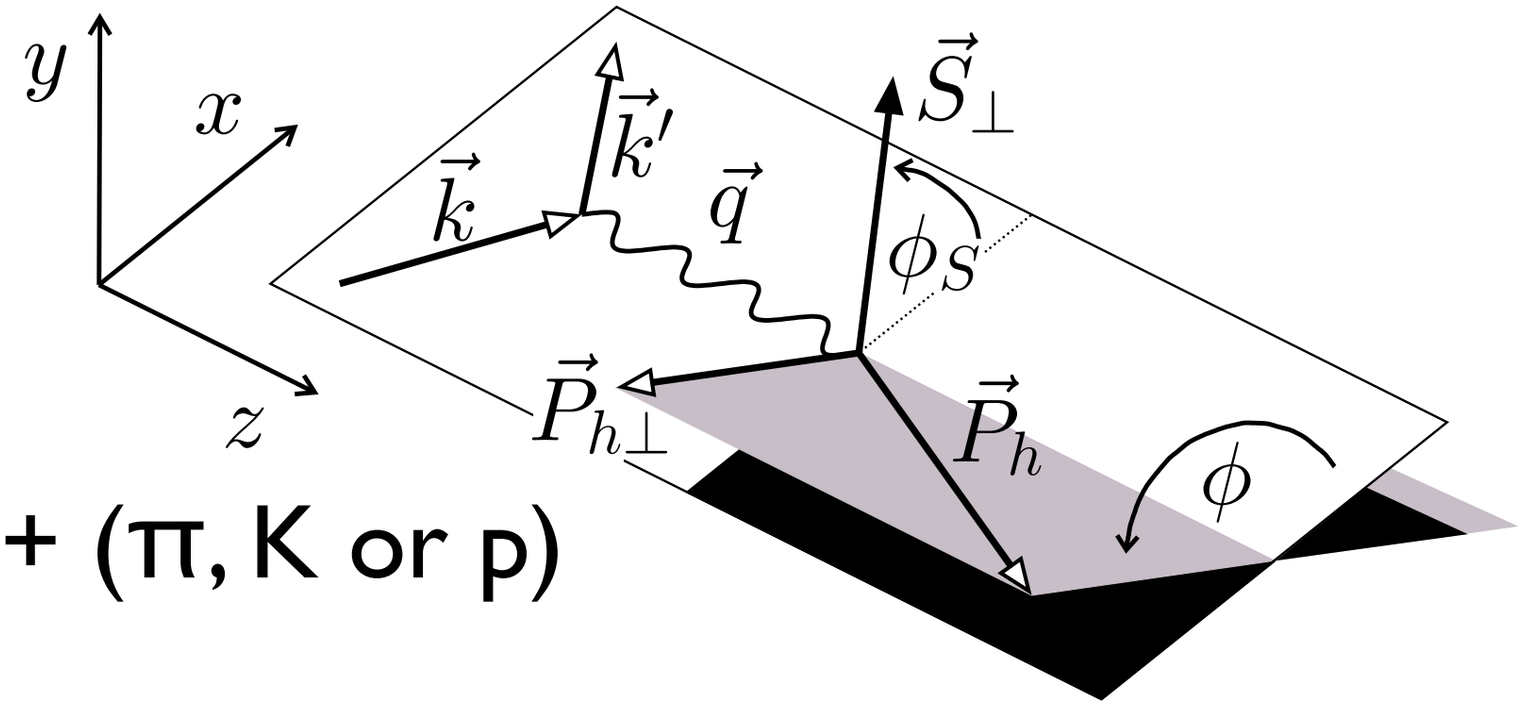
gmc_trans

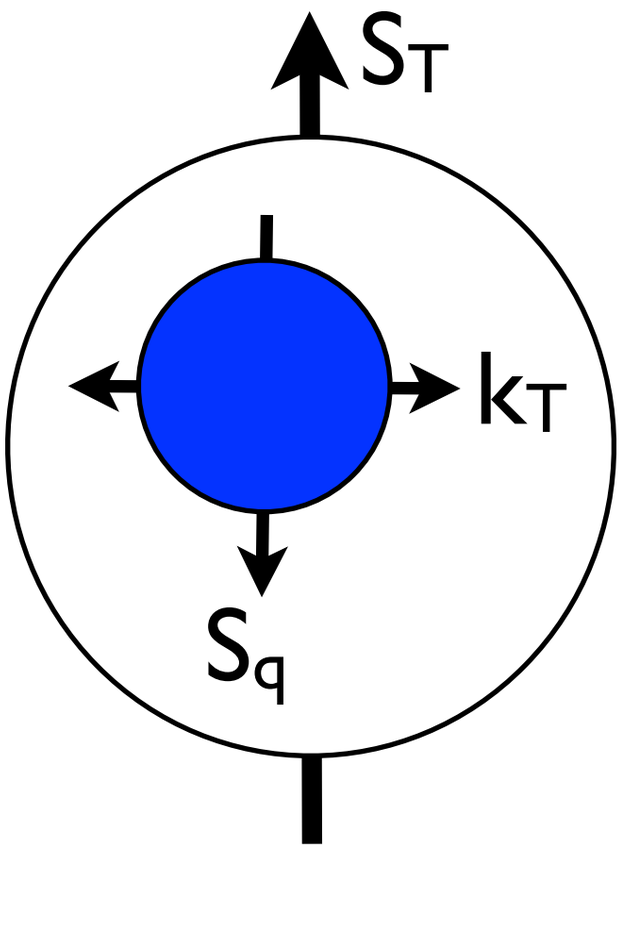
SIDIS transverse spin asymmetry monte carlo

The code



- Fortran
- e + p → e + p + (π, K or p)
- HERMES →
  - Fixed target
  - Different computing infrastructure

Transverse spin asymmetries

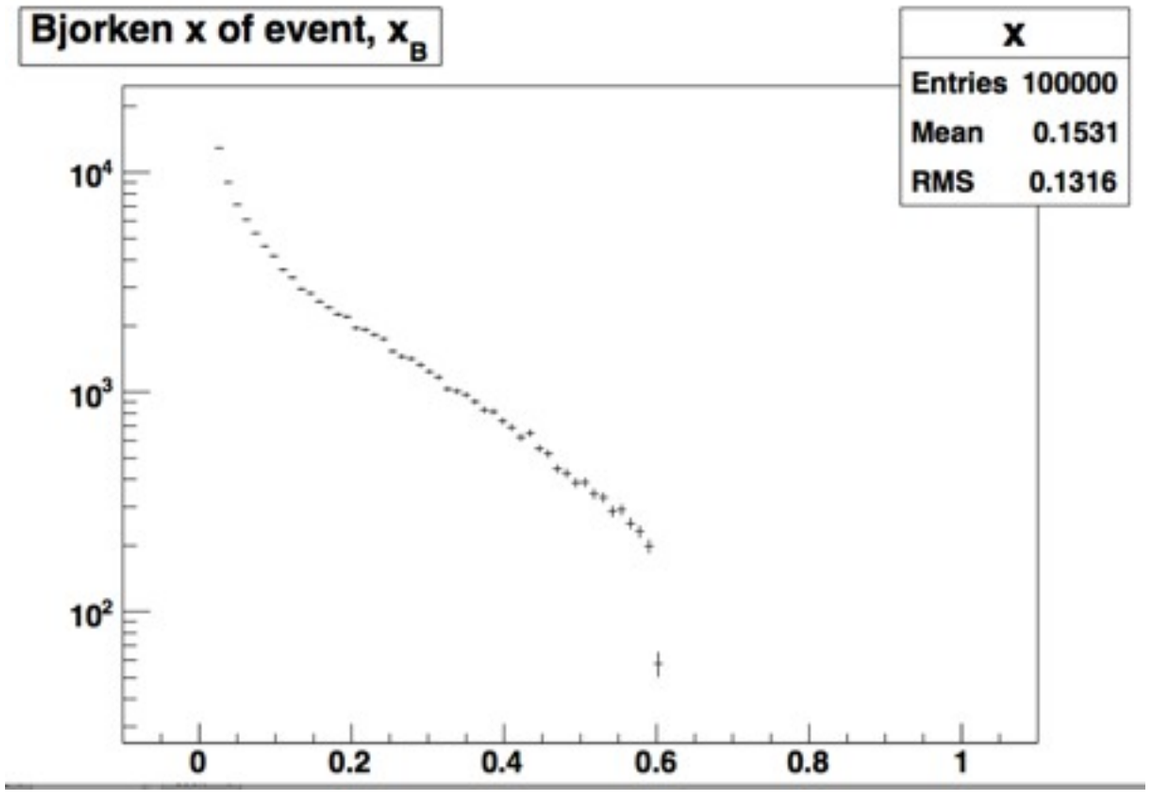


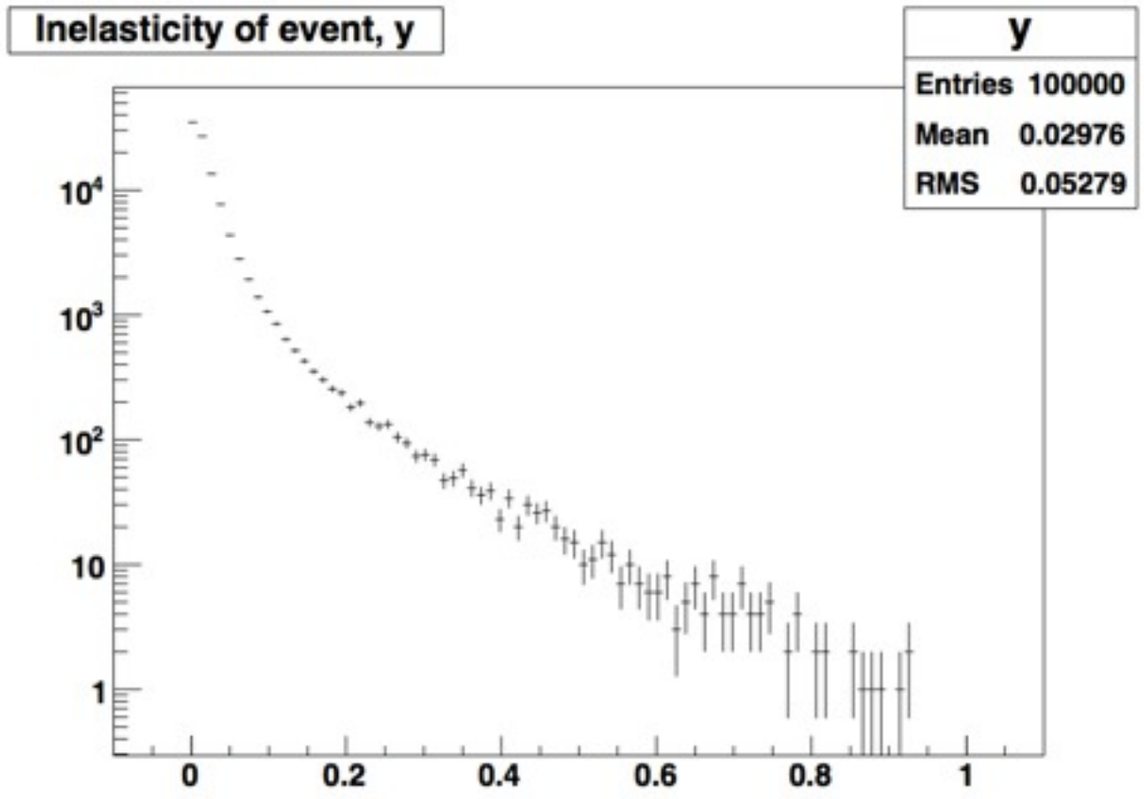
- Sivers: correlation between S_T and k_T
- Collins: correlation between fragmentation and S_q.
- Boer-Mulders: correlation between k_T and S_q
- Use input parameterisations to generate events with asymmetries

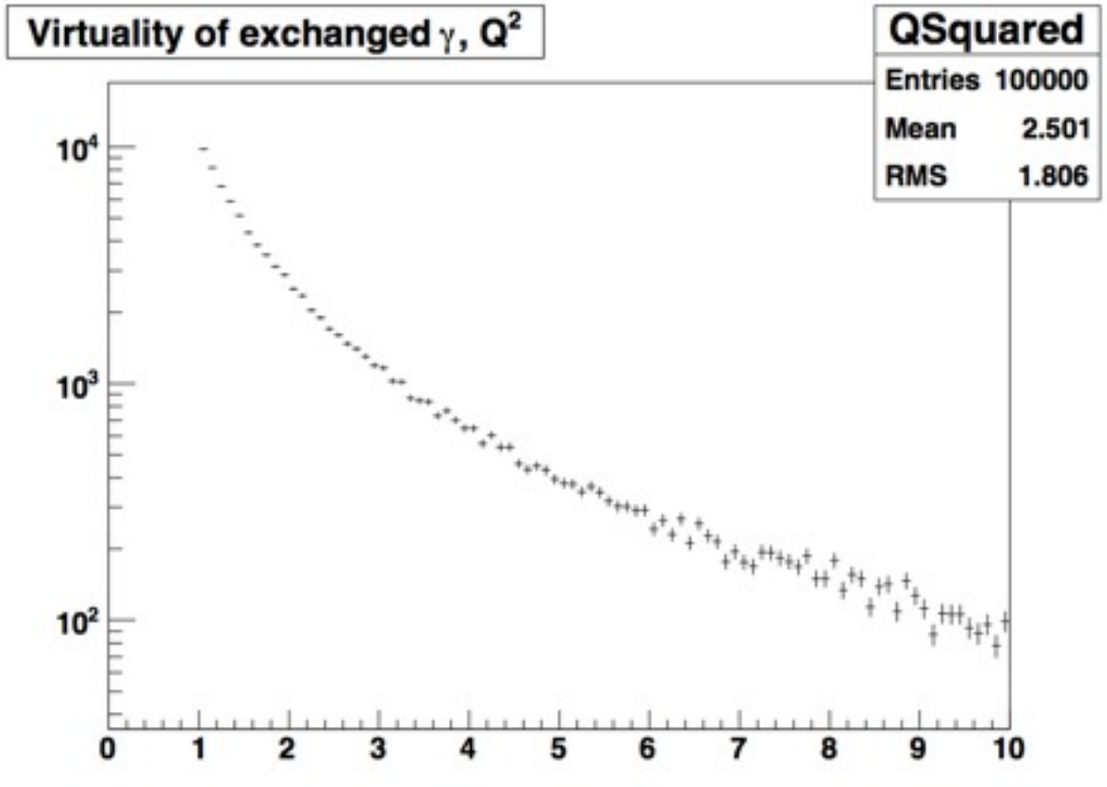
Different parameterisations

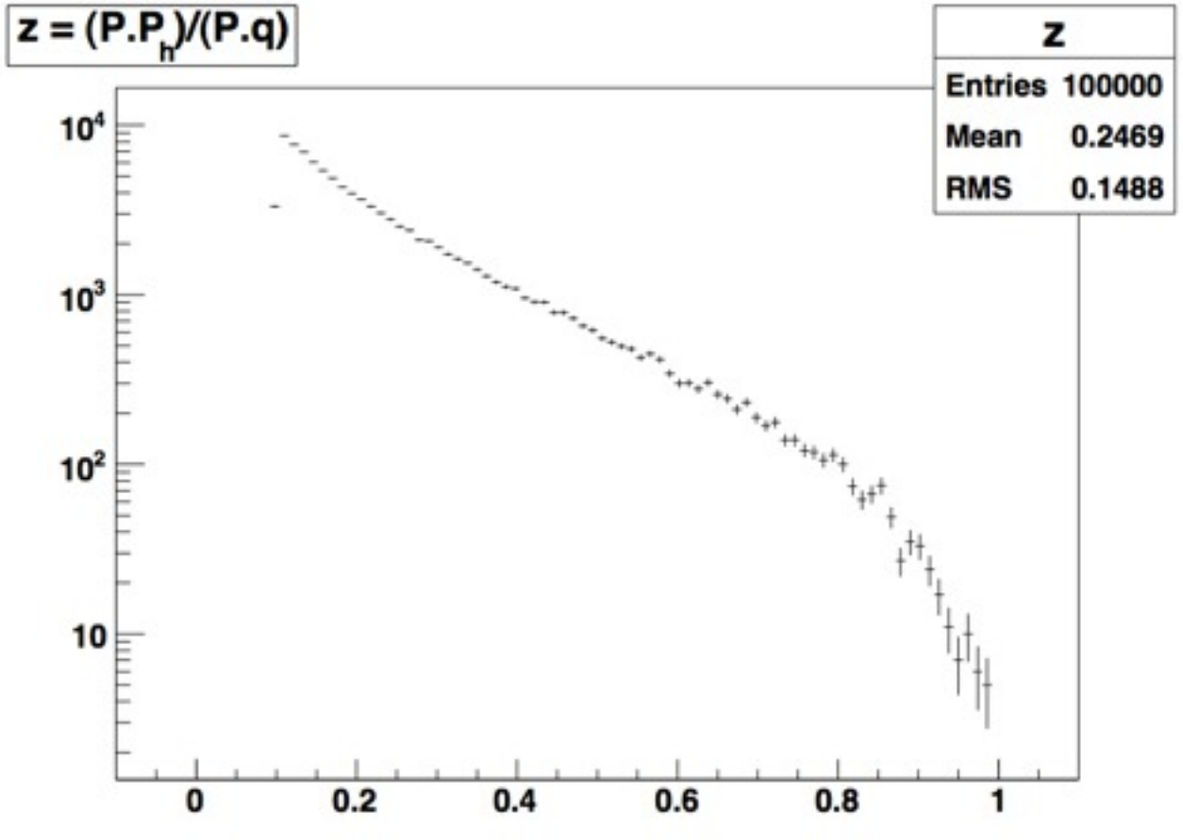
```
CollinsSet     6  INTE  MODEL H1p: 0=0 1=D1 2=ABM 3=Leader 4=Bacc 5:H1p(1)=D1 6:z^Pow*D1 7-
CollinsZPow 1.0   REAL  MODEL H1p: z power for multiplier in CollinsSet 6
CollinsNormFav 0.2   REAL  MODEL H1p: normalization factor for favored Collins FF
CollinsNormDis 0.2   REAL  MODEL H1p: normalization factor for disfav. Collins FF
CollinsUseRatio   F  LOGI  MODEL H1p: don't norm disfav.FF separately, use CollinsDisFac
CollinsDisFac  -1.   REAL  MODEL H1p: norm. factor for  Disfav = Fav * CollinsDisFac
HtildeSet   0  INTE  MODEL H1p: 0=0 1=D1 2:H1p(1)=D1 3:z^Pow*D1
HtildeZPow  1.0   REAL  MODEL H1p: z power for multiplier in HtildeSet 6
HtildeNormFav  1.5   REAL  MODEL H1p: normalization factor for favored Htilde FF
HtildeNormDis  1.5   REAL  MODEL H1p: normalization factor for disfav. Htilde FF
HtildeUseRatio F  LOGI  MODEL H1p: don't norm disfav.FF separately, use HtildeDisFac
HtildeDisFac   -1.   REAL  MODEL H1p: norm. factor for  Disfav = Fav * HtildeDisFac
TransvSet      0  INTE  MODEL h1: 0=0 1=g1 2=SofferBound 3-6=Leader 7-9=Anselmino
TransvNormU 0.4   REAL  MODEL h1: norm factor for h1_u, all sets
TransvNormD 0.4   REAL  MODEL h1: norm factor for h1_d, all sets
TransvNormQbar 0.0   REAL  MODEL h1: norm factor for h1_qbar, all sets
SiversSet   6  INTE  MODEL f1Tp: 0=0 1=f1 2=g1 3=Mulders 4:f1tp(1)=f1 5,6=Anselmino
SiversNormU -.4   REAL  MODEL f1Tp: norm factor for u quark
SiversNormD 0.8   REAL  MODEL f1Tp: norm factor for d quark
SiversNormQbar 0.0   REAL  MODEL f1Tp: norm factor for antiquarks
SiversUseRatio F  LOGI  MODEL f1Tp: don't norm, use Andy ratios
```

Different normalisations

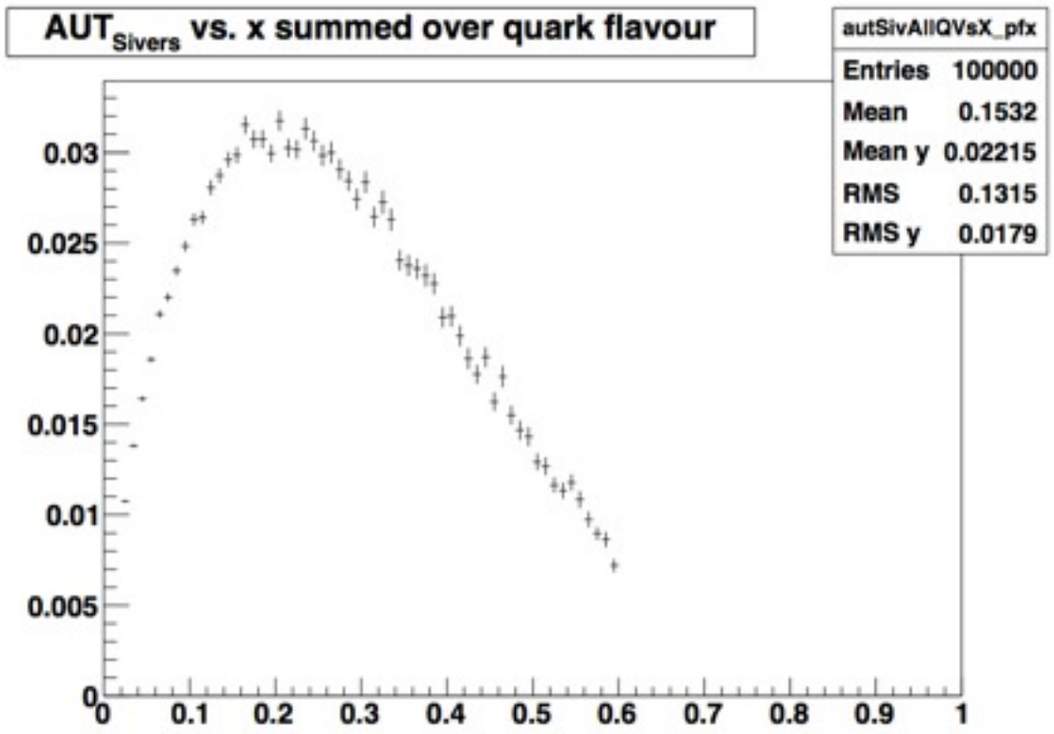

Bjorken x of event, x_B
x
Entries 100000
Mean 0.1531
RMS 0.1316
10^4
10^3
10^2
0
0.2
0.4
0.6
0.8
1

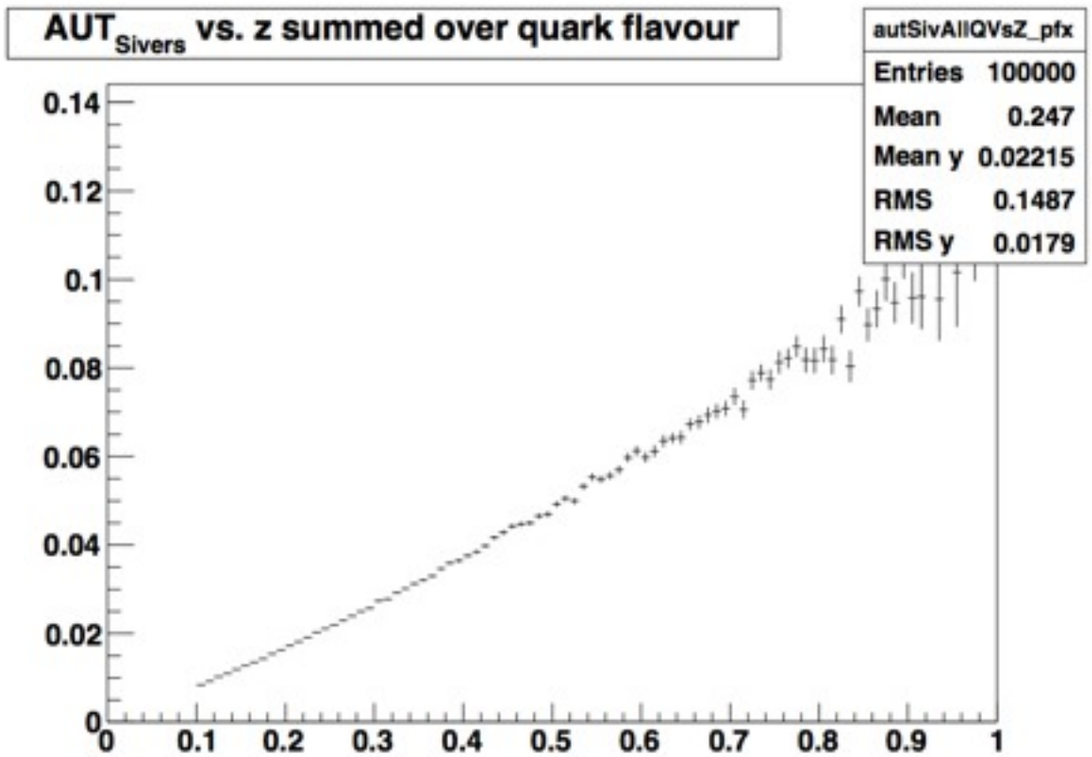

Inelasticity of event, y
y
Entries 100000
Mean 0.02976
RMS 0.05279
10^4
10^3
10^2
10
1
0
0.2
0.4
0.6
0.8
1

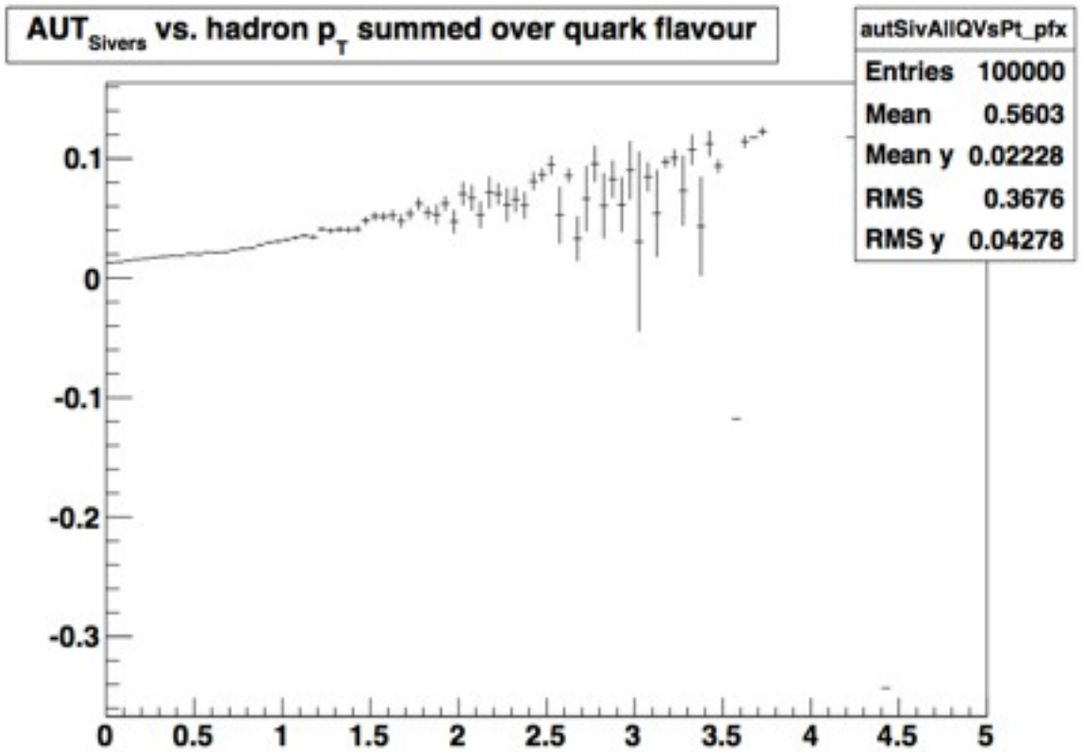

Virtuality of exchanged γ, Q^2
QSquared
Entries 100000
Mean 2.501
RMS 1.806
10^4
10^3
10^2
0
1
2
3
4
5
6
7
8
9
10


z = $(P.P_h)/(P.q)$
z
Entries 100000
Mean 0.2469
RMS 0.1488
10^4
10^3
10^2
10
0
0.2
0.4
0.6
0.8
1

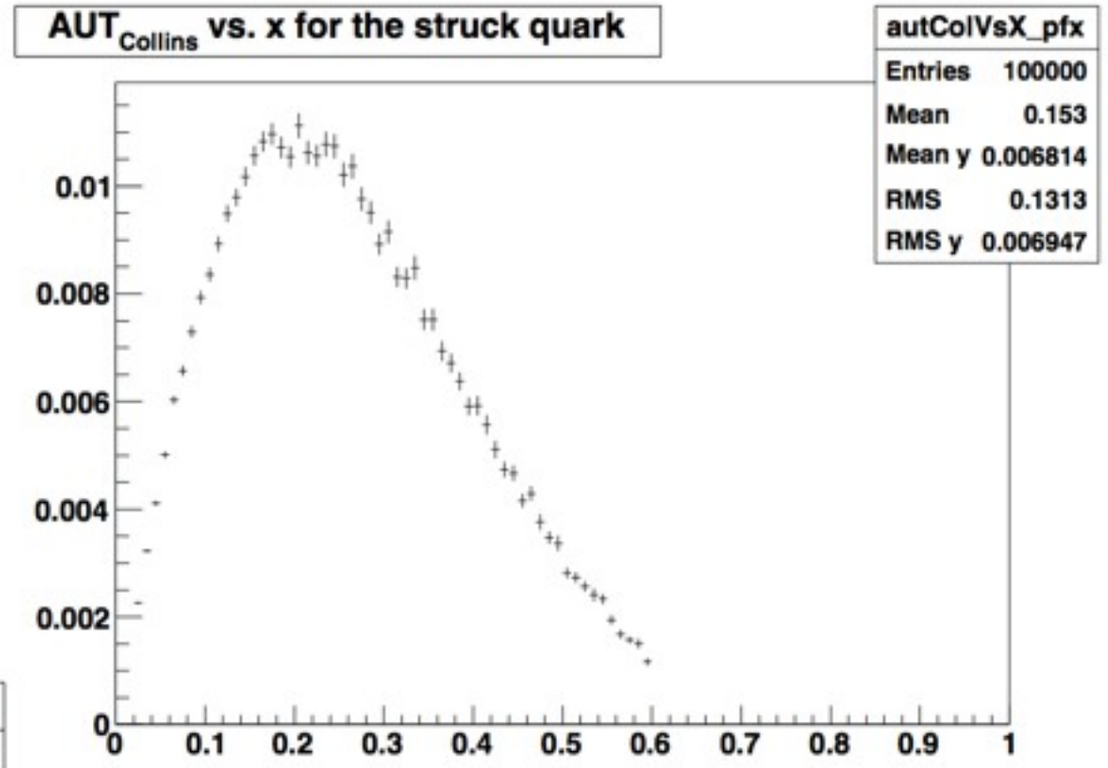
Sivers

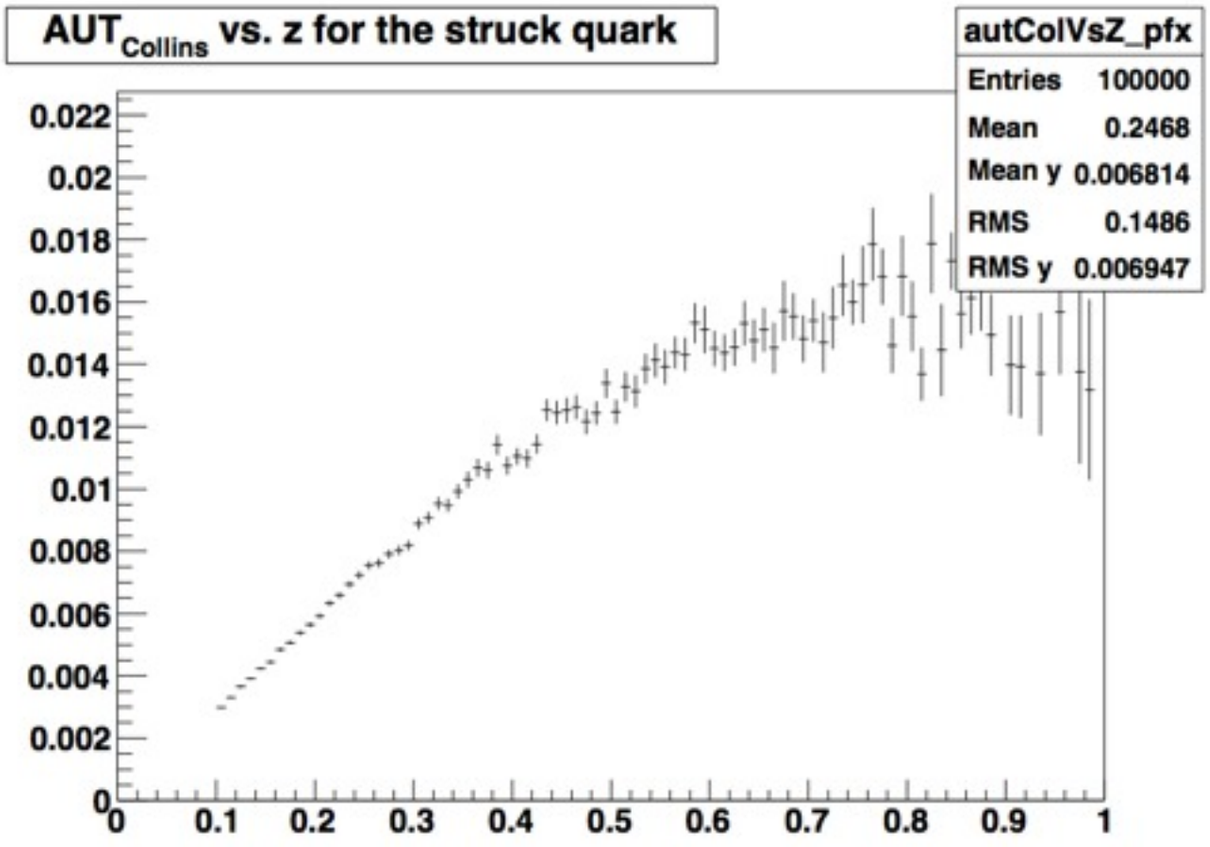

AUT Sivers vs. x summed over quark flavour
autSivAllQVsX_pfx
Entries 100000
Mean 0.1532
Mean y 0.02215
RMS 0.1315
RMS y 0.0179
0.03
0.025
0.02
0.015
0.01
0.005
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

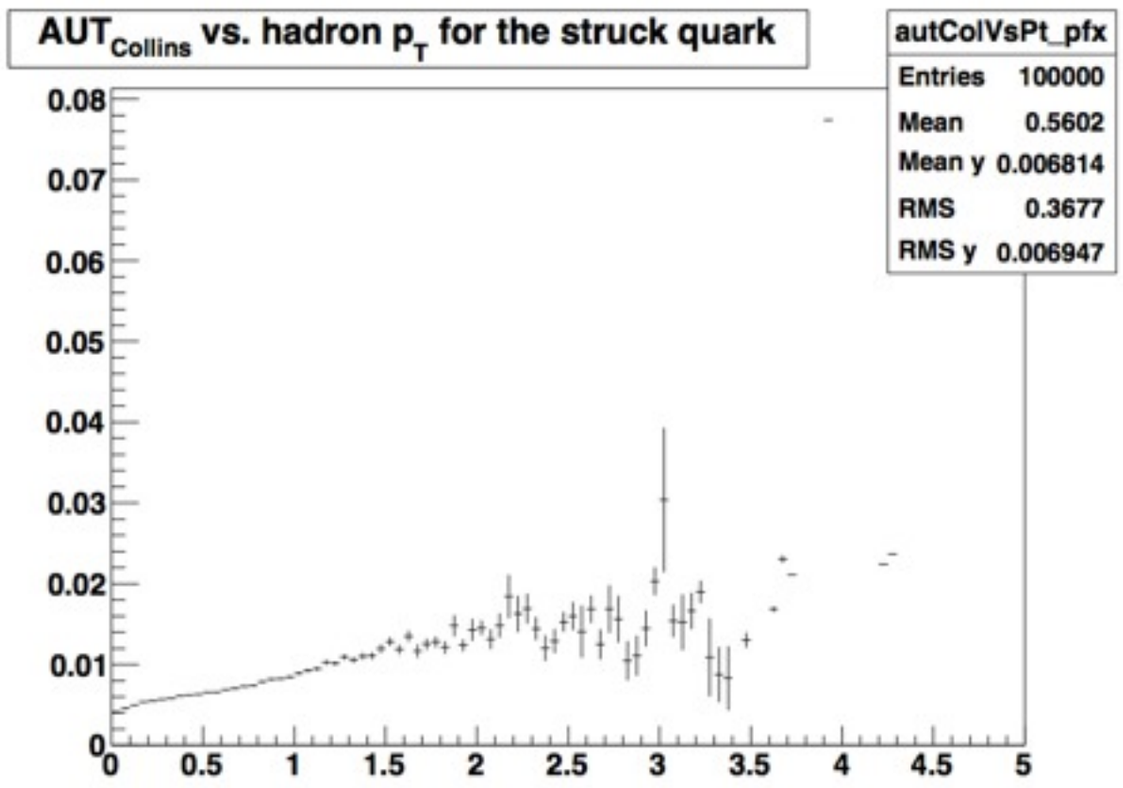

AUT Sivers vs. z summed over quark flavour
autSivAllQVsZ_pfx
Entries 100000
Mean 0.247
Mean y 0.02215
RMS 0.1487
RMS y 0.0179
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


AUT Sivers vs. hadron p T summed over quark flavour
autSivAllQVsPt_pfx
Entries 100000
Mean 0.5603
Mean y 0.02228
RMS 0.3676
RMS y 0.04278
0.1
0
-0.1
-0.2
-0.3
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Collins


AUT_Collins vs. x for the struck quark
autColVsX_pfx
Entries 100000
Mean 0.153
Mean y 0.006814
RMS 0.1313
RMS y 0.006947
0.01
0.008
0.006
0.004
0.002
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

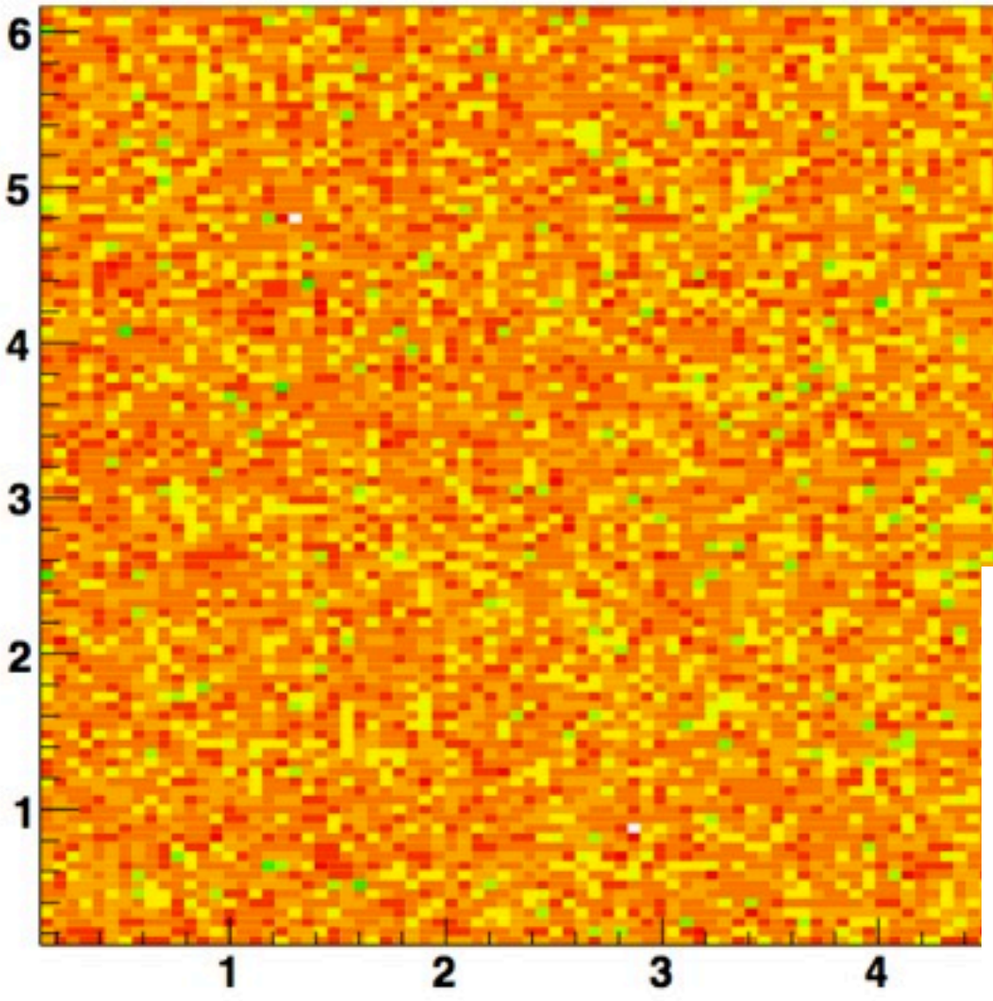

AUT_Collins vs. z for the struck quark
autColVsZ_pfx
Entries 100000
Mean 0.2468
Mean y 0.006814
RMS 0.1486
RMS y 0.006947
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


AUT_Collins vs. hadron p_T for the struck quark
autColVsPt_pfx
Entries 100000
Mean 0.5602
Mean y 0.006814
RMS 0.3677
RMS y 0.006947
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

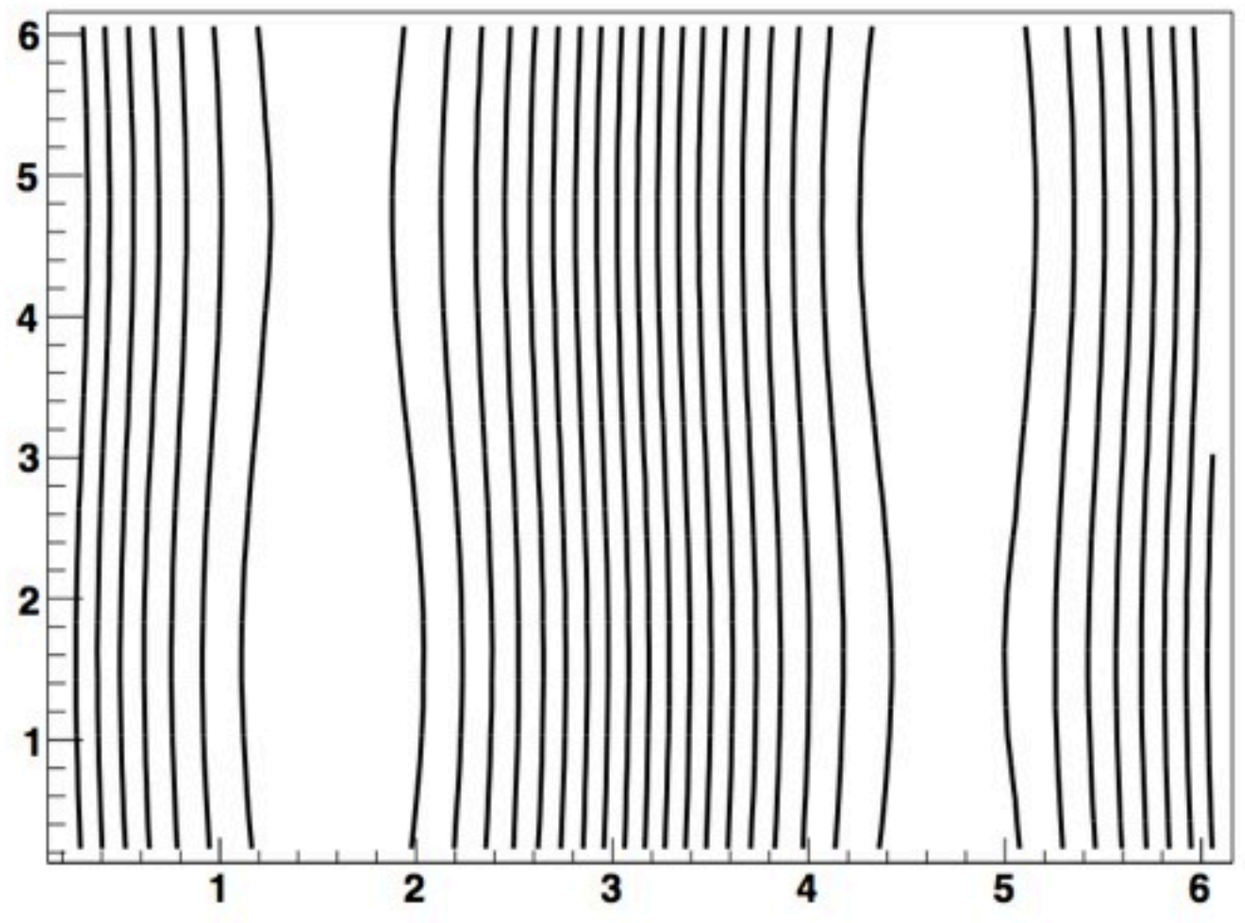
$\phi+\phi_S$ vs. $\phi-\phi_S$

collinsAngleVsSiversAngle

Entries	100000
Mean x	3.113
Mean y	3.141
RMS x	1.749
RMS y	1.744



[0]+[1]*sin(x)+[2]*sin(y)



- Check/QA results to make sure they look sensible, squash bugs...
- Run for different energies (4x50, 4x100, 4x250)
- Run tracks through detector GEANT