ITTF Implementation Specifics: HitErrorCalculator

Andrew Rose

ITTF Review, Sept 23, 02 - BNL

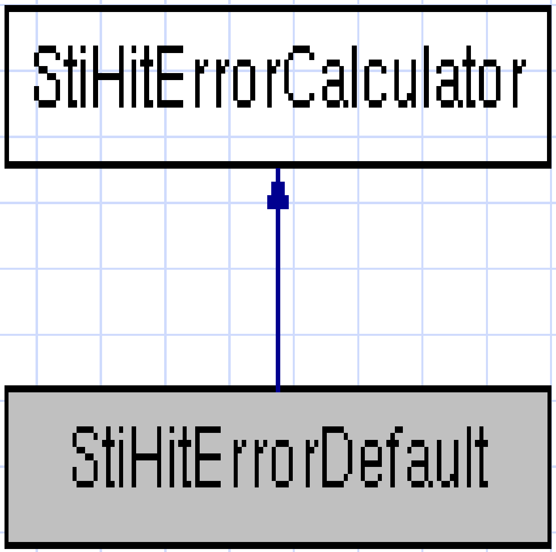
WAYNE STATE UNIVERSITY

HitErrorCalculator

- Self contained, independent class
- User sets:
  - User input parameterization (z, cross, dip)
  - DB, default, etc.

Class Hierarchy

- Virtual base class StiHitErrorDefault defines the interface
- Implementation in StiHitErrorCalculator
- Encapsulated functions represent each detector



Hit Error Parameterization

- **Hit errors calculated with M. Lisa parameterization**

  Fct of "z", crossing and pitch angles: θ_y, θ_z

- **All detectors currently parameterized using a similar scheme. In tpc,**

$$\left(\begin{array}{l} y = {}_{y,0} - {}_{y,1} \dfrac{200 - hit.z()}{\cos{}_{y}} + {}_{y,2} \tan^2{}_{y} \\ z = {}_{z,0} - {}_{z,1} \dfrac{200 - hit.z()}{\cos{}_{z}} + {}_{z,2} \tan^2{}_{z} \end{array} \right)$$

- **Requires further studies for SVT, SSD, FTPC**

ITTF Implementation Specifics: dE/dx Calculator

Andrew Rose

ITTF Review, Sept 23, 02 - BNL

WAYNE STATE UNIVERSITY

StiDedxCalculator

- Independently functioning class
- Single instance for each detector
- User sets:
  - Detector for Calculation
  - fraction for the truncated mean

GetDedx

- Arguments: StiKalmanTrack, return arguments dedx, dedx error, number of fit points used in the calculation.

Get track nodes (detector)

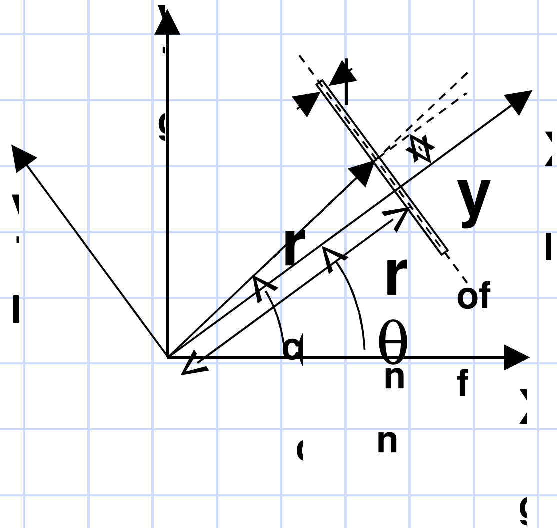
Calculate dE/dx for each node

Truncated mean: mFraction*(number of points)

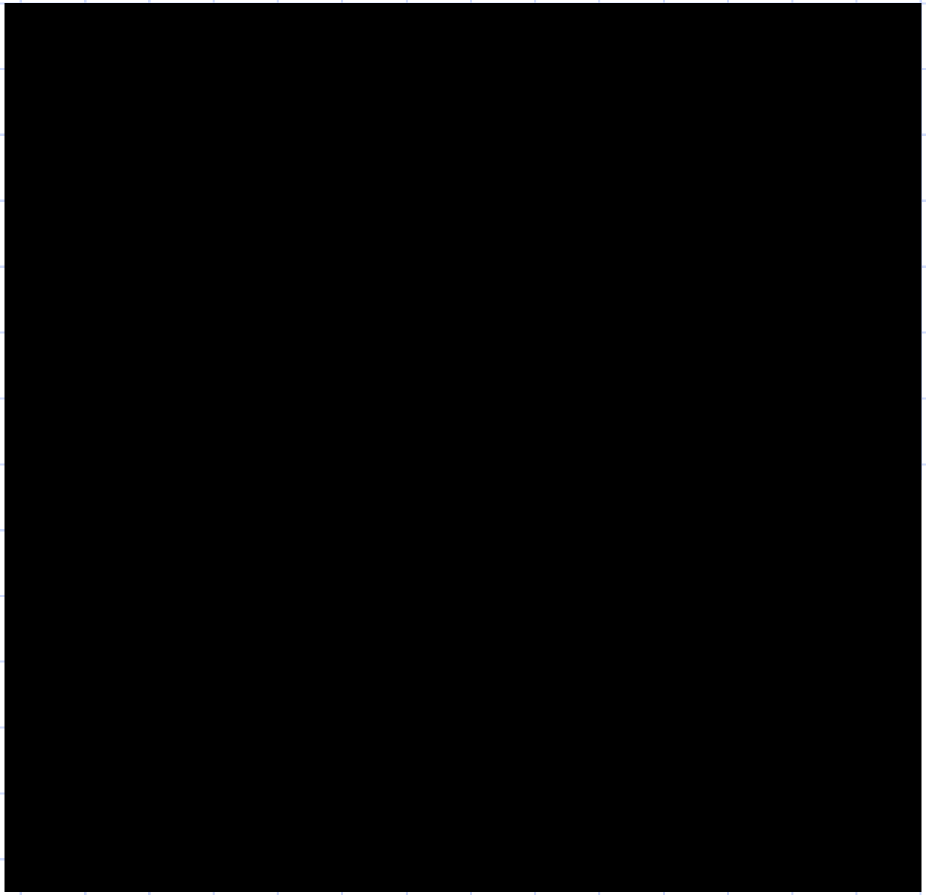
dE/dx Hit Calculation

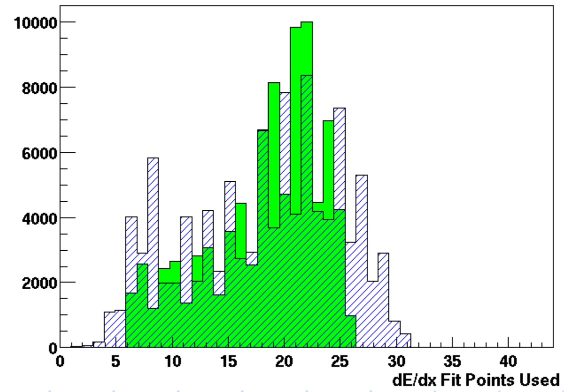
- dE from previously calibrated hit
- dx calculated from track, detector characteristics:

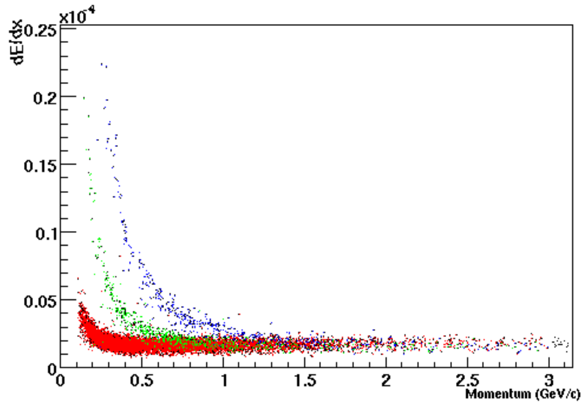
$$dx = \frac{t}{\cos(\ ) \cdot \cos(\ )}$$



dE/dx Results







Future Work

- Current dE/dx LOW! (~.6*Tpt)