TOF Simulation Meeting

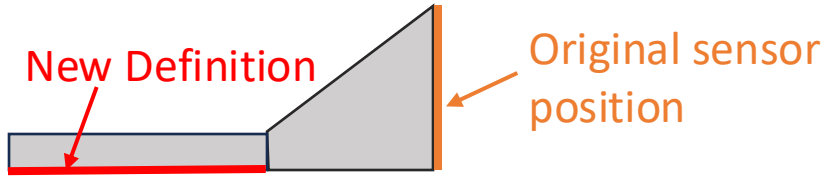
2024/10/08 (Tue)

Shunichiro Muraoka

Hiroshima University

Simulation

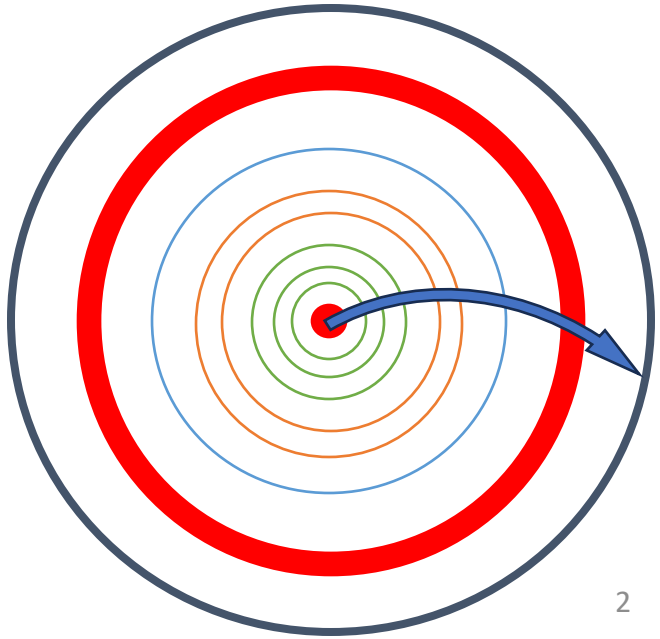
1. Define the surface of hpDIRC using Geant4 (hpDIRC surface)
   - Because no particles are detected on the surface of the hpDIRC
   - To obtain MC information of particles at hpDIRC surface



2. Calculate angle of incident particles from hit information (momentum) to detector
   - MC info(mc.theta)： Use MC information of the detector defined in 1.
   - Track info(track.theta)： Get values from propagation of reconstructed tracks from tracker

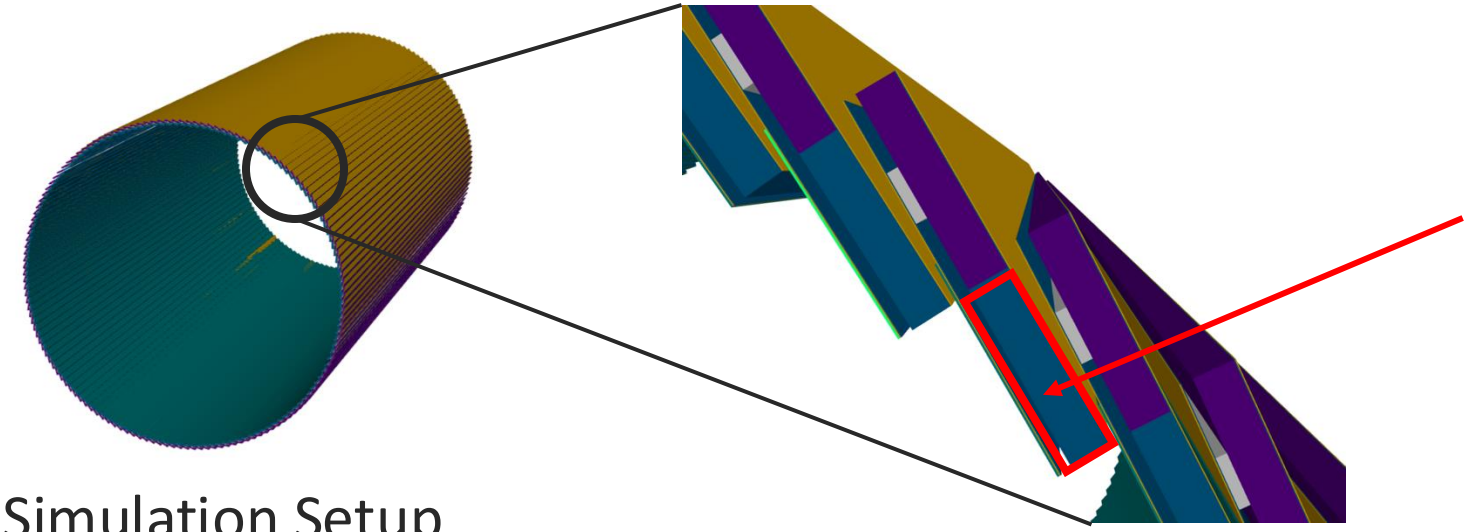
Tracker Detectors (Barrel)
- Silicon Vertex Tracker × 3
- Silicon Barrel Tracker × 2
- Inner MPGD (Micro Pattern Gas Detector)
- Barrel TOF
- Outer MPGD

Kalman filter to reconstruct tracks
Using track parameter information on hpDIRC surface (r = 75.5 cm)



Simulation

3. Change the BTOF material and perform the same simulation



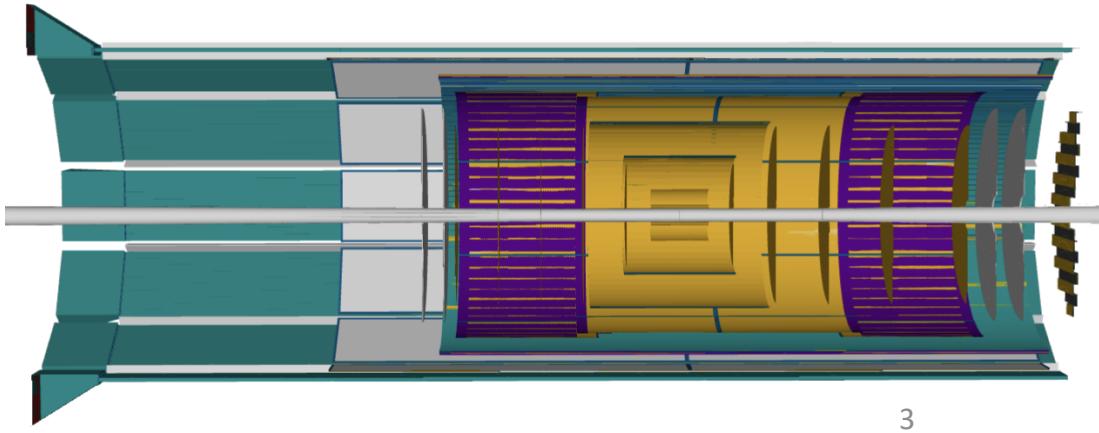
Carbon foam : 0.09 g/cm^3, 5.8mm
~ 0.7 % (X/X0)

↓Change material

Aluminum : 2.65 g/cm^3
~ 7.1 % (X/X0)

■Simulation Setup

- EICrecon : Particle collision reconstruction using phythia8, Geant4, and track reconstruction algorithms using ePIC detector structures are available.
- Single particle : π^-
- Momentum : 1, 2, 4, 6, 8, 10 [GeV/c]
- Direction : $0^\circ \leq \phi \leq 360^\circ$, $92^\circ \leq \theta \leq 94^\circ \rightarrow \langle\eta\rangle = -0.05$



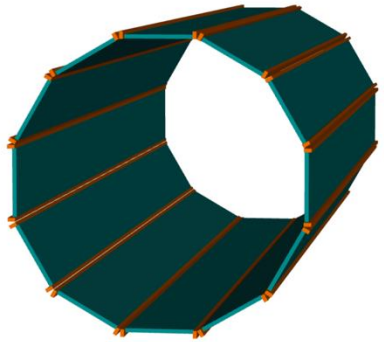
Results - Hit position on hpDIRC surface

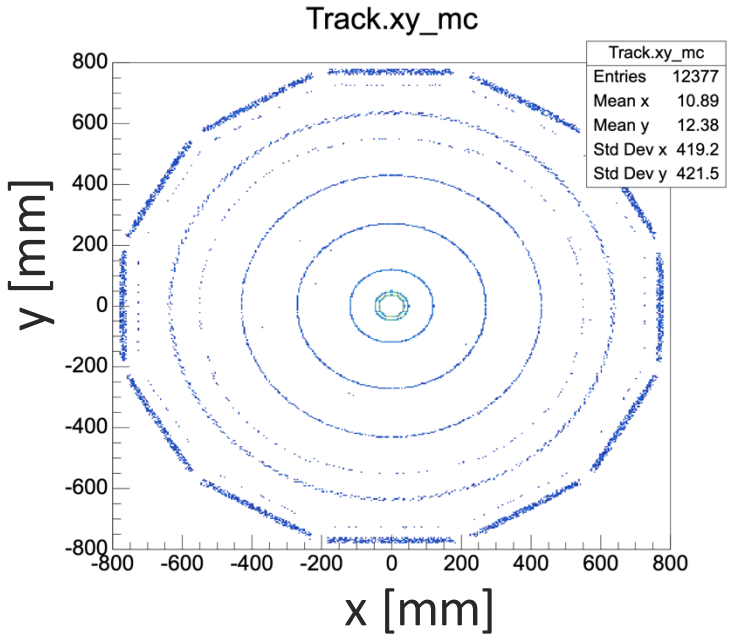
■Generates 1000event single particles (6 GeV/c, $0^\circ \leq \phi \leq 360^\circ, 92^\circ \leq \theta \leq 94^\circ$)

- Get particle information on hpDIRC surface defined by Geant4 (MC info)

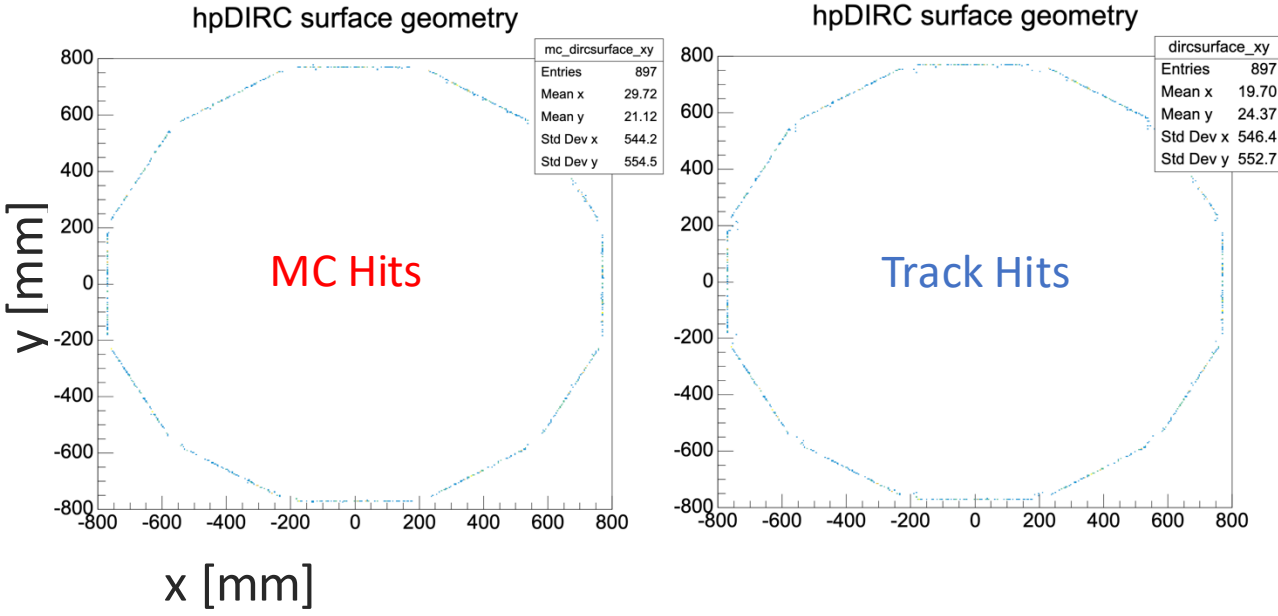
➢ Can get as well as MC information for epic's tracker detector

➢ The outermost layer is the hpDIRC surface defined by myself





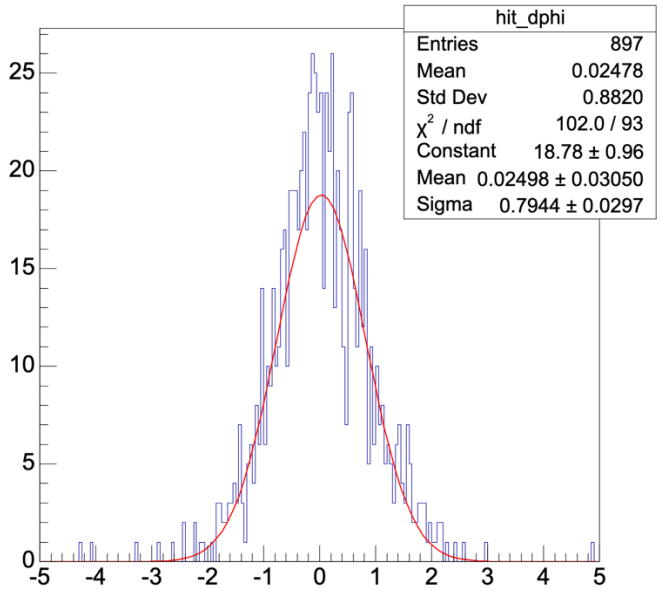
➢ MC information and particle information obtained by propagating tracks on hpDIRC surface



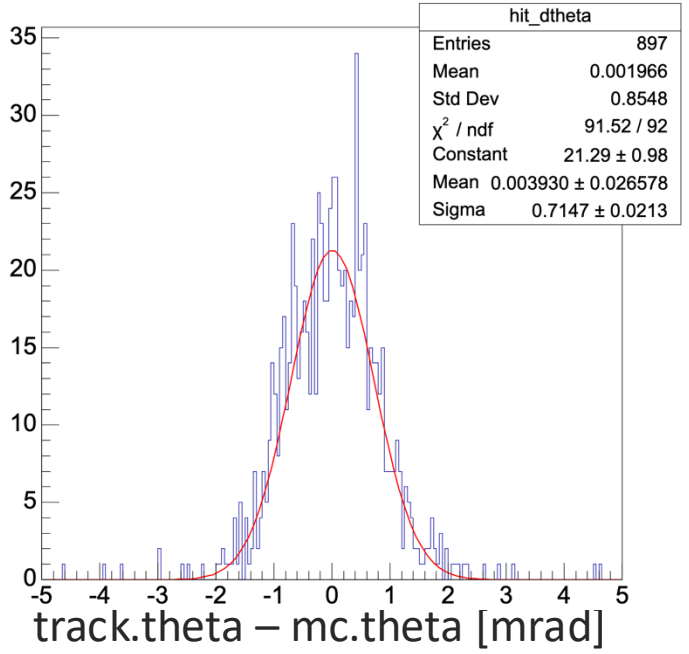
Results - Angular distribution on hpDIRC surface

- Angular resolution calculated from particle momentum in MC and track information
  - $\Delta\theta = \theta(track) - \theta(MC), \qquad \Delta\phi = \phi(track) - \phi(MC)$
  - $\theta = arctan2\left(\sqrt{{p_x}^2 + {p_y}^2}, p_z\right), \qquad \phi = arctan2(p_y, p_x)$
  - Angular resolution is given by Gaussian sigma
  - The higher the momentum, the better the angular resolution.

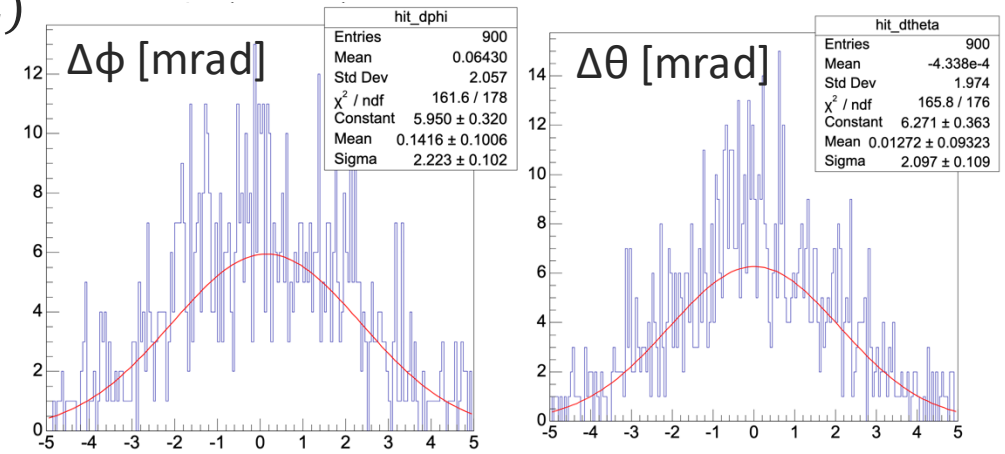
Δϕ [mrad] , 92 < θ < 94 , p = 6 GeV



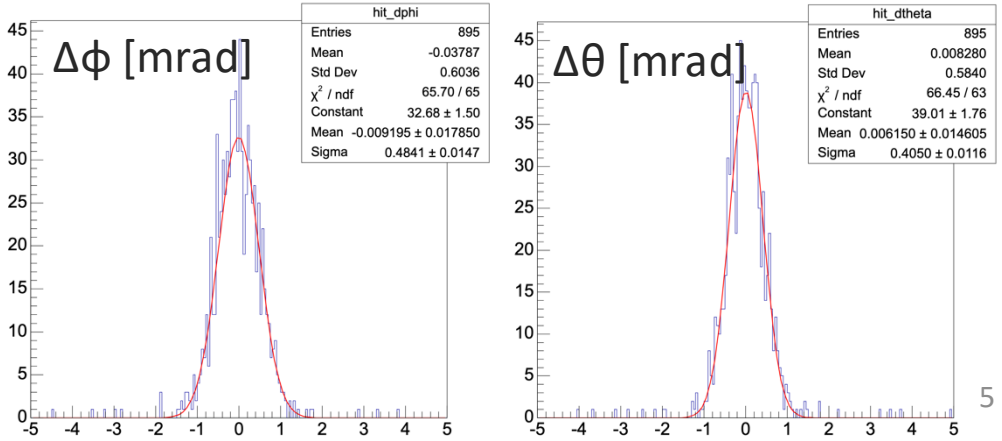
Δθ [mrad] , 92 < θ < 94 , p = 6 GeV



92 < θ < 94 , p = 2 GeV



92 < θ < 94 , p = 10 GeV



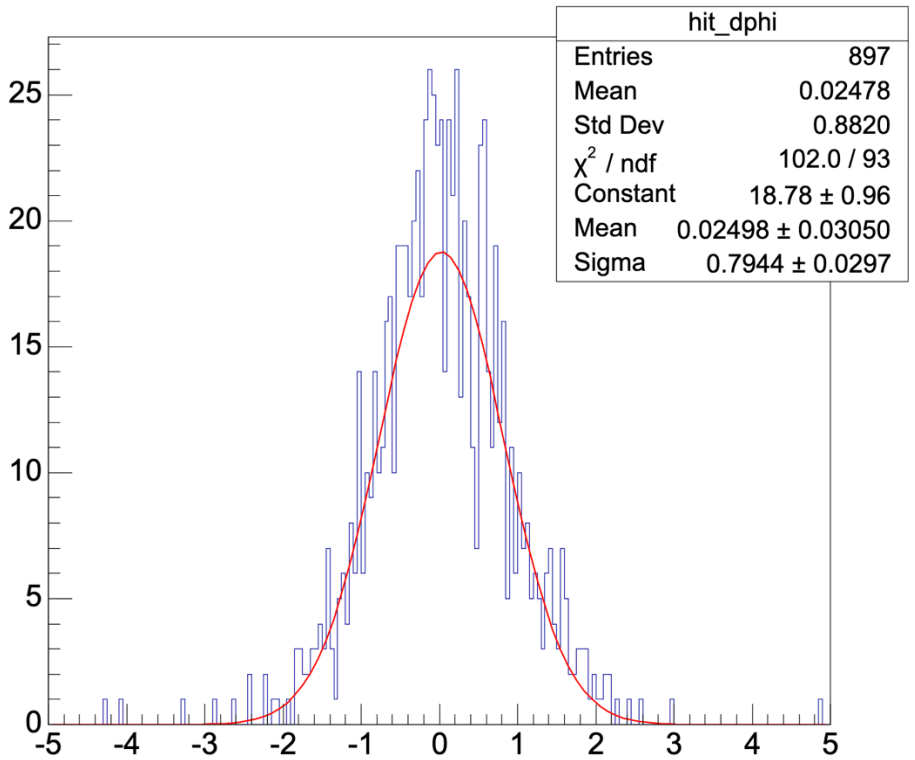
Angular distribution on hpDIRC surface

- Single particle : π^- (1000 events)
- Momentum : 1,2,4,6.8,10 [GeV/c]
- Direction : $0^\circ \leq \phi \leq 360^\circ$, $92^\circ \leq \theta \leq 94^\circ \rightarrow \langle\eta\rangle = -0.05$

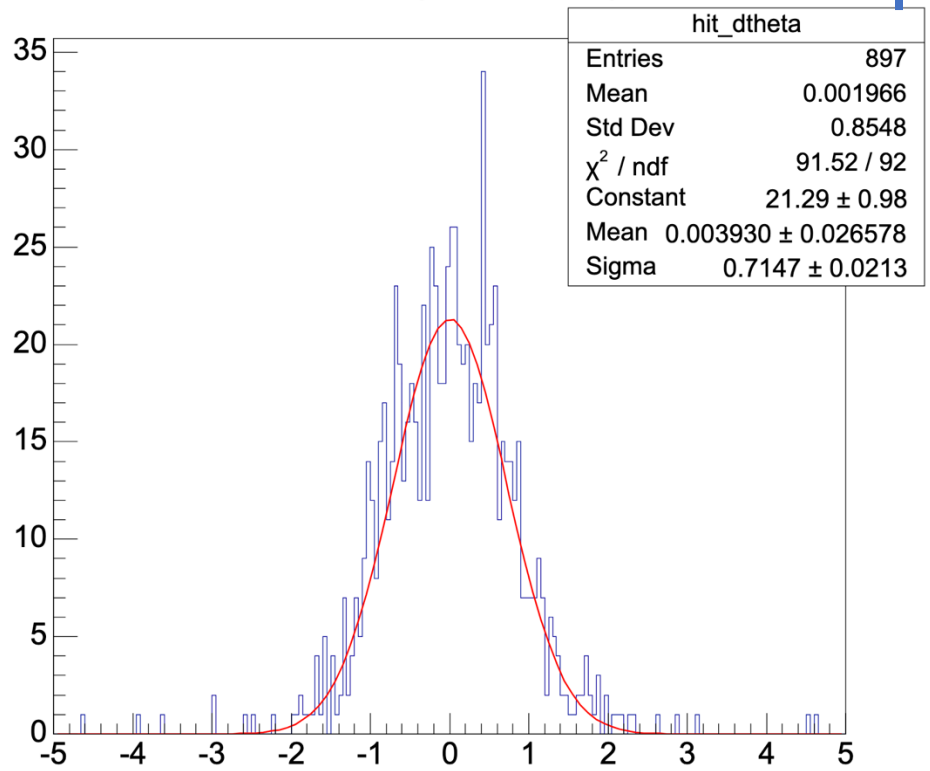
$\sigma = 0.7147 \pm 0.0213\ [mrad]$

Even if we use the carbon form (default setting), we cannot fulfill the request from hpDIRC
(delta_angle=0.5mrad@6GeV)

Δϕ [mrad] , 92 < θ < 94 , p = 6 GeV



Δθ [mrad] , 92 < θ < 94 , p = 6 GeV



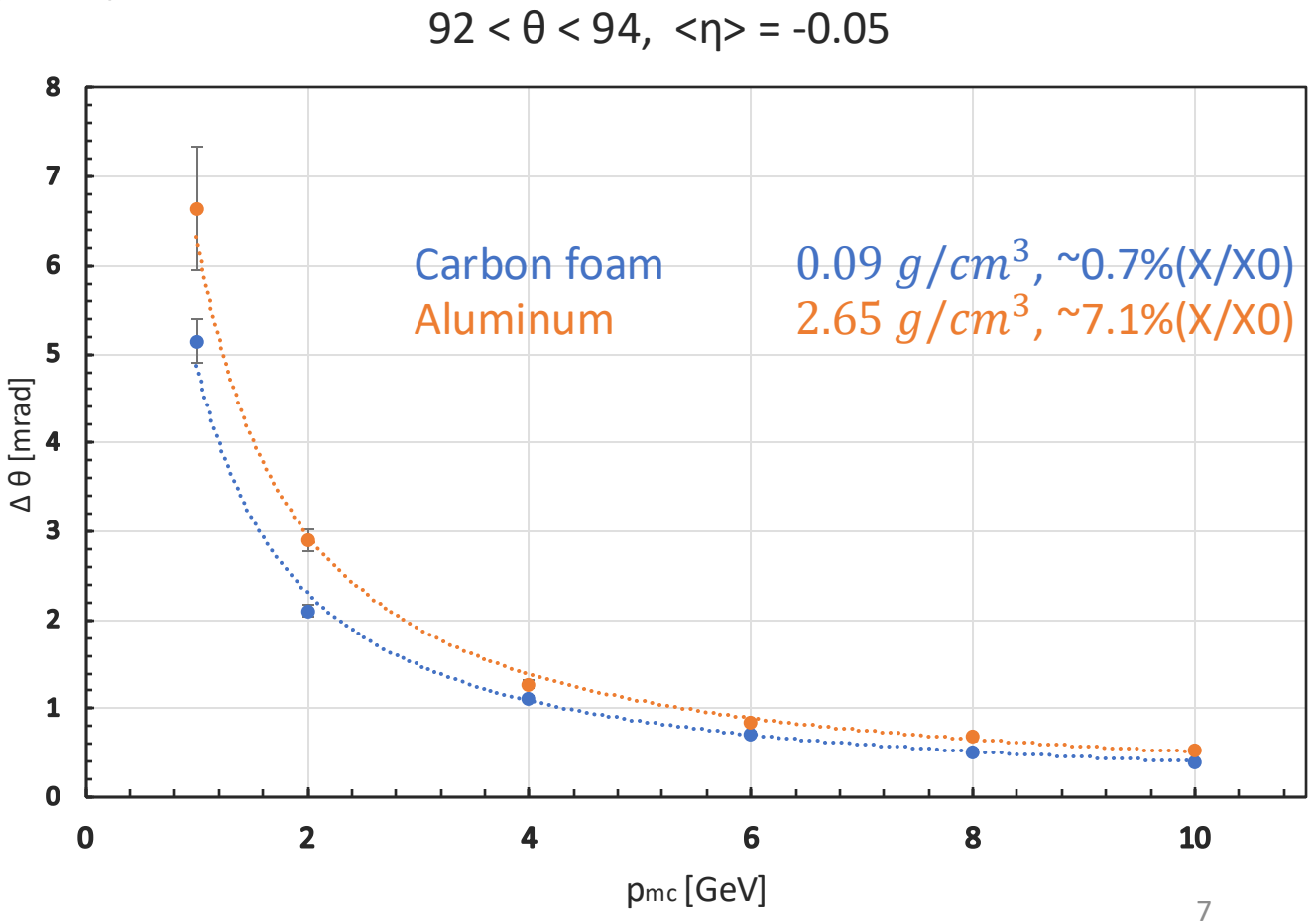
Results - Angular resolution on hpDIRC surface

■Results for some changes in particle momentum and BTOF material.

- Part of BTOF Carbon foam changed to Aluminum
- The value of Δθ increases

But…

- Current structure (Carbon foam) does not satisfy hpDIRC requirements.
- Please tell me your opinion on this simulation



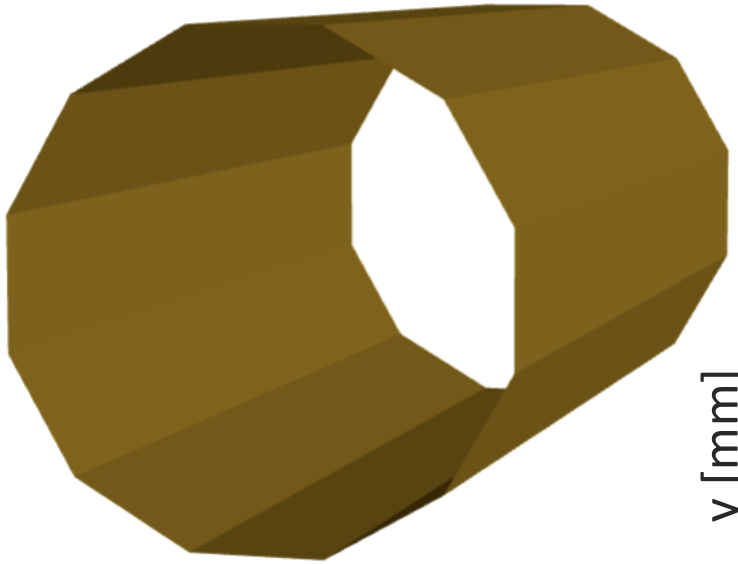
TOF General Meeting

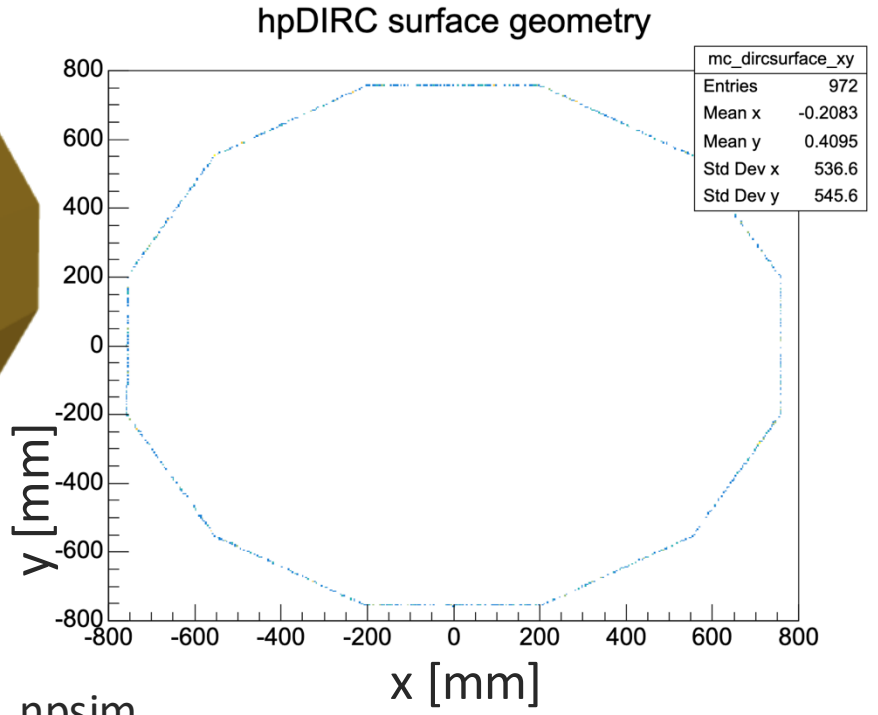
■Got some comments.

1. Define the exact location for this calculation of resolution (https://github.com/sourav-tarafdar/epic/blob/main/compact/tracking/dirc_ref.xml)
2. Use trackless of MPGD+BTOF?
3. Use Kalman filter inside-out
4. Resolution with and without BTOF material at all
5. Resolution with carbon fiber BTOF,
6. +1-2% material to BTOF and study its impact

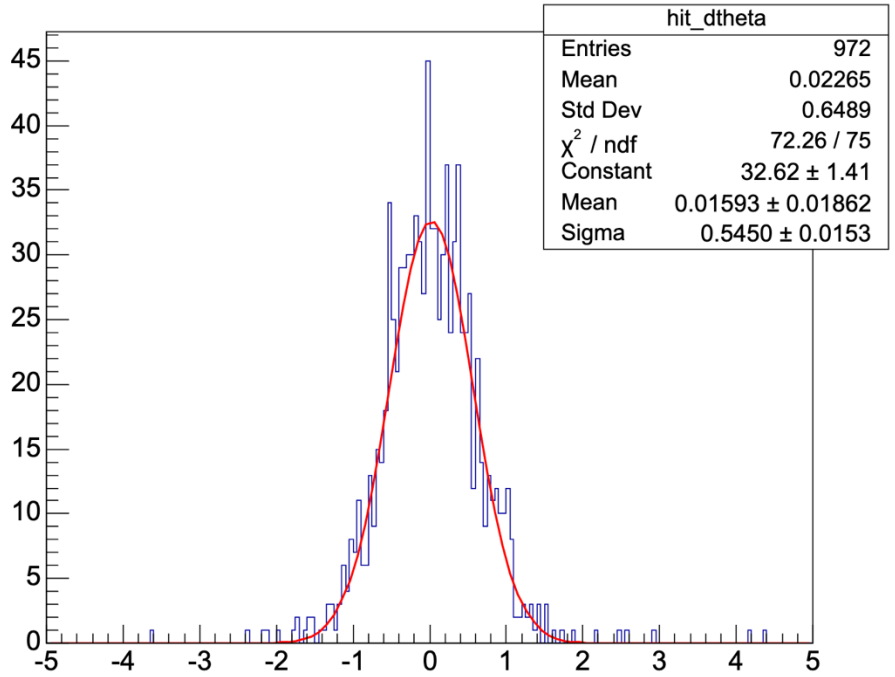
Define the exact location for this calculation of resolution

- More precise hpDIRC surface configuration (https://github.com/sourav-tarafdar/epic/blob/main/compact/tracking/dirc_ref.xml)





$\Delta\theta\ [mrad]$



Single particle : π^- , 1000events, npsim
Momentum : 6 [GeV/c]
Direction : $0^\circ \leq \phi \leq 360^\circ$,
$92^\circ \leq \theta \leq 94^\circ \rightarrow \langle\eta\rangle = -0.05$

$$\sigma = 0.5450 \pm 0.0153\ [mrad]$$

TOF General Meeting

■Got some comments.

1. Define the exact location for this calculation of resolution (https://github.com/sourav-tarafdar/epic/blob/main/compact/tracking/dirc_ref.xml)
2. Use trackless of MPGD+BTOF?
3. Use Kalman filter inside-out → I'm trying to figure out what I need to do.
4. Resolution with and without BTOF material at all
5. Resolution with carbon fiber BTOF,
6. +1-2% material to BTOF and study its impact

Back up

Data generation

Data Generation Settings

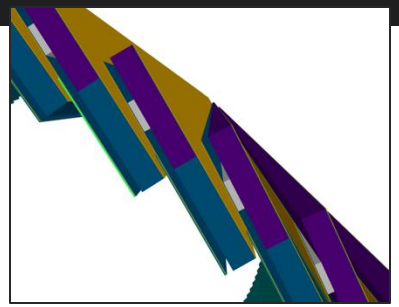
- EICrecon : Particle collision reconstruction using phythia8, Geant4, and track reconstruction algorithms using ePIC detector structures are available.
- ddsim
- Single particle : π^-(1000 events)
- Momentum : 1, 2, 4, 6, 8, 10 [GeV/c]
- Direction : $0^\circ \leq \phi \leq 360^\circ$,
  $92^\circ \leq \theta \leq 94^\circ \rightarrow \langle\eta\rangle = -0.05$

```
ddsim --compactFile=$DETECTOR_PATH/epic.xml
      -N=1000
      --random.seed 1
      --enableGun
      --gun.particle "pi-"
      --gun.momentumMin "10*GeV" --gun.momentumMax "10*GeV"
      --gun.thetaMin 92.0*deg --gun.thetaMax 94.0*deg
      --gun.distribution uniform
      --outputFile epic_fakedirc_pi_10GeV_1000event_ddsim.edm4hep.root
```

Detector settings

■BTOF material (epic/compact/tracking/tof_barrel.xml)

```
<module name="BarrelTOF_Module1" vis="TOFBarrelModuleVis">
  <module_component name="sensor" material="Silicon" sensitive="true" width="BarrelTOF_Sensor_width" length="BarrelTOF_length1" thickness="BarrelTOF_Sensor_thickness" vis="TOFSensorVis" >
    <position x="BarrelTOF_Sensor_position" y="0" z="0" />
  </module_component>
  <module_component name="hybridtop" material="Kapton" sensitive="false" width="BarrelTOF_Module_width" length="BarrelTOF_length1" thickness="BarrelTOF_Hybrid_thickness" vis="TOFHybridVis" >
    <position x="BarrelTOF_Service_position" y="0" z="0" />
  </module_component>
  <module_component name="cfskintop" material="CFRPMix2" sensitive="false" width="BarrelTOF_Module_width" length="BarrelTOF_length1" thickness="BarrelTOF_CFSkin_thickness" vis="TOFCSkinVis" >
    <position x="BarrelTOF_Service_position" y="0" z="0" />
  </module_component>
  <module_component name="coolingtube" material="Aluminum" sensitive="false" width="BarrelTOF_CoolingTube_width" length="BarrelTOF_length1" thickness="BarrelTOF_CoolingTube_thickness" vis="TOFCoolingTubeVis" >
    <position x="BarrelTOF_CoolingTube_position" y="0" z="0" />
  </module_component>
  <module_component name="coolant" material="NOVEC7200" sensitive="false" width="BarrelTOF_Coolant_width" length="BarrelTOF_length1" thickness="BarrelTOF_Coolant_thickness" vis="TOFCoolantVis" >
    <position x="BarrelTOF_Coolant_position" y="0" z="0" />
  </module_component>
  <module_component name="cfoam" material="Aluminum5083" sensitive="false" width="BarrelTOF_CFoam_width" length="BarrelTOF_length1" thickness="BarrelTOF_CFoam_thickness" vis="TOFCFoamVis" >
  <!-- <module_component name="cfoam" material="CarbonFoam" sensitive="false" width="BarrelTOF_CFoam_width" length="BarrelTOF_length1" thickness="BarrelTOF_CFoam_thickness" vis="TOFCFoamVis" > -->
    <position x="BarrelTOF_CFoam_position" y="0" z="0" />
  </module_component>
  <module_component name="choneycomb" material="CFRPMix" sensitive="false" width="BarrelTOF_CHoneycomb_width" length="BarrelTOF_length1" thickness="BarrelTOF_CHoneycomb_thickness" vis="TOFCHoneycombVis" >
    <position x="BarrelTOF_CHoneycomb_position" y="0" z="-1*BarrelTOF_CFoam_thickness" />
  </module_component>
  <module_component name="cfskinbottom" material="CFRPMix2" sensitive="false" width="BarrelTOF_Module_width" length="BarrelTOF_length1" thickness="BarrelTOF_CFSkin_thickness" vis="TOFCSkinVis" >
    <position x="BarrelTOF_Service_position" y="0" z="0" />
  </module_component>
  <module_component name="hybridbottom" material="Kapton" sensitive="false" width="BarrelTOF_Module_width" length="BarrelTOF_length1" thickness="BarrelTOF_Hybrid_thickness" vis="TOFHybridVis" >
    <position x="BarrelTOF_Service_position" y="0" z="0" />
  </module_component>
```



Changed material

Default material

Detector settings

- hpDIRC surface (Referring to https://github.com/eic/athena/compact/fake_dirc.xml)

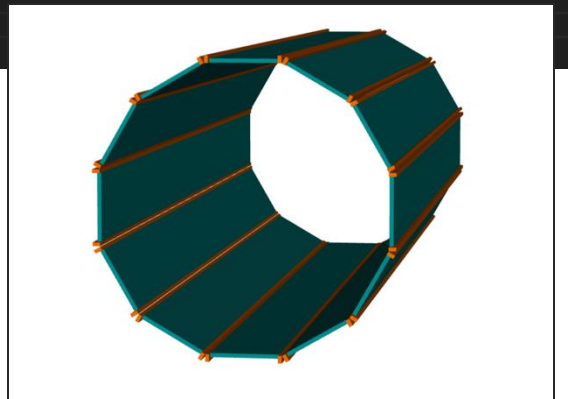
```xml
<lccdd>

  <define>
    <constant name="DIRCModule_count"          value="12" />
    <constant name="DIRCBar_thickness"         value="17*mm" />
    <constant name="DIRCBar_length"            value="DIRC_length-8*cm" />
    <constant name="DIRCSkinFront_thickness"   value="2.5*mm" />
    <constant name="DIRCSkinBack_thickness"    value="2.5*mm" />
    <constant name="DIRCFrame_width"           value="20*mm"/>
    <constant name="DIRCFrame_length"          value="DIRC_length"/>
    <constant name="DIRCFrame_thickness"       value="40*mm"/>
    <constant name="DIRCModule_rmax"           value="DIRC_rmin + DIRCBar_thickness + DIRCSkinFront_thickness + DIRCSkinBack_thickness"/>
    <constant name="DIRCModule_thickness"      value="DIRCModule_rmax - DIRC_rmin" />
    <constant name="DIRCFrame_rmax"            value="DIRC_rmin + DIRCFrame_thickness" />
    <constant name="DIRCModule_halfangle"      value="180*degree/DIRCModule_count" />
    <constant name="DIRCModule_width"          value="2*DIRC_rmin * tan(DIRCModule_halfangle)"/>
    <constant name="DIRCModule_rmin"           value="DIRC_rmin + 1*cm"/>
  </define>

  <detectors>
    <detector id="BarrelDIRC_ID" name="FakeDIRC" type="epic_FakeDIRC" readout="DIRCBarHits" vis="DIRCVis">
      <dimensions rmin="DIRC_rmin" rmax="DIRC_rmax" length="DIRC_length" />
      <position x="0" y="0" z="DIRC_offset" />
      <comment> Fake DIRC modules </comment>
      <module name="DIRCModule" vis="DIRCModuleVis">
        <module_component name="FrontSkin"
                material="CarbonFiber"
                sensitive="false"
                width="DIRCModule_width"
                thickness="DIRCSkinFront_thickness"
                vis="DIRCSupportVis"
                length="DIRCBar_length" />
        <module_component name="QuartzBar"
                material="Quartz"
                sensitive="true"
                width="DIRCModule_width"
                thickness="DIRCBar_thickness"
                vis="DIRCBarVis"
                length="DIRCBar_length" />
        <module_component name="BackSkin"
                material="CarbonFiber"
                sensitive="false"
                width="DIRCModule_width"
                thickness="DIRCSkinBack_thickness"
                vis="DIRCSupportVis"
                length="DIRCBar_length" />
        <frame material="StainlessSteel"
                width="DIRCFrame_width"
                length="DIRCFrame_length"
                vis="DIRCFrameVis"
                thickness="DIRCFrame_thickness" />
      </module>
      <comment> Fake DIRC layers </comment>
      <layer module="DIRCModule" id="1" vis="DIRCLayerVis">
        <barrel_envelope
          inner_r="DIRC_rmin"
          outer_r="DIRC_rmax"
          z_length="DIRC_length" />
        <rphi_layout
          phi_tilt="0"
          nphi="DIRCModule_count"
          phi0="0"
          rc="0.5*(DIRCModule_rmin+DIRCModule_rmax)"
          dr="0" />
        <z_layout
          dr="0.0*mm"
          z0="0.0*mm"
          nz="1" />
      </layer>
    </detector>
  </detectors>

  <readouts>
    <readout name="DIRCBarHits">
      <segmentation type="CartesianGridXY" grid_size_x="3.0*mm" grid_size_y="3.0*mm" />
      <id>system:8,layer:4,module:8,section:4,x:32:-16,y:-16</id>
    </readout>
  </readouts>

</lccdd>
```



About Analysis Code

- Track information (Referring to https://github.com/eic/EICrecon/src/algorithms/tracking/TrackPropagation.cc)

```
const auto &dirc = *static_cast<const edm4hep::SimTrackerHitCollection*>(event->GetCollectionBase("DIRCBarHits"));
for(auto hit: dirc) {
    auto& mcinfo = hit.getMCParticle();
    auto& mcpos = hit.getPosition();
    auto& mcmom = hit.getMomentum();

    if (mcinfo.getGeneratorStatus() != 1) continue;
    if (i != 0 ) continue;

    auto mcr = sqrt(mcpos.x*mcpos.x + mcpos.y*mcpos.y + mcpos.z*mcpos.z);
    auto mcrt = sqrt(mcpos.x*mcpos.x + mc
    auto mcp = sqrt(mcmom.x * mcmom.x + m                    om.z * mcmom.z);
    auto mcpt = sqrt(mcmom.x * mcmom.x + mcmom.y * mcmom.y);
    auto mceta = -log(tan(atan2(sqrt(mcmom.x*mcmom.x+mcmom.y*mcmom.y),mcmom.z)/2.));
    auto mctheta = atan2(sqrt(mcmom.x*mcmom.x + mcmom.y*mcmom.y),mcmom.z);
    auto mcphi = atan2(mcmom.y, mcmom.x);
    // auto mceta = -log(tan(atan2(sqrt(mcpos.x*mcpos.x+mcpos.y*mcpos.y),mcpos.z)/2.));
    // auto mctheta = atan2(sqrt(mcpos.x*mcpos.x + mcpos.y*mcpos.y),mcpos.z);
    // auto mcphi = atan2(mcpos.y, mcpos.x);

    std::unique_ptr<edm4eic::TrackPoint> hpdirc_point;
    const auto hpdircR = mcrt;
    const auto hpdircMinZ = -2526.0;
    const auto hpdircMaxZ = 1850.0;
    auto hpdircBounds = std::make_shared<Acts::CylinderBounds>(hpdircR, (hpdircMaxZ-hpdircMinZ)/2);
    auto hpdirc_t = Acts::Translation3(Acts::Vector3(0, 0, (hpdircMaxZ+hpdircMinZ)/2));
    auto hpdirc_trf = Acts::Transform3(hpdirc_t);
    hpdirc_surface = Acts::Surface::makeShared<Acts::CylinderSurface>(hpdirc_trf, hpdircBounds);
    try {hpdirc_point =  m_propagation_algo.propagate(edm4eic::Track{}, trajectory, hpdirc_surface);}
    catch(std::exception &e) {throw JException(e.what());}
    if(!hpdirc_point) {
    fprintf (outputfileTrack,"could not propagate!\n");
    continue;
    }
        auto pos = hpdirc_point->position;
        auto mom = hpdirc_point->momentum;
        auto length = hpdirc_point->pathlength;
        auto theta = hpdirc_point->theta;
        auto phi = hpdirc_point->phi;
        auto eta = -log(tan(atan2(sqrt(mom.x*mom.x + mom.y*mom.y),mom.z)/2.));
        auto r = sqrt(pos.x*pos.x + pos.y*pos.y + pos.z*pos.z);
        auto rt = sqrt(pos.x*pos.x + pos.y*pos.y);
        Track_xy -> Fill(pos.x, pos.y);
        Hitpoints_dircsurface_xy -> Fill(pos.x, pos.y);
        Hitpoints_dircsurface_z -> Fill(pos.z);
        if(pos.x > 0)Track_rz -> Fill(pos.z, rt);
               else Track_rz -> Fill(pos.z, -rt);
        hit_propagate_eta -> Fill(eta);
        hit_propagate_phi -> Fill(phi);
        hit_propagate_theta -> Fill(theta);
        fprintf (outputfileTrack,"Reconstructed DIRC hit  x:%f, y:%f, z:%f, r:%f, rt:%f, theta:%f, phi:%f\n", pos.x, pos.y, pos.z, length, rt, theta, phi);
        fprintf (outputfileTrack, "MC DIRCsurface x:%f, y:%f, z:%f, r:%f, rt:%f, theta:%f, phi:%f, p:%f\n", mcpos.x, mcpos.y, mcpos.z, mcr, mcrt, mctheta, mcphi, mcp);
```

<error-type> &mcmom

```
        auto reso_eta = (eta - mceta)/mceta * 100;
        auto reso_phi = (phi - mcphi)/mcphi * 100;
        auto reso_theta = (theta - mctheta)/mctheta * 100;
        auto deta = (eta - mceta)* 1000;
        auto dphi = (phi - mcphi)* 1000;
        auto dtheta = (theta - mctheta)* 1000;

        hit_true_eta -> Fill(mceta);
        hit_true_phi -> Fill(mcphi);
        hit_true_theta -> Fill(mctheta);
        hit_resolution_eta -> Fill(reso_eta);
        hit_resolution_phi -> Fill(reso_phi);
        hit_resolution_theta -> Fill(reso_theta);
        hit_deta -> Fill(deta);
        hit_dphi -> Fill(dphi);
        hit_dtheta -> Fill(dtheta);

        Hitpoints_dircsurface_mc_xy -> Fill(mcpos.x, mcpos.y);
        Hitpoints_dircsurface_mc_z -> Fill(mcpos.z);

        fprintf (outputfileMC, "DIRCsurface  r:%f, rt:%f, p:%f, eta:%f, phi:%f, theta:%f\n", mcr, mcrt, mcp, mceta, mcphi, mctheta);
        i++;
    }
```

Get the parameters of the reconstructed track at any r.

(Trajectory
→ CentralCKFActsTrajectories)

About Analysis Code

■MC information (edm4hep::SimTrackerHit)

```
304   std::vector<std::string> sim_data_names = {
305       "VertexBarrelHits",         // Vertex
306       "SiBarrelHits",             // Barrel Tracker
307       "MPGDBarrelHits",           // MPGD
308       "TOFBarrelHits",             // Barrel TOF
309       "OuterMPGDBarrelHits",      // MPGD DIRC
310       "TrackerEndcapHits",        // End Cap tracker
311       "TOFEndcapHits",            // End Cap TOF
312       "DIRCBarHits",
313   };
314   for(int i = 0; i < sim_data_names.size(); i++ ) {
315       auto data_name = sim_data_names[i];
316       const auto &hits = *static_cast<const edm4hep::SimTrackerHitCollection*>(event->GetCollectionBase(data_name));
317       const char* Data_name = data_name.c_str();  //fprintfに出力する際に文字化けを防ぐための措置
318       for(auto hit: hits) {
319           auto& mcinfo = hit.getMCParticle();
320           auto& pos = hit.getPosition();
321           auto& mom = hit.getMomentum();
322           auto r = sqrt(pos.x*pos.x + pos.y*pos.y + pos.z*pos.z);
323           auto rt = sqrt(pos.x*pos.x + pos.y*pos.y);
324           auto p = sqrt(mom.x * mom.x + mom.y * mom.y + mom.z * mom.z);
325           auto pt = sqrt(mom.x * mom.x + mom.y * mom.y);
326           auto eta = -log(tan(atan2(sqrt(mom.x*mom.x+mom.y*mom.y),mom.z)/2.));
327           auto phi = atan2(mom.y, mom.x);
328           auto theta = atan2(sqrt(mom.x*mom.x + mom.y*mom.y),mom.z);
329
330
331           if (mcinfo.getGeneratorStatus() != 1) continue;
332           auto pdg = mcinfo.getPDG();
333           auto& mcmom = mcinfo.getMomentum();
334           auto status = mcinfo.getGeneratorStatus();
335           auto mcp = sqrt(mcmom.x * mcmom.x + mcmom.y * mcmom.y + mcmom.z * mcmom.z);
336           auto mceta = -log(tan(atan2(sqrt(mcmom.x*mcmom.x+mcmom.y*mcmom.y),mcmom.z)/2.));
337           Track_xy_mc -> Fill(pos.x, pos.y);
338           if(pos.x > 0)Track_rz_mc -> Fill(pos.z, rt);
339               else Track_rz_mc -> Fill(pos.z, -rt);
340           fprintf (outputfileMC, "%s  r:%f, rt:%f, p:%f, eta:%f, phi:%f, theta:%f, cp:%f, mceta:%f, pdg:%d\n",Data_name, r, rt, p, eta, phi, theta, mcp, mceta, pdg);
341       }
342   }
```

About Analysis Code

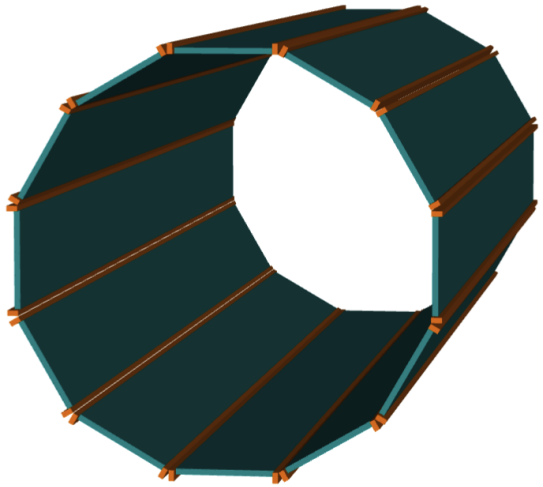
■MC information (edm4hep::SimTrackerHit)

ddsim → ~sim.edm4hep.root ファイル
検出器ヒットのMC情報（DIRCBarHitsがhpDIRC surfaceのヒット情報）

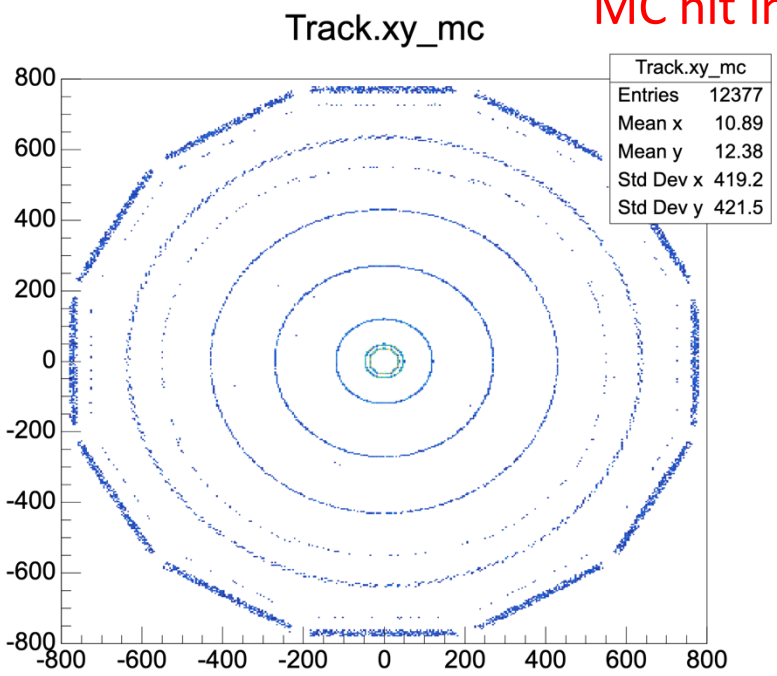
その他、 Barrel and Endcapのトラッカー検出器ヒットのMC情報が含まれる

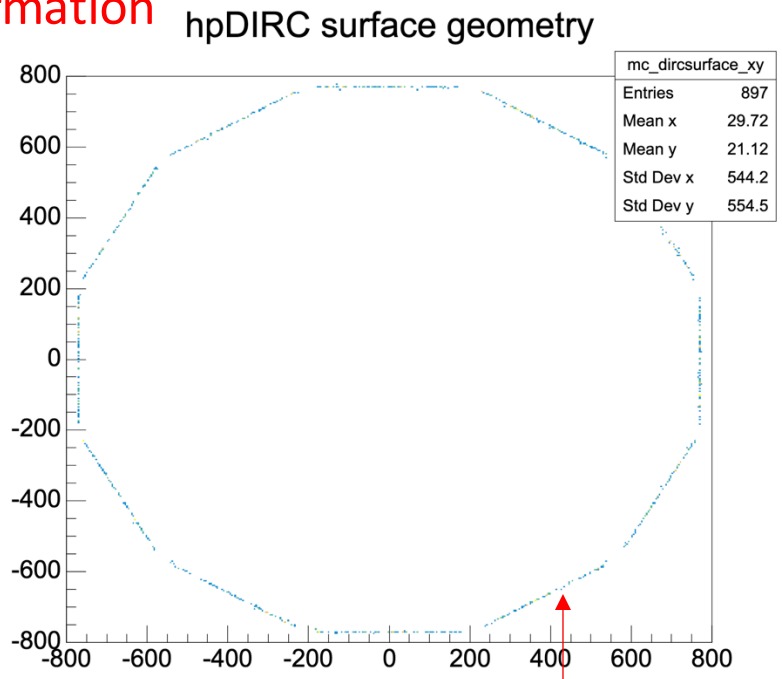
hpDIRC surface geometry (defined by Geant4)

■New geometry



MC hit information






```
EVENT 4
MCParticle    pdg:-211, p:6.000000, eta:-0.069659, phi:-0.134677, vtx:0.000000, end:1557.169237
VertexBarrelHits  r:36.089468, rt:36.002210, p:5.999233, eta:-0.069247, phi:-0.137460, theta:1.639988, cp:6.000000, mceta:-0.069659, pdg:-211
VertexBarrelHits  r:48.118721, rt:48.002713, p:5.999222, eta:-0.069207, phi:-0.138491, theta:1.639948, cp:6.000000, mceta:-0.069659, pdg:-211
VertexBarrelHits  r:120.292002, rt:120.003523, p:5.999197, eta:-0.069202, phi:-0.144673, theta:1.639943, cp:6.000000, mceta:-0.069659, pdg:-211
SiBarrelHits  r:271.622498, rt:270.972416, p:5.999155, eta:-0.069175, phi:-0.157668, theta:1.639916, cp:6.000000, mceta:-0.069659, pdg:-211
SiBarrelHits  r:430.369157, rt:429.339817, p:5.998950, eta:-0.069222, phi:-0.171259, theta:1.639964, cp:6.000000, mceta:-0.069659, pdg:-211
TOFBarrelHits  r:637.265306, rt:635.741938, p:5.998363, eta:-0.069097, phi:-0.189478, theta:1.639838, cp:6.000000, mceta:-0.069659, pdg:-211
DIRCBarHits  r:783.996336, rt:782.080762, p:5.972156, eta:-0.073050, phi:-0.204004, theta:1.643782, cp:6.000000, mceta:-0.069659, pdg:-211
DIRCBarHits  r:840.156027, rt:774.041517, p:0.000208, eta:-0.179534, phi:0.388139, theta:1.749374, cp:6.000000, mceta:-0.069659, pdg:-211
DIRCBarHits  r:839.313887, rt:772.840684, p:0.000215, eta:-0.236910, phi:2.882551, theta:1.805521, cp:6.000000, mceta:-0.069659, pdg:-211
DIRCBarHits  r:844.772741, rt:780.567620, p:0.000171, eta:3.002465, phi:-1.134712, theta:0.099247, cp:6.000000, mceta:-0.069659, pdg:-211
DIRCBarHits  r:1014.476148, rt:784.720317, p:0.051137, eta:-0.114422, phi:2.279752, theta:1.684970, cp:6.000000, mceta:-0.069659, pdg:-211
DIRCBarHits  r:841.641765, rt:769.022910, p:0.000467, eta:-0.334031, phi:-2.599846, theta:1.898783, cp:6.000000, mceta:-0.069659, pdg:-211
DIRCsurface  r:783.996336, rt:782.080762, p:5.972156, eta:-0.073050, phi:-0.204004, theta:1.643782
```

原因がわかっていない問題（今の所問題なし？）

Generated Particleのみを選んでいるはずなのに複数のMCヒット情報が得られてしまう

検出器上の
MC情報

→複数のDIRCBarHits

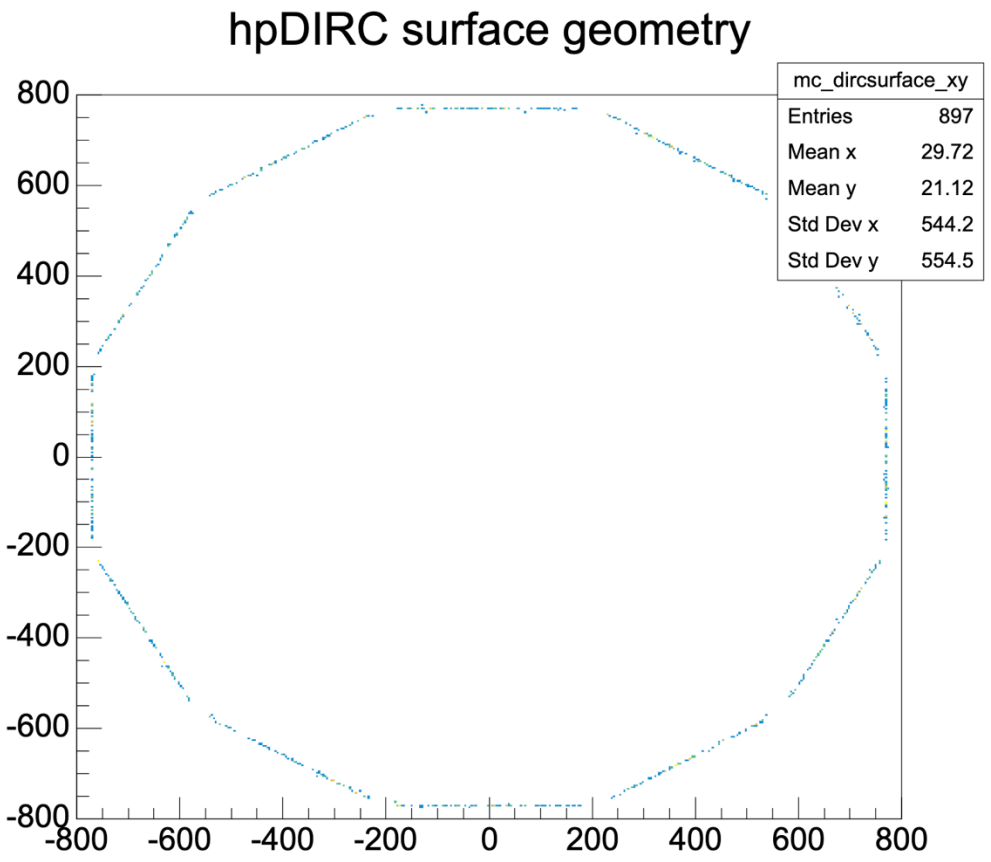
採用した値のみの場合

それらしい値のみを採用して
角度分解能の計算に使用している

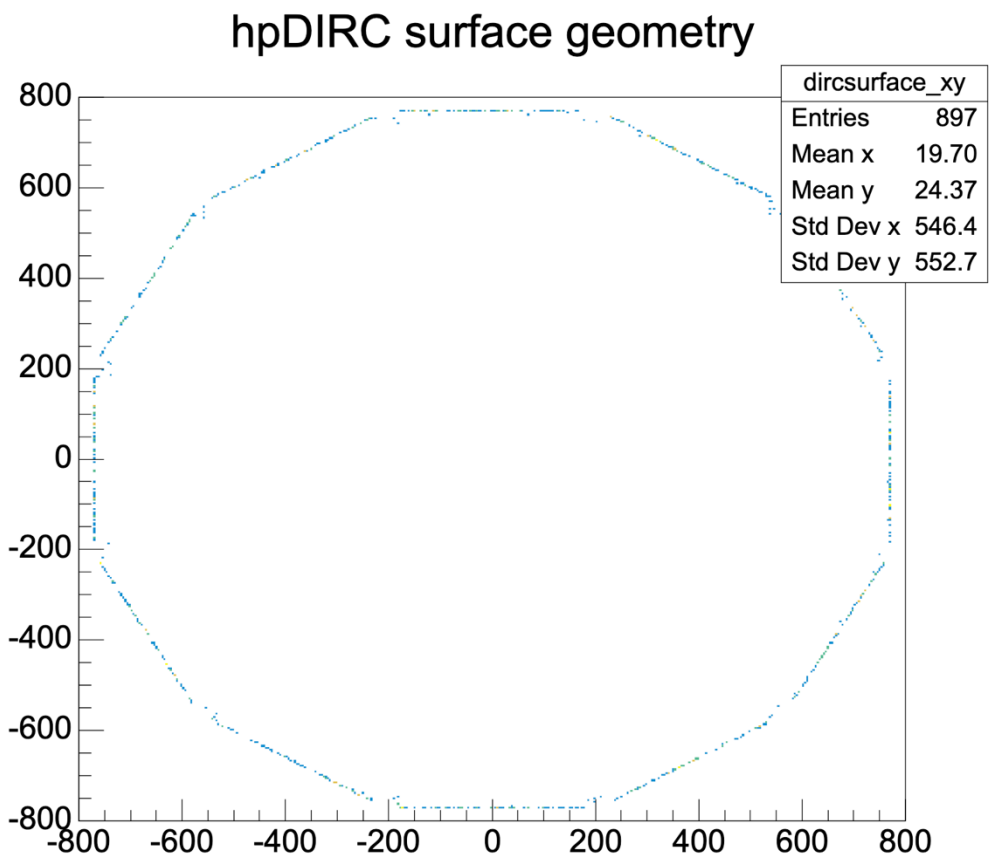
Trajectory to hpDIRC surface

■Track propagate　→　track上で、MC情報と同じ r の時の粒子情報を取得

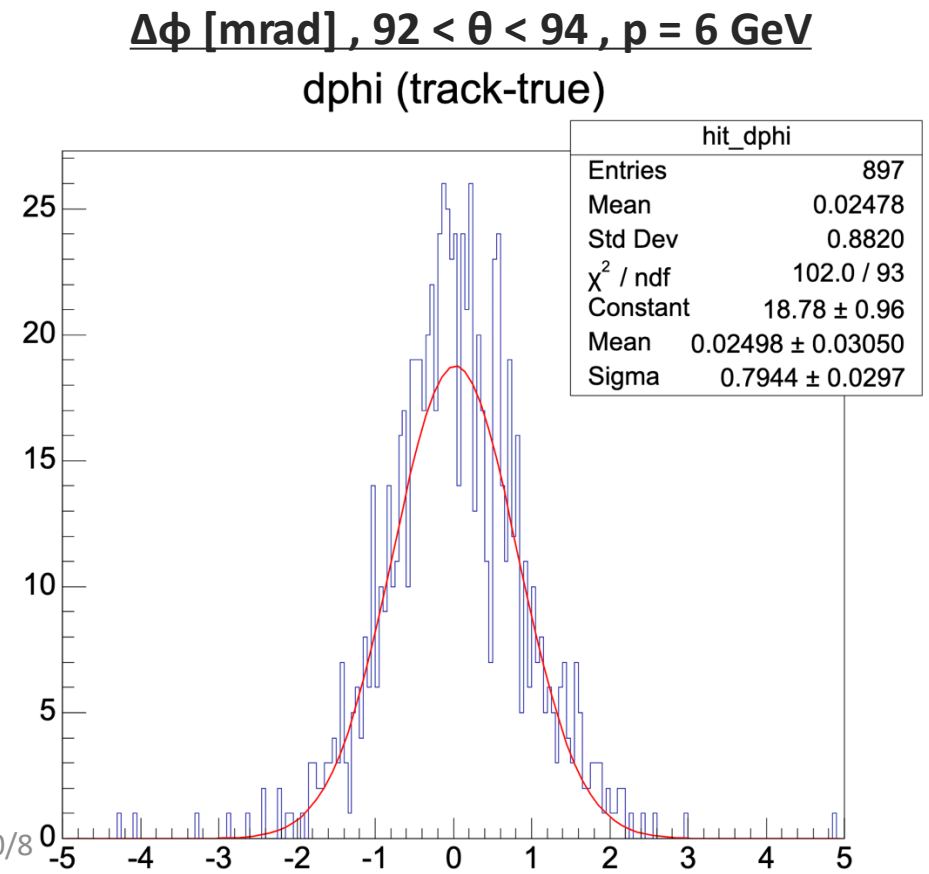
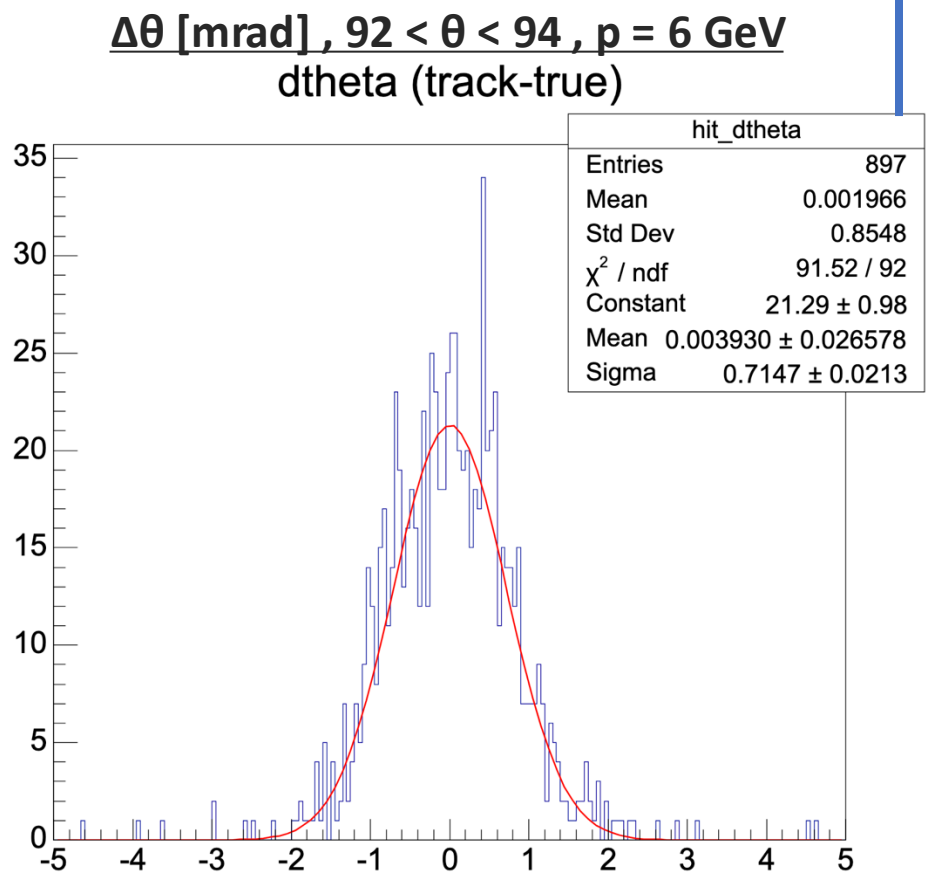
前スライドと同じMCヒット情報



Track propagateから得られた情報



Angular distribution on hpDIRC surface

- Single particle : π^- (1000 events)
- Momentum : 6 [GeV/c]
- Direction : $0^\circ \leq \phi \leq 360^\circ$, $92^\circ \leq \theta \leq 94^\circ \rightarrow \langle\eta\rangle = -0.05$

$\sigma = 0.7147 \pm 0.0213$
要求を満たしていない