Vertex fitting

Franco Bedeschi, INFN

FCC-ee topical meeting

February, 2021

OUTLINE

- Intro/Caveats
- Vertex fitting
  - Performance checks
- Way forward

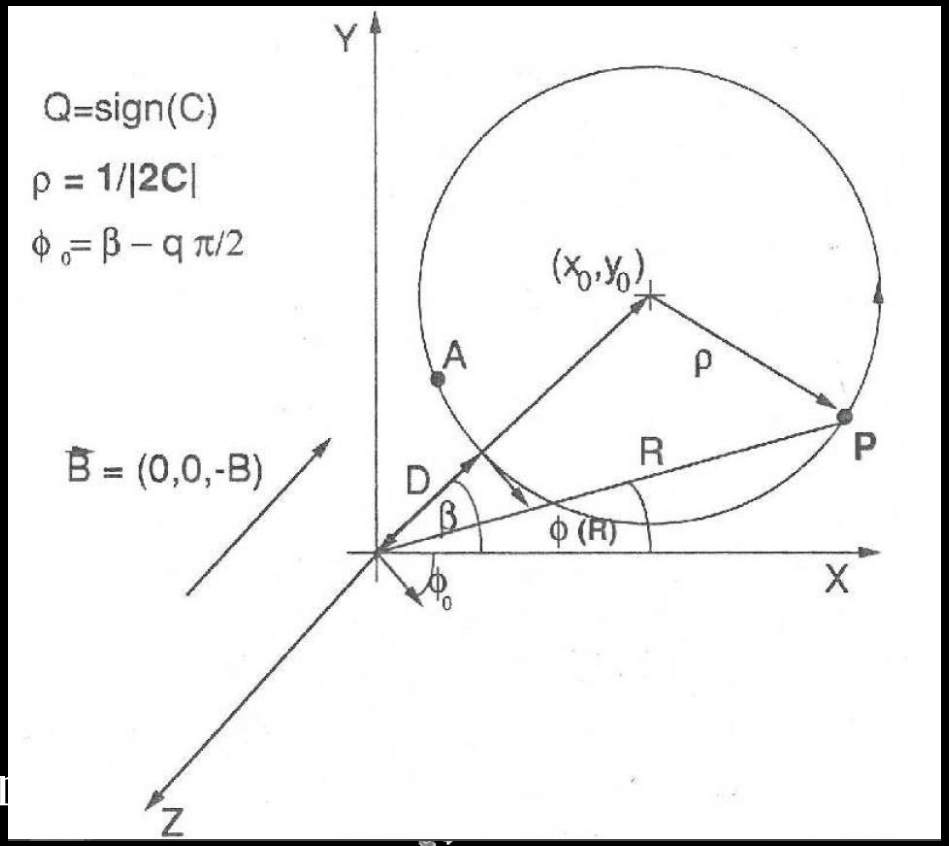
Introduction

- 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
  - More complex implementations can come later

Vertex fit (1)

- Input:
  - Array of observed tracks
    - 5 parameters + Cov. Matrix
    - Assume perfect helix
      - D, ϕ_0, C, cot θ, z_0
- Output:
  - 3D vertex + covariance
- Method:
  - χ^2 minimization with constrai
    - Vary parameters and force all tracks to cross at same point



Vertex fit (2)

- The χ^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:

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

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

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

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

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

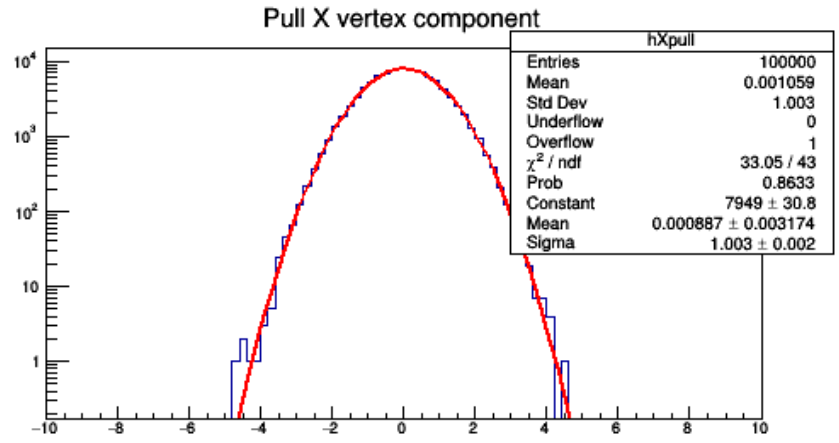
$$D = \sum_{i=1}^{N} D_i$$

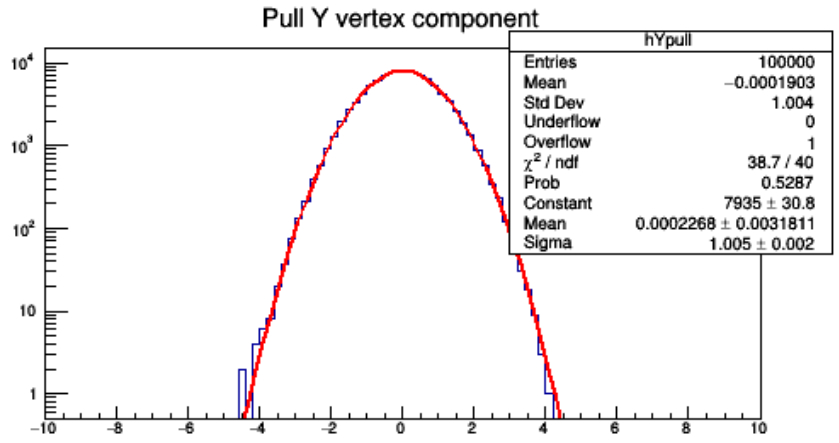
$$\chi^2 = \sum_{i=1}^{N} \vec{\lambda}_i^{\,t} W_i^{-1} \vec{\lambda}_i$$

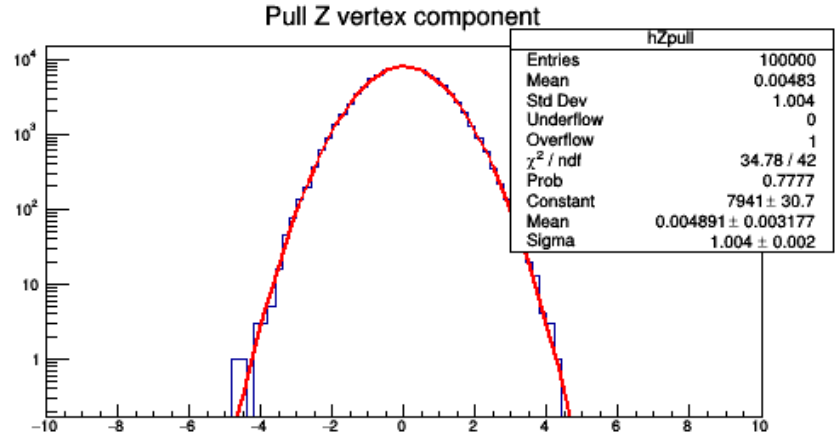
$$\chi^2 = \sum_{i=1}^{N} (\vec{x}(\vec{\alpha}_i, s_i) - \vec{x}_V)^t W_i (\vec{x}(\vec{\alpha}_i, s_i) - \vec{x}_V)$$

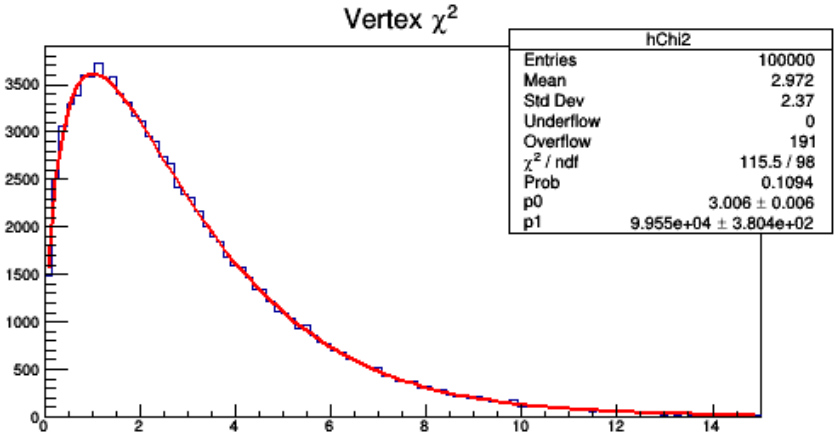
Performance (1)

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



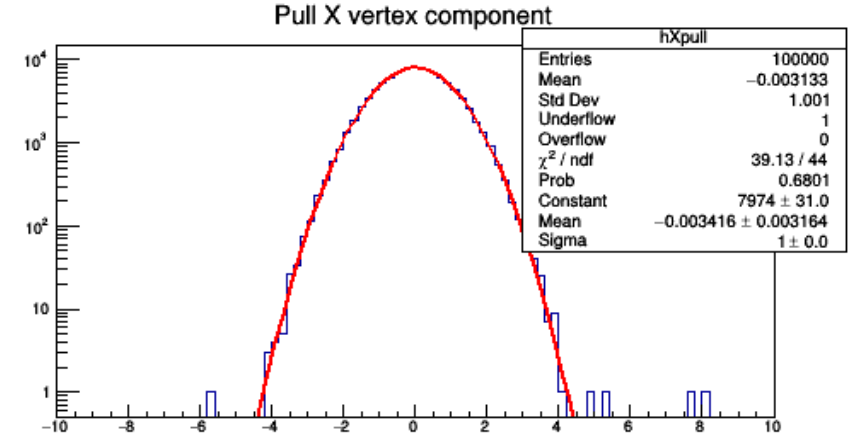


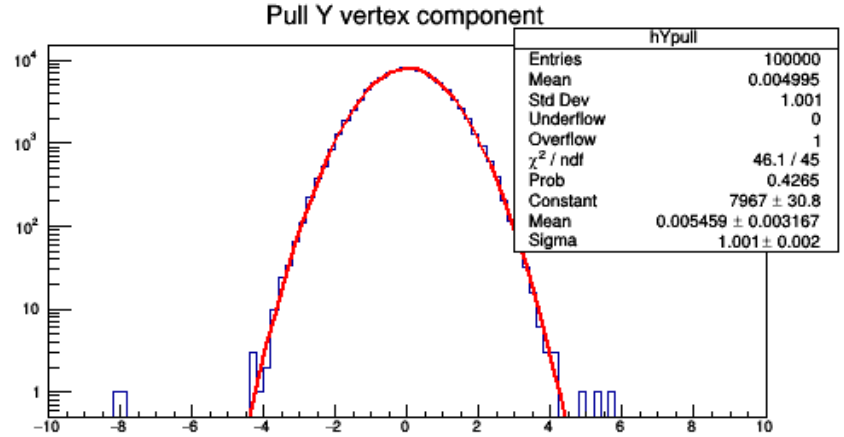


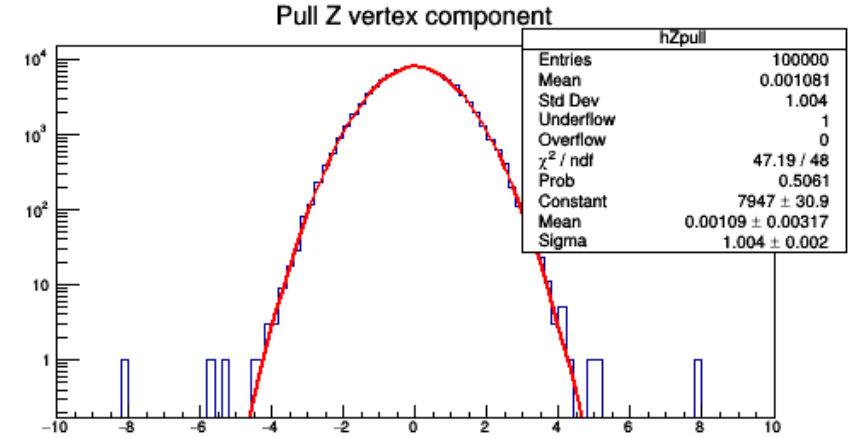


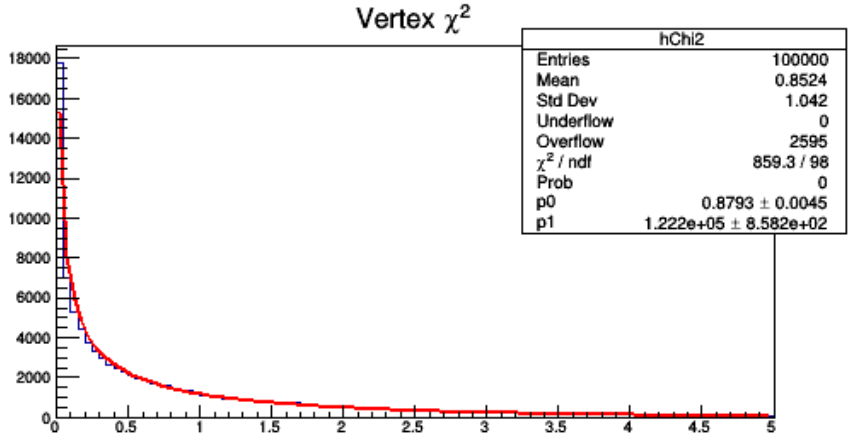
Performance (2)

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









Way forward

- Can turn routine into a class with additional features:
  - External vertex constraint (easy): useful for primary vertex
  - Adding/removing individual tracks (easy)
  - Calculate updated track parameters and their error matrix
    - Large matrix with correlations (easy)
  - Setup for tertiary vertices (more complex)
    - Calculate total vertex momentum /covariance
    - Allow for vertices with neutral tracks
- In general strong interaction with track class format
- Simple track/vertex structure to be refined