの座標を再構成マクロから取得する
- INTT のGeometry を作成する
  1. セグメントサイズに合わせた箱を作成
  2. セグメントの座標から接線方向の角度を計算
  3. セグメントの座標と角度を設定して箱を複製

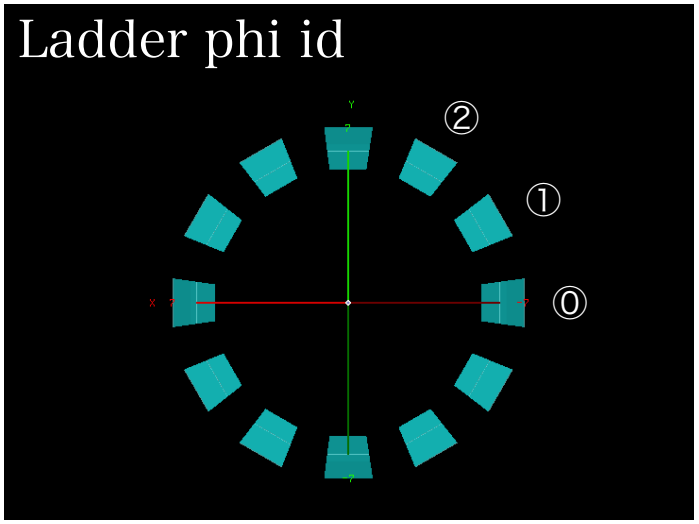
INTT Geometry の作成

セグメントの座標の取得

```
1187      //typeA position
1188      ladder_z_id = 0;
1189      for(uint8_t ladder_index=0;ladder_index<ladder_index_max;ladder_index++){
1190  auto genhitkeyintt = InttDefs::genHitSetKey(lyr,ladder_z_id,ladder_phi_id,
      time_bucket);
1191  auto surfintt = m_tGeometry->maps().getSiliconSurface(genhitkeyintt);
1192  double ladderLocation[3] = {0.,0.,0.};
1193  m_geom->find_segment_center(surfintt,m_tGeometry,ladderLocation);
1194  //cout << "ladderLocationA=(" << ladderLocation[0] << "," << ladderLocation
     [1] << "," << ladderLocation[2] << ")" << endl;
```

層, ladder z id,
ladder phi id を指定して
セグメントの座標を取得





INTT Geometry の作成

```
56
57      // segment
58      double segment_thickness = 0.32; // mm
59      double segment_thicknessrphi = 1.0;
60      double segment_widthY = 19.9;   // mm
61      double segmentA_widthZ = 128.0; // mm
62      double segmentB_widthZ = 100.0; // mm
63      TGeoVolume *segmentA = geom->MakeBox("BOX", NULL, segment_thickness /
        10 / 2, segment_widthY / 10 / 2, segmentA_widthZ / 10 / 2);
64      TGeoVolume *segmentB = geom->MakeBox("BOX", NULL, segment_thickness /
        10 / 2, segment_widthY / 10 / 2, segmentB_widthZ / 10 / 2);
```

セグメントのサイズを設定

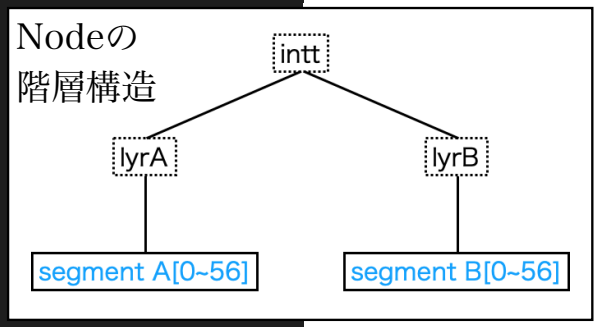
セグメントサイズに
合わせた箱を作成

```
119         for (int i = 0; i < sum_segmentA; i++)
120         {
121             phi = atan2(segmentALocationY[i], segmentALocationX[i])
                * 180 / M_PI;
122             rotA[i] = new TGeoRotation(Form("rotA[%d]", i), phi,
                0., 0.);
123             combitranslyrA[i] = new TGeoCombiTrans(segmentALocationX
                [i], segmentALocationY[i], segmentALocationZ[i], rotA
                [i]);
124             lyrA->AddNode(segmentA, i, combitranslyrA[i]);
125         }
```

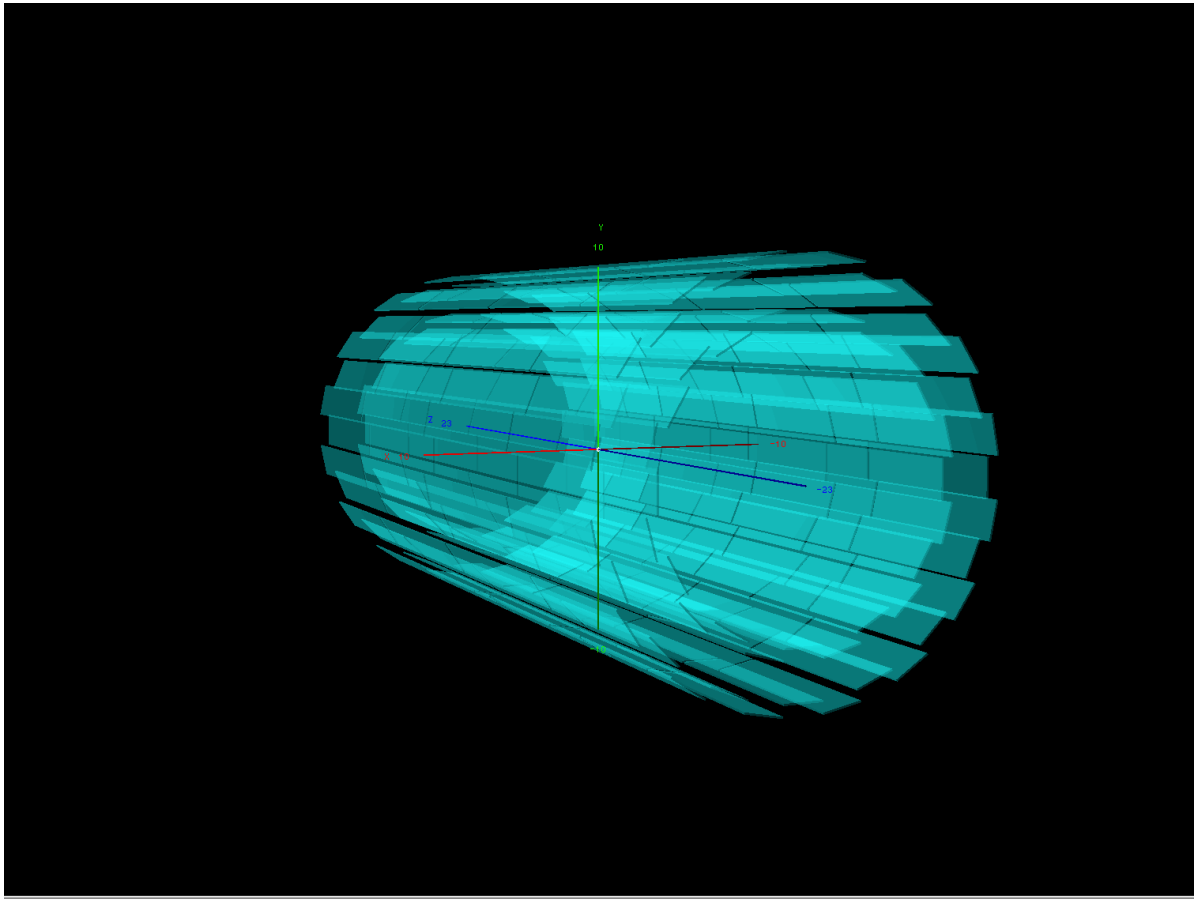
回転角度を計算

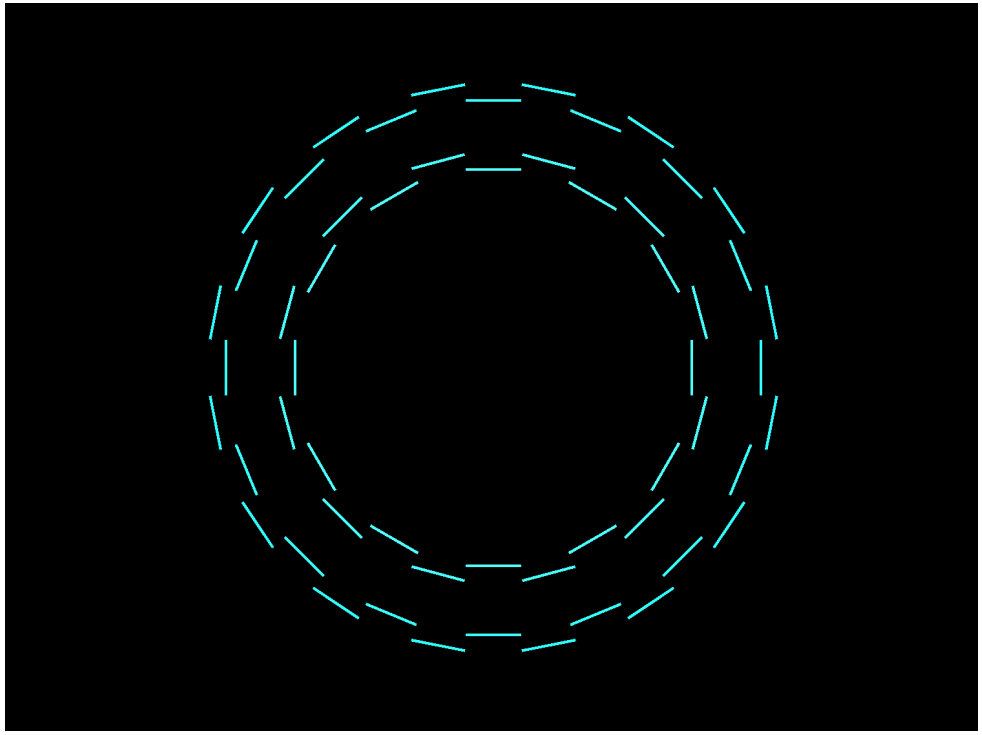
回転角度と
座標を設定

設定を反映してセグメントを複製



INTT Geometry



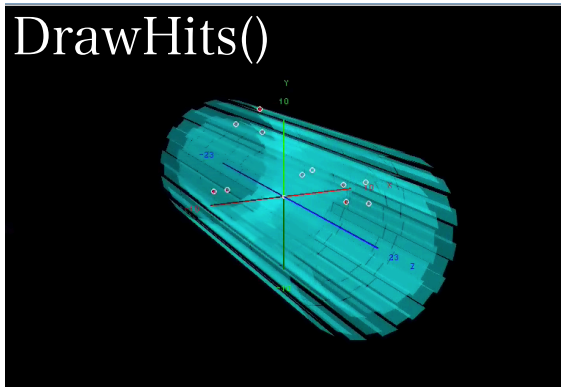


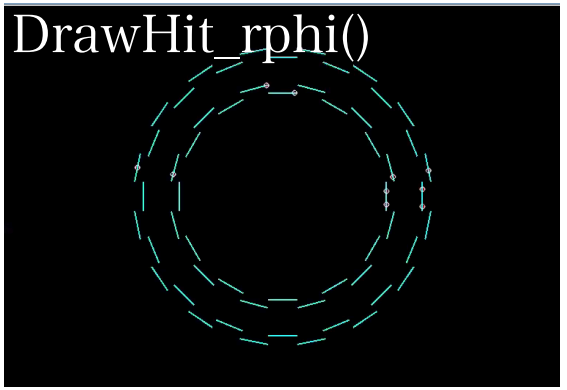
Event Display表示の仕方

1. root Loadfile.C でファイルを読み込む
2. anaTutorial->DrawHits() or DrawHit_rphi() でEvent Displayを表示

   DrawHits()は3D表示、DrawHit_rphi()は輪切りのような表示
3. 次のイベントを見るにはse->run(1) の後に2. のコマンドをもう一度入力

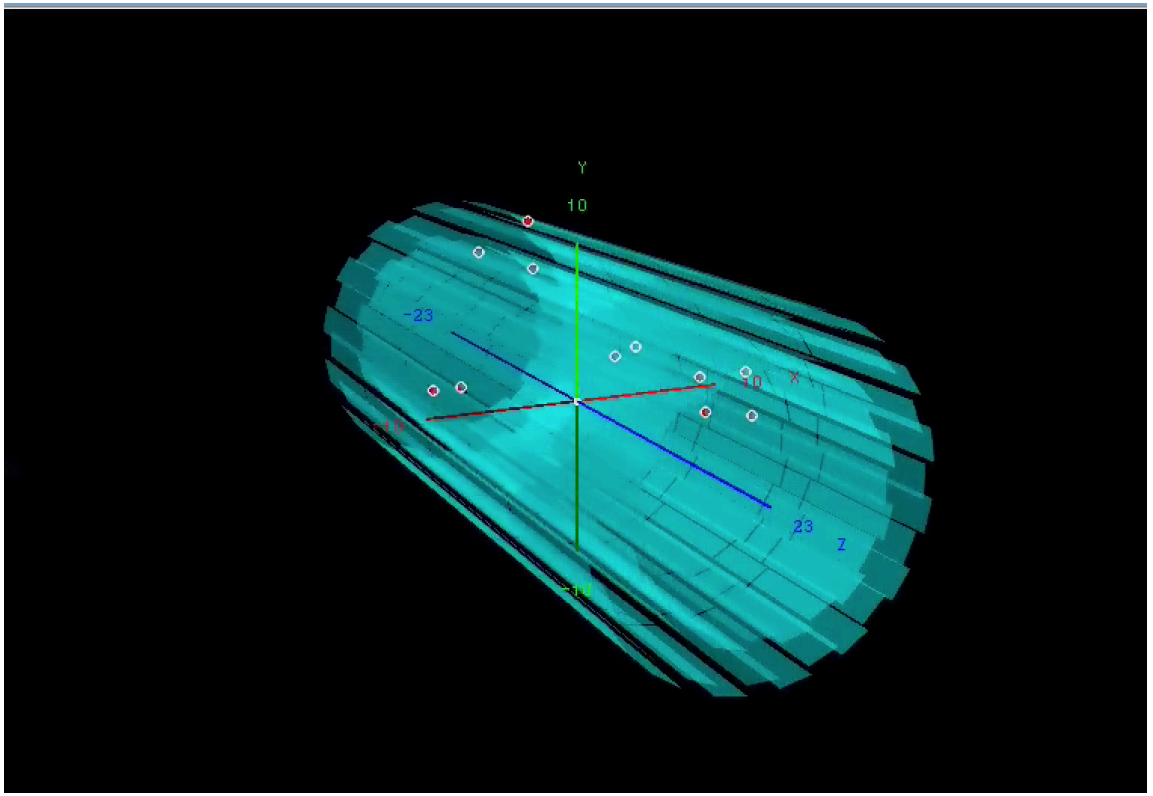
※前のイベントには戻れない





pp衝突のEvent Display

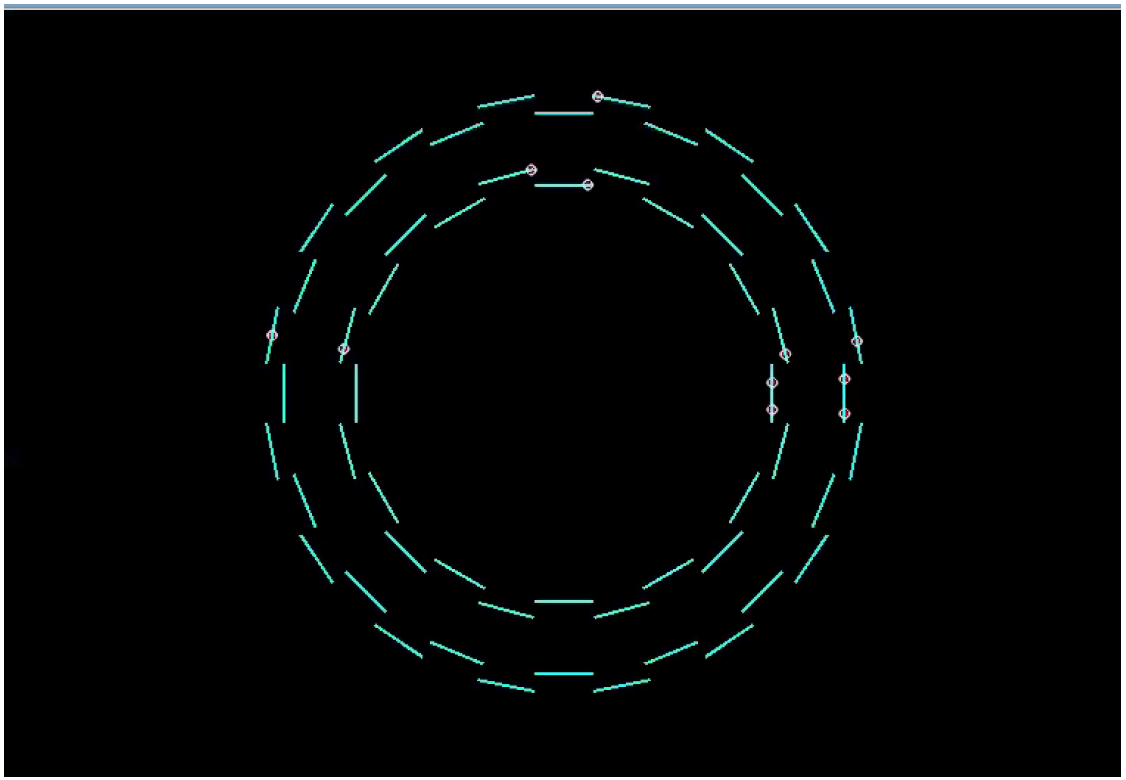
3D表示



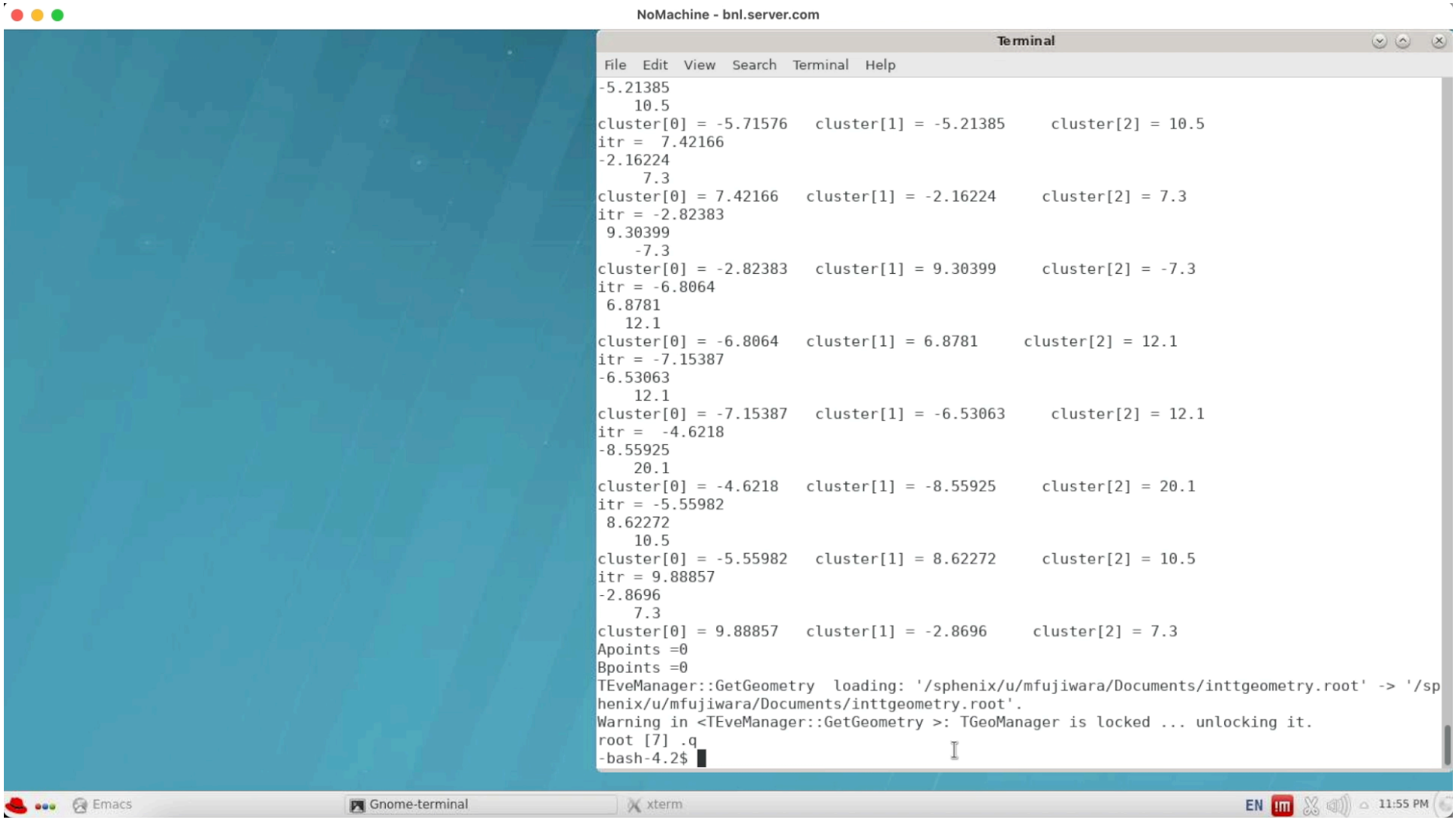
- 水色がINTT
- 赤の点がヒット位置

pp衝突のEvent Display

R-φプロジェクション



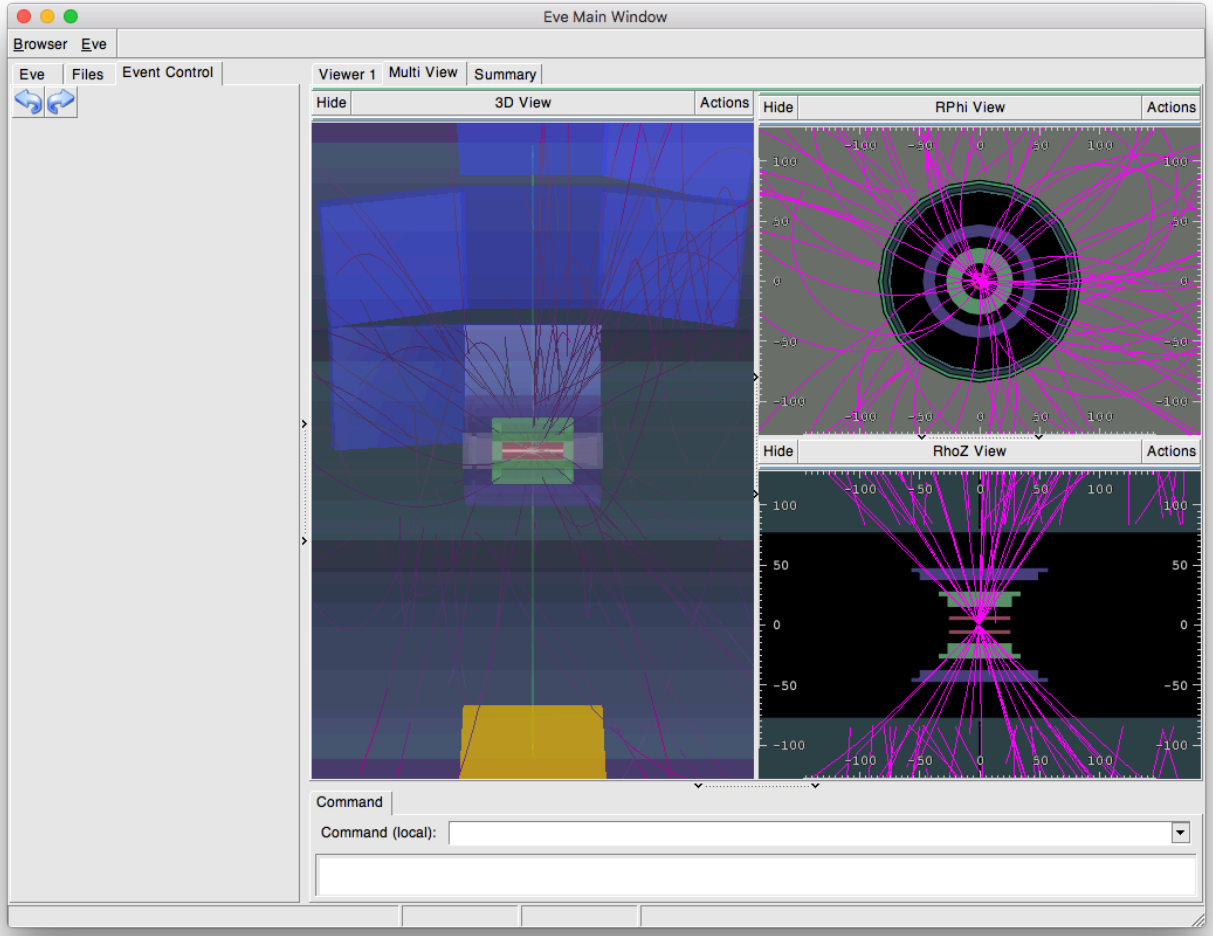
pp衝突のEvent Display



4. 今後の課題とまとめ

今後の課題

- Au+Au 衝突のデータの可視化
- ρ -z プロジェクションの実装
- Tracking の実装



まとめ

- INTT の情報を可視化し、確認するためのツールであるEvent Display を開発した
- 今後も ρ -z プロジェクション、 Tracking など必要な機能を随時実装予定である

Back Up

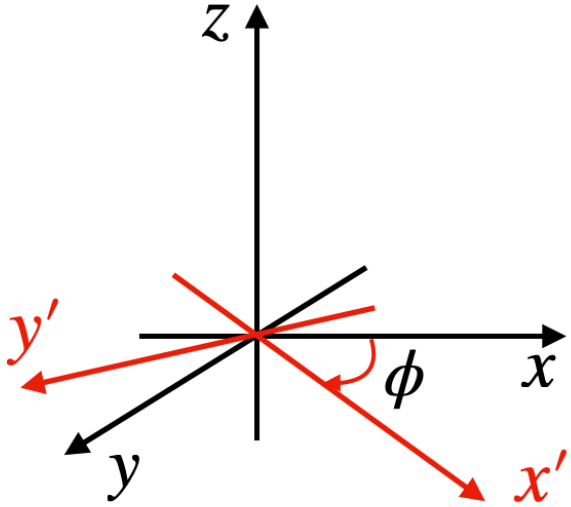
Event Display 開発

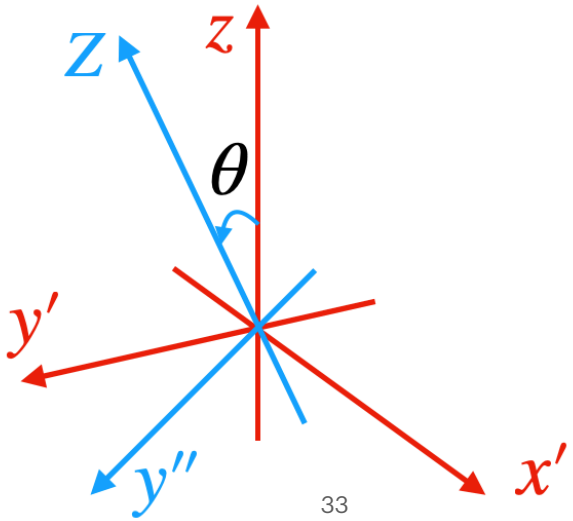
動作環境

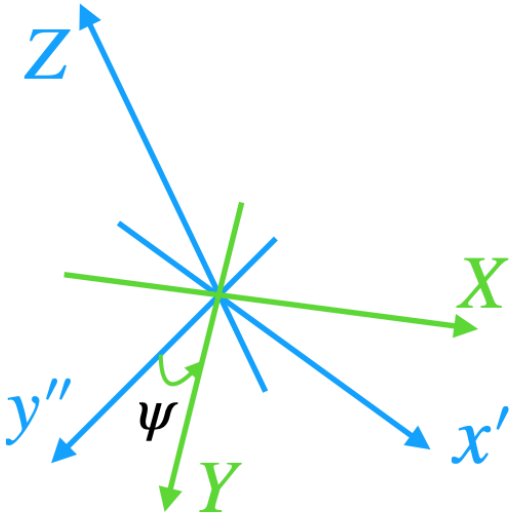
- ROOT 6.24/06
- g++ (GCC) 8.3.0
- リモートデスクトップ用ソフト NoMachine を通して実行可能

オイラー角

- 最初にz軸を中心に角度ϕ 回転、2番目にx軸を中心に角度θ 回転、最後に回転後のz軸を中心に角度ψ 回転させる。
- TGeoRotation では、単位は度

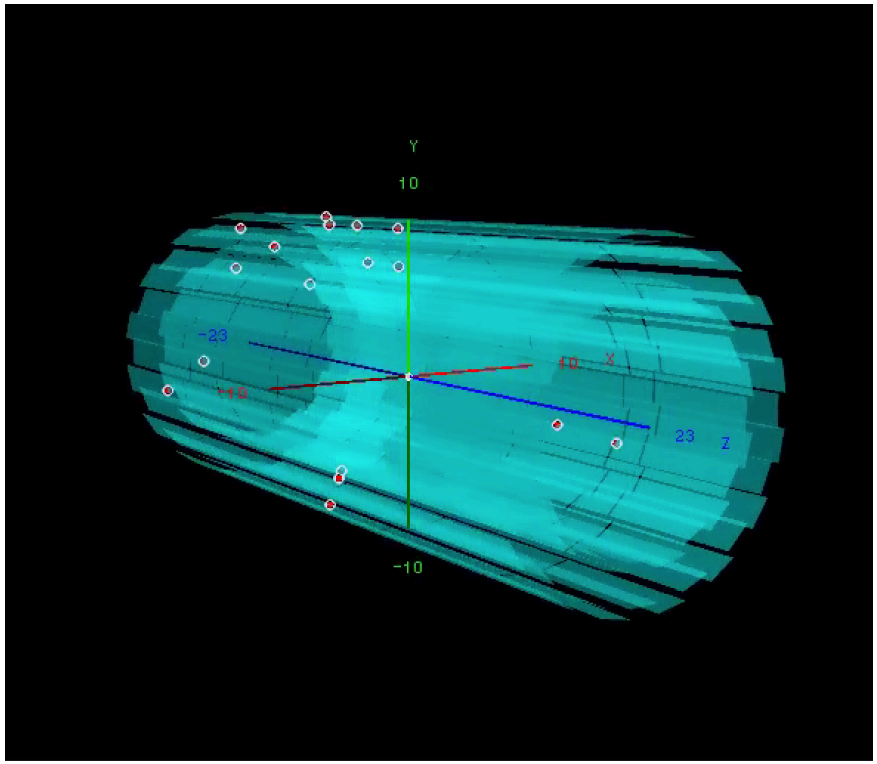


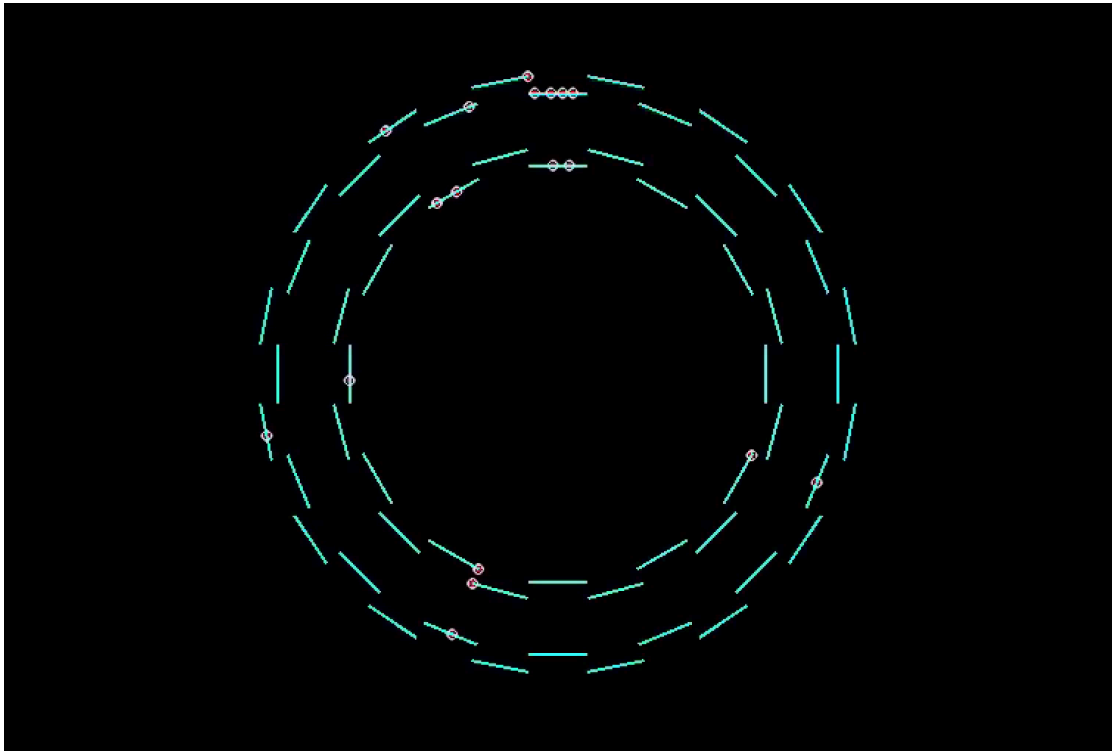




pp衝突のEvent Display

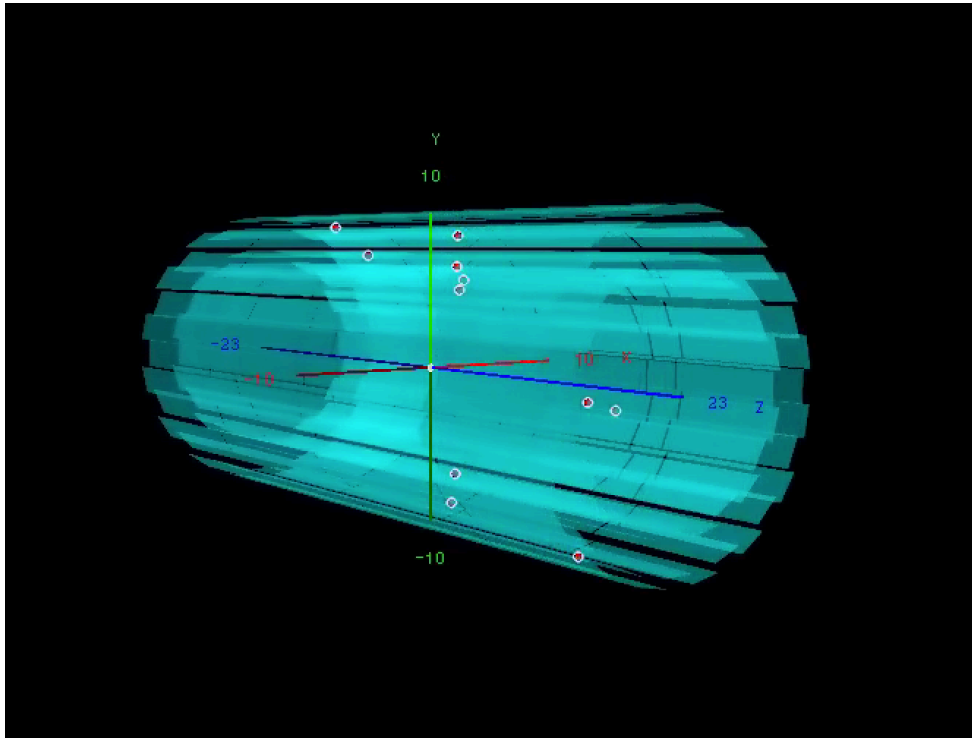
Event2

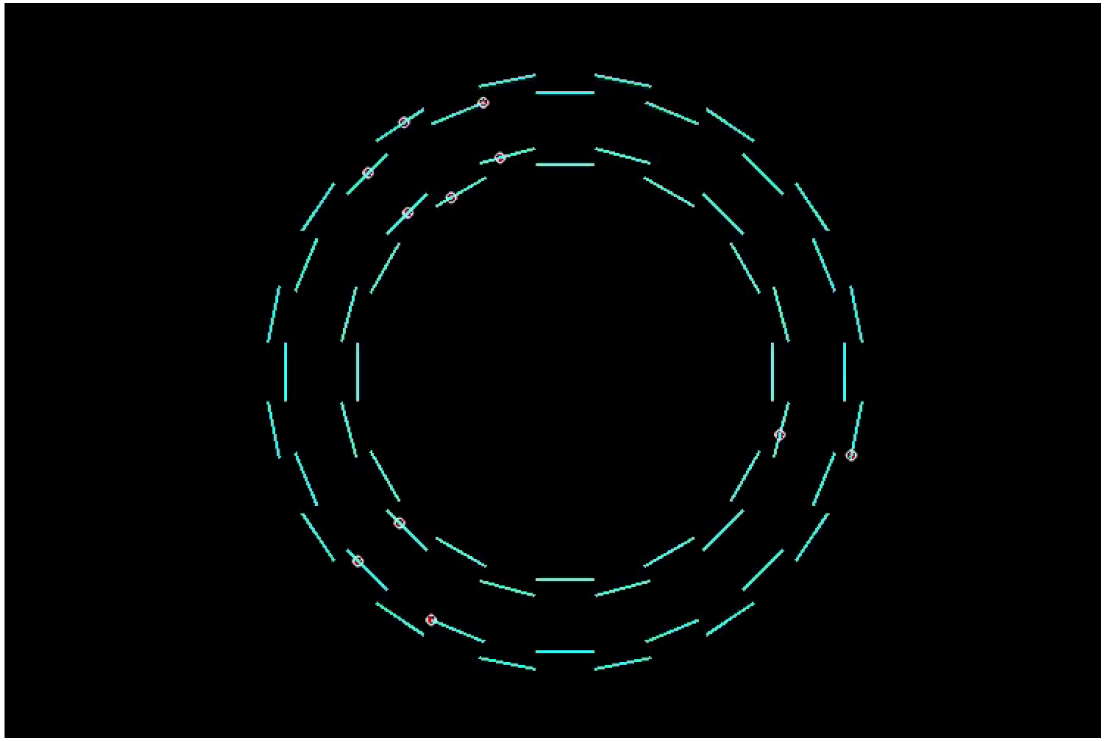




pp衝突のEvent Display

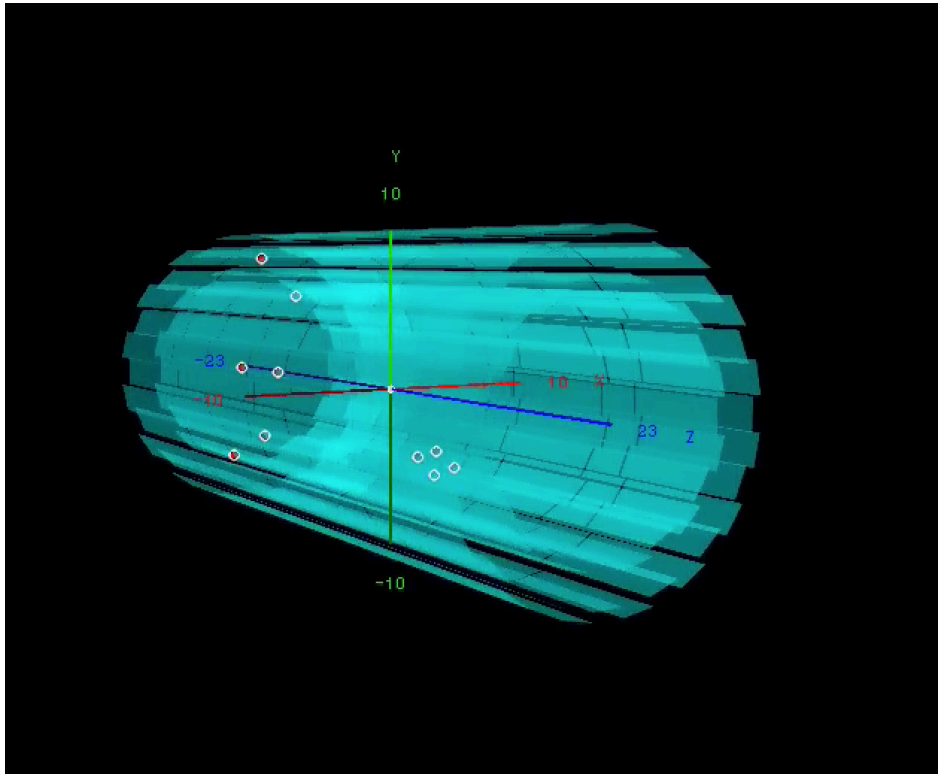
Event3

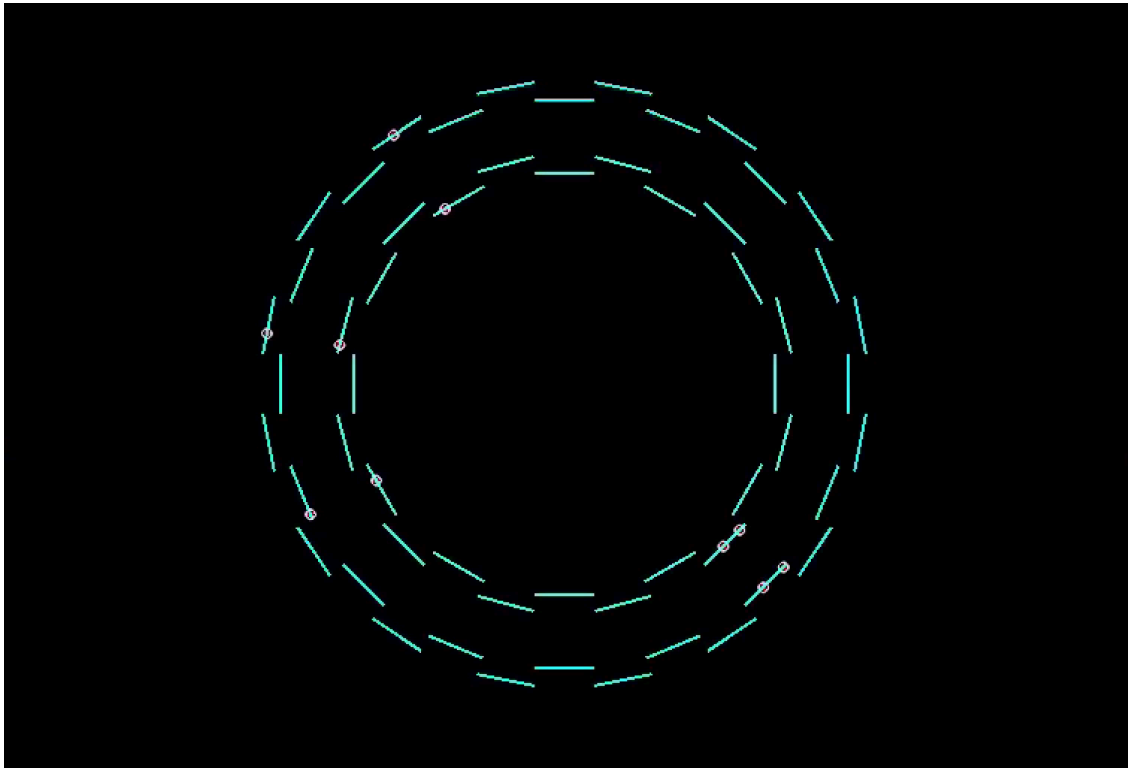




pp衝突のEvent Display

Event4





データの読み込み

```
43
44  void Loadfile(const char*inputfile="G4sPHENIX_10event.root"){
45    se = Fun4AllServer::instance();
46    se->Verbosity(0);
47    const string &outputFile = "G4sPHENIX";
48    string outputroot = outputFile;
49    const int nEvents = 1;
50
51    Fun4AllInputManager*in = new Fun4AllDstInputManager("DSTin");
52    in->fileopen(inputfile);
53    se->registerInputManager(in);
54
55
56    Enable::MICROMEGAS = true;
57
58    TrackingInit();
59    Tracking_Reco();
60
61    cout<<"before Anatutorial"<<endl;
62    anaTutorial = new AnaTutorial("anaTutorial", outputroot + "_anaTutorial.
      root");
```

↑データファイルの名前

ファイルデータ読み込み

```
63
64    anaTutorial->setMinJetPt(10.);
65    cout<<"setMinJetPt"<<endl;
66    anaTutorial->Verbosity(0);
67    cout<<"Verbosity(0)"<<endl;
68    anaTutorial->analyzeTracks(true);
69    cout<<"analyzeTracks(true)"<<endl;
70    anaTutorial->analyzeClusters(true);
71    cout<<"analyzeClusters(true)"<<endl;
72    anaTutorial->analyzeJets(true);
73    cout<<"analyzeJets"<<endl;
74    anaTutorial->analyzeTruth(false);
75    cout<<"analyzeTruth(false)"<<endl;
76    se->registerSubsystem(anaTutorial);
77
78    cout<<"after Anatutorial"<<endl;
79
80    se->run(nEvents);
81    return 0;
82  }
```

イベントディスプレイ描画

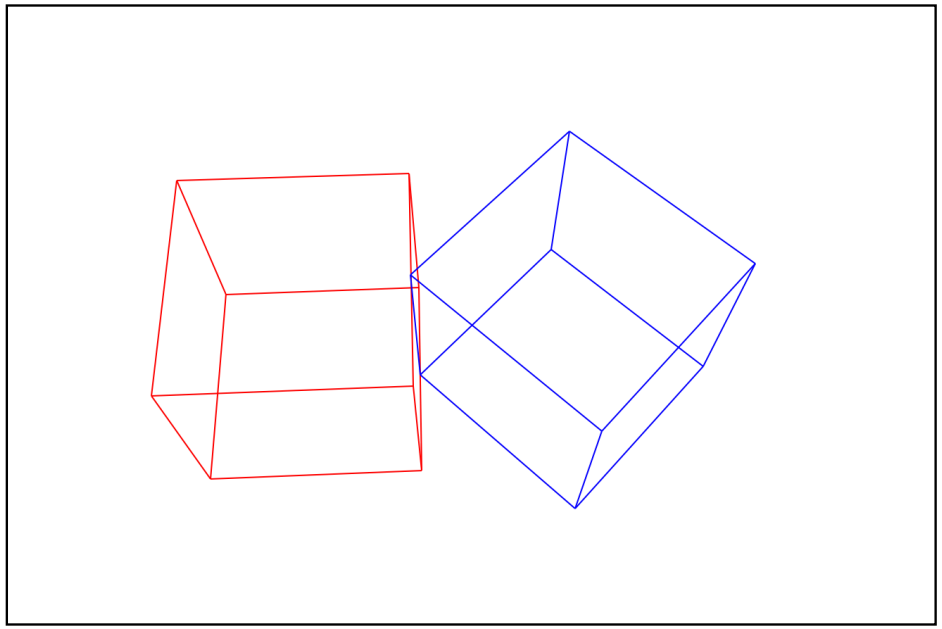
マクロにシミュレーション

データを渡す

TGeoManager

Example

```
void makebox()
{
    TGeoManager *gGeoManager = new TGeoManager("boxes", "boxes");
    TGeoVolume *top = gGeoManager->MakeBox("Top", NULL, 100., 100., 100.);
    gGeoManager->SetTopVolume(top);
    top->SetVisibility(kFALSE);

    TGeoVolume * boxred = gGeoManager->MakeBox("box",NULL,1.,1.,1.);
    TGeoVolume *boxblue = gGeoManager->MakeBox("box", NULL, 1., 1., 1.);

    boxred->SetLineColor(kRed);
    boxblue->SetLineColor(kBlue);

    TGeoRotation *rot = new TGeoRotation("rot", 45., 0., 0.);
    TGeoCombiTrans *box0 = new TGeoCombiTrans(1.0+sqrt(2), 0., 0., rot);

    top->AddNode(boxred, 1, 0);
    top->AddNode(boxblue, 2, box0);

    gGeoManager->CloseGeometry();
    top->Draw();
}
```



ヒット位置の再構成

```
1127  std::vector<Acts::Vector3> AnaTutorial :: writeInttClusters(PHCompositeNode
      *topNode,
1128                                TrkrClusterContainer *m_clusterMap,
1129                                ActsGeometry *m_tGeometry)
1130  {
1131    std::vector<TrkrDefs::cluskey> matchedClusters;
1132    std::vector<Acts::Vector3> clusters;
1133    for (unsigned int inttlayer = 0; inttlayer < m_nInttLayers; inttlayer++)
1134      {
1135   for( const auto& hitsetkey : m_clusterMap->getHitSetKeys
      (TrkrDefs::TrkrId::inttId, inttlayer+3))
1136    {
1137      //cout <<"hitsetkey="<<hitsetkey<<endl;
1138      auto range = m_clusterMap->getClusters(hitsetkey);
1139
1140
1141      for (auto clusIter=range.first; clusIter!= range.second; ++clusIter)
1142        {
1143    const auto cluskey = clusIter->first;
1144    cout <<"cluskey="<<cluskey<<endl;
1145    const auto cluster = clusIter->second;
1146
1147    cout<<"X = "<<cluster->getLocalX()<<", Y = "<<cluster->getLocalY()<<endl;
1148
1149    const auto globalPos = m_tGeometry->getGlobalPosition(cluskey, cluster);
1150    cout<<"gX = "<<globalPos(0)<<", gY = "<<globalPos(1)<<" , gZ = "<<globalPos
      (2)<<endl;
1151
1152    clusters.push_back(globalPos);
1153          }
1154    }
```

- 鳴ったch→ヒット位置に変換

INTT Event Display 作成の流れ

1. シミュレーション
2. ヒット位置を再構成、描画

---

3. INTTのGeometry を作成する
4. シミュレーションデータをファイルに書き出し、そのファイルを読み込んでヒット位置を描画することで、作ったソフトが実験データでも使えるかを確認

Viewer機能

- r-ϕ プロジェクションの実装

画像

- QGP http://alice-j.org/クォーク・グルーオンプラズマ-qgp/
- RICH https://www.bnl.gov/newsroom/news.php?a=119262
- sPHENIX 検出器 https://www.bnl.gov/rhic/sphenix.php
- INTT ladder https://webhepl.cc.nara-wu.ac.jp/old_HP/thesis/4kaisei/2020/nishimori_b4thesis.pdf
- INTT barrel https://wiki.sphenix.bnl.gov/index.php/File:20220504_Geant4_ladder_ID.pdf
- Event Display https://root.cern/doc/v610/group__TEve.html
- Rho-z https://root.cern/doc/v610/alice__esd_8C.html