E70 meeting

2020/07/14

Ohashi

Beam Line観測値再現

- 前回まで、BL観測値を右のように作っていた
- LocalTrackのFitを移植したので以下に変更
  - 実験データでLocalTrackとBC layerの交点をとる（想定真値）
  - 交点にN(0, 200 um)のノイズを乗せる
  - その点をヒットポイントとみなし運動量δ^oを算出
- また、$\delta x_{BFT} = 0.18$ mmとした。（実験時の使用値に揃えた）

Beam x observables: X^o_{out}

- Calculate $x_{3(,4)}$
  : the cross points of Local Track and BC3-x-1 (, BC4-x-1)
- $x^o_{3(,4)} = \mathrm{x}_{3(,4)} + \mathrm{N}(0, \delta x_{BC3(,4)})$
  - $\delta x_{BC3(,4)} = 200\mu m$
- $u^o = u \times (x^o_3 - x^o_4)/(x_3 - x_4)$
- $x^o = x^o_3 - u^o \times z_3$
- $X^o_{out} = (x^o, y, u^o, v)$

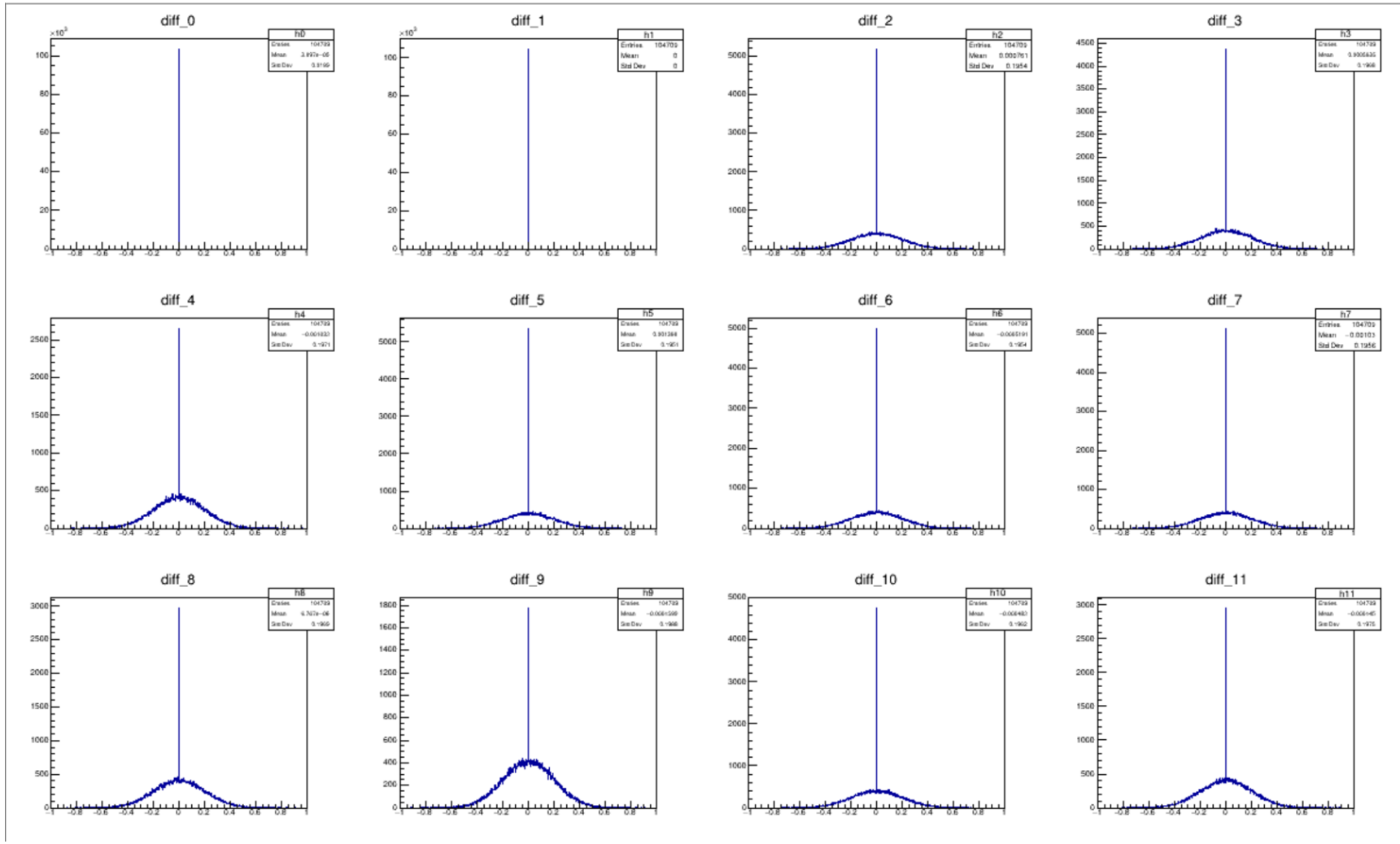
作成した観測データの整合性
LocalTrack Hitの差分：妥当

- BC3, 4の12面分の標準偏差

```
0 : 0.01989893384569642
1 : 0.0
2 : 0.1953683244598564
3 : 0.19681689569515945
4 : 0.1971207457016245
5 : 0.1950943539263969
6 : 0.19541549063441097
7 : 0.19561910482735112
8 : 0.19691879737582005
9 : 0.19877446467477394
10 : 0.19615892512168226
11 : 0.1975311592312629
```

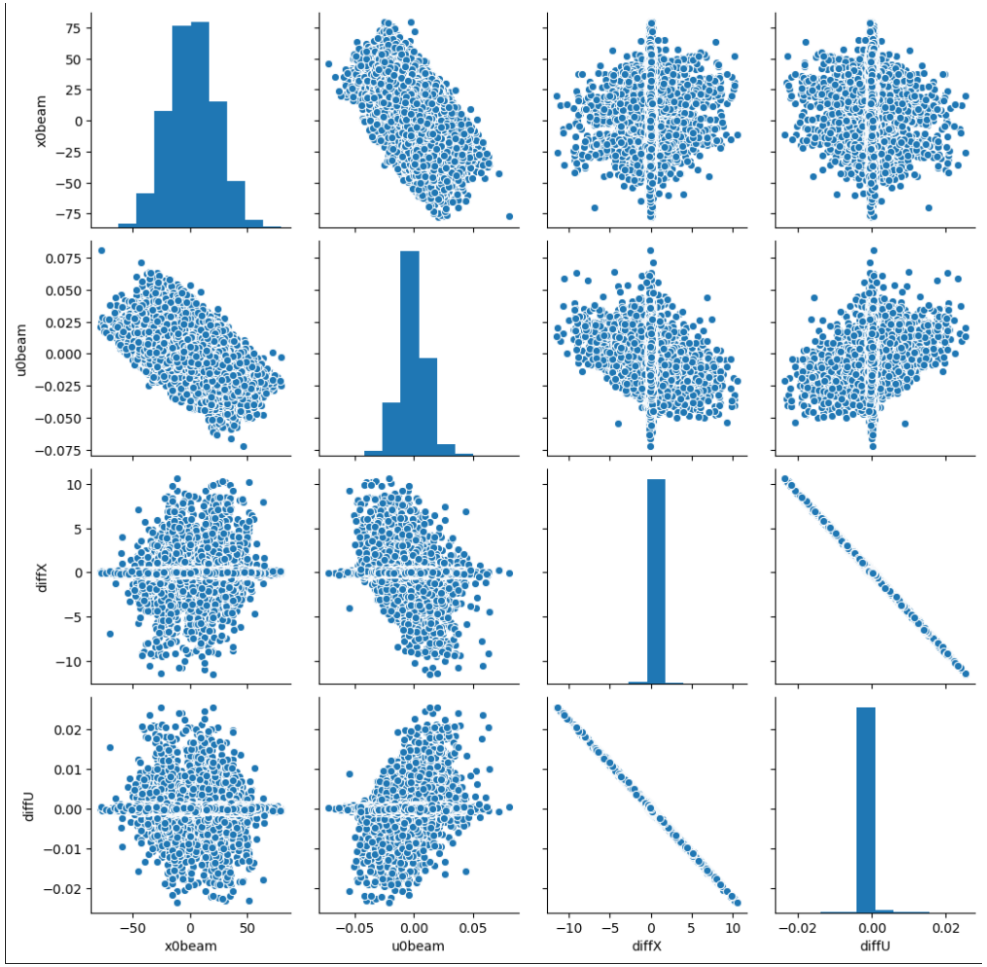
- 差が0のものはそのイベントはHitはなし
- Hitのあるイベント数

```
0 : 1030.0
1 : 0.0
2 : 99925.0
3 : 100737.0
4 : 102422.0
5 : 99758.0
6 : 100132.0
7 : 99955.0
8 : 102150.0
9 : 103339.0
10 : 100354.0
11 : 102169.0
```



作成した観測データの整合性
FPでのXYUV差分：妥当性に疑問

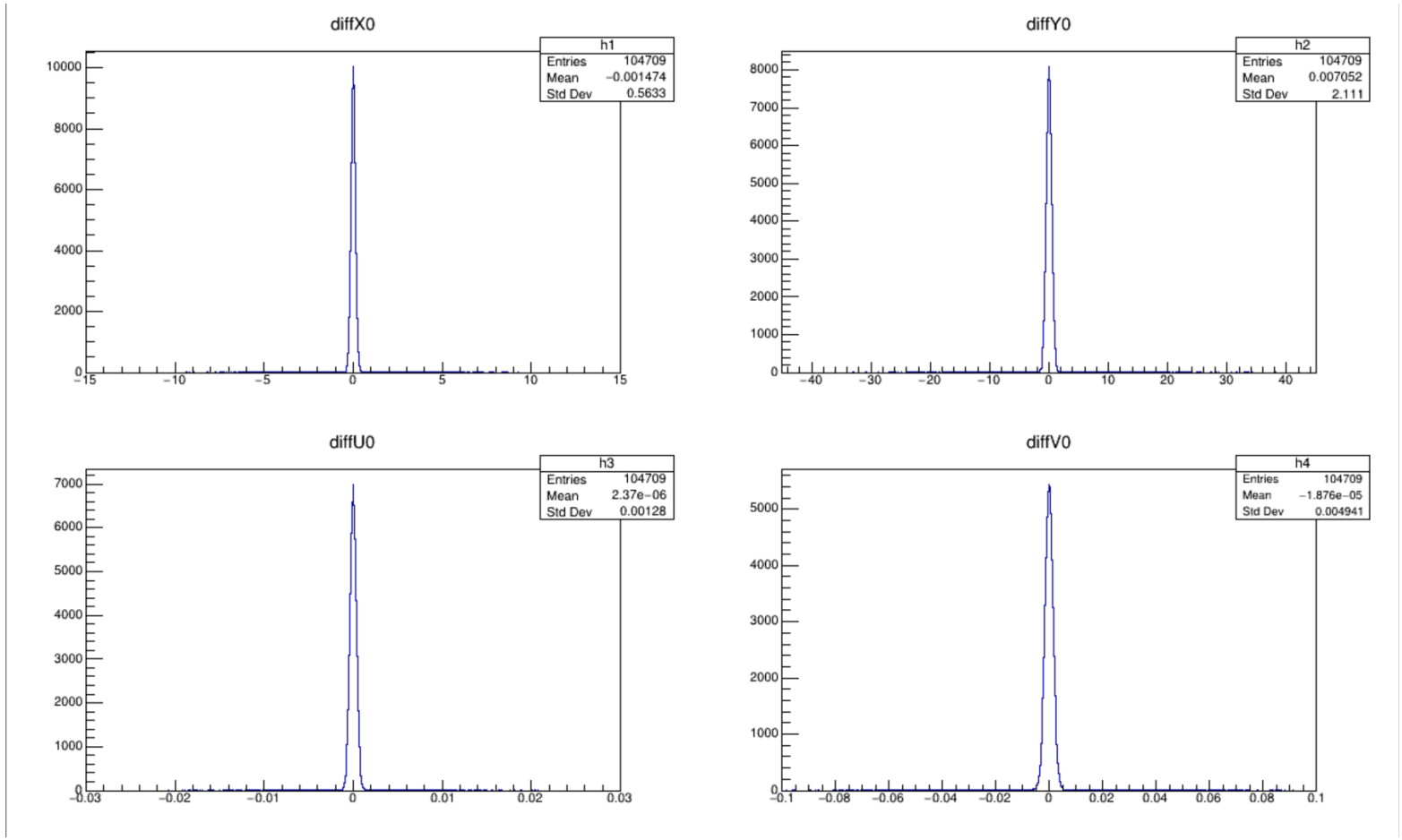
- LocalTrackのFPでの差分
- BC3, 4間の距離 ~ 400 mm
- FP, BC3間の距離 ~ 20 mm
- FPのXが10 mmも差が生じるのは妥当？
- 傾きU~0.02で、BC3,4 の中間点~200 mmを固定すればdX~4 mm
- 1桁大きい気がする




```
>>> df['diffX0'].describe()
count    104709.000000
mean         -0.001474
std           0.563304
min         -11.525861
25%          -0.084433
50%           0.000773
75%           0.084478
max          10.606711
Name: diffX0, dtype: float64
>>> df['diffY0'].describe()
count    104709.000000
mean          0.007052
std           2.110722
min         -40.395087
25%          -0.317883
50%           0.000093
75%           0.316525
max          43.258930
Name: diffY0, dtype: float64
>>> df['diffU0'].describe()
count    1.047090e+05
mean     2.369504e-06
std      1.280089e-03
min     -2.359637e-02
25%     -2.475941e-04
50%     -9.933496e-07
75%      2.439973e-04
max      2.544358e-02
Name: diffU0, dtype: float64
>>> df['diffV0'].describe()
count    1.047090e+05
mean    -1.876002e-05
std      4.941176e-03
min     -9.836490e-02
25%     -1.040727e-03
50%      2.672487e-08
75%      1.034657e-03
max      9.127915e-02
Name: diffV0, dtype: float64
>>>
```

作成した観測データの整合性 FPでのXYUV差分

- FPでの、元の実験データとの座標差分

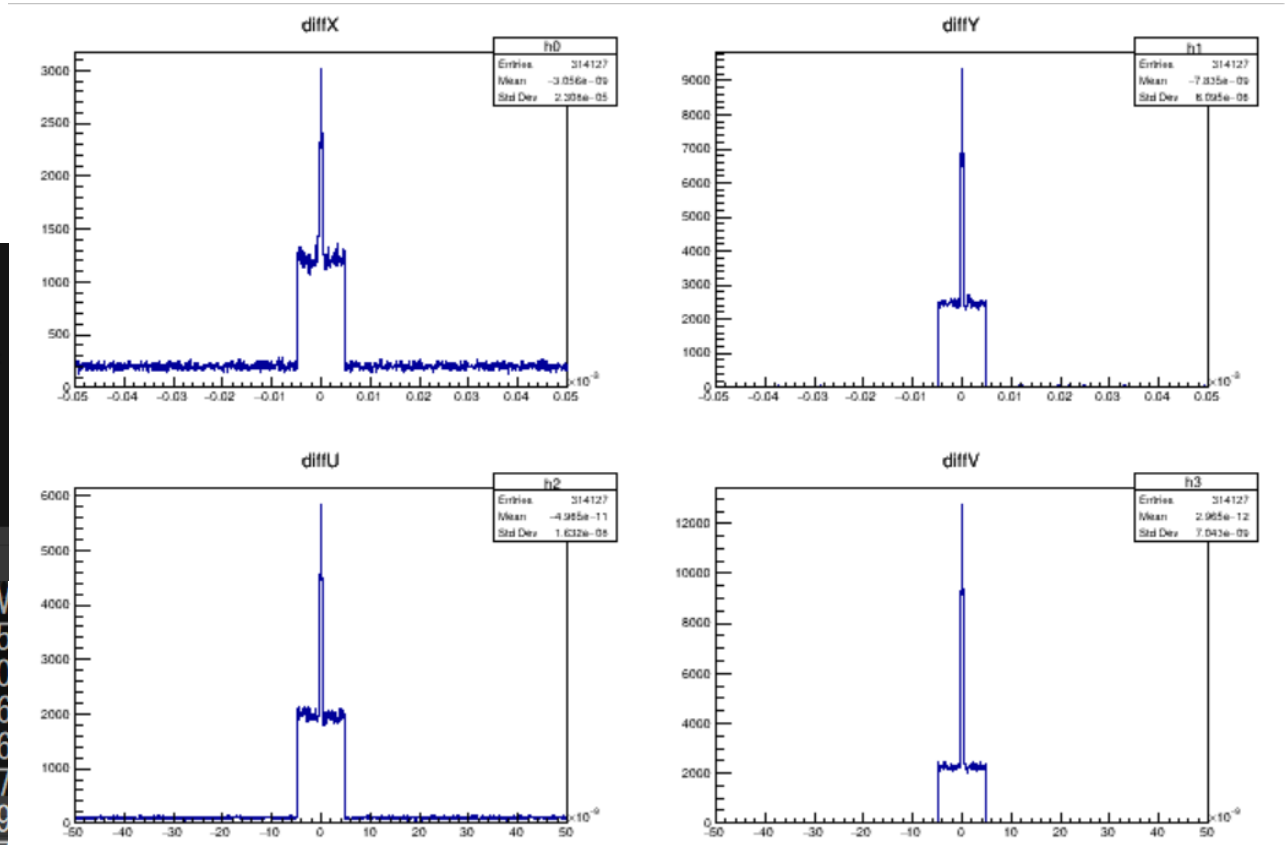


作成した観測データの整合性
ノイズ入れずに同じ解析をした際の
FPでのXYUV差分

- FPのXYUV算出が誤っているか？
- ノイズなしで同じ算出
- 差分（上）と、元の値で割ったもの（下）
  - 計算誤差（？）

	diffX	diffY	diffU	diffV
count	1.047090e+05	1.047090e+05	1.047090e+05	1.047090e+05
mean	-3.056000e-09	-7.835341e-09	-4.965082e-11	2.965352e-12
std	2.308115e-05	8.094788e-06	1.632113e-08	7.042669e-09
min	-4.999973e-05	-4.999198e-05	-4.999893e-08	-4.999980e-08
25%	-1.082455e-05	-2.222408e-06	-3.310069e-09	-1.836764e-09
50%	-1.089214e-08	4.108069e-10	-1.443136e-11	2.321402e-13
75%	1.076285e-05	2.211495e-06	3.228883e-09	1.829732e-09
max	4.999817e-05	4.999128e-05	4.999980e-08	4.997269e-08

	diffX/X	diffY/Y	diffU/U	diffV/V
count	1.047090e+05	1.047090e+05	1.047090e+05	1.047090e+05
mean	2.364489e-09	4.162893e-09	-2.933520e-10	5.405196e-10
std	1.397150e-06	1.216119e-06	1.385280e-06	1.295665e-06
min	-4.962902e-06	-4.941124e-06	-4.958764e-06	-4.955887e-06
25%	-7.841086e-07	-6.031576e-07	-6.559814e-07	-7.121731e-07
50%	2.917665e-09	4.369935e-09	5.995857e-10	-3.165184e-09
75%	7.894954e-07	6.049102e-07	6.514317e-07	7.150652e-07
max	4.929398e-06	4.962612e-06	4.985919e-06	4.983466e-06

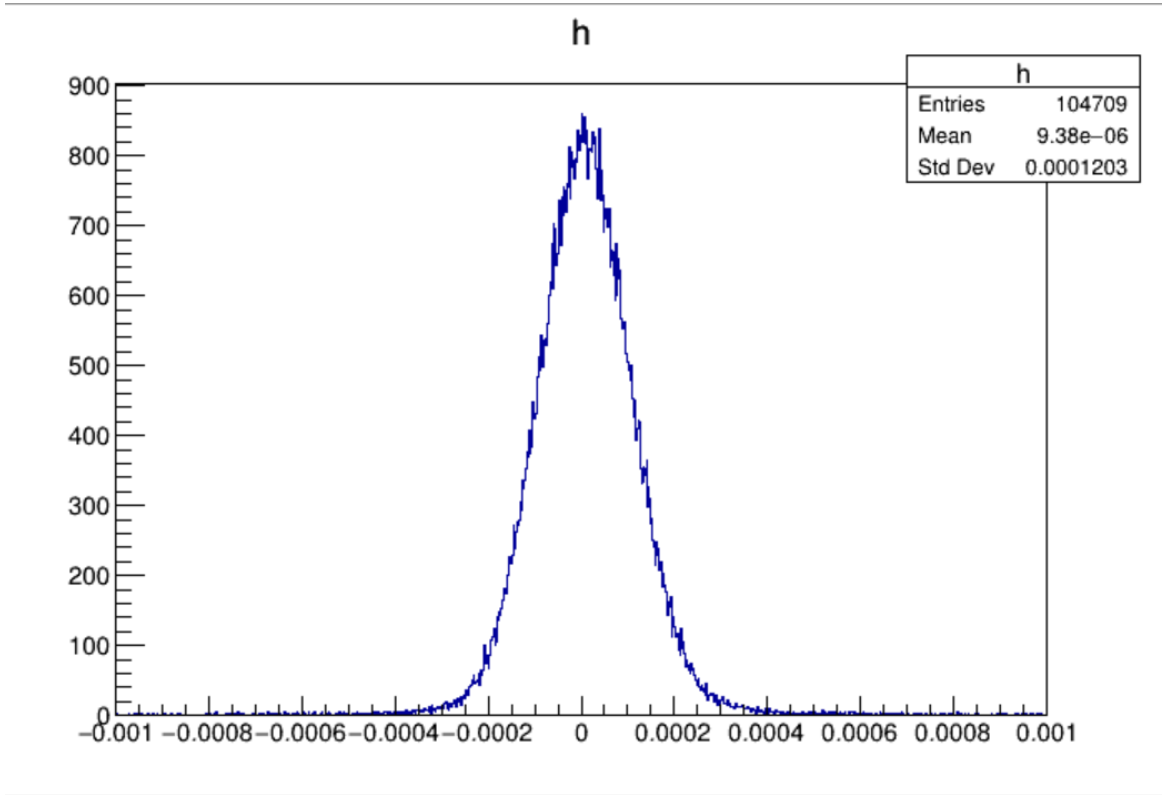


作成した観測値からTMで運動量

- 作成観測値から輸送行列で運動量を算出
- 元の運動量（想定真値）との差分
  - FWHM~2.2×10^{-4}
- 前回までの結果

$\delta x_{BFT}\ [\mu m]$	$\delta^o_{TM} - \delta$	$\delta^o_{lgb} - \delta$
200	3.3×10^{-4}	3.0×10^{-4}

- 差分のFWHMは小さくなっている




```
fwhm(h2)
In [38]: my_function.fwhm(h)
Out[38]: 0.00021799999999999999
```

作成した観測データからTMで運動量差分FWHMが小さくなった考察

- （作成観測値）－（想定真値）
- 前回データ

Out[14]:

	diff_xout	diff_yout	diff_uout	diff_vout	diff_xin	diff_yin	diff_uin	diff_vin
count	104709.000000	104709.0	104709.0	104709.0	104709.000000	104709.0	1.047090e+05	104709.0
mean	-0.000518	0.0	0.0	0.0	0.000332	0.0	5.925436e-07	0.0
std	0.200710	0.0	0.0	0.0	0.209058	0.0	6.427335e-04	0.0
min	-0.897600	0.0	0.0	0.0	-0.884660	0.0	-2.963702e-03	0.0
25%	-0.136000	0.0	0.0	0.0	-0.140900	0.0	-4.324850e-04	0.0
50%	0.000200	0.0	0.0	0.0	-0.000200	0.0	2.380000e-06	0.0
75%	0.135300	0.0	0.0	0.0	0.140900	0.0	4.325000e-04	0.0
max	1.018810	0.0	0.0	0.0	0.829270	0.0	2.712800e-03	0.0

- 今回データ

In [5]: df[df.columns[df.columns.str.contains('diff')]].describe()

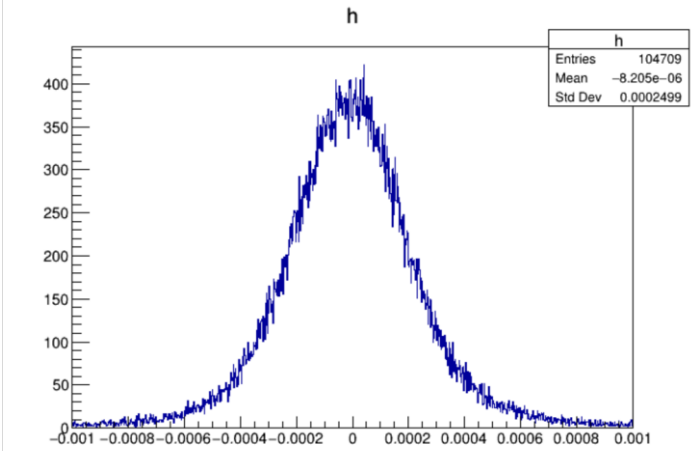
Out[5]:

	diff_bft_o	diff_xin_o	diff_yin_o	diff_uin_o	diff_vin_o
count	104709.000000	104709.000000	104709.000000	1.047090e+05	1.047090e+05
mean	0.000005	-0.001474	0.007052	2.369553e-06	-1.876002e-05
std	0.179769	0.563304	2.110722	1.280089e-03	4.941176e-03
min	-0.750300	-11.525900	-40.395100	-2.359637e-02	-9.836490e-02
25%	-0.121700	-0.084400	-0.317880	-2.476000e-04	-1.040726e-03
50%	-0.000470	0.000800	0.000093	-1.000000e-06	3.000000e-08
75%	0.121420	0.084500	0.316520	2.440000e-04	1.034655e-03
max	0.848200	10.606700	43.258900	2.544358e-02	9.127920e-02

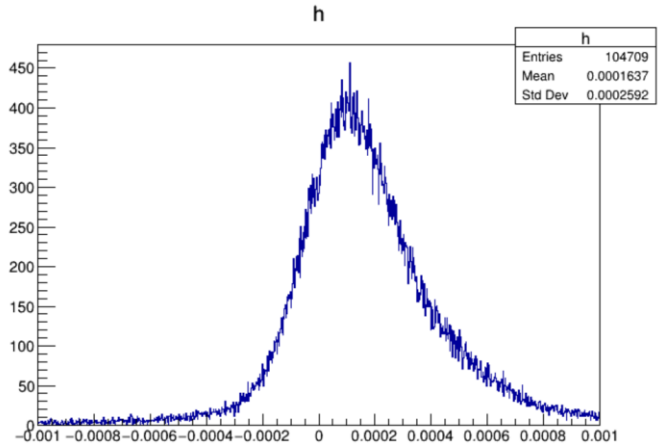
- Xinの差分は四分位範囲は小さくなっている
  - →運動量の差分は小さくなる

作成観測値から算出した運動量 lgb, TM比較

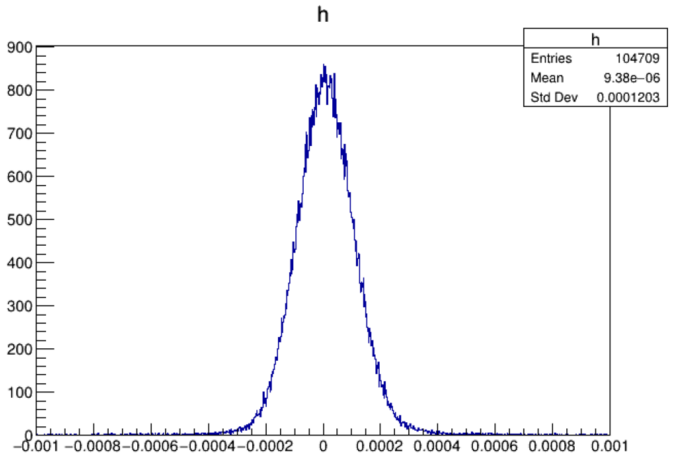
- 左上：lgb
  - 想定真値とのずれ
  - FWHM ~ 4.3×10^{-4}
- 右上：DNN
  - 3.8×10^{-4}
- 右下：TM
  - 2.2×10^{-4}




```
In [22]: fwhm(h)
Out[22]: 0.00042599999999999984
```




```
In [29]: fwhm(h)
Out[29]: 0.00037799999999999997
```




```
In [38]: my_function.fwhm(h)
Out[38]: 0.00021799999999999999
```

前回までの結果

$\delta x_{BFT}\ [\mu m]$	$\delta^o_{TM} - \delta$	$\delta^o_{lgb} - \delta$
200	3.3×10^{-4}	3.0×10^{-4}

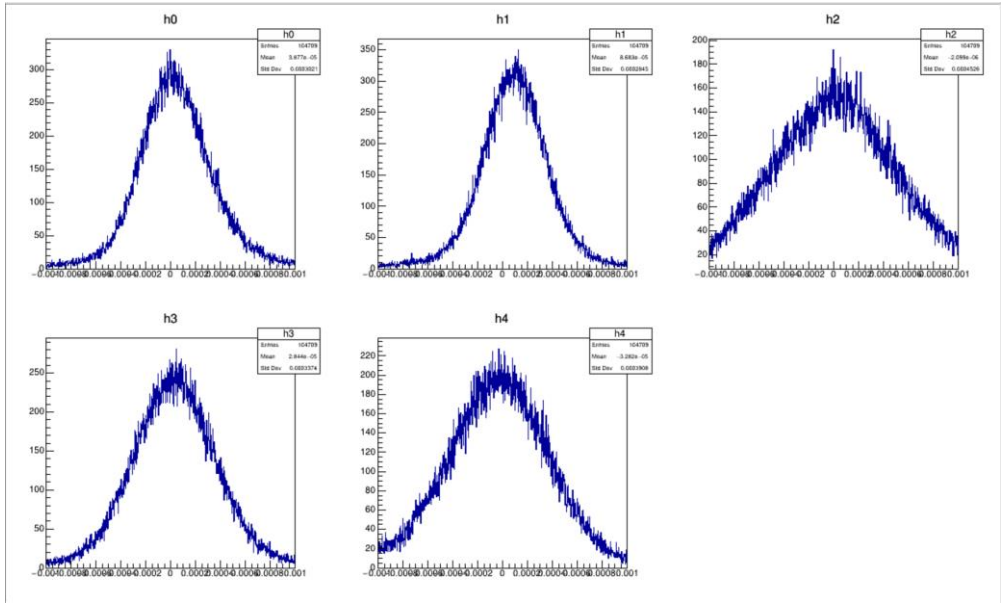
Model詳細

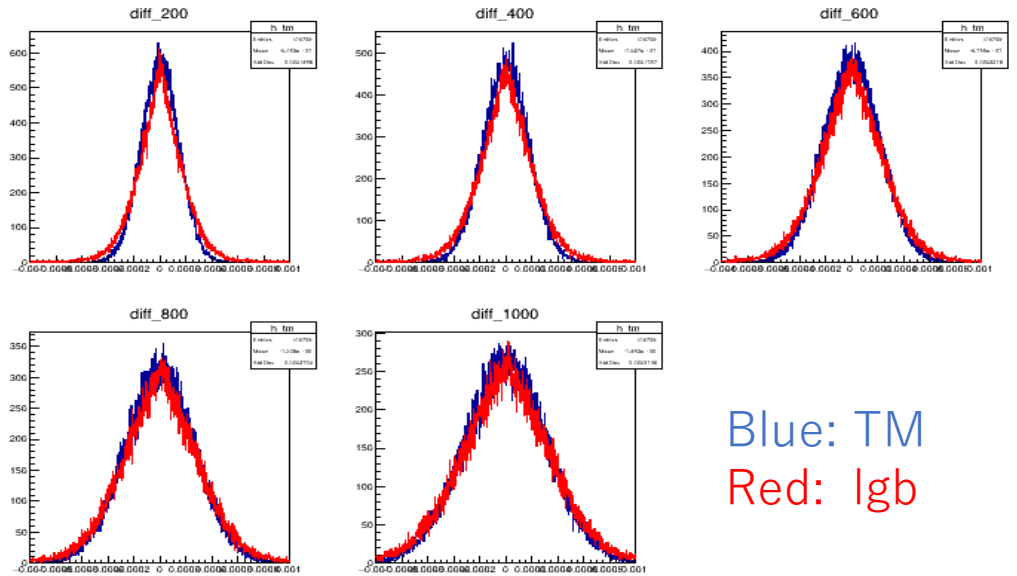
- Lgb
  - ハイパーパラメータ：デフォルト
  - Num_round = 100000, Early_stopping_round = 100

- DNN
  - 隠れ層：4, ユニット数：64[/1隠れ層], バッチサイズ：128
  - Early_stopping_round = 100

Back up
以前のデータ（今回以降不使用、参考程度）

- 以前までのデータによる差分に、DNNでの結果を加えたもの

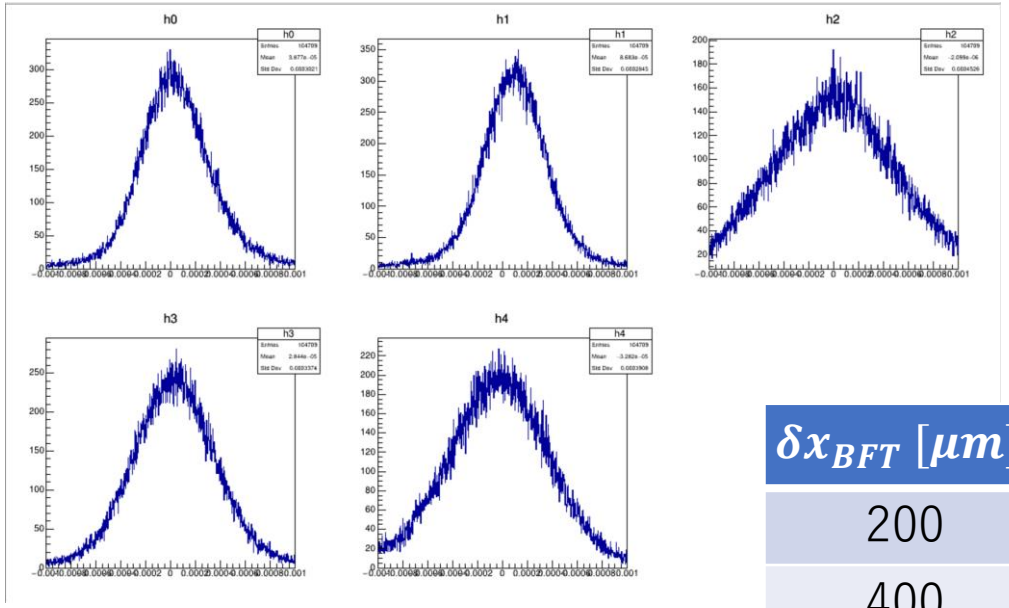


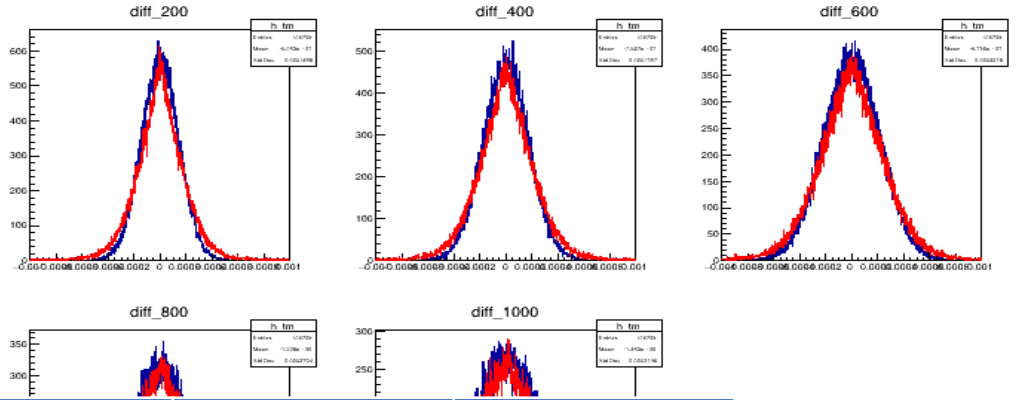


Blue: TM
Red: lgb

Back up
以前のデータ（今回以降不使用、参考程度）

- 以前までのデータによる差分に、DNNでの結果を加えたもの





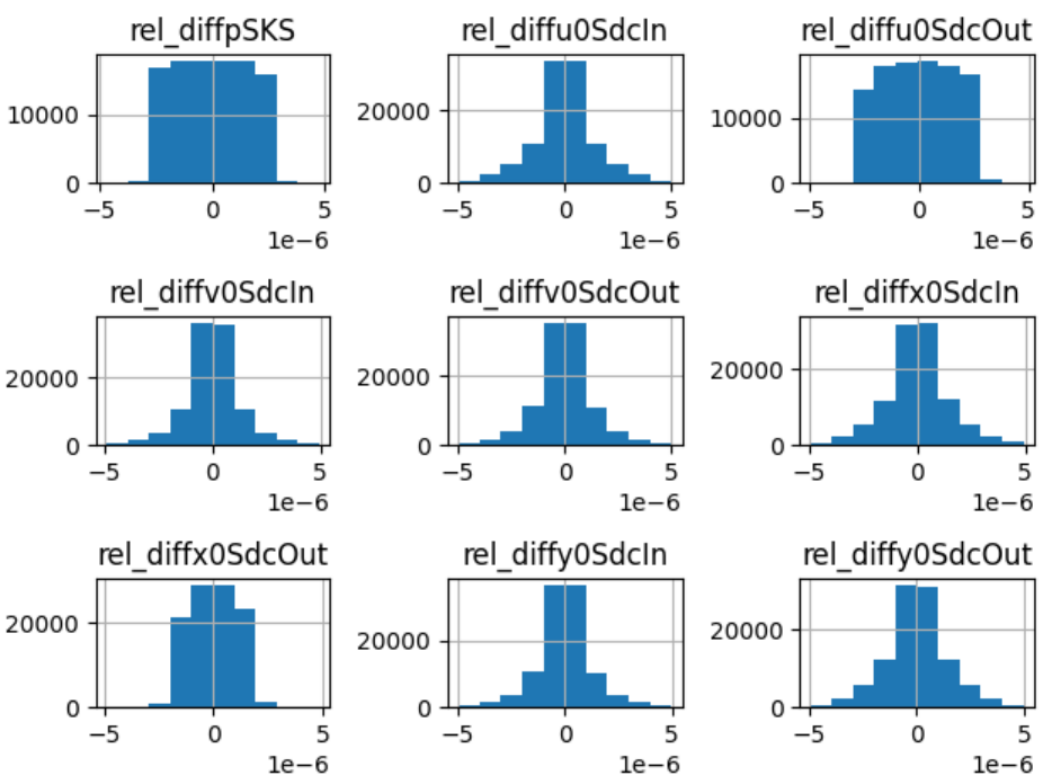
δx_{BFT} $[\mu m]$	$\delta^o_{TM} - \delta$	$\delta^o_{lgb} - \delta$	$\delta^o_{DNN} - \delta$
200	3.3×10^{-4}	3.0×10^{-4}	6.1×10^{-4}
400	4.0×10^{-4}	3.9×10^{-4}	5.4×10^{-4}
600	5.0×10^{-4}	5.3×10^{-4}	1.1×10^{-3}
800	6.3×10^{-4}	6.0×10^{-4}	7.4×10^{-4}
1000	7.5×10^{-4}	7.4×10^{-4}	9.1×10^{-4}

Blue: TM
Red: lgb

学習がうまくいっていない
乱数Seed,
early_stopping, …

SKS観測値作成

- BL時のHit -> LocalTrackを適用
- その後、RKで運動量を算出
- 整合性評価
  - 元の実験データとの相対誤差
  - $\sim 10^{-6}$
  - BLの時のLocalTrackFitが誤っていればこの値も誤り
  - 妥当性に疑問



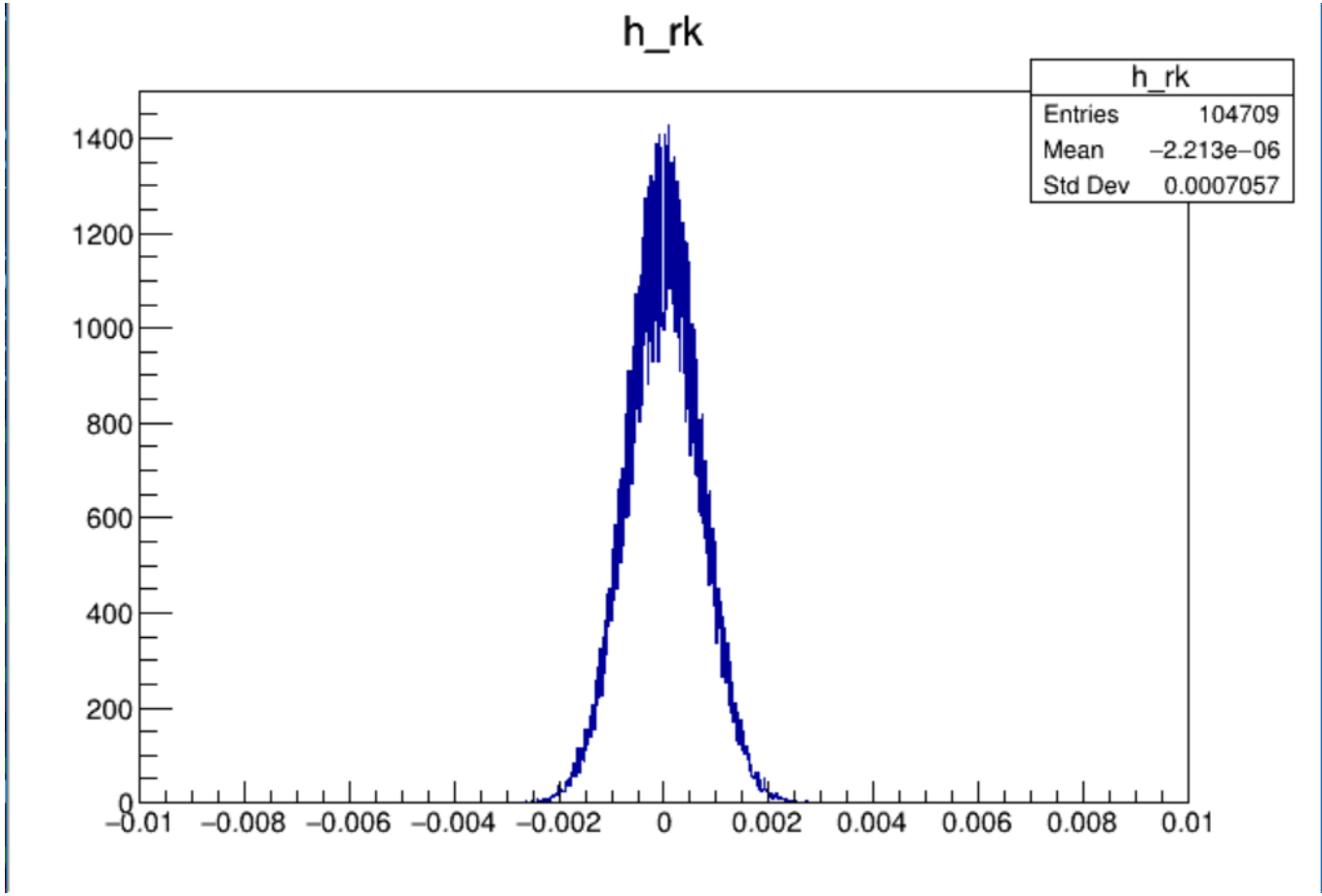

```
In [26]: df_no['rel_diffpSKS'].describe()
Out[26]:
count    1.047090e+05
mean    -1.322110e-09
std      1.605595e-06
min     -4.706402e-06
25%     -1.388671e-06
50%      1.932343e-09
75%      1.390526e-06
max      4.762128e-06
Name: rel_diffpSKS, dtype: float64
```

	rel_diffx0SdcIn	rel_diffy0SdcIn	rel_diffu0SdcIn	rel_diffv0SdcIn	rel_diffx0SdcOut	rel_diffy0SdcOut	rel_diffu0SdcOut	rel_diffv0SdcOut
count	1.047090e+05	1.047090e+05	1.047090e+05	1.047090e+05	1.047090e+05	1.047090e+05	1.047090e+05	1.047090e+05
mean	3.607413e-09	-2.394654e-09	-2.281841e-10	-4.892138e-09	-3.293763e-09	-3.290209e-09	-4.525516e-09	-4.274071e-09
std	1.417734e-06	1.186156e-06	1.393349e-06	1.212362e-06	1.056720e-06	1.422323e-06	1.608797e-06	1.236880e-06
min	-4.957777e-06	-4.965781e-06	-4.985126e-06	-4.922420e-06	-4.895781e-06	-4.953208e-06	-4.934446e-06	-4.982847e-06
25%	-7.174408e-07	-6.132958e-07	-6.570602e-07	-6.247506e-07	-8.806902e-07	-7.406175e-07	-1.381947e-06	-6.432501e-07
50%	3.224122e-10	-1.370903e-09	-2.184181e-09	-4.694294e-09	-1.155075e-09	1.285341e-09	-3.743217e-09	-2.456265e-09
75%	7.225451e-07	6.103104e-07	6.584320e-07	6.185406e-07	8.719414e-07	7.425816e-07	1.365897e-06	6.366565e-07
max	4.919489e-06	4.969143e-06	4.988634e-06	4.933790e-06	4.817023e-06	4.973248e-06	4.770133e-06	4.986412e-06

$\delta^o_{RK} - \delta^t_{RK}$

- FWHM~1.5×10^{-3}
- 大きな外れ値が気になる

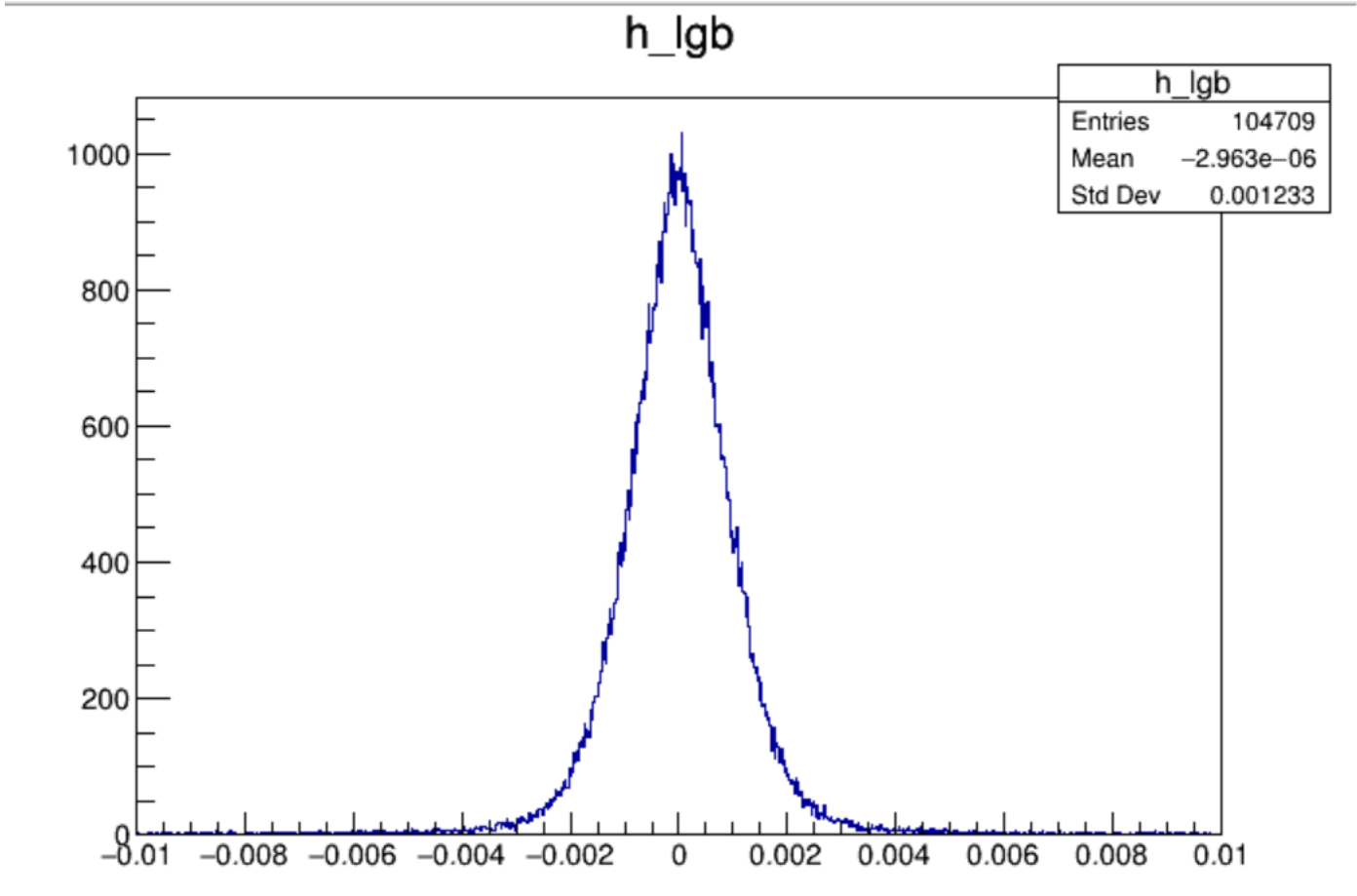
```
In [10]: df_rk['diff'].describe()
Out[10]:
count    104709.000000
mean         -8.309372
std         288.152263
min      -10000.919120
25%          -0.000480
50%           0.000000
75%           0.000470
max           0.003330
Name: diff, dtype: float64
```



$\delta^{o}_{lgb} - \delta^{t}_{RK}$

- FWHM~1.7×10^{-3}
- 汎用のtunerでパラメータチューニング済み

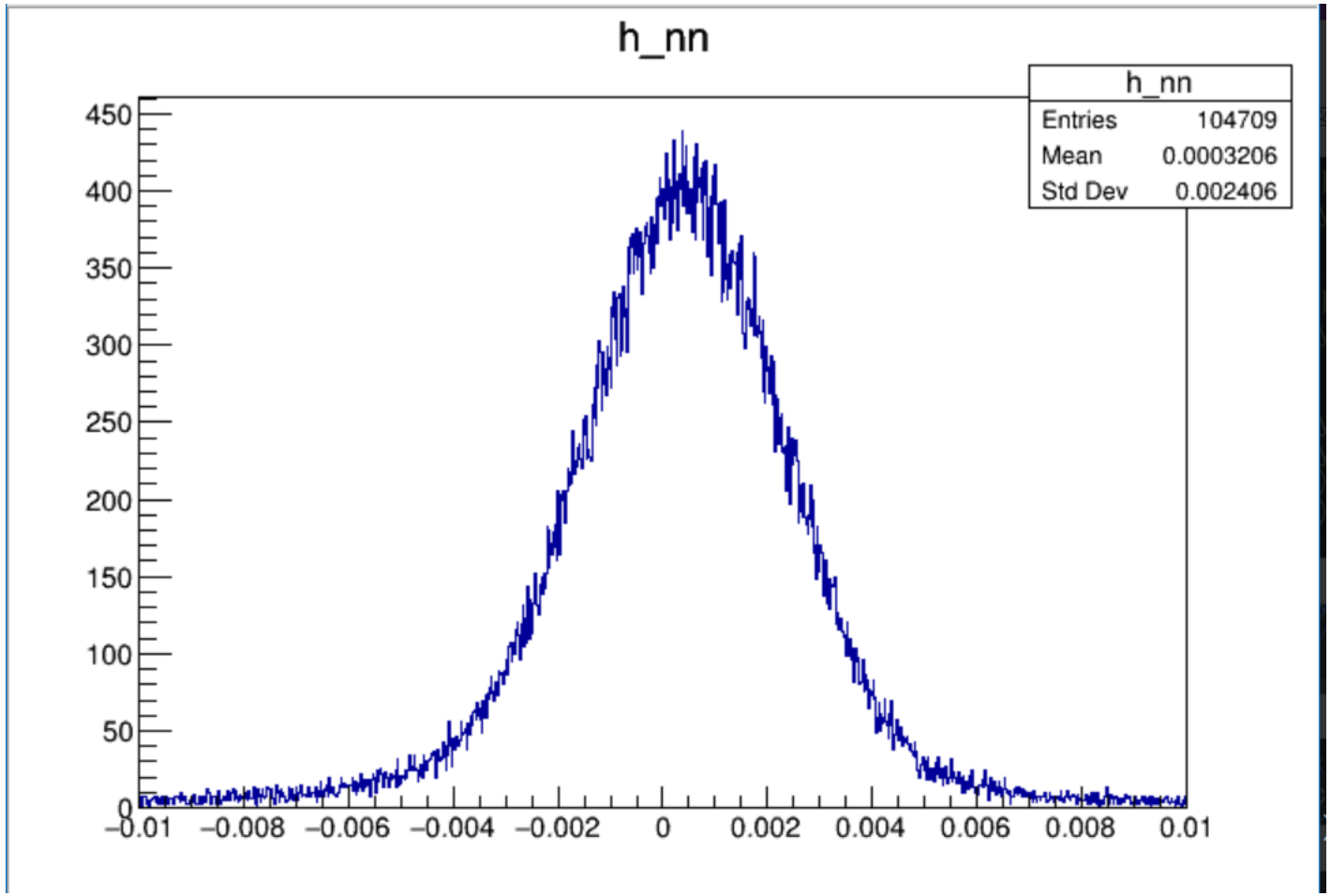
```
In [11]: df_lgb['diff'].describe()
Out[11]:
count    104709.000000
mean          0.000009
std           0.002249
min          -0.099406
25%          -0.000615
50%          -0.000002
75%           0.000615
max           0.112238
Name: diff, dtype: float64
```



$\delta^o_{NN} - \delta^t_{RK}$

- FWHM~4.4×10^{-3}
- １通りしか試しておらず、これ以上よくならないとは言い切れない

```
In [34]: df_nn['diff'].describe()
Out[34]:
count    104709.000000
mean          0.000278
std           0.004130
min          -0.088256
25%          -0.001077
50%           0.000375
75%           0.001792
max           0.211759
Name: diff, dtype: float64
```



SKS 3手法比較

Blue: RK
Red: lgb
Green: NN

- RKが最も適している
- NNは学習工夫が残されている

- ノイズを乗せる前のデータをMLが知っている状況は好ましくない
- 現実的使えるデータを考慮すれば、lgbの性能はより下がる