

Franco Bedeschi, INFN

Phys. Performance meeting  
July, 2022

## OUTLINE

- ❖ **Intro/Caveats**
- ❖ **Vertex fitting**
- ❖ **Recent updates**
- ❖ **More to come**

- ❖ **Vertex fit is already available in many existing packages**
  - ..... That's the problem:
  - Using it usually means you also buy a complex infrastructure
- ❖ **Goal here is to provide a simple standalone code**
  - Easy to insert in FCC framework
  - Based on ROOT libraries only
  - More complex implementations can come later

## ❖ Input:

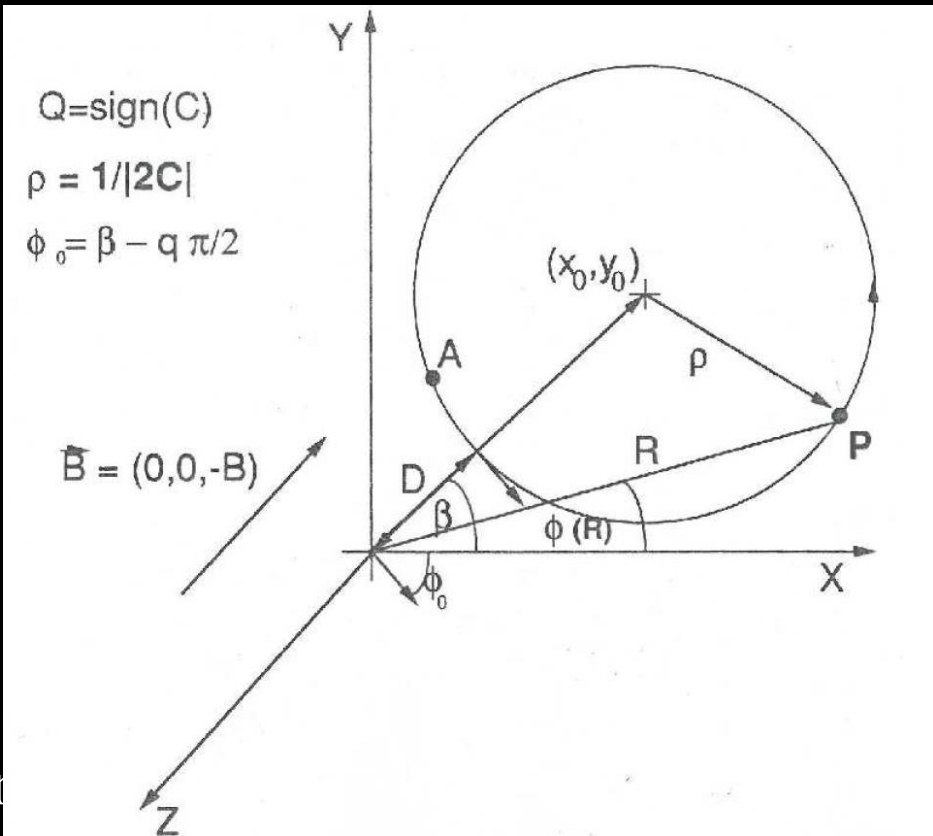
- Array of observed tracks
  - 5 parameters + Cov. Matrix
  - Assume perfect helix
    - $D, \phi_0, C, \cot \theta, z_0$

## ❖ Output:

- 3D vertex + covariance

## ❖ Method:

- $\chi^2$  minimization with constraint
  - Vary parameters and phase and force all tracks to cross at same point



## ❖ The $\chi^2$ :

$$\chi^2 = \sum_{i=1}^N \{ \delta \vec{\alpha}_i^t C_i^{-1} \delta \vec{\alpha}_i + 2(\vec{x}(\vec{\alpha}_i, s_i) - \vec{x}_V)^t \vec{\lambda}_i \}$$

- $\vec{\alpha}_i$ ,  $C_i$  = track parameters and covariance matrix
- $s_i$  = helix phases,  $\vec{\lambda}_i$  = Lagrange multipliers
- Solve by linearization and iteration on  $s_i$  (mean is 4 iterations)

## ❖ Solution:

$$\vec{x}_V = D^{-1} \sum_{i=1}^N D_i \vec{x}_i^0$$

$$C_v = Cov(\vec{x}_V) = D^{-1} \left( \sum_{i=1}^N D_i W_i^{-1} D_i \right) D^{-1}$$

$$\vec{x}_i = \vec{x}(\vec{\alpha}_i, s_i) = \vec{x}_i^0 + A_i \delta \vec{\alpha}_i + \vec{a}_i \delta s_i$$

$$W_i^{-1} = A_i C_i A_i^t = Cov(\vec{x}_i) \text{ at fixed } s_i$$

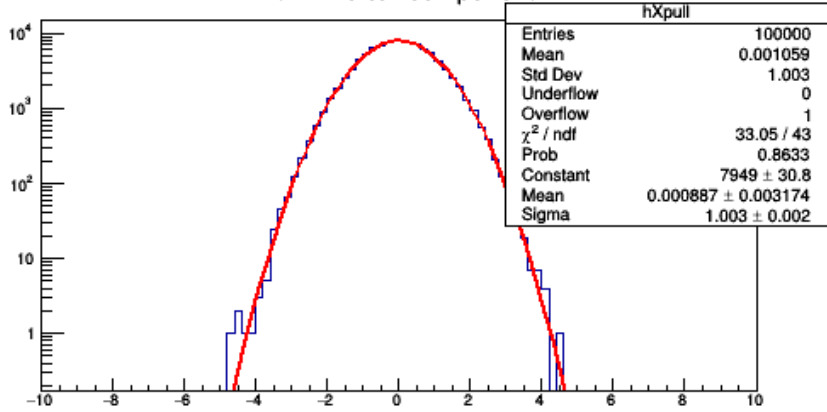
$$D_i = W_i - W_i \frac{\vec{a}_i \vec{a}_i^t}{a_i} W_i, \text{ where } a_i = \vec{a}_i^t W_i \vec{a}_i$$

$$D = \sum_{i=1}^N D_i$$

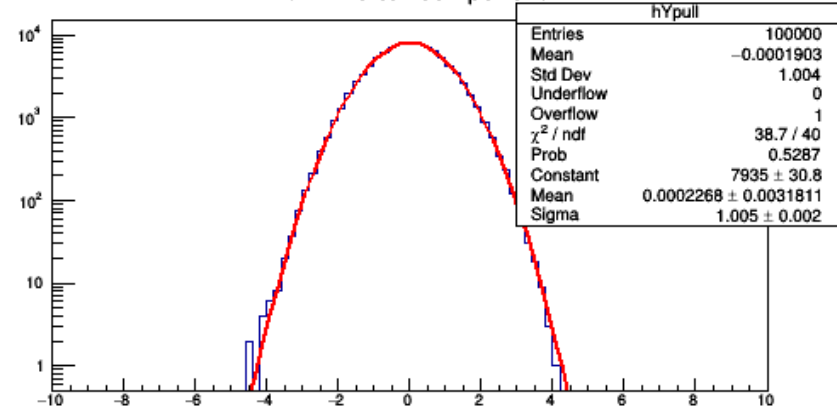
$$\chi^2 = \sum_{i=1}^N \vec{\lambda}_i^t W_i^{-1} \vec{\lambda}_i$$

❖ Example: 100,000 events, 3 tracks,  $R_V$  (0-1.5 cm),  $pt$  (0.5-1.0 GeV)

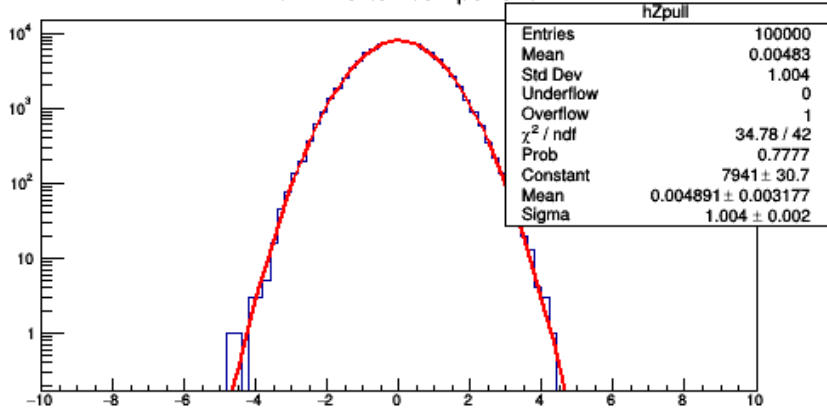
Pull X vertex component



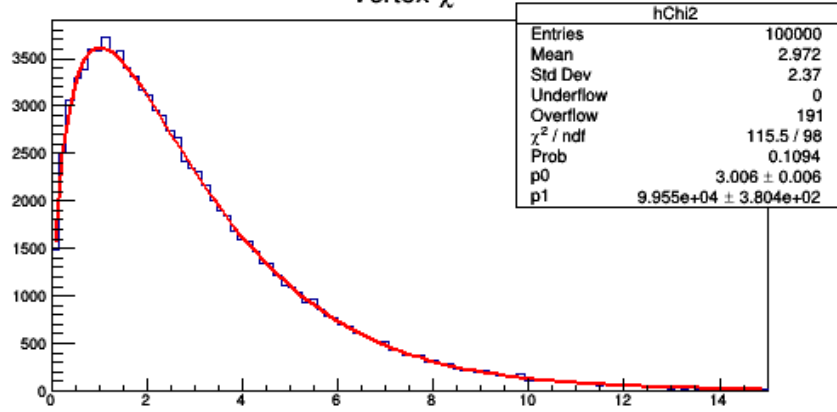
Pull Y vertex component



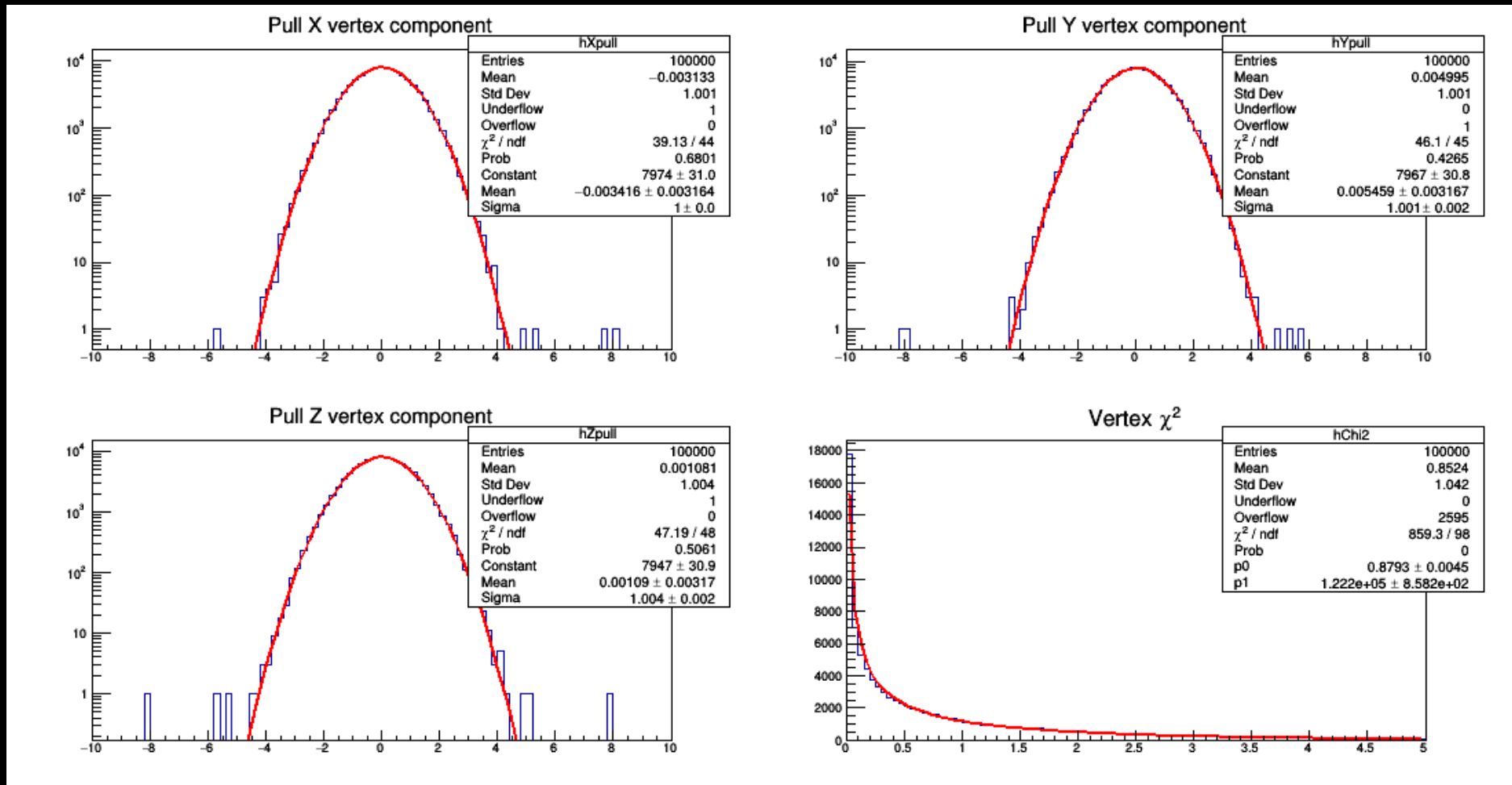
Pull Z vertex component



Vertex  $\chi^2$



❖ Example: 100,000 events, 2 tracks,  $R_V$  (0-1.5 cm),  $pt$  (0.5-1.0 GeV)



## ❖ Code location:

➤ <https://github.com/delphes/delphes/tree/master/external/TrackCovariance>

■ VertexFit.cc, VertexFit.h (basic vertex fitting)

■ VertexMore.cc, VertexMore.h (additional features)

## ❖ How to call:

➤ Track loop:

```
pr[Ntr] = new TVectorD(obsPar); // Parameters
```

```
cv[Ntr] = new TMatrixDSym(covMat); // Error matrices
```

```
➤ VertexFit* Vtx = new VertexFit(Ntr, pr, cv); // Vertex fit
```

```
TVectorD xvtx = Vtx->GetVtx(); // Vertex found
```

```
TMatrixDSym covX = Vtx->GetVtxCov(); // Vertex covariance
```

```
Double_t Chi2 = Vtx->GetVtxChi2(); // Fit Chi2
```

## ❖ Add/remove track from existing fit:

- AddTrk(TVectorD \*par, TMatrixDSym \*Cov);
- RemoveTrk(Int\_t iTrk);
  - Useful in vertex finding loops

## ❖ Set starting radius:

- SetStartR(Double\_t R);
  - Separate seagulls and cowboys
  - Default is expand at P.C.A.
  - New feature in DELPHES to provide lowest hit coordinates



## ❖ Add vertex constraint:

- AddVtxConstraint(TVectorD xv, TMatrixDSym cov);
  - Useful in primary vertex finding



## ❖ Updated track parameters after vertex fit:

- GetNewPar(Int\_t i)
- GetNewCov(Int\_t i) // Diagonal covariance
- Correlations:
  - GetNewCov(Int\_t i, Int\_t j) //  $\langle \text{Par}_i * \text{Par}_j \rangle$
  - GetNewCovXvPar(Int\_t i) //  $\langle X_v * \text{Par}_i \rangle$

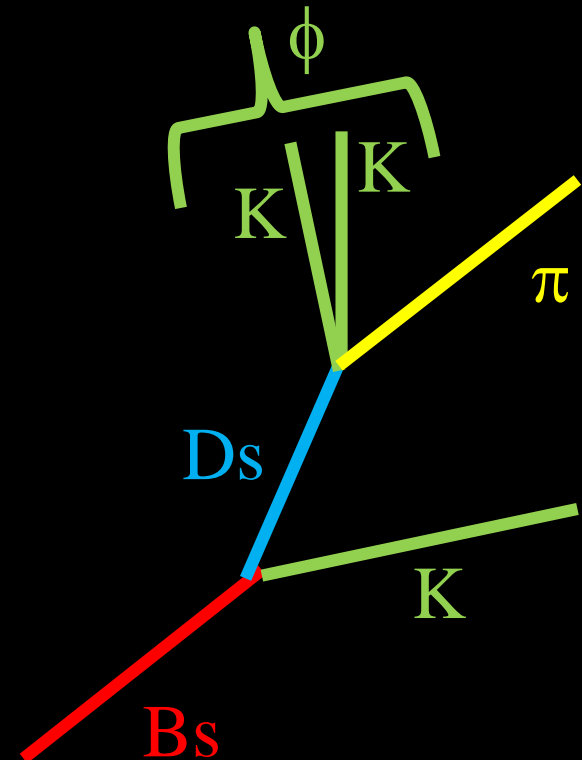
❖ **Invocation:** `VertexMore* VM = new VertexMore(Vtx);`

❖ **Track momentum/errors at vertex:**

- `TVector3 GetMomentum(Int_t i)`
- `TMatrixDSym GetMomentumC(Int_t i)`
- Total vertex momentum/error:
  - `TVector3 GetTotalP()`
  - `TMatrixDSym GetTotalPcov()`
- `TMatrixDSym GetBigCov()`
  - Full  $x_v, p_1, p_2, \dots, p_n$  covariance matrix

❖ **Charged vertex only for now!**

- Vertex track parameters/errors:
  - `TVectorD GetVpar()`
  - `TMatrixDSym GetVcov()`
  - Can then use as track in vertexing chain (e.g.  $B_s \rightarrow D_s K$ )
  - Problem with charged vertex tracing in DELPHES



## ❖ Multiple mass constraint:

- AddMassConstraint (Double\_t Mass, Int\_t Ntr, Double\_t\* masses, Int\_t\* list)
- Eg.  $D_s \rightarrow \phi \pi$  ( $\phi \rightarrow KK$ )
  - Constrain simultaneously  $\phi$  and  $D_s$  mass in vertex fit
    - Call twice before refit
- Coded and tested locally
  - Waiting for fixing tracking problem in DELPHES

## ❖ Including neutrals

## ❖ Fast vertex fitting code dependent only on ROOT libraries is provided

- Very simple to use without any overhead
- Most common features included and more coming

## ❖ Coming (hopefully) soon:

- Easy new features:
  - Mass constraining (basically complete)
  - Pointing constraint (math done)
- Hard new features:
  - Dealing with neutrals