Vertex fitting update

Franco Bedeschi, INFN

FCC software meeting

March, 2023

OUTLINE

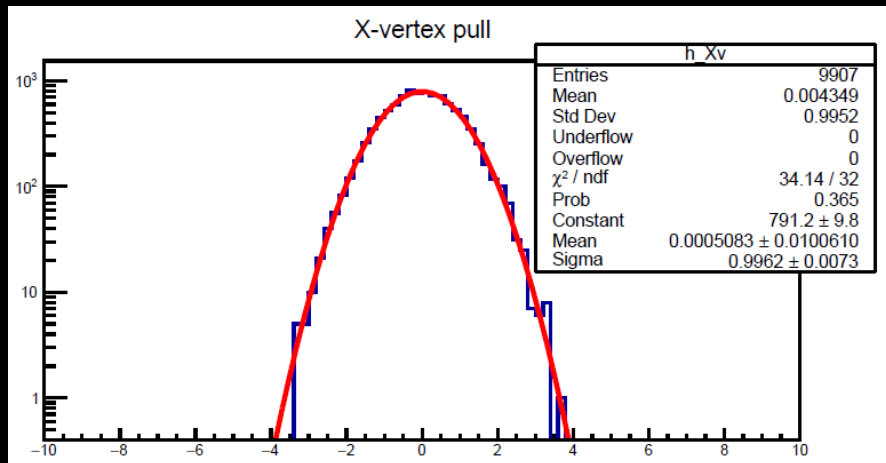
- Primary vertex finding/fitting
- Bs→Ds π example

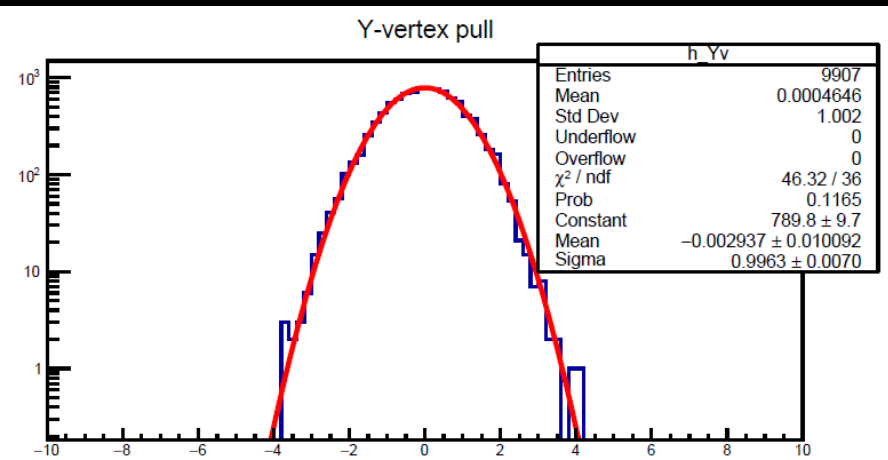
Example of primary fitting

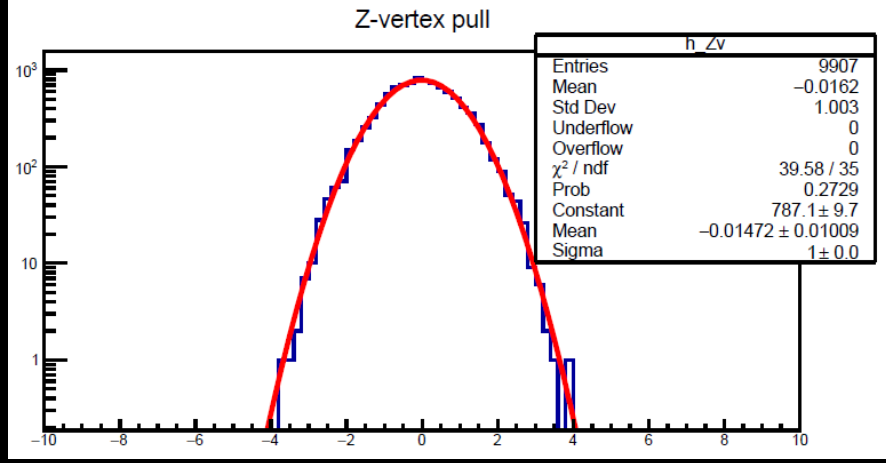
- Vertex fit with external beam spot constraint

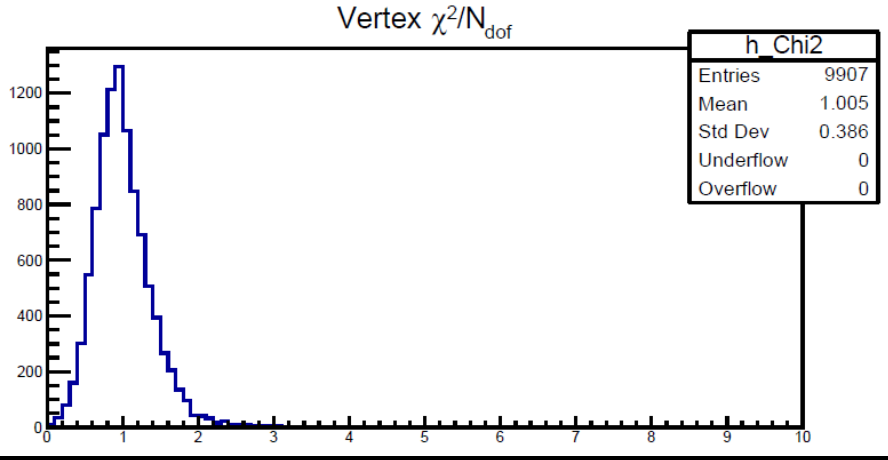
```
//
// Fit primary vertex with beam constraint
//
Int_t MinTrk = 1;   // Minumum # tracks for vertex fit
if (Ntr >= MinTrk) {
   VertexFit* Vtx = new VertexFit(Ntr, pr, cv);
   Vtx->AddVtxConstraint(xpvc, covpvc);
   TVectorD xvtx = Vtx->GetVtx();
   delete Vtx;
}
```

Vertex pulls



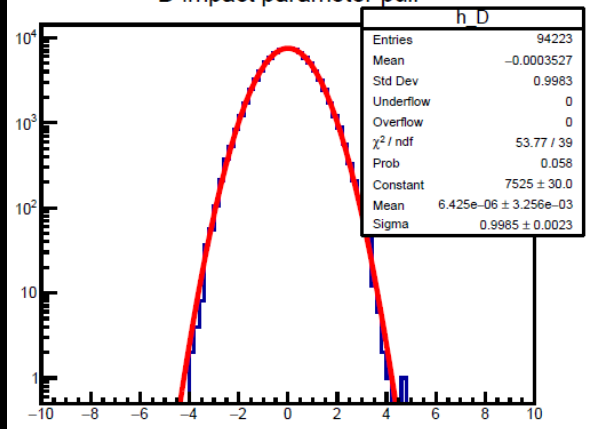




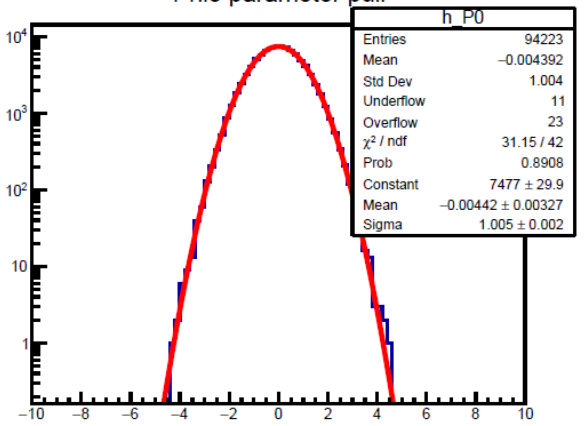


Updated parameter pulls

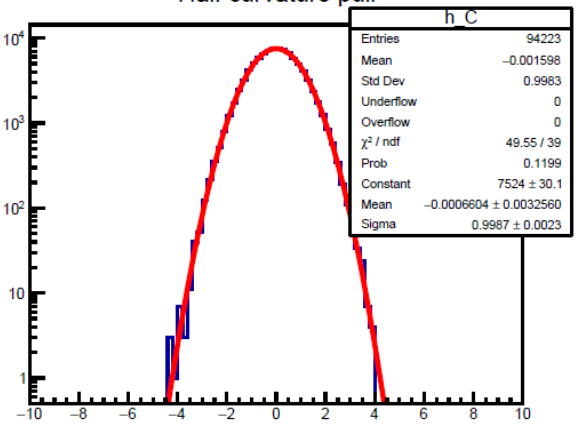
D impact parameter pull



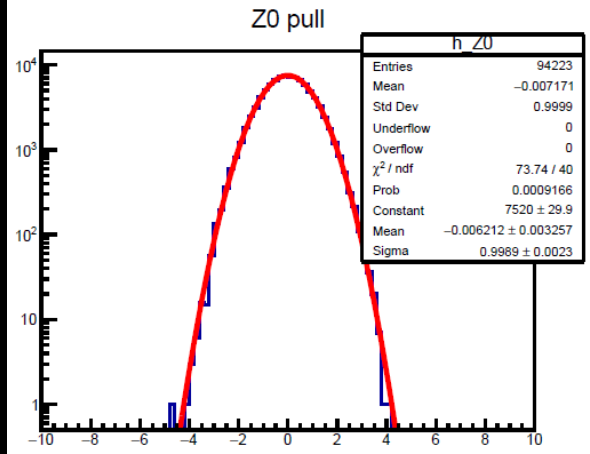
Phi0 parameter pull



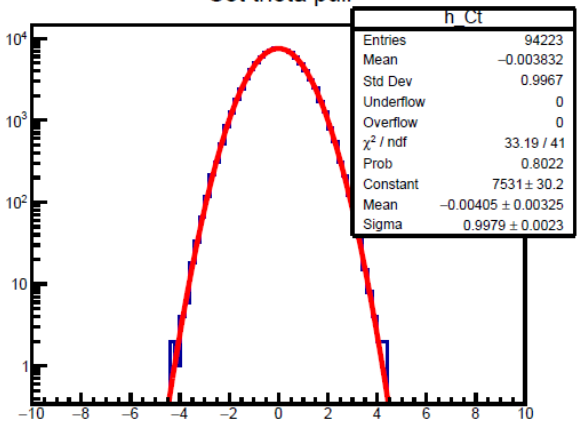
Half curvature pull



Z0 pull



Cot theta pull

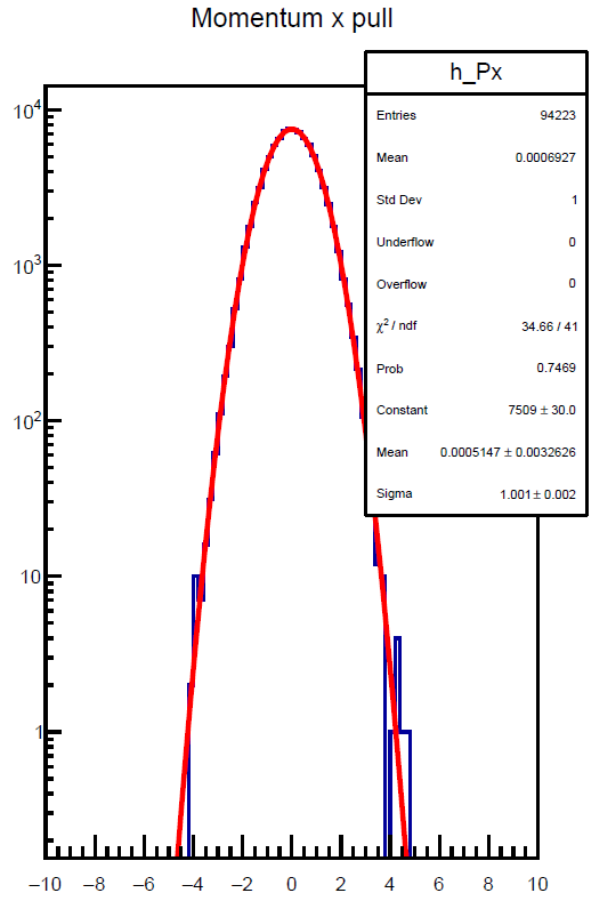


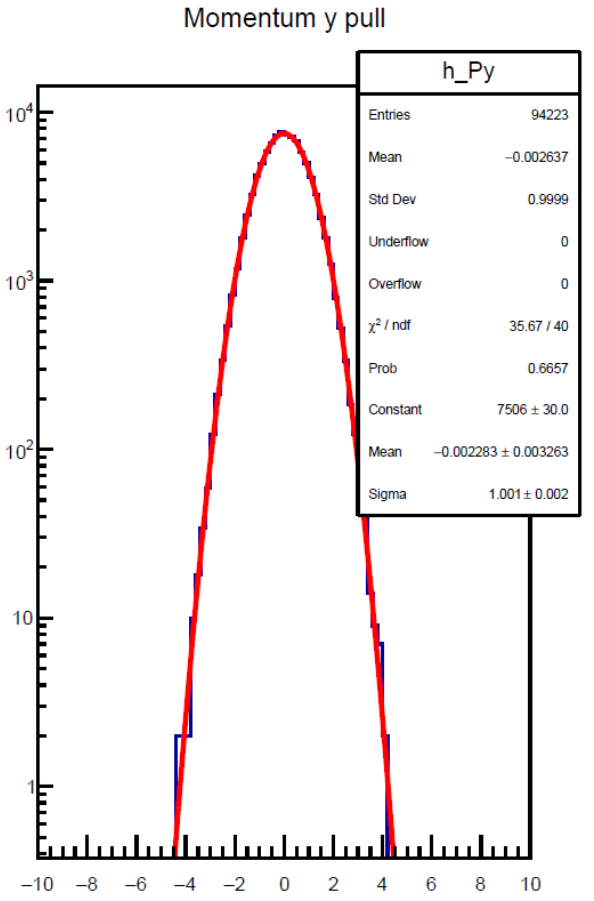
Track momenta

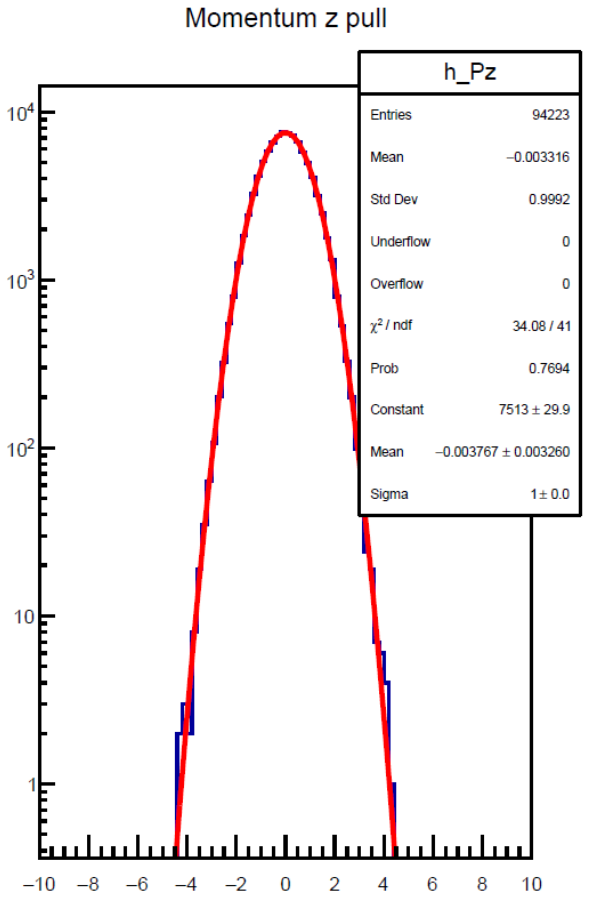
- Get track momenta at vertex

```
Bool_t Mm = kTRUE;
VertexMore* Vmore = new VertexMore(Vfit, Mm);
    //
    // Get fit momenta momenta
    //
    TVector3 pRec = Vmore->GetMomentum(i);
    TMatrixDSym pCov = Vmore->GetMomentumC(i);
```

Track momenta pulls







Primary finder (1)

- Skim tracks

```
//
// Skim tracks
Int_t nSkim = 0;
Int_t* nSkimmed = new Int_t[NtrG];
TVectorD** PrSk = new TVectorD * [1];
TMatrixDSym** CvSk = new TMatrixDSym * [1];
Double_t MaxChi2 = 9.;
for (Int_t n = 0; n < NtrG; n++) {
    PrSk[0] = new TVectorD(*pr[n]);
    CvSk[0] = new TMatrixDSym(*cv[n]);
    VertexFit* Vskim = new VertexFit(1,PrSk, CvSk);
    Vskim->AddVtxConstraint(xpvc, covpvc);
    Double_t Chi2One = Vskim->GetVtxChi2();
    if(fprvx[n])hChi2SngP->Fill(Chi2One);
    else hChi2SngN->Fill(Chi2One);
    //
    if (Chi2One < MaxChi2) {
        nSkimmed[nSkim] = n;
        nSkim++;
    }
```

Primary finder (2)

- Fit skimmed

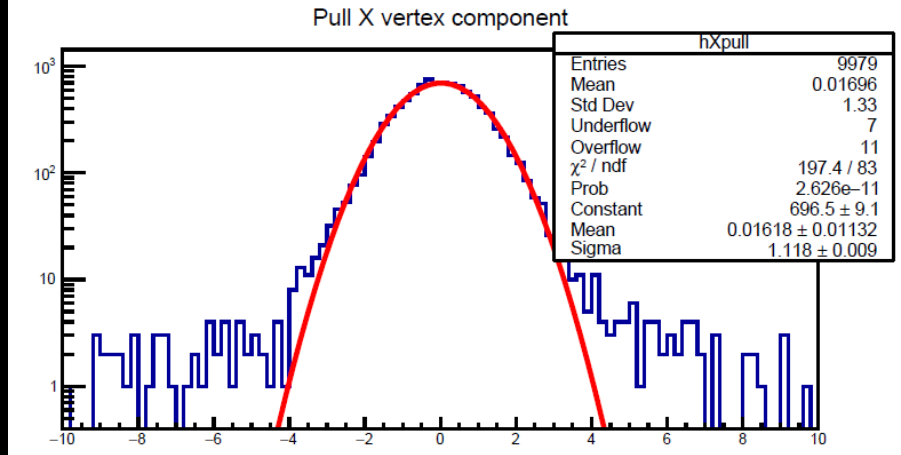
```
//
// Load tracks for primary fit
Int_t MinTrk = 1;   // Minumum # tracks for vertex fit
std::vector<Int_t> trnum;
if (nSkim >= MinTrk) {
    TVectorD** PrFit = new TVectorD * [nSkim];
    TMatrixDSym** CvFit = new TMatrixDSym * [nSkim];
    for (Int_t n = 0; n < nSkim; n++) {
        PrFit[n] = new TVectorD(*pr[nSkimmed[n]]);
        CvFit[n] = new TMatrixDSym(*cv[nSkimmed[n]]);
        trnum.push_back(nSkimmed[n]);
    }
    delete[] nSkimmed;
    Int_t Nfound = nSkim;
    if(entry%500 ==0)std::cout << "Found tracks "<<Nfound << std::endl;
    const Int_t MaxFound = 100;
    Double_t Chi2LL[MaxFound]; Double_t *Chi2L = Chi2LL;
    VertexFit* Vtx = new VertexFit(nSkim, PrFit, CvFit);
    //std::cout << "Vertex fit created " << std::endl;
    Vtx->AddVtxConstraint(xpvc, covpvc);
```

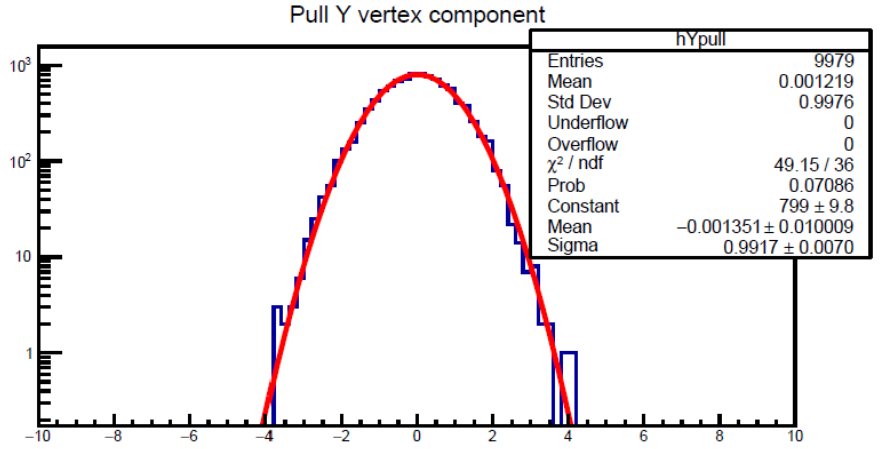
Primary finder (3)

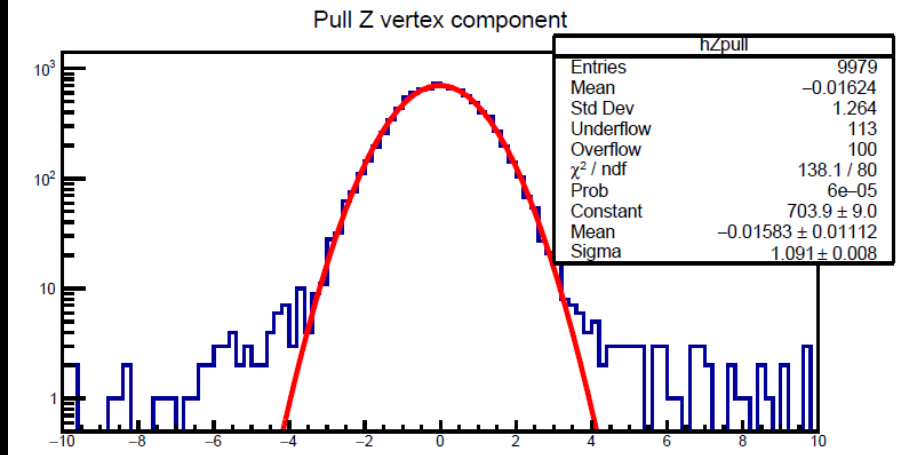
- Remove iteratively large Chi2 contrubutions

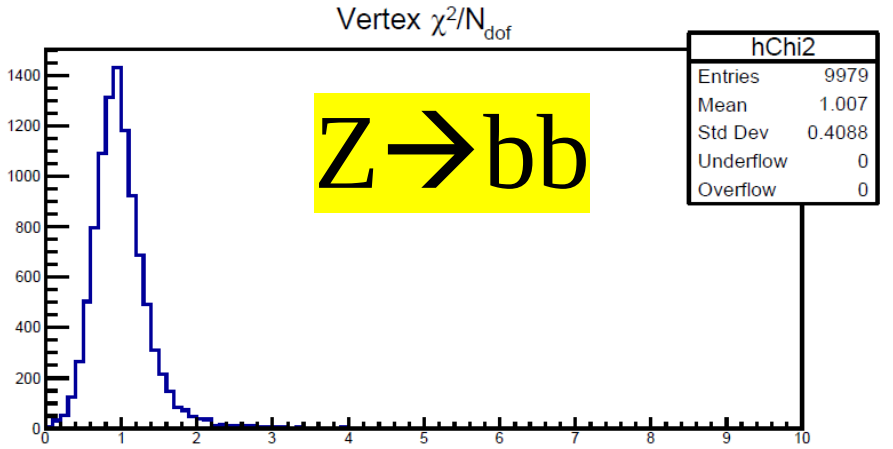
```
//
// Remove tracks with large chi2
Double_t MaxChi2Fit = 8.0;
Bool_t Done = kFALSE;
while (!Done) {
    // Find largest Chi2 contribution
    TVectorD Chi2List = Vtx->GetVtxChi2List();  // Get contributions to Chi2
    Chi2L = Chi2List.GetMatrixArray();
    Int_t iMax = TMath::LocMax(Nfound, Chi2L);
    Double_t Chi2Mx = Chi2L[iMax];
    if(fprvx[trnum[iMax]])hChi2MaxP->Fill(Chi2Mx);
    else hChi2MaxN->Fill(Chi2Mx);
    if (Chi2Mx > MaxChi2Fit && Nfound > 1) {
        Vtx->RemoveTrk(iMax);
        trnum.erase(trnum.begin() + iMax);
        Nfound--;
    }
    else {
        Done = kTRUE;
    }
}
```

Found vertex pulls

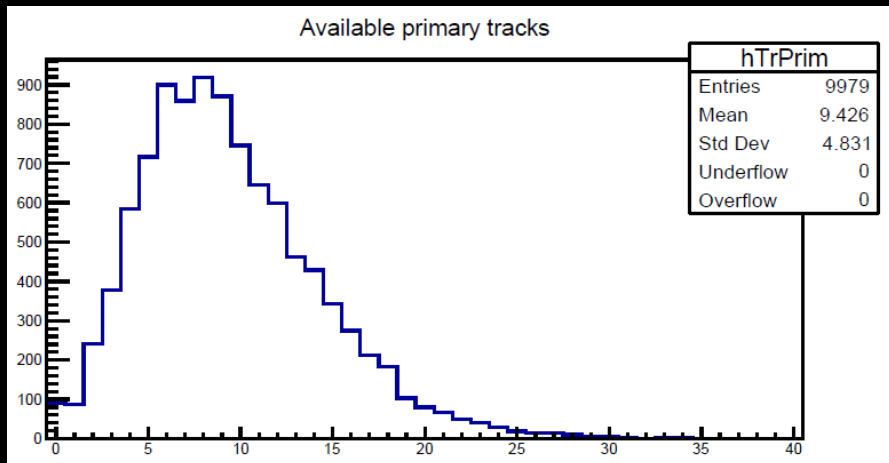


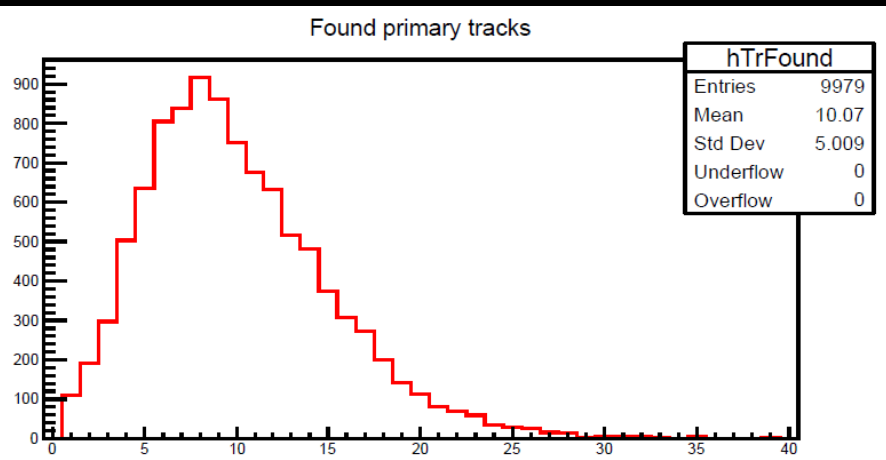


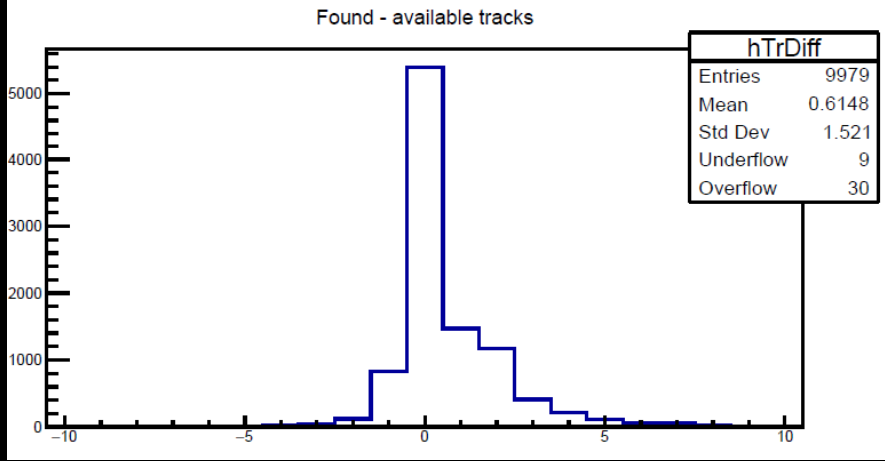




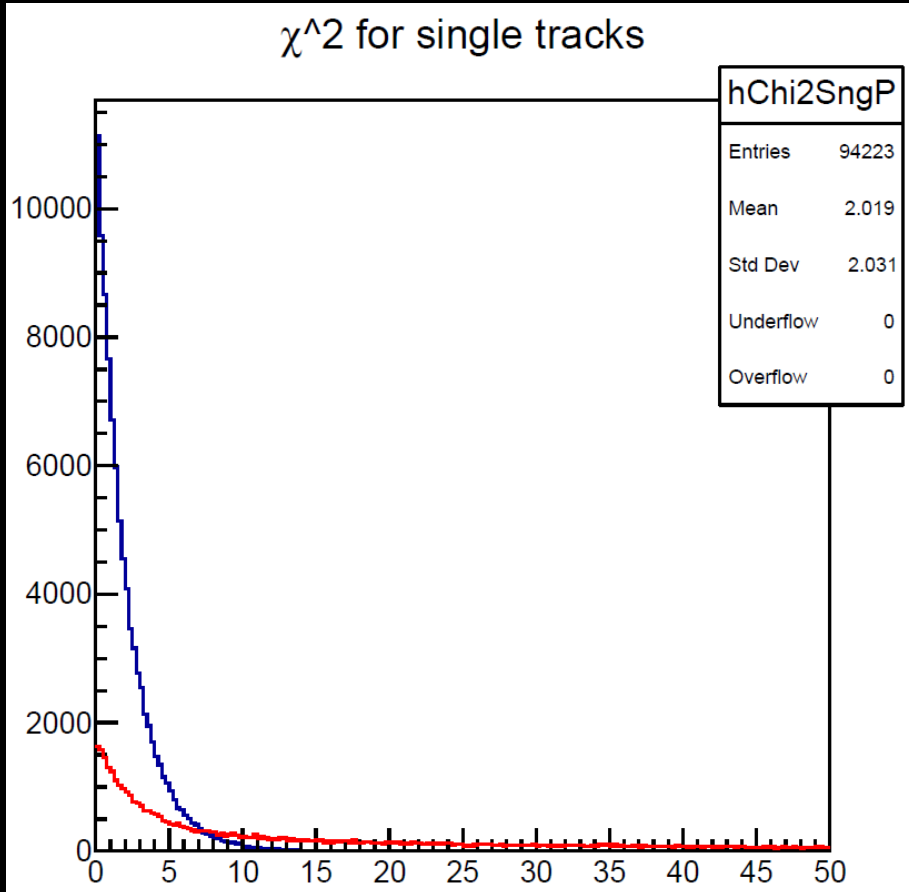
Found vertex tracks

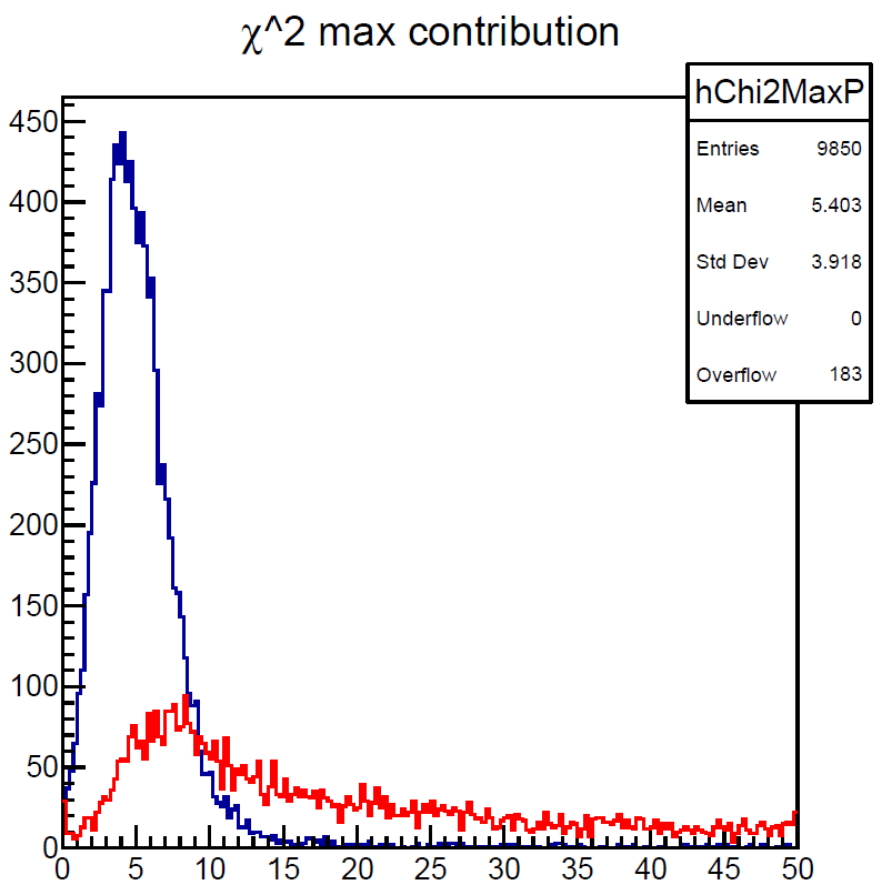






Found vertex Chi2's



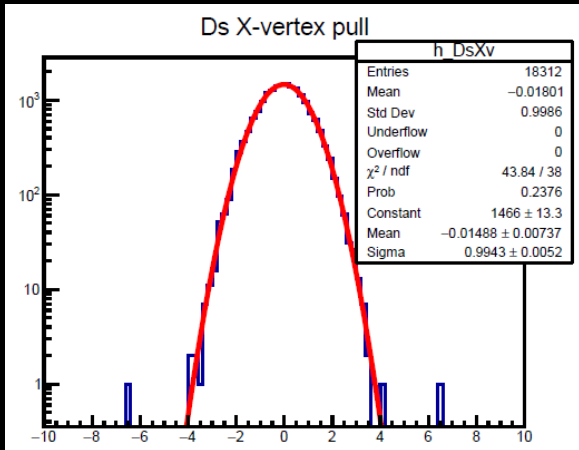


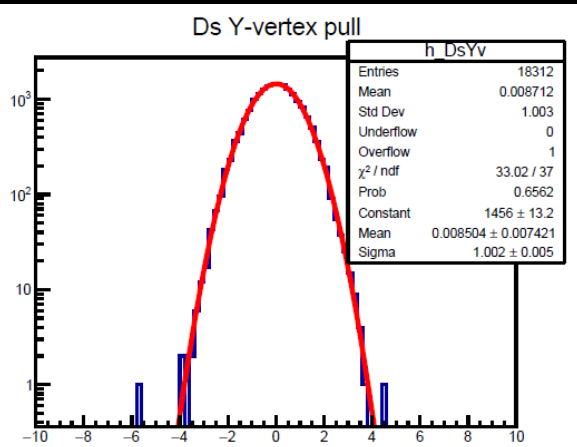
Fit Ds vertex

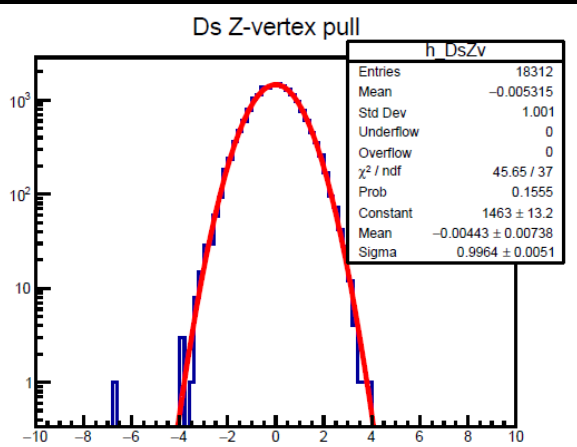
```
// Fit Ds vertex
VertexFit* vDs = new VertexFit(nDsT, tDsPar, tDsCov);
Double_t DsChi2 = vDs->GetVtxChi2();        // Ds fit Chi2
```

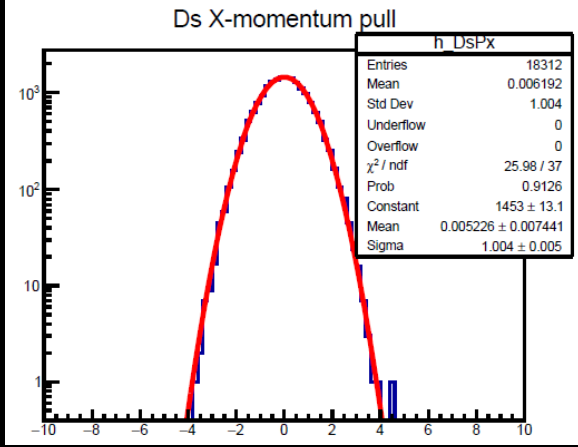
```
// More fitting
Bool_t Units = kTRUE;        // Set to mm
VertexMore* VMDs = new VertexMore(vDs,Units);
// Mass constraint if requested
Bool_t MCst = kTRUE;              // Mass constraint flag
if(MCst){
    Double_t DsMass = pBsDs->Mass;  // Ds mass
    Double_t DsMasses[nDsT]; Int_t DsList[nDsT];
    for(Int_t k=0; k<nDsT; k++){
      DsMasses[k] = pDs[k]->Mass;
      DsList[k]   = k;
    }
    VMDs->AddMassConstraint(DsMass,  nDsT,  DsMasses,  DsList);
    VMDs->MassConstrFit();
}
TVectorD rDv = VMDs->GetXv();           // Ds vertex
```

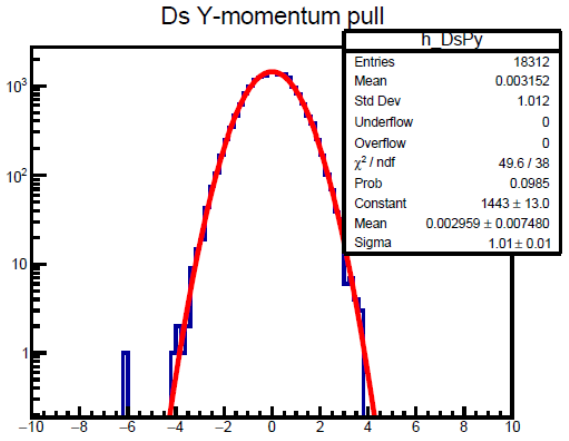
Ds pulls

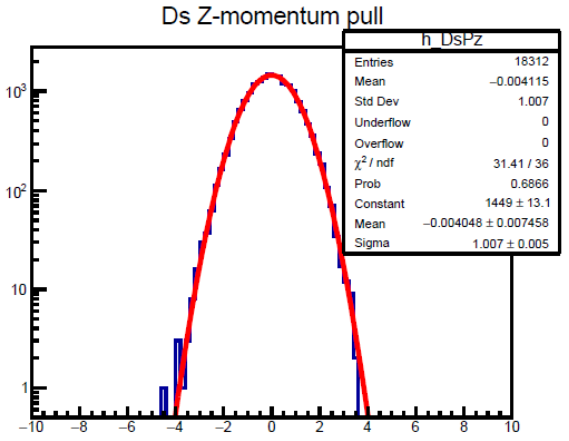

Ds X-vertex pull
h_DsXv
Entries 18312
Mean −0.01801
Std Dev 0.9986
Underflow 0
Overflow 0
χ² / ndf 43.84 / 38
Prob 0.2376
Constant 1466 ± 13.3
Mean −0.01488 ± 0.00737
Sigma 0.9943 ± 0.0052


Ds Y-vertex pull
h_DsYv
Entries 18312
Mean 0.008712
Std Dev 1.003
Underflow 0
Overflow 1
χ² / ndf 33.02 / 37
Prob 0.6562
Constant 1456 ± 13.2
Mean 0.008504 ± 0.007421
Sigma 1.002 ± 0.005


Ds Z-vertex pull
h_DsZv
Entries 18312
Mean −0.005315
Std Dev 1.001
Underflow 0
Overflow 0
χ² / ndf 45.65 / 37
Prob 0.1555
Constant 1463 ± 13.2
Mean −0.00443 ± 0.00738
Sigma 0.9964 ± 0.0051


Ds X-momentum pull
h_DsPx
Entries 18312
Mean 0.006192
Std Dev 1.004
Underflow 0
Overflow 0
χ² / ndf 25.98 / 37
Prob 0.9126
Constant 1453 ± 13.1
Mean 0.005226 ± 0.007441
Sigma 1.004 ± 0.005


Ds Y-momentum pull
h_DsPy
Entries 18312
Mean 0.003152
Std Dev 1.012
Underflow 0
Overflow 0
χ² / ndf 49.6 / 38
Prob 0.0985
Constant 1443 ± 13.0
Mean 0.002959 ± 0.007480
Sigma 1.01 ± 0.01


Ds Z-momentum pull
h_DsPz
Entries 18312
Mean −0.004115
Std Dev 1.007
Underflow 0
Overflow 0
χ² / ndf 31.41 / 36
Prob 0.6866
Constant 1449 ± 13.1
Mean −0.004048 ± 0.007458
Sigma 1.007 ± 0.005

Fit Bs

```
//
// Load Bs tracks
TVectorD* tBsPar[nBsT];
TMatrixDSym* tBsCov[nBsT];
TVectorD par(5); TMatrixDSym cov(5);
TrkToVector(tBs[0], par, cov);
tBsPar[0] = new TVectorD(par);              // Bs pion
tBsCov[0] = new TMatrixDSym(cov);
tBsPar[1] = new TVectorD(VMDs->GetVpar());  // Ds from previous fit
tBsCov[1] = new TMatrixDSym(VMDs->GetVcov());
//
// Fit Bs vertex
VertexFit* vBs = new VertexFit(nBsT, tBsPar, tBsCov);
vBs->SetStartR(RDs);
Double_t BsChi2 = vBs->GetVtxChi2();
```

Bs fitted mass