Likelihood of elongated shower

Angle issues

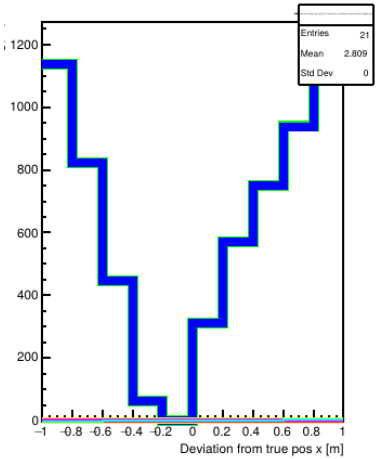
Jordan Seneca

November 27th 2020, Reconstruction Meeting

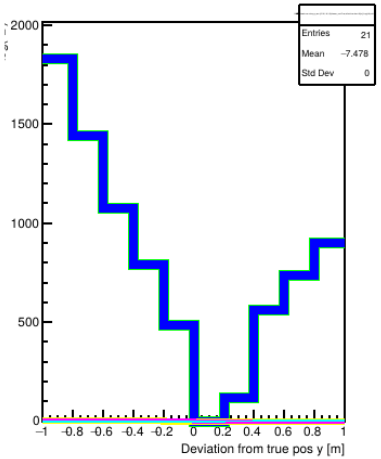
Generate shower with elongation.

Plot likelihood of PDF components about shower maximum.

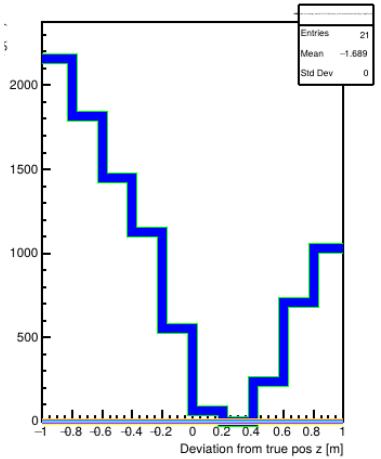
unhit likelihood, lik, x



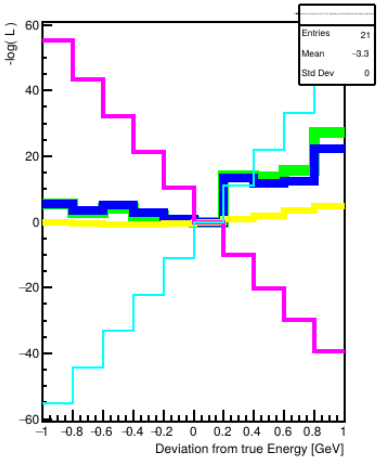
unhit likelihood, lik, y



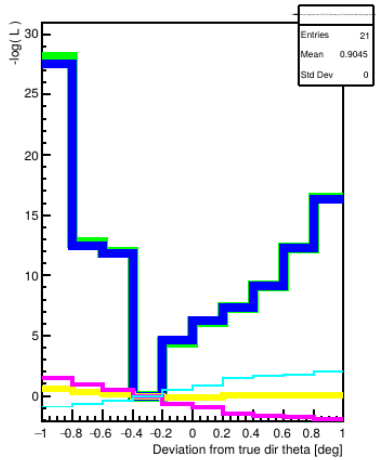
unhit likelihood, lik, z



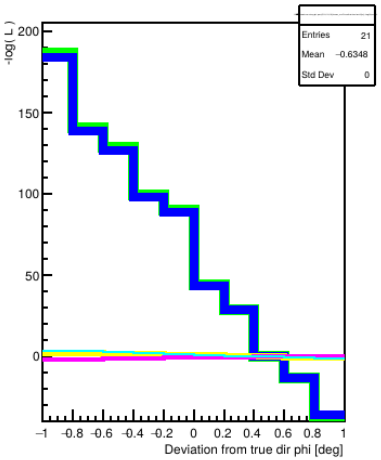
unhit likelihood, lik, E



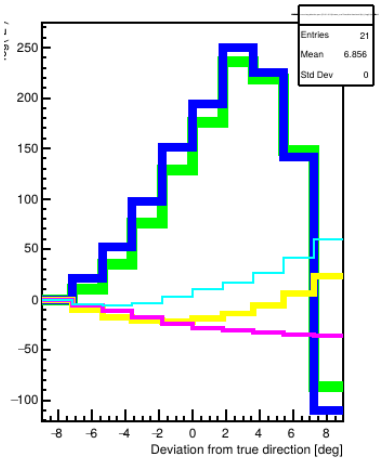
unhit likelihood, lik, theta



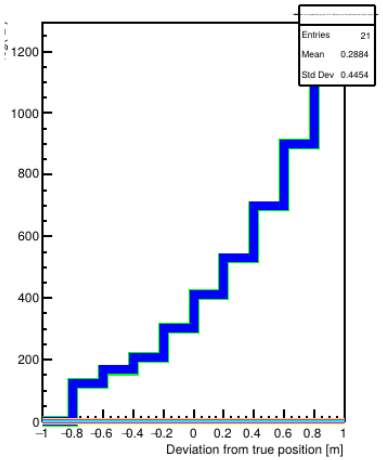
unhit likelihood, lik, phi



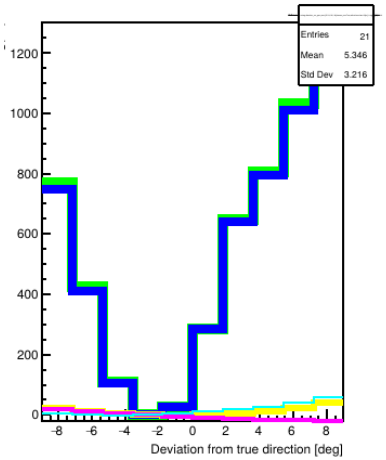
unhit likelihood, lik, direction



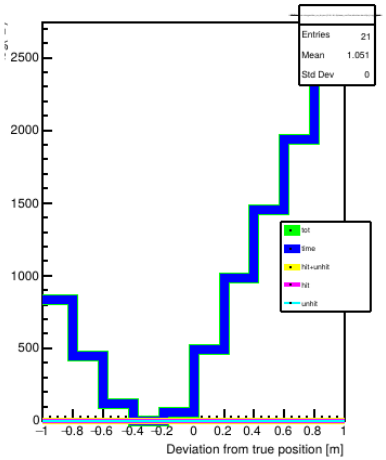
unhit likelihood, lik, position

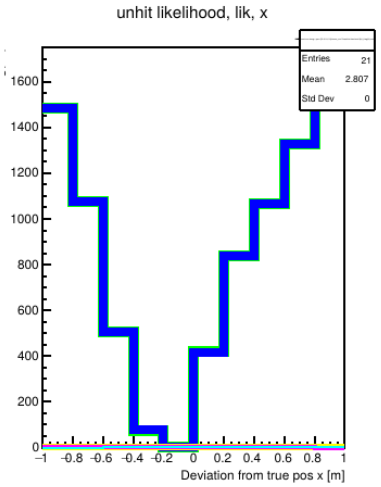


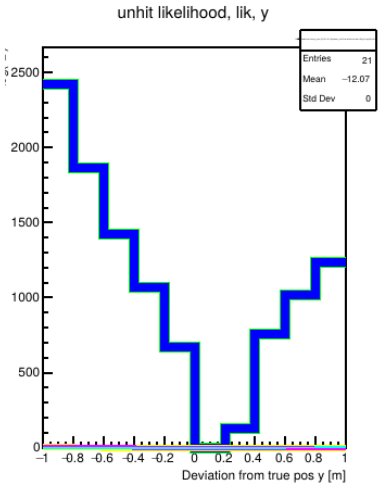
unhit likelihood, lik, direction_mc_pos

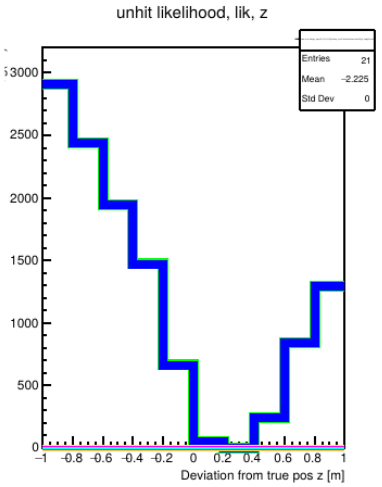


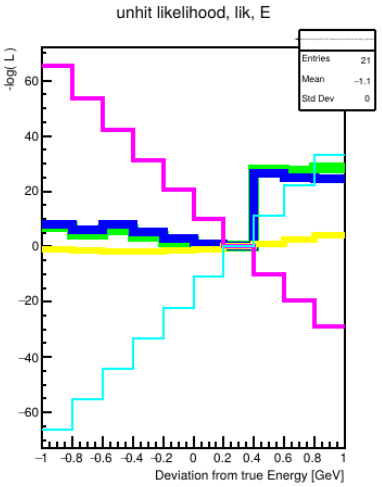
unhit likelihood, lik, position_mc_dir

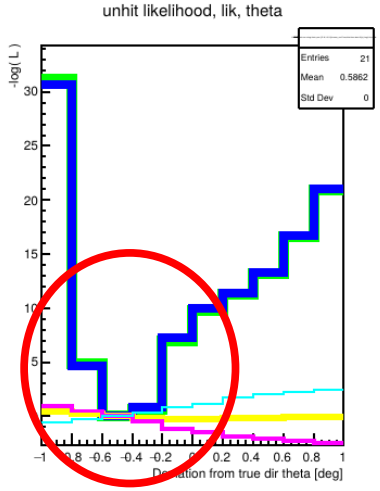


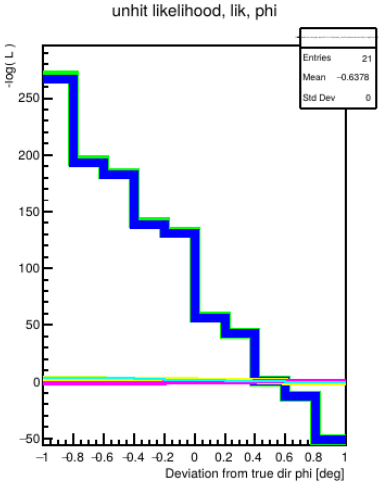

unhit likelihood, lik, x
Entries 21
Mean 2.807
Std Dev 0
Deviation from true pos x [m]

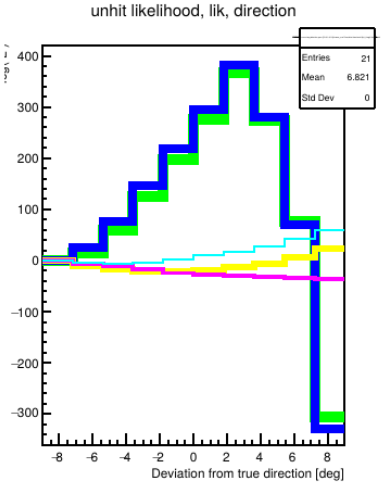

unhit likelihood, lik, y
Entries 21
Mean −12.07
Std Dev 0
Deviation from true pos y [m]

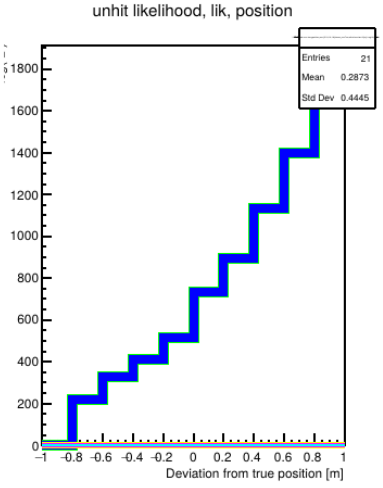

unhit likelihood, lik, z
Entries 21
Mean −2.225
Std Dev 0
Deviation from true pos z [m]

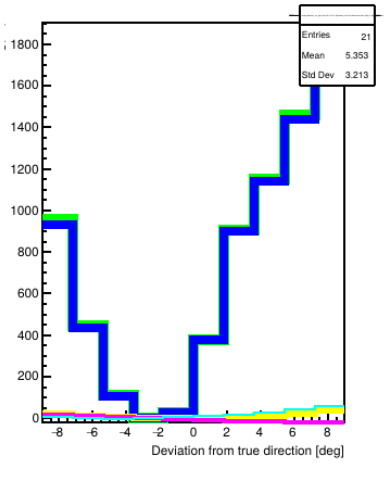

unhit likelihood, lik, E
-log(L)
Entries 21
Mean −1.1
Std Dev 0
Deviation from true Energy [GeV]

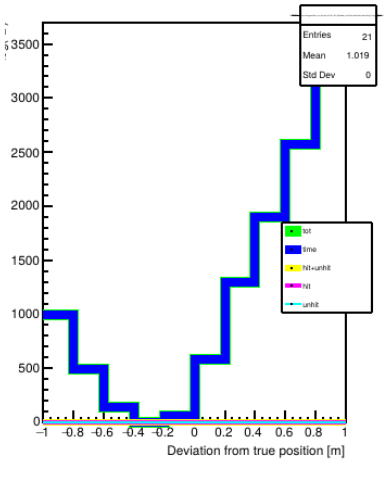

unhit likelihood, lik, theta
-log(L)
Entries 21
Mean 0.5862
Std Dev 0
Deviation from true dir theta [deg]


unhit likelihood, lik, phi
-log(L)
Entries 21
Mean −0.6378
Std Dev 0
Deviation from true dir phi [deg]


unhit likelihood, lik, direction
Entries 21
Mean 6.821
Std Dev 0
Deviation from true direction [deg]


unhit likelihood, lik, position
Entries 21
Mean 0.2873
Std Dev 0.4445
Deviation from true position [m]


unhit likelihood, lik, direction_mc_pos
Entries 21
Mean 5.353
Std Dev 3.213
Deviation from true direction [deg]


unhit likelihood, lik, position_mc_dir
Entries 21
Mean 1.019
Std Dev 0
tot
time
hit+unhit
hit
unhit
Deviation from true position [m]

“aanet way”

```
void init( const Trk& trk, const Vec& pos, const Vec& dir )
{
  const Vec    v = pos - trk.pos;
  const double y = v.dot( trk.dir );
  const double u = dir.dot( trk.dir );

  E     = trk.E;
  d     = v.len();
  z     = y/d;
  theta = acos(u);
  phi   = angle_between ( v - y * trk.dir, dir - u * trk.dir );
}
```

“jpp way”

```
void set_hit_dir( JPosition3D Jpos, const JDirection3D& Jdir, const Vec& hit_dir ){

  const JRotation3D Rot( Jdir );

  Jpos.rotate(Rot);

  JAxis3D axis( JPosition3D(  _hit_pos.x, _hit_pos.y, _hit_pos.z ),
                JDirection3D( hit_dir.x, hit_dir.y, hit_dir.z ) );
  axis.transform( Rot, Jpos);

  R     = axis.getPosition().getX();
  cd    = axis.getZ()/D;
  theta = axis.getTheta();
  phi   = fabs(axis.getPhi());
}
```

Next: Generate and plot likelihood of each step separating toy MC from real MC.