CHANGES TO SIMULATION

OLD VERSION

```
1688 //F2kMean Q2 vs F2k
1689 TCanvas *CHf2 = new TCanvas();
1690 TH1F *HF2kCut = new TH1F("HF2kCut", "F2K1D_Histo", 100, 0.0, 1.0);
1691
1692 double GetF2kForCut(int NEvts, Double_t xBjData[],
1693                       Double_t Q2Data[], Double_t F2NData[],
1694                       Double_t Q2CutMin, Double_t Q2CutMax,
1695                       Double_t XCutMin, Double_t XCutMax, Int_t& count){
1696 count=0;
1697          for (Int_t i = 0; i < NEvts; i++) {
1698                if (Q2Data[i] > Q2CutMin &&
1699                    Q2Data[i] < Q2CutMax &&
1700                    xBjData[i] >  XCutMin &&
1701                    xBjData[i] < XCutMax) {
1702 count++;
1703
1704                        HF2kCut->Fill(F2NData[i]);
1705         cout<< count << "-- " "F2N= "<< F2NData[i] << endl;
1706 }
1707         }
1708 cout<< "mean= "<< HF2kCut->GetMean()<< endl;
1709
1710 return HF2kCut->GetMean(2);
1711 }
1712
```

UPDATED VERSION

```
1636 //f2kmean for f2k vs q2
1637 double GetF2kForCut(int NEvts, Double_t xBjData[],
1638                       Double_t Q2Data[], Double_t F2NData[],
1639                       Double_t Q2CutMin, Double_t Q2CutMax,
1640                       Double_t XCutMin, Double_t XCutMax, Int_t& count){
1641 count=0;
1642 Double_t F2kMean= .0;
1643          for (Int_t n = 0; n < NEvts; n++) {
1644                if (Q2Data[n] > Q2CutMin &&
1645                    Q2Data[n] < Q2CutMax &&
1646                    xBjData[n] >  XCutMin &&
1647                    xBjData[n] < XCutMax) {
1648                count++;
1649                F2kMean+=(F2NData[n]);
1650                cout<< count << "-- " "F2N= "<< F2NData[n] << endl;
1651 }
1652         }
1653
1654                        F2kMean= F2kMean/(double)count;
1655 cout<< "mean= "<< F2kMean<< endl;
1656 return F2kMean;
1657 }
1658
1659
```

1)MAKES CUTS TO F2NDATA
AND PUTS IN HISTOGRAM.
2)FINDS MEAN OF HISTOGRAM.
WRONG

1)MAKES CUTS TO F2NDATA
2)DIRECTLY TAKES MEAN
CORRECT

```
1-- F2N= 0.00835195
mean= 0.371169
unc= 0 x= 5.145 y= 0
1-- F2N= 0.00336018
2-- F2N= 0.00346438
mean= 0.370747
```

```
1-- F2N= 0.042476
2-- F2N= 0.0699475
mean= 0.0562117
unc= 0.00177757 x= 5.145 y= 0.0562117
1-- F2N= 0.0857332
mean= 0.0857332
```

OLD VERSION

```
sigma_tdis = sigma_dis * (fpi / f2N);

// #### important !!!!
// cross-section unit "nanobarn" will be applied later... 10^-33 cm2
```

UPDATED VERSION

```
sigma_tdis = sigma_dis * (fpi / f2N)* 10e33;

// #### important !!!!
// (cross-section unit "nanobarn" will be applied later... 10^-33 cm2)HAS BEEN APPLIED
```

TDIS CROSS SECTION WAS IN CM^-2. KIJUN ADVISED TO CONVERT TO NANOBARNS

```
double lum = 100;
```

→

```
double lum = 10e-5;
```

LUMINOSITY WAS ALSO CONVERTED FROM CM^2 TO NANOBARNS TO MATCH CROSS SECTION

OLD VERSION

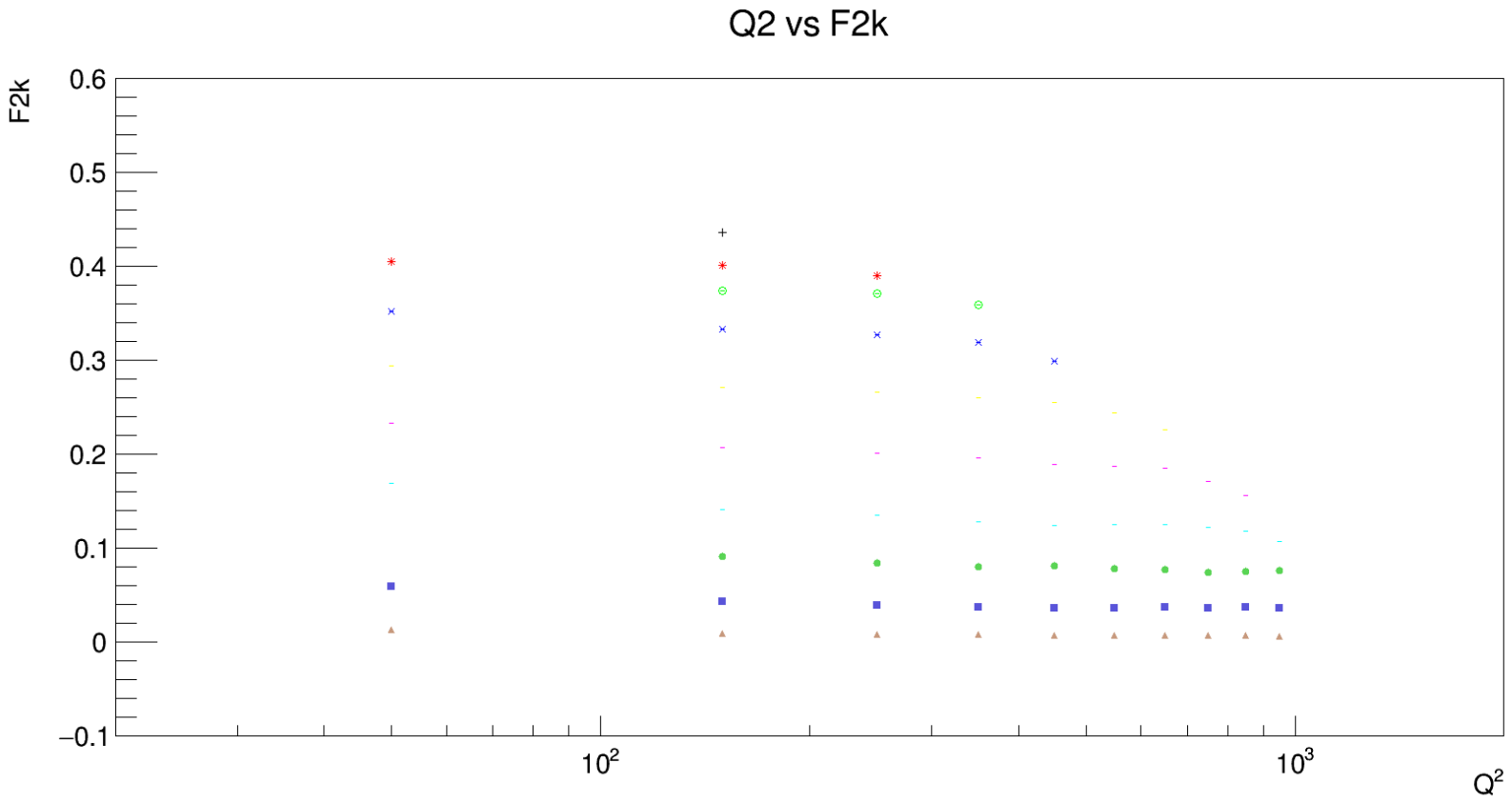
```
double delX = uncerWL(((xminbin[i] + xmaxbin[i]) / 2), 20); //0.2563=std
double lum = 100e39;
double delQ2 = uncerWL(((q2minbin[j] + q2maxbin[j]) / 2), 10);//26.38=std
double uncerVal[3] = {lum,delX,delQ2};
delf2k = uncerW(F2kMean, 10, uncerVal);
```

UPDATED VERSION

```
double delX = xbint[i+1]-xbint[i];
double delQ2 = q2bint[j+1]-q2bint[j];
double delt = tbin[k+1]-tbin[k];
double lum = 10e-5;
double uncertanty = lum*delX*delQ2*delt*(sqrt(count)/count);
```

UNCERTAINTY CALCULATION WAS ADJUSTED:
LUMINOSITY * ΔX * ΔQ * ΔT * $\frac{\sqrt{N}}{N}$
UNCERTAINTIES ARE NOW 10^-7

```
0.1unc= 8.82523e-07 x= 150 y= 0.436054
1.0unc= 2.80056e-07 x= 50 y= 0.404729
1.1unc= 3.08607e-07 x= 150 y= 0.401234
1.2unc= 6.32456e-07 x= 250 y= 0.389893
2.1unc= 2.91386e-07 x= 150 y= 0.374216
2.2unc= 3.01511e-07 x= 250 y= 0.370784
2.3unc= 2.12132e-06 x= 350 y= 0.358517
3.0unc= 5.90879e-07 x= 50 y= 0.352472
3.1unc= 6.09557e-07 x= 150 y= 0.332771
3.2unc= 6.0678e-07 x= 250 y= 0.326966
3.3unc= 6.54654e-07 x= 350 y= 0.318955
3.4unc= 1.26025e-06 x= 450 y= 0.298625
```

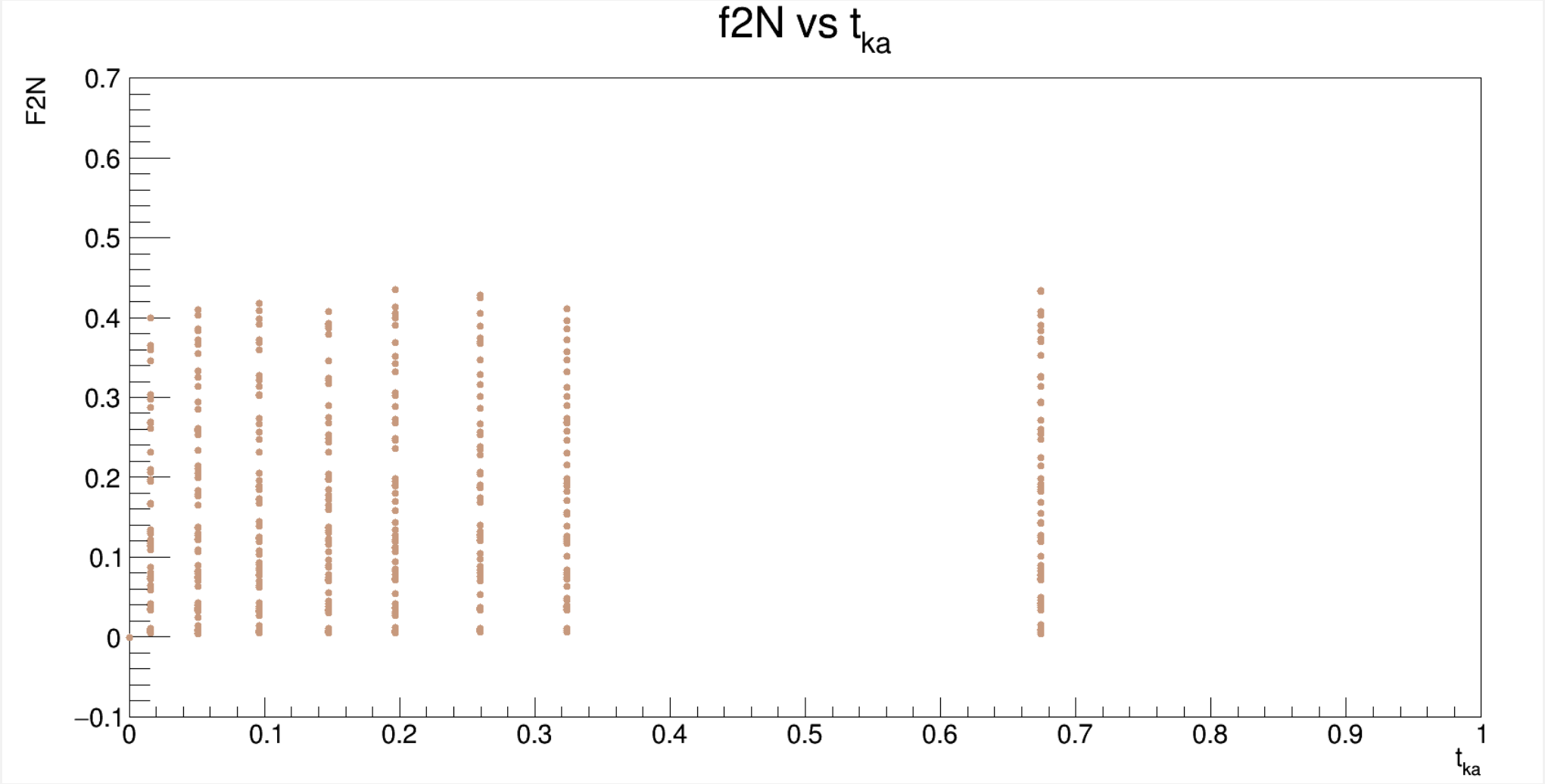



```
/*2d*/  if (!isnan(F2kMean)) {
        if (((q2binq2[j] + q2binq2[j+1]) / 2)!=0){
        Gf2NQ2[i]->SetPoint(j, ((q2binq2[j] + q2binq2[j+1]) / 2), F2kMean);
        Gf2NQ2[i]->SetPointError(j, 0., uncerVal);

/*3D points*/g1->SetPoint(k, i, (q2binq2[j] + q2binq2[j+1]) / 2, F2kMean);
        g1->SetPointError(k , 0., 0., uncerVal);

/*3Dlines */ line3d[i]->SetPoint(j, (q2binq2[j] + q2binq2[j+1]) / 2, F2kMean);
        line3d[i]->SetPointError(j , 0., uncerVal);
```

CUT OUT "NAN" DATA
("NAN" ARISES FROM TAKING MEAN OF EMPTY BIN OR INVALID F2N DATA)



BINS NEED ADJUSTMENT
ADD TDIS CUT
(WORK DIRECTLY FROM ROOT FILE)