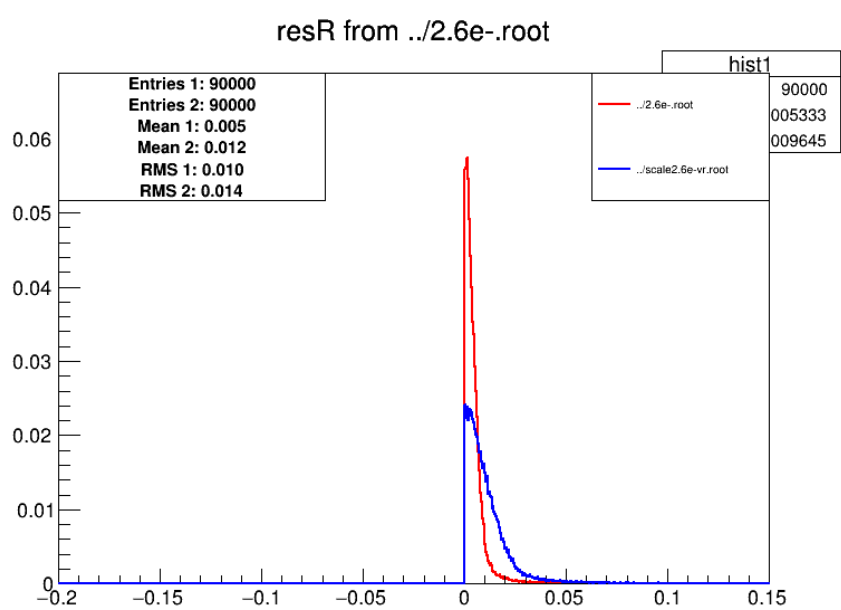


RecChargedTool.cc添加 scaling

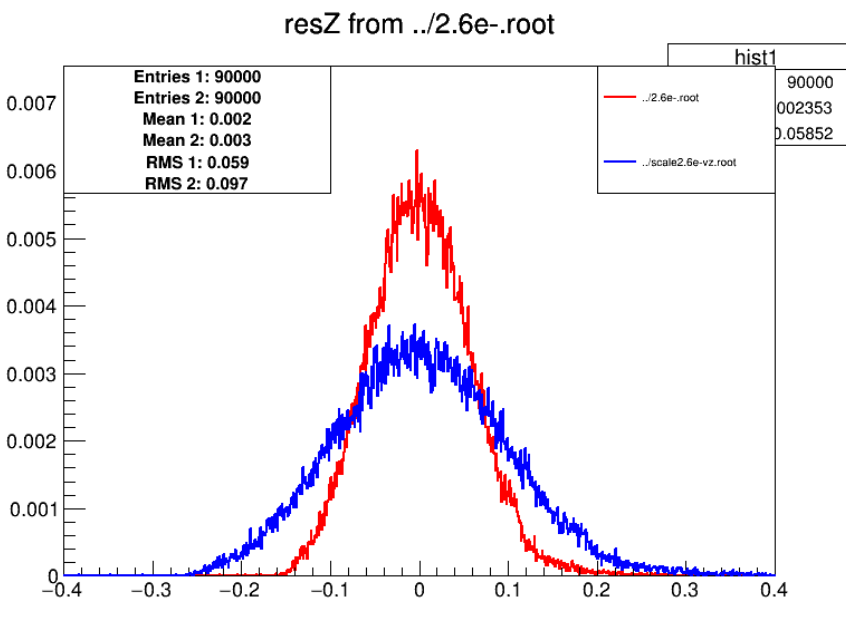
e- 2.6GeV, costheta -0.6to0.5

Double ResXY = 130
Double ResZ = 2480



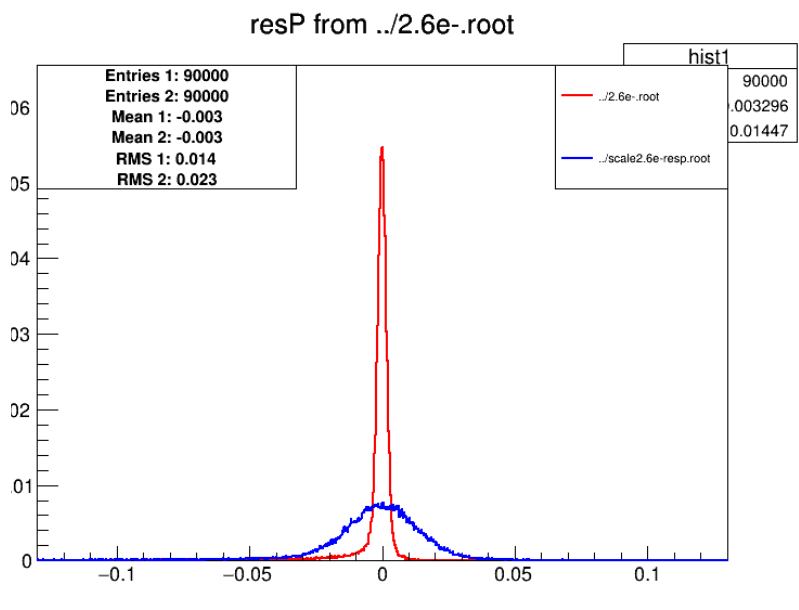
VrScale = 2.5

将峰值附近拟合得到scale $\approx$ 2.47



VzScale = 1.7

将峰值附近拟合得到scale $\approx$ 1.55

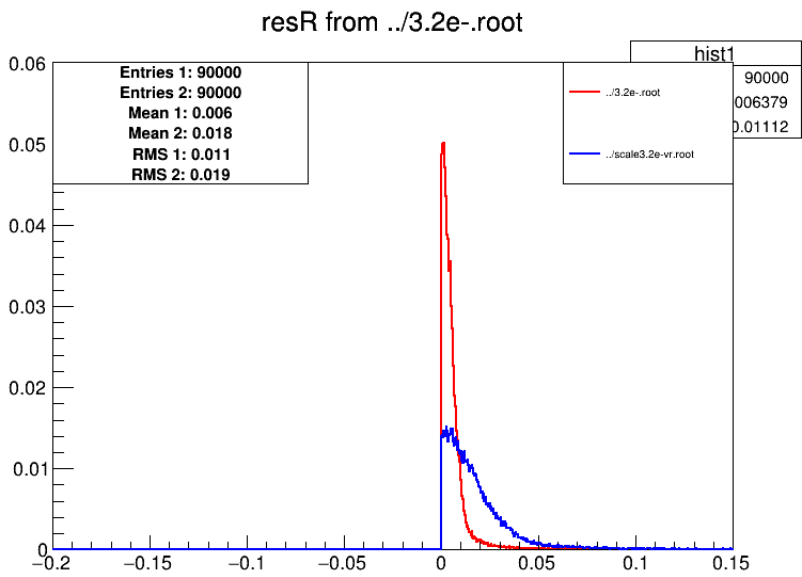


ResxyScale = 260
ResZScale = 4960

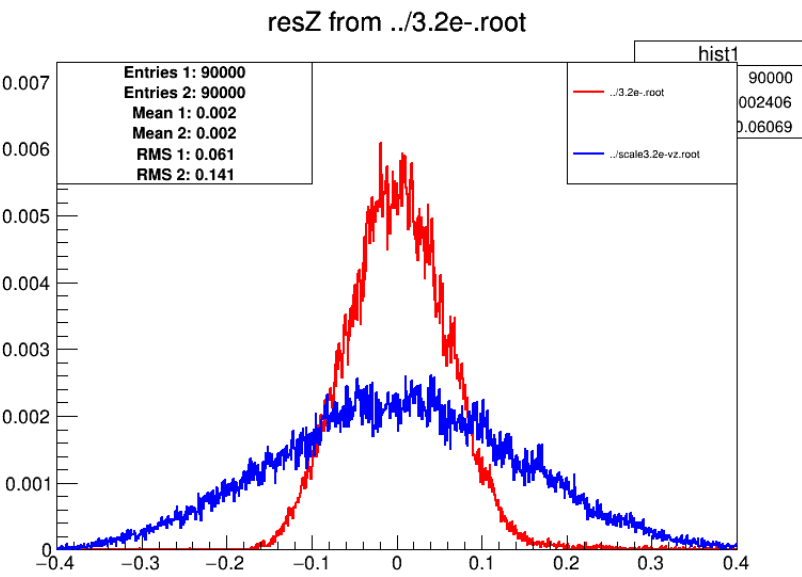
将峰值附近拟合得到scale $\approx$ 2.22

RecChargedTool.cc添加 scaling

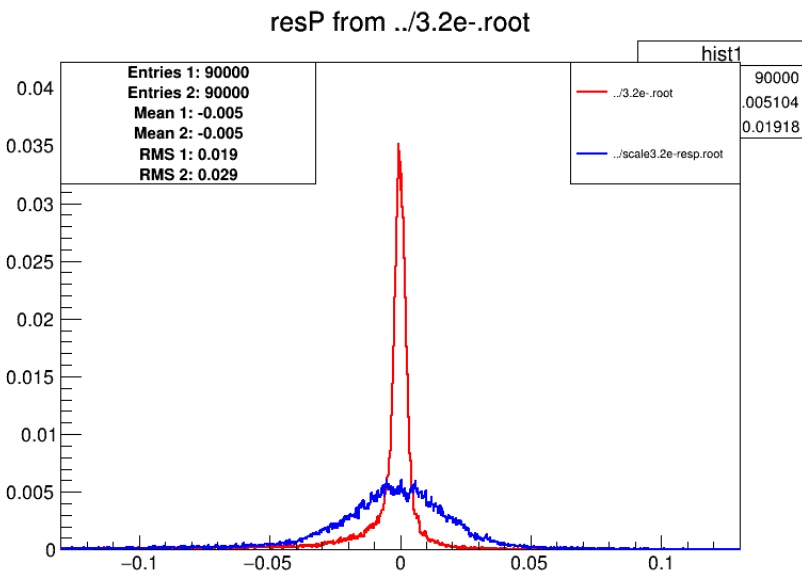
e- 3.2GeV, costheta 0.5to1



VrScale = 3.0



VzScale = 2.5



ResXYScale = 325
ResZScale = 6200

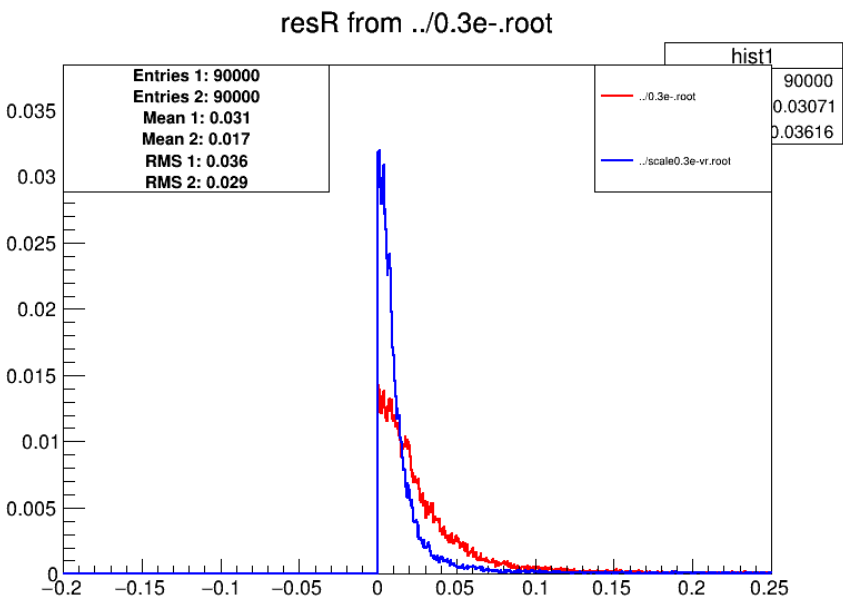
RecChargedTool.cc添加 scaling

e- 0.3GeV, costheta -1to-0.6

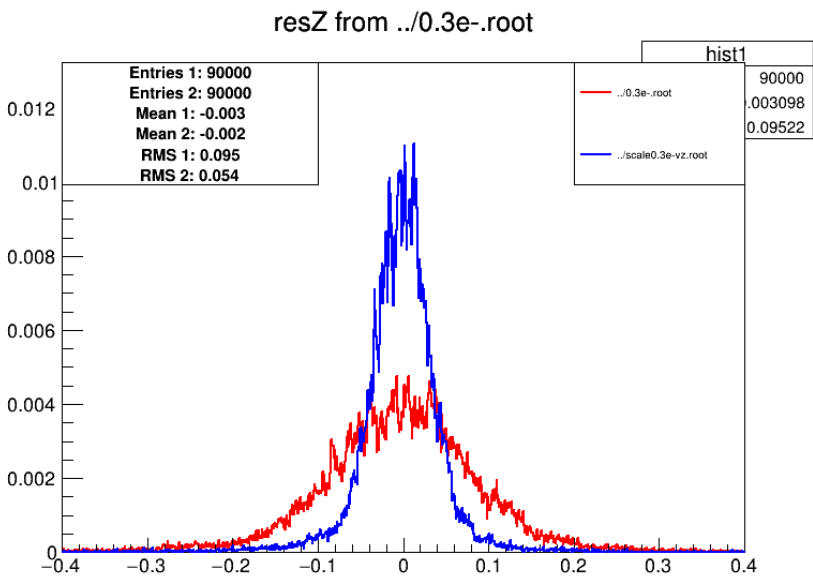
调整m_scaleVr ;m_scaleVz 改变Vr Vz分辨率

```
m_chgTrk.recx0 = m_vrfac*cos(randomTheta)*m_scaleVr+begPos.x();
m_chgTrk.recy0 = m_vrfac*sin(randomTheta)*m_scaleVr+begPos.y();
m_chgTrk.recz0 = m_vzfac*m_scaleVz + begPos.z();
```

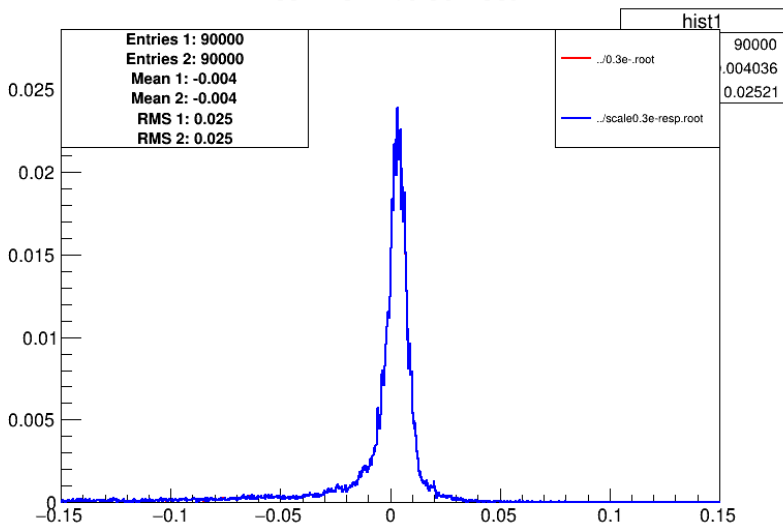
Double ResXY = 130
Double ResZ = 2480



VrScale = 0.4



VzScale = 0.5



ResxyScale = 26
ResZScale = 496

代码功能添加 scaling

RecChargedTool.cc:

ResXY影响Resrho & ResPhi → scaleRho & scalePhi ; ResZ 影响 ResThe → scaleThe

```
642 if ( !resChgTrack( m_ResKeysName[0], curYene, binXthe, ibinYenes, m_rhofac ) ){
643     LogError << "failed to retrieve rho correction factor" << std::endl;
644     LogError << "p:the      " << mc_p4.rho() << ":" << mc_p4.theta() << std::endl;
645     return false;
646 }
647 m_chgTrk.recmon = m_chgTrk.recmon * ( m_rhofac*scaleRho+ 1 );
648 // --- do theta smearing;
649 if ( !resChgTrack( m_ResKeysName[1], curYene, binXthe, ibinYenes, m_thefac ) ){
650     LogError << "failed to retrieve theta correction factor" << std::endl;
651     return false;
652 }
653 m_chgTrk.recthe = m_chgTrk.recthe + m_thefac*scaleThe ;
654 // --- do phi smearing;
655 if ( !resChgTrack( m_ResKeysName[2], curYene, binXthe, ibinYenes, m_phifac ) ){
656     LogError << "failed to retrieve phi correction factor" << std::endl;
657     return false;
658 }
659 m_chgTrk.recphi = m_chgTrk.recphi + m_phifac*scalePhi ;
```

```
m_ResKeysName.push_back( "ResP" );
m_ResKeysName.push_back( "ResThe" );
m_ResKeysName.push_back( "ResPhi" );
```

Conclusion:

ResXY调整ResP & ResPhi

ResZ 调整 ResThe

$$\text{ResRho} = \sqrt{(20.17 \text{ ResXY } p_T)^2 + \left(\frac{a}{0.3 p_T \sqrt{\sin \theta}}\right)^2}$$

$$\text{ResThe} = \sqrt{(0.64 \text{ ResZ } \sin^2 \theta)^2 + \left(\frac{a}{\beta} \sqrt{\sin \theta}\right)^2}$$

$$\text{ResPhi} = \sqrt{(2.5 \text{ ResXY})^2 + \left(\frac{a}{\beta p_T \sqrt{\sin \theta}}\right)^2}$$

$$\text{ResRho}_0 = \sqrt{(20.17 \text{ ResXY}_0 p_T)^2 + \left(\frac{a}{0.3 p_T \sqrt{\sin \theta}}\right)^2}$$

$$\text{ResThe}_0 = \sqrt{(0.64 \text{ ResZ}_0 \sin^2 \theta)^2 + \left(\frac{a}{\beta} \sqrt{\sin \theta}\right)^2}$$

$$\text{ResPhi}_0 = \sqrt{(2.5 \text{ ResXY}_0)^2 + \left(\frac{a}{\beta p_T \sqrt{\sin \theta}}\right)^2}$$

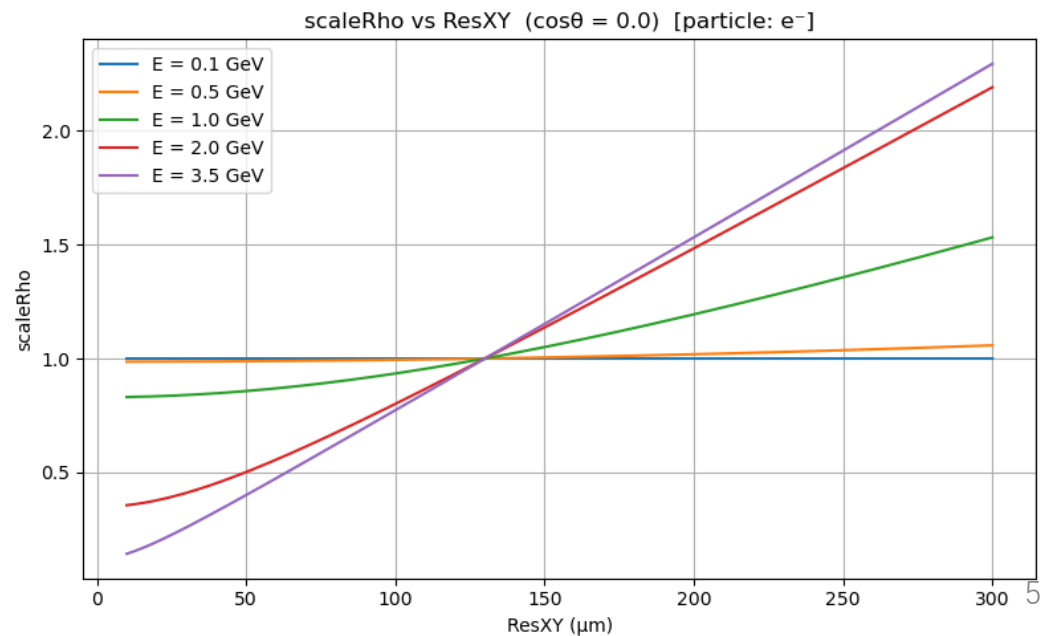
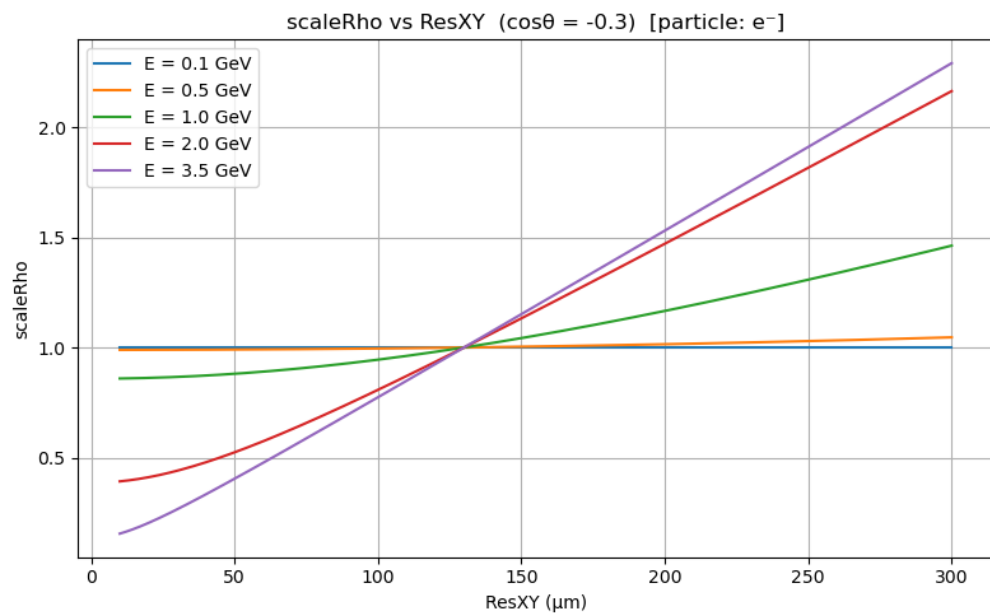
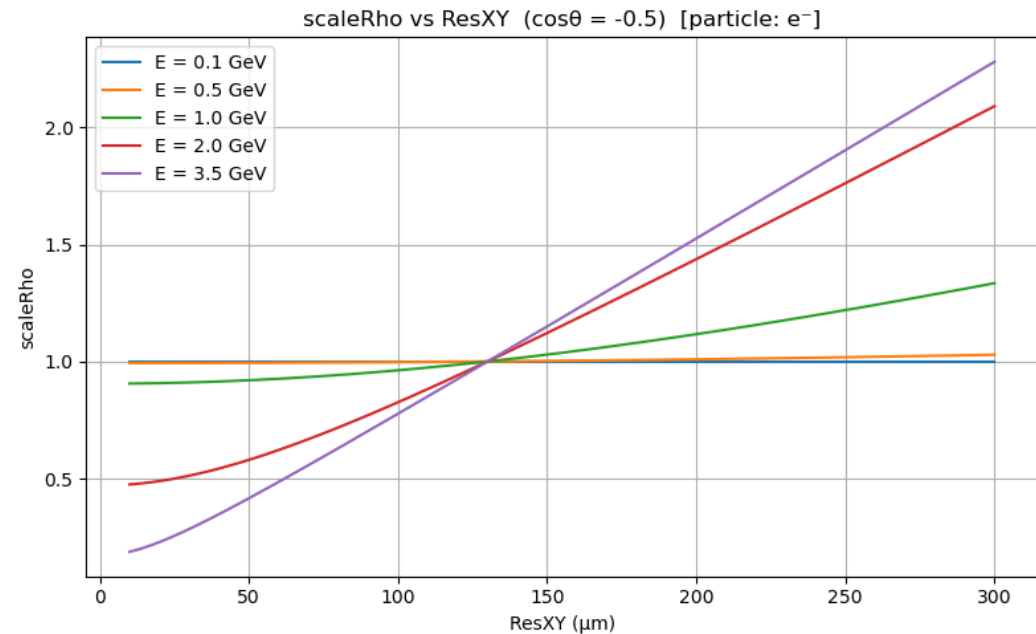
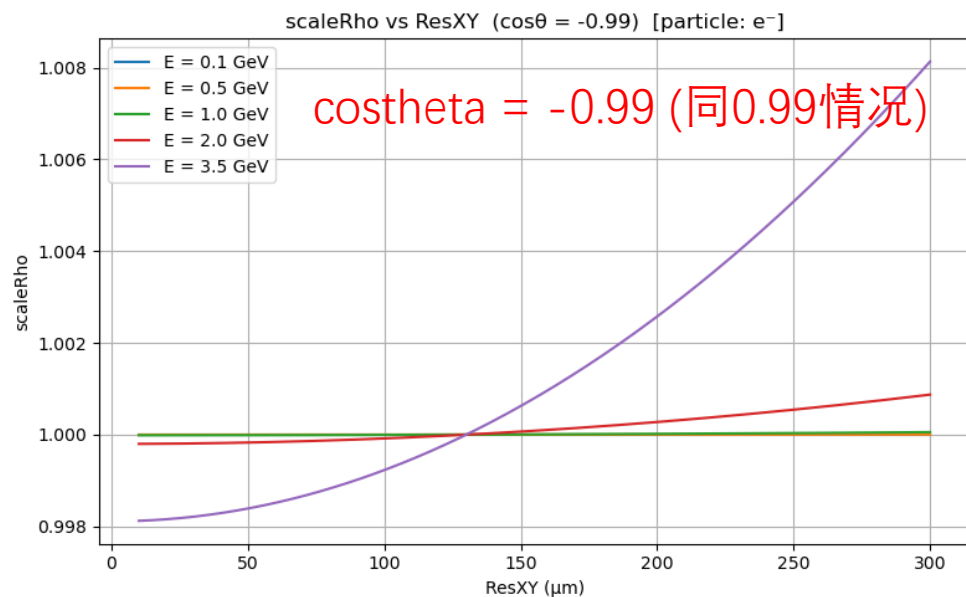
$$\text{scaleRho} = \frac{\text{ResRho}}{\text{ResRho}_0}$$

$$\text{scaleThe} = \frac{\text{ResThe}}{\text{ResThe}_0}$$

$$\text{scalePhi} = \frac{\text{ResPhi}}{\text{ResPhi}_0}$$

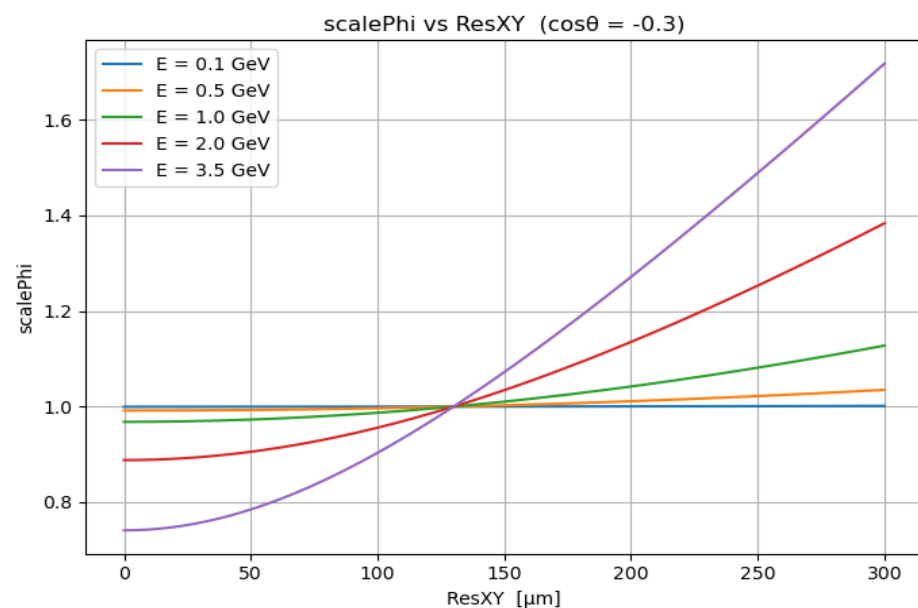
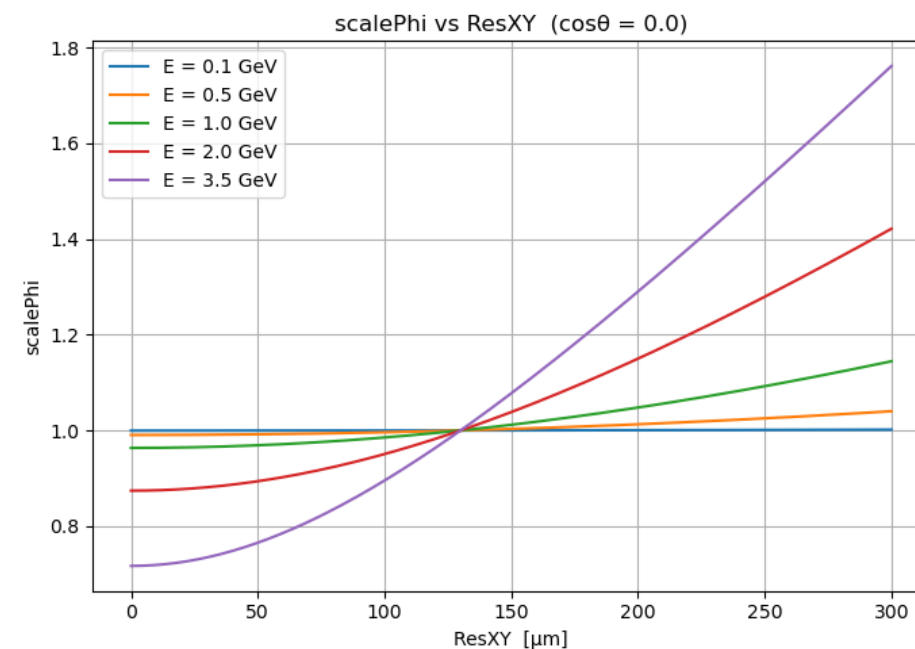
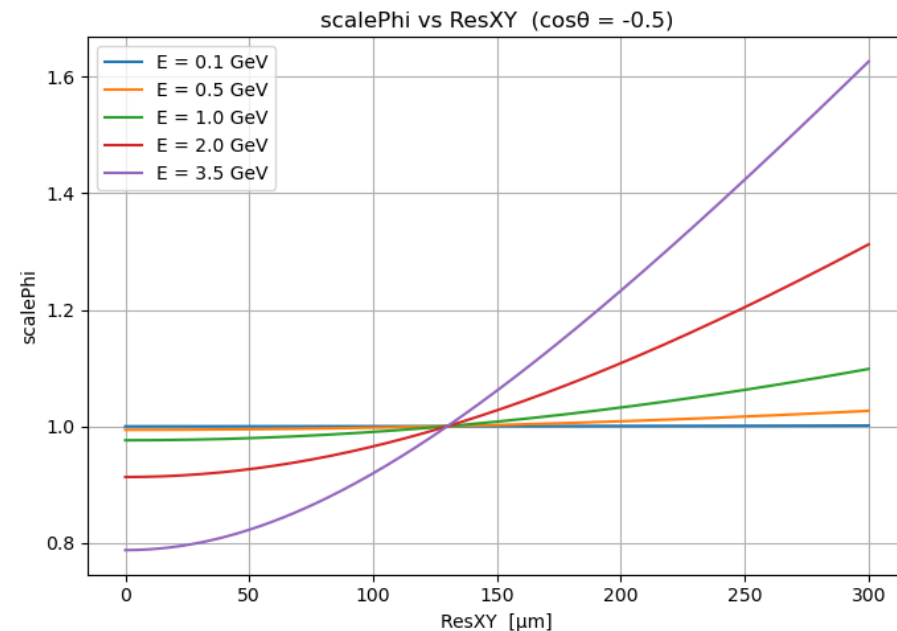
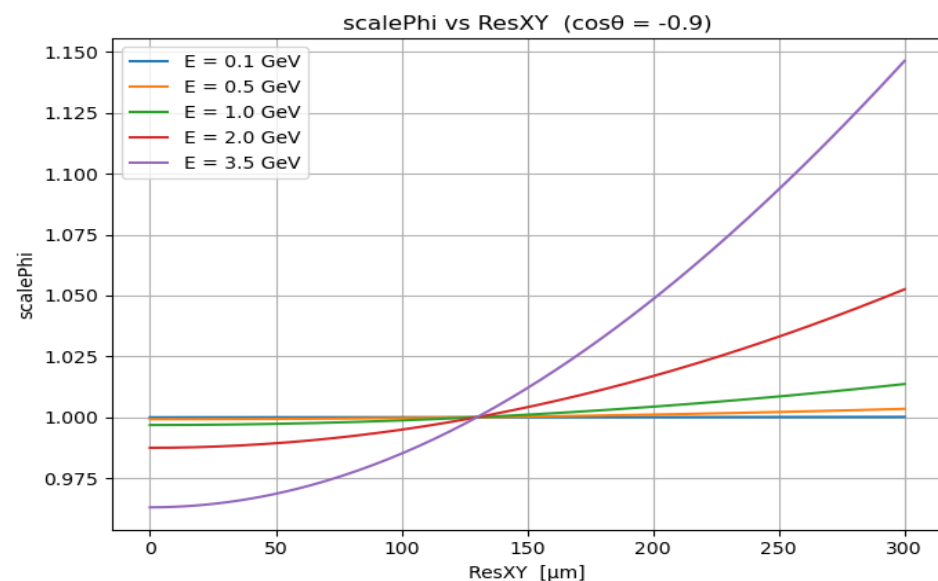
代码功能添加 scaling

RecChargedTool.cc: 对resP, 检查scalerho随resXY的变化趋势



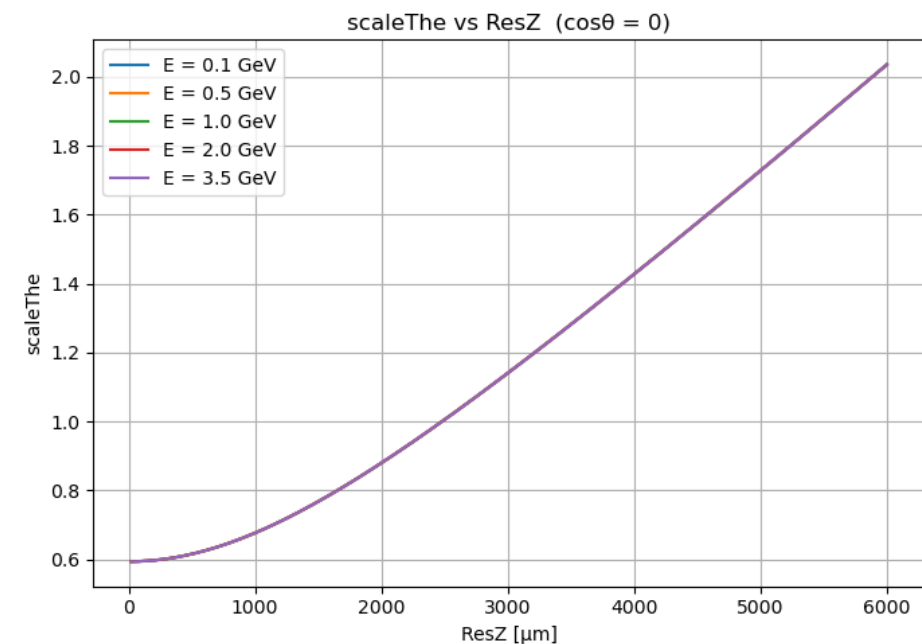
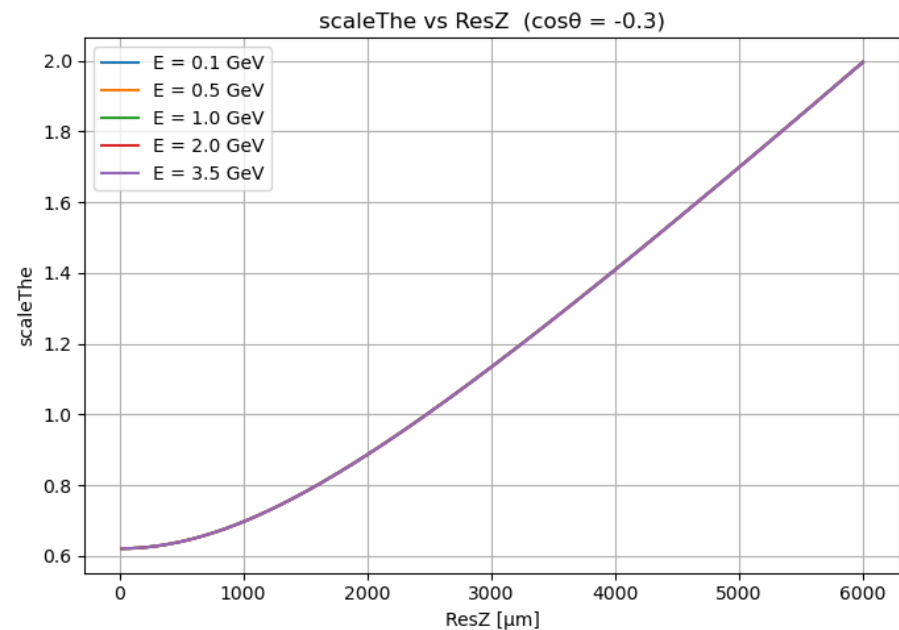
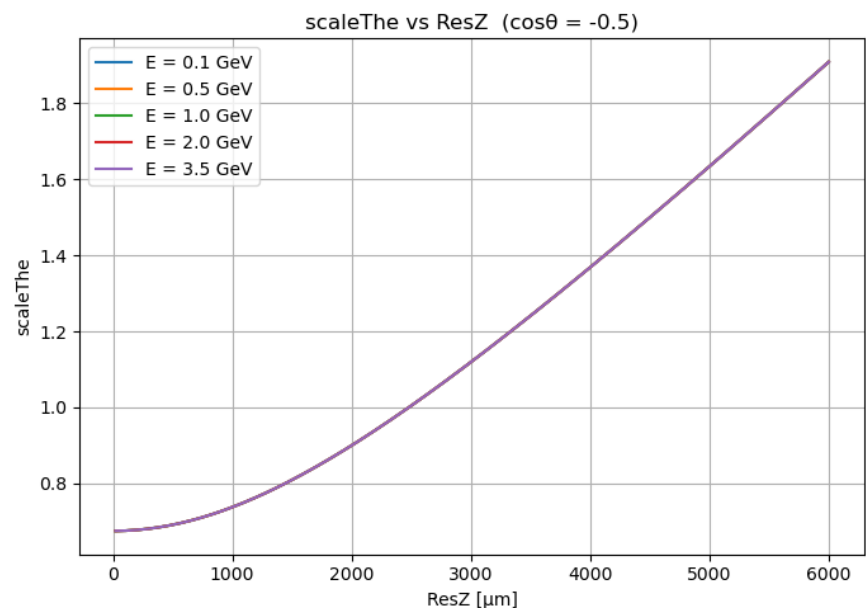
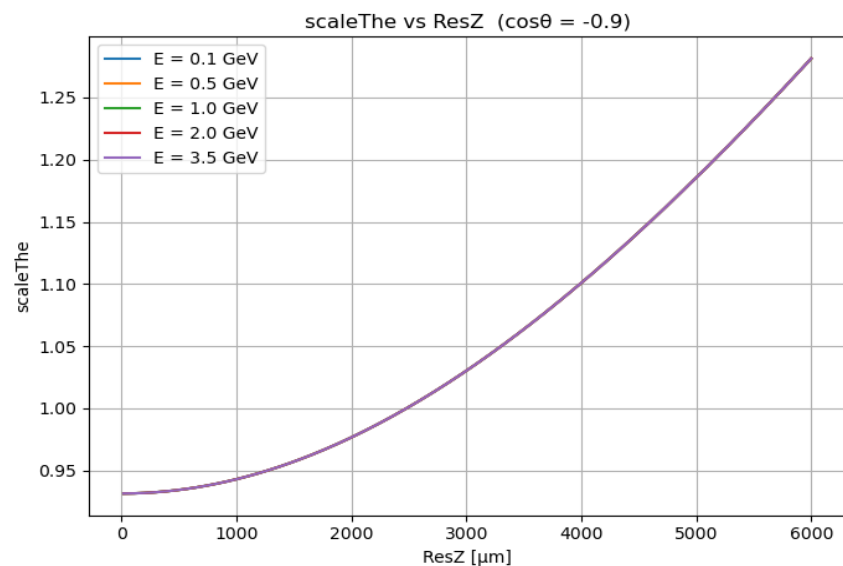
代码功能添加 scaling

RecChargedTool.cc: 检查scalephi随resXY的变化趋势



代码功能添加 scaling

RecChargedTool.cc: 检查scalethe随resZ的变化趋势

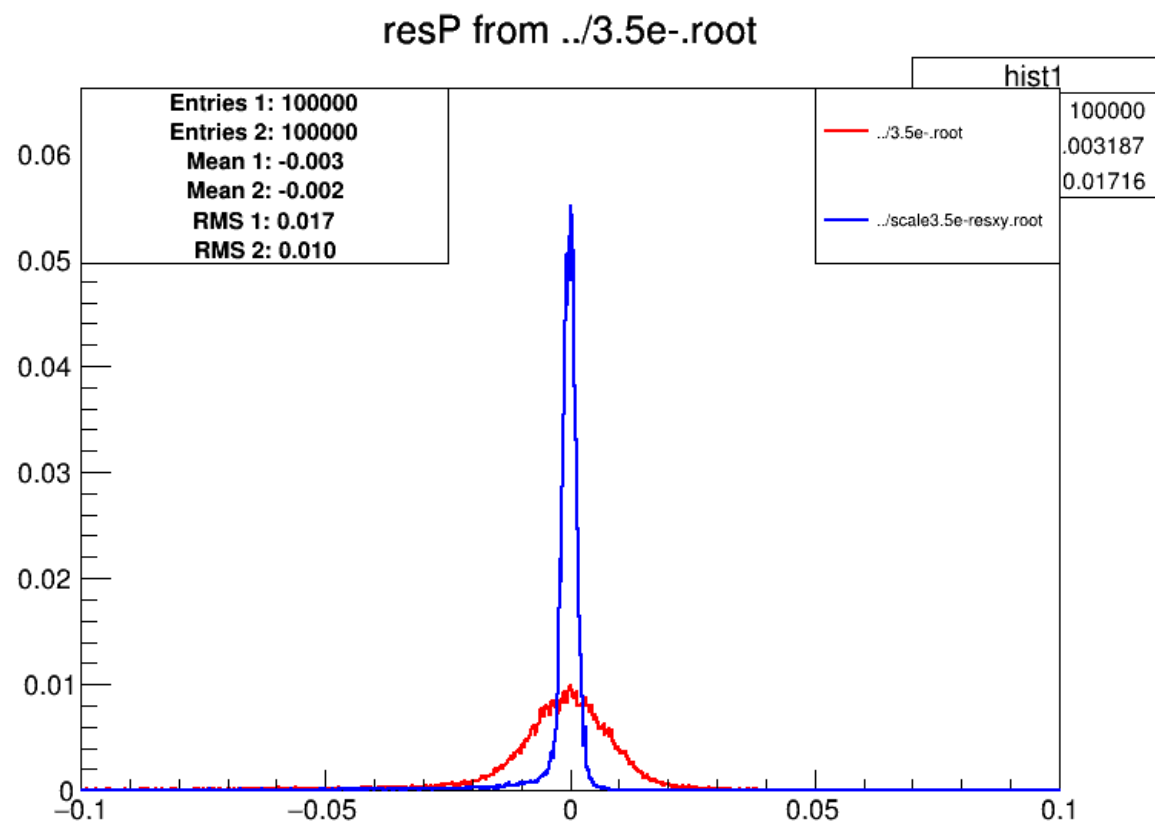


代码功能添加 scaling

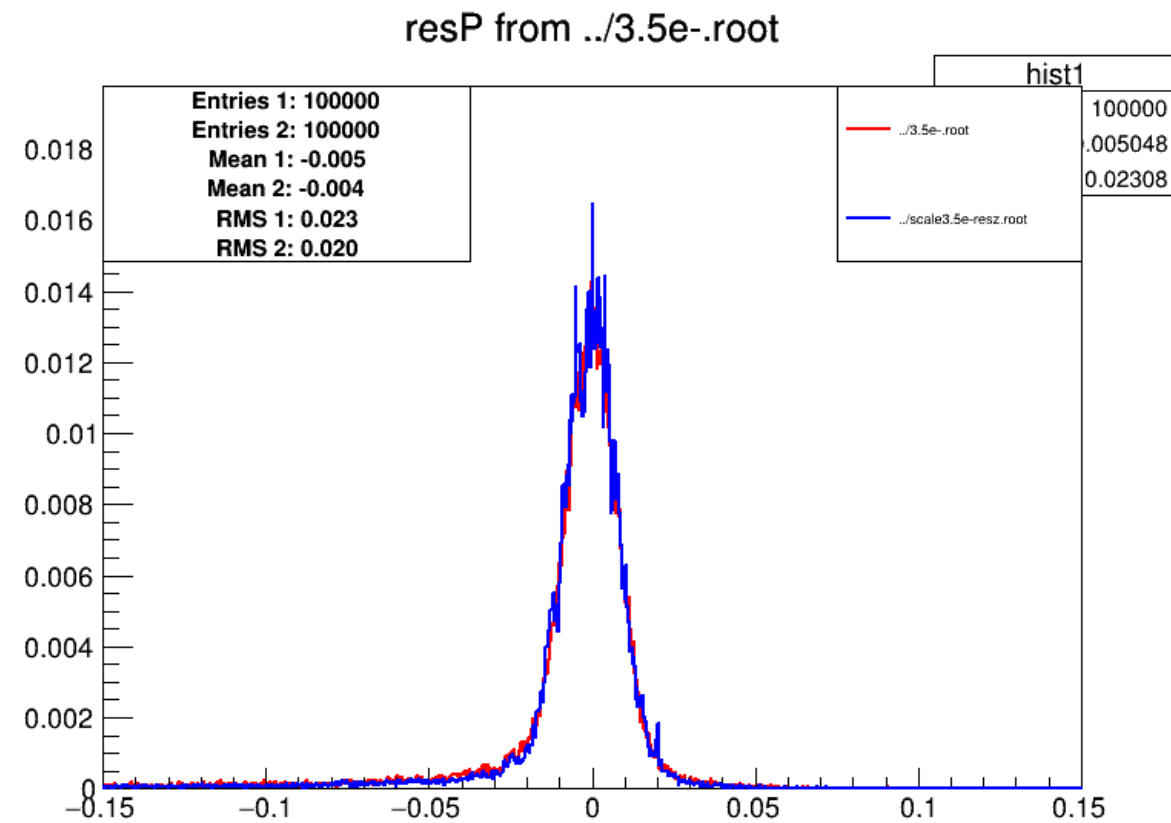
RecChargedTool.cc: double ResXY = 130 ; double ResZ = 2480

要使scalerho显著小于1，产生3.5GeV， costheta -0.3to0的e-进行测试；

令resXY = 13， resZ = 2480;同时测试resXY = 130, resZ =24.8 进行对比， resZ的设置对ResP几乎没有影响



resXY = 13, resZ = 2480;



resXY = 130, resZ =24.8;

代码功能添加 scaling

RecChargedTool.cc:

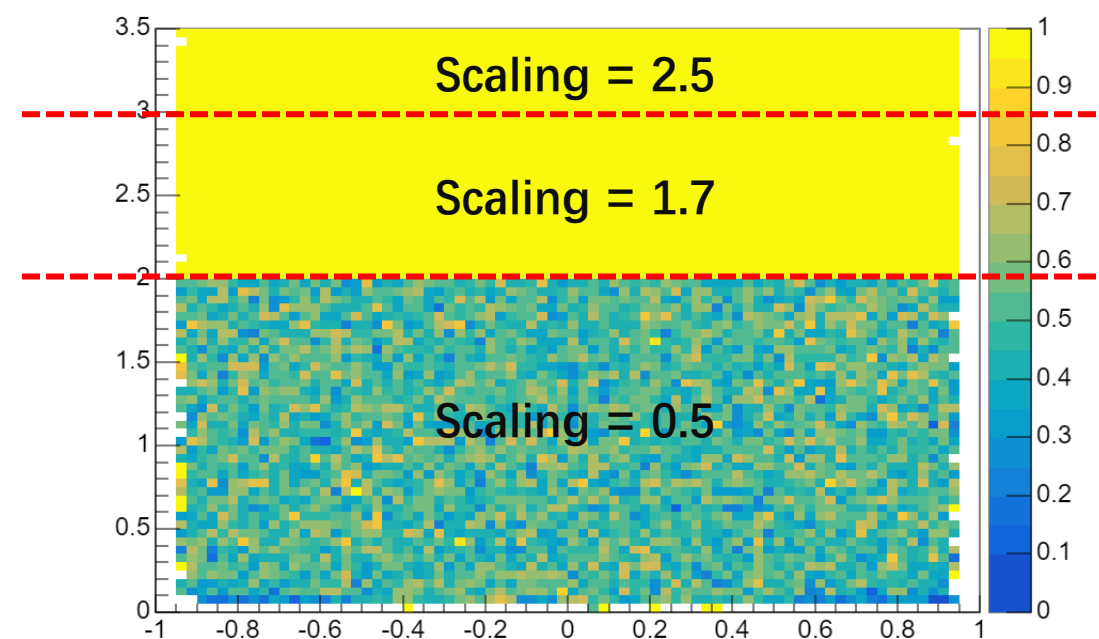
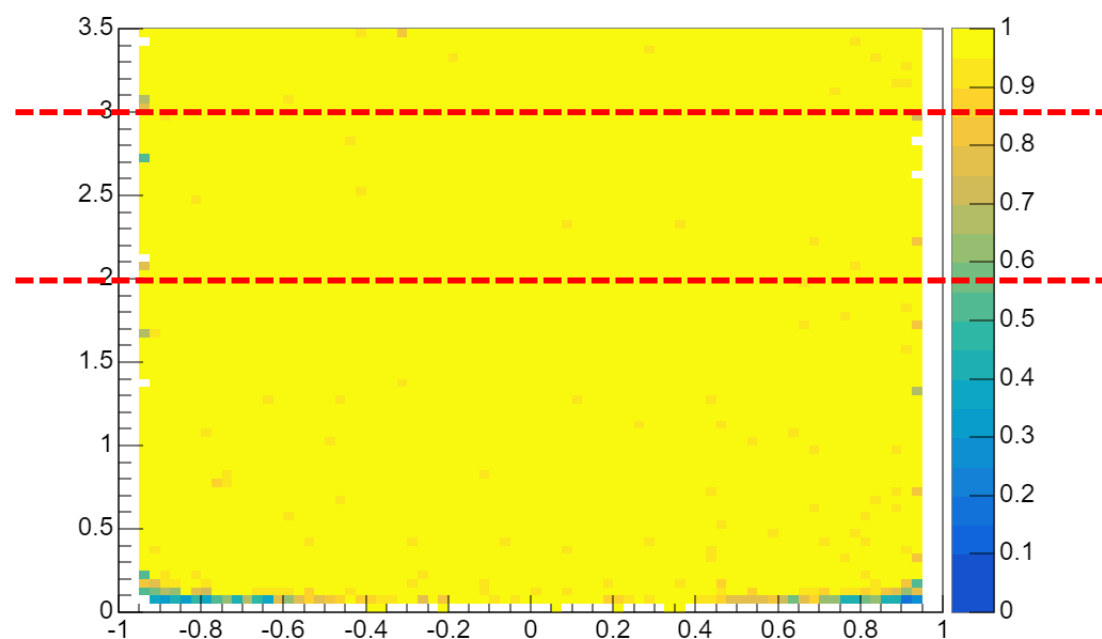
m_scaleRecEff :

对比cos-1to1 , 能量0.05-3.5GeV , Scaling前后重建效率,e-样本10w个

if ($E < 2.0$) scaling = 0.5;

else if ($E < 3.0$) scaling = 1.7;

else scaling = 2.5;



代码功能添加 scaling

RecChargedTool.cc:

代码功能添加：通过replace_hist.cc脚本，用户可以自由修改root文件中任一TH2D的scale值，其他不变

```
if (rule_id == 0) {  
    if (x_center < 1.0) return 1.0;  
    else if (x_center < 2.0) return 5.0;  
    else if (x_center < 3.0) return 1.0;  
    return 5.0;  
}
```

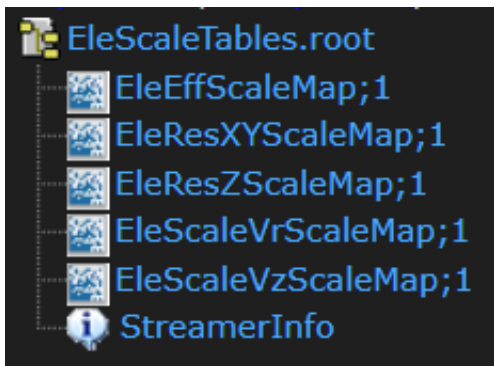
```
if (rule_id == 1) {  
    if (y_center < 0.5) return 1.0;  
    else if (y_center < 1.5) return 5.0;  
    else if (y_center < 2.5) return 1.0;  
    return 5.0;  
}
```

```
const string infile = "EleScaleTables.root";  
const string outfile = "EleScaleTables_modified.root";
```

可设置不同文件路径或选择覆盖原文件

可根据需要添加至任意数量的rule_id
或更改TH2D的scale值

代码功能添加 scaling



```
root [0] .x replace_hist.cc
```

```
in EleScaleTables.root find TH2D:
```

```
EleEffScaleMap
```

```
EleResXYScaleMap
```

```
EleResZScaleMap
```

```
EleScaleVrScaleMap
```

```
EleScaleVzScaleMap
```

```
Enter the TH2D to modify along with your rule numbers(e.g. EleEffScaleMap 0 GamEffScaleMap 1)
```

```
> EleEffScaleMap 0 EleResXYScaleMap 1
```

```
Modified: EleEffScaleMap (rules No.0)
```

```
Modified: EleResXYScaleMap (rules No.1)
```

```
Success, saved to EleScaleTables_modified.root
```

代码功能添加 scaling

RecChargedTool.cc:

```
if (rule_id == 0) {  
    if (x_center < 1.0) return 1.0;  
    else if (x_center < 2.0) return 5.0;  
    else if (x_center < 3.0) return 1.0;  
    return 5.0;  
}
```

```
if (rule_id == 1) {  
    if (y_center < -0.6) return 1.0;  
    else if (y_center < 0) return 5.0;  
    else if (y_center < 0.6) return 1.0;  
    return 5.0;  
}
```

