Status of online filter at the SPD straw tracker

Morozova Svetlana
Samara University
SPD Physics and MC meeting
18.09.2024

Acknowledgements

The author is grateful to A. Shipilova,
A. Allakhverdieva, A. S. Zhemchugov

The goals of study

- Development of methods and creation of software for modelling the response of SPD tracker in the trigger-free regime.
- Development of algorithms for reconstruction of events in the trigger-free regime.
- Investigation of reconstruction efficiency and purity on MC simulation data.
- Development of prototype software for event reconstruction at the stage of online data filtering.

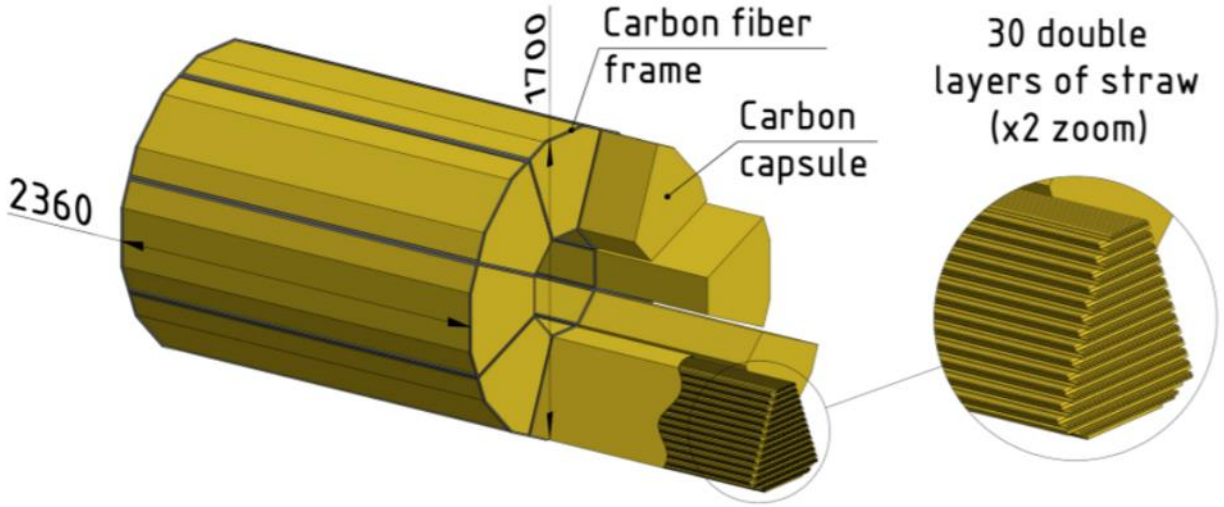


Fig.1. Schematic representation of the straw tracker

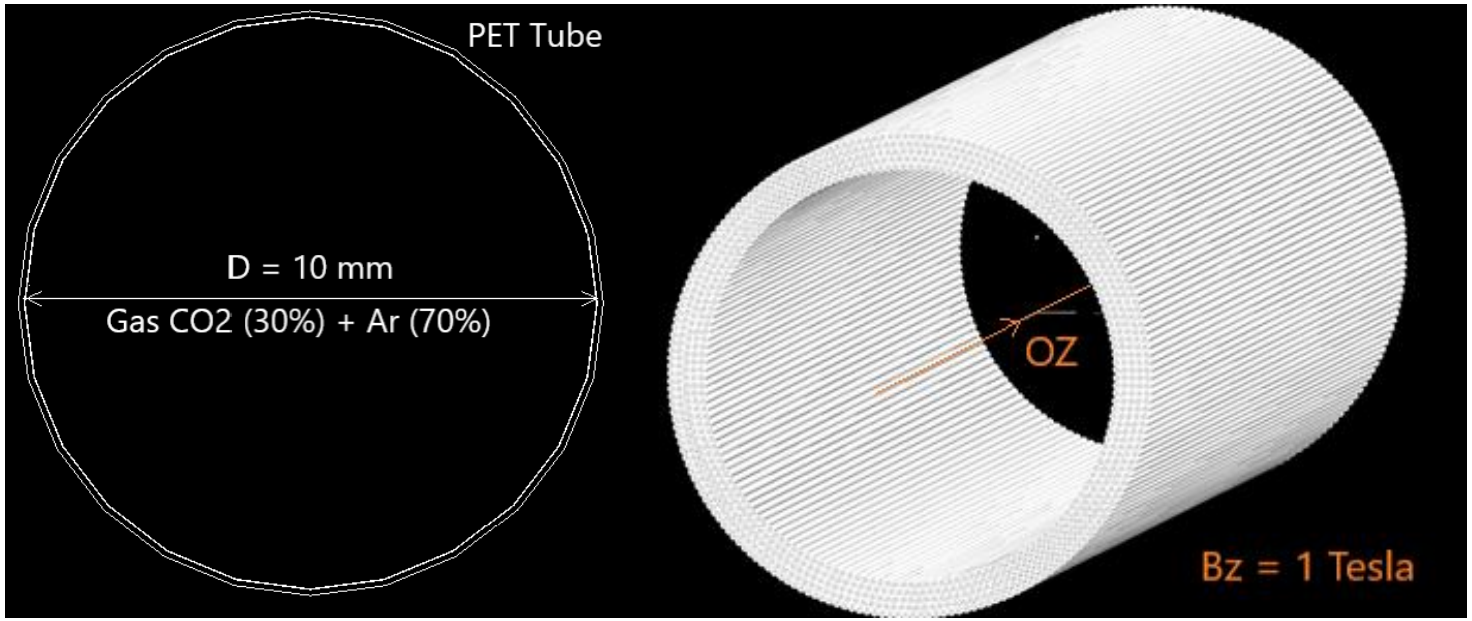


Fig.2. Straw-tube segment model(left), full detector model (5 layers) (right) built by Geant4 from the GeoModel file.

Event Generation

Time slice: 10 μs, proton bunch crossings occur every 76 ns

Primary vertex $(0, 0, Z_{pr})$ where Z_{pr} : $f(Z) = \frac{1}{\sigma\sqrt{\pi}} e^{-\frac{1}{2}\left(\frac{Z-Z_0}{\sigma}\right)}$, σ = 30 cm, Z_0 = 0

Probability of pp beam crossings: $f(k) = \frac{\lambda^k}{k!} e^{-\lambda}$, λ = 0.3

Muons:

- $E = 1\ GeV$,
- uniformly distributed in 4π space,
- number of particles: $f(k) = \frac{\lambda^k}{k!} e^{-\lambda}$, λ = 7

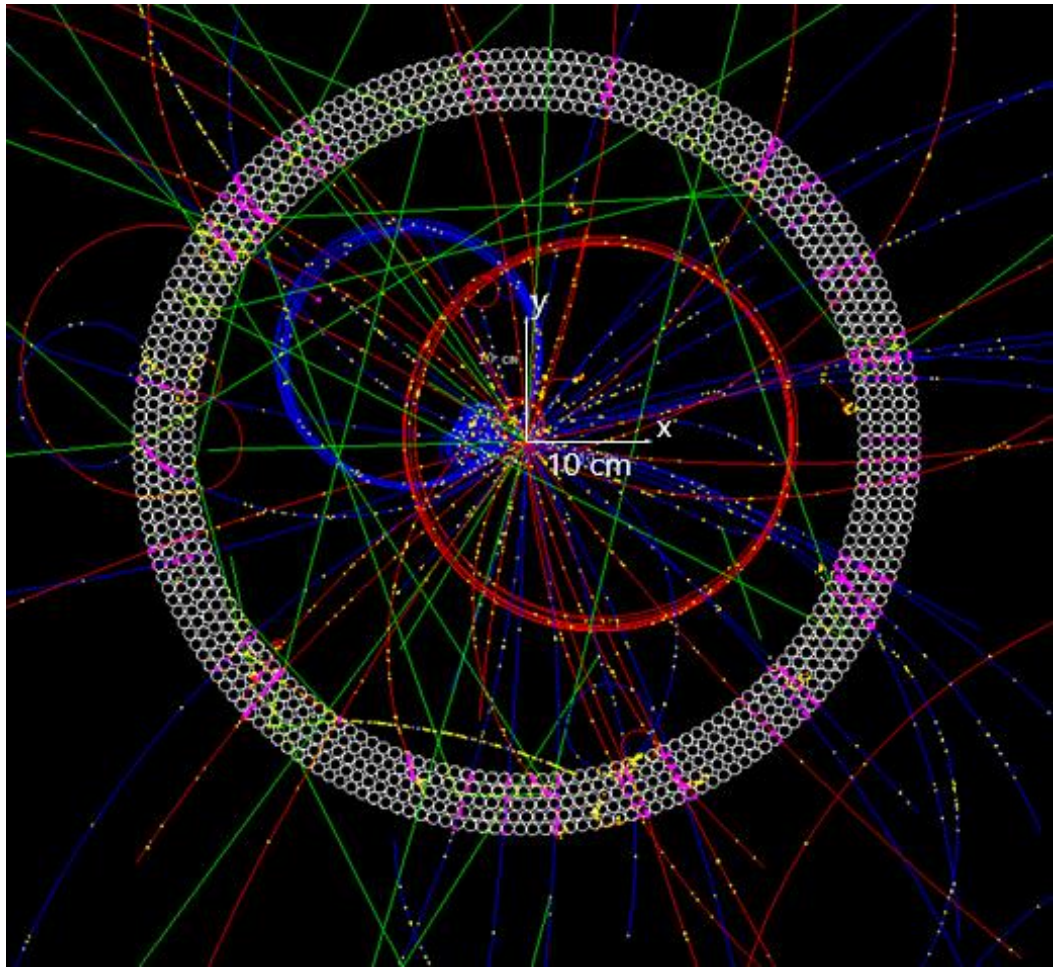
Pythia8:

- $\sqrt{s} = 27\ GeV$,
- softQCD = on

Launch 1 event: Oxy plane.

Pythia8

1 GeV muons



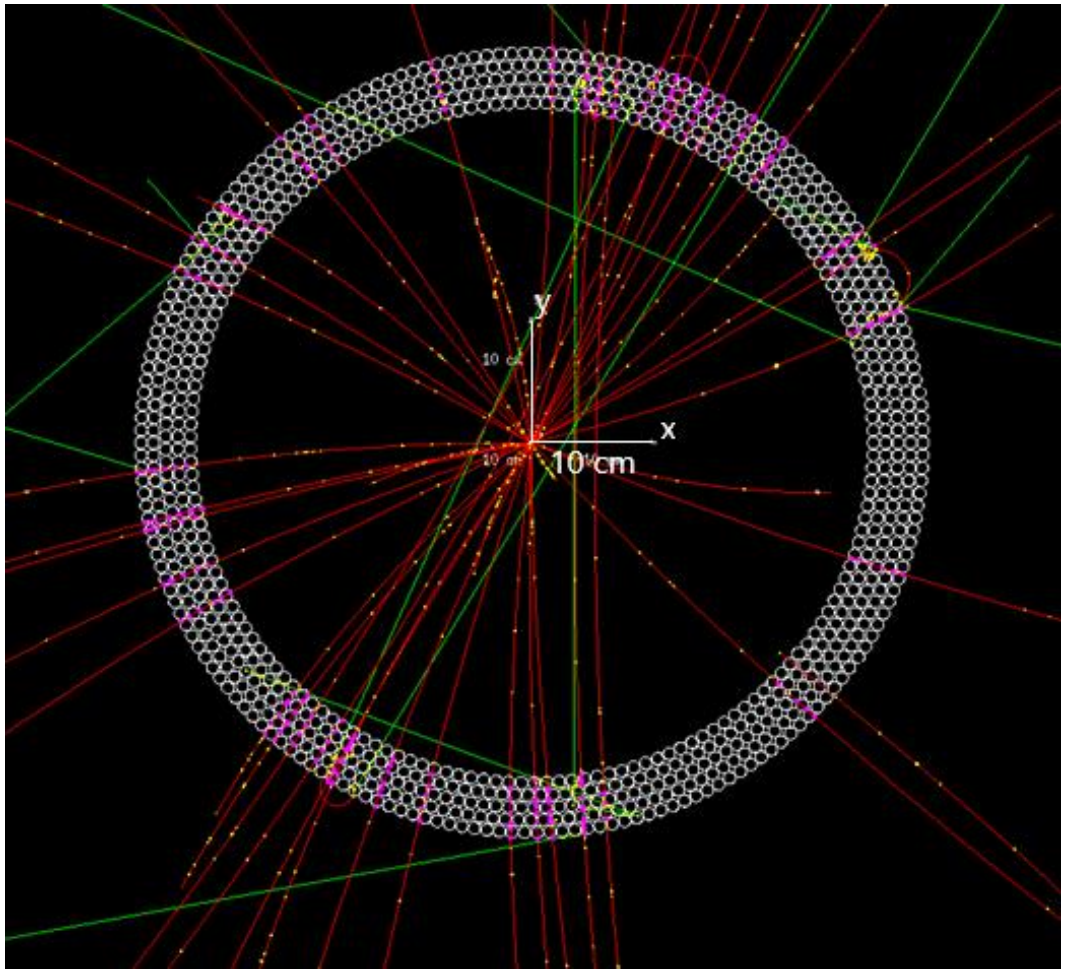
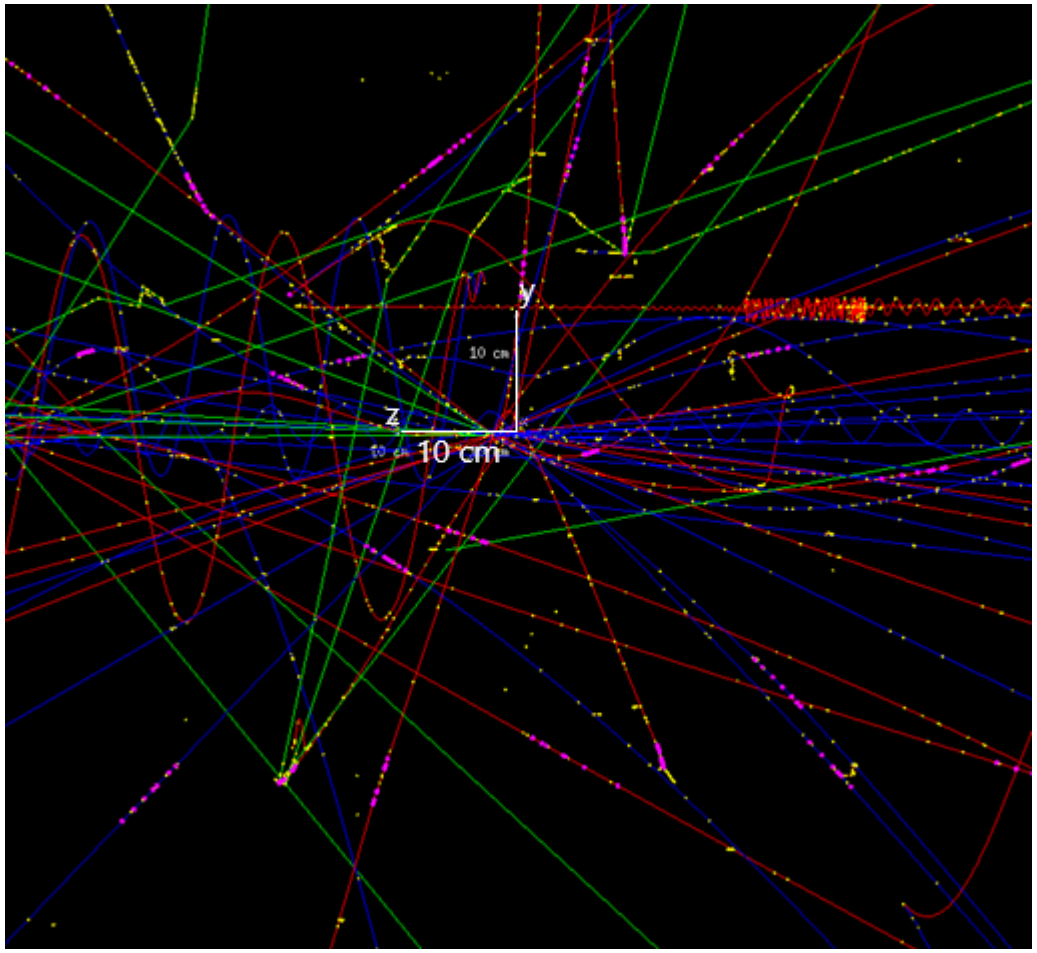


Fig.3. pp-interaction: Pythia8 event (generation) without photons (left), only muons E = 1 GeV (right). Pink dots – Hits (points of energy loss in the sensitive volume).

Launch 1 event: Ozy plane.

Pythia8

1 GeV muons



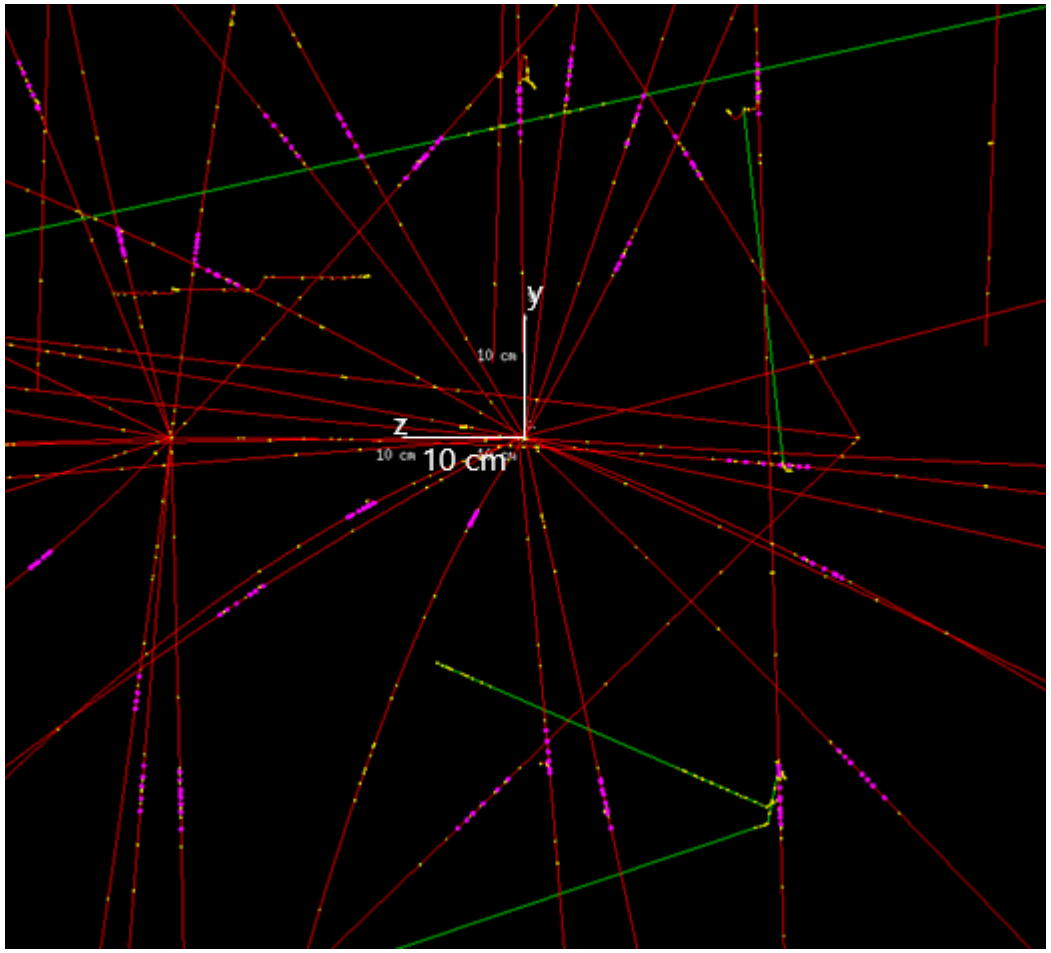


Fig.4. pp-interaction: Pythia8 event (generation) without photons (left), only muons E = 1 GeV (right). Pink dots – Hits (points of energy loss in the sensitive volume).

Selection criteria

Pythia8 establish hit:

- All primary particles except e^-, e^+
- E_k > 100 MeV

Vertex reconstruction

Step 1. Parabola approximation in Oxy

$$y(x) = a_2x^2 + a_1x + a_0$$

Step 2. Calculating the length of the parabolic segment

$$l = \int_0^X \sqrt{1 + (2a_2x + a_1)^2}\, dx =$$

$$= \frac{1}{4a_2}\left(\begin{array}{c}\log\left[\left|\sqrt{((2a_2x + a_1)^2 + 1)} + 2a_2x + a_1\right|\right] + \\ + (2a_2x + a_1)\sqrt{(2a_2x + a_1)^2 + 1}\end{array}\right)$$

Step 3. Linear function approximation

$$z(l) = b_1l - b_0$$

Step 4. Extrapolation to the Z-axis

$$Z_0 = z(0) = b_0$$

Features of approximation

In case of non-monotonic behavior of either x or y it is chosen as a function of another one

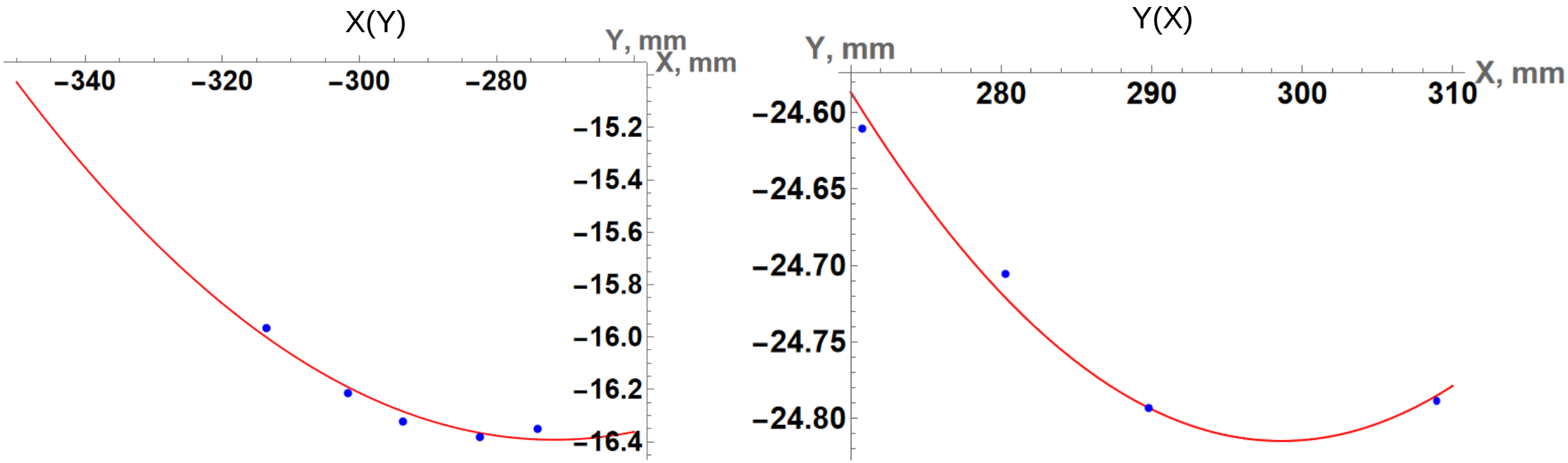


Fig.5. Parabolic approximation choices of track in XOY plane

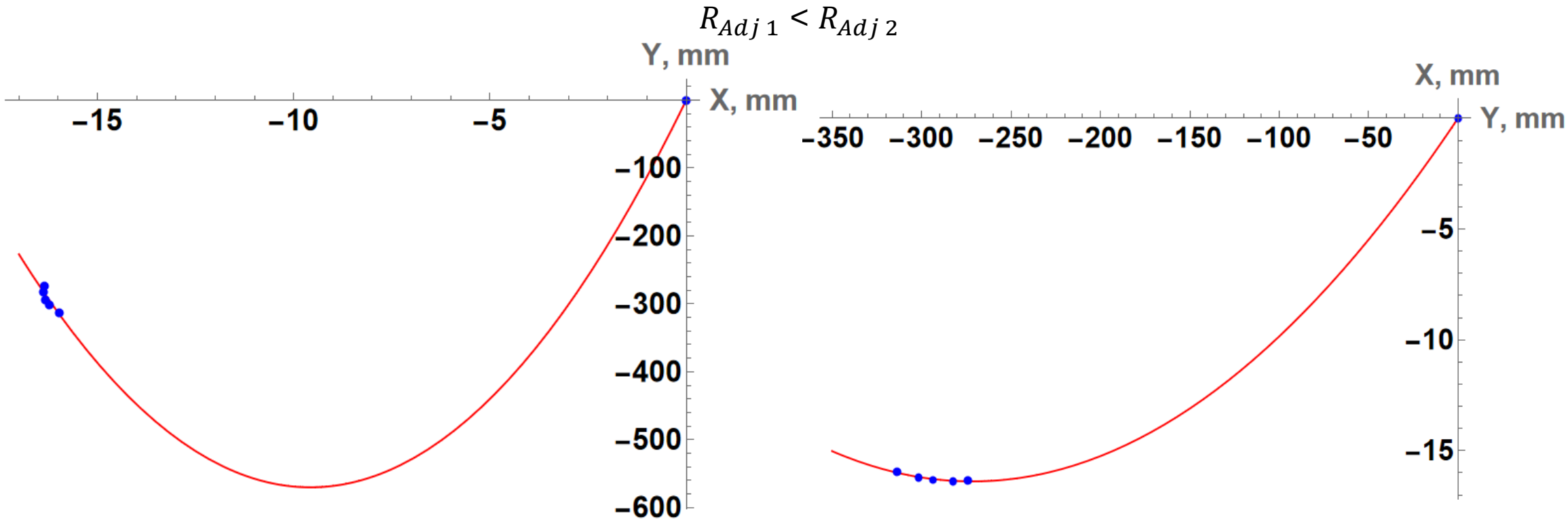


Fig.6. Approximation of the same track Y(X) (left), X(Y) (right).

$$R^2 = 1 - \frac{\sum_i \left(y_i - f(x_i)\right)^2}{\sum_i (y_i - \bar{y})^2} \qquad R^2_{Adj} = 1 - \frac{(1 - R^2)(n-1)}{n - k - 1}$$

k – number of hit points, n – polynomial degree

Track selection after approximation

$$RSS \text{ ("regression residuals squares")} = \sum_{i=1}^{n} \left(z_i - (b_0 + b_1 l_i)\right)^2$$

$RSS = 0.5$ $RSS = 1.0$ $RSS = 1.5$

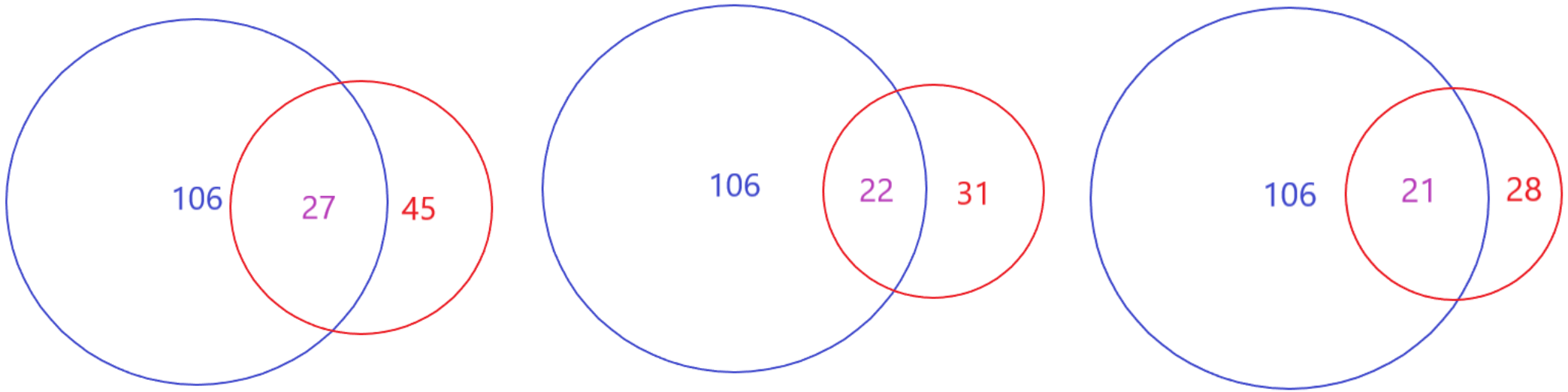
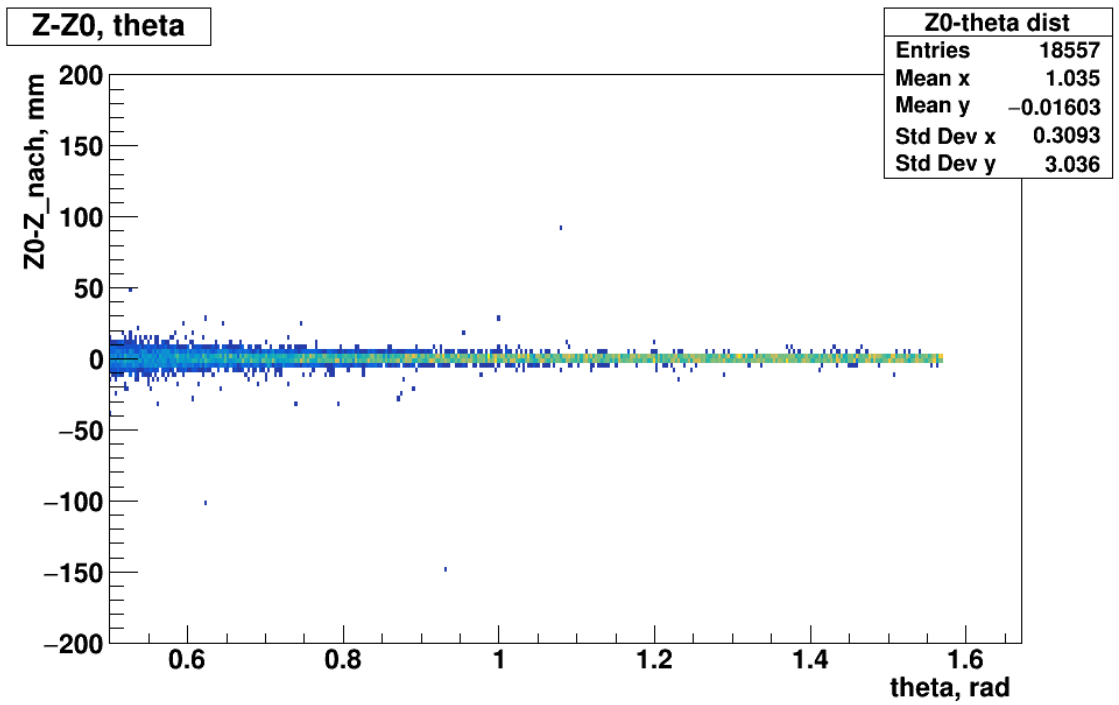


Fig. 7. Blue circle - total number of "bad" tracks ($|Z_{pr} - Z_0| > 1$ cm), red circle - number of dropped tracks by RSS $z(l)$

Muons



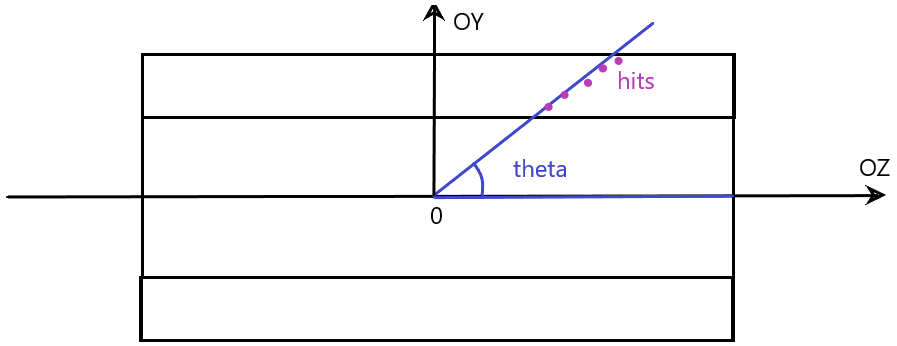


Fig.9. Visualisation of the theta angle

Pythia8

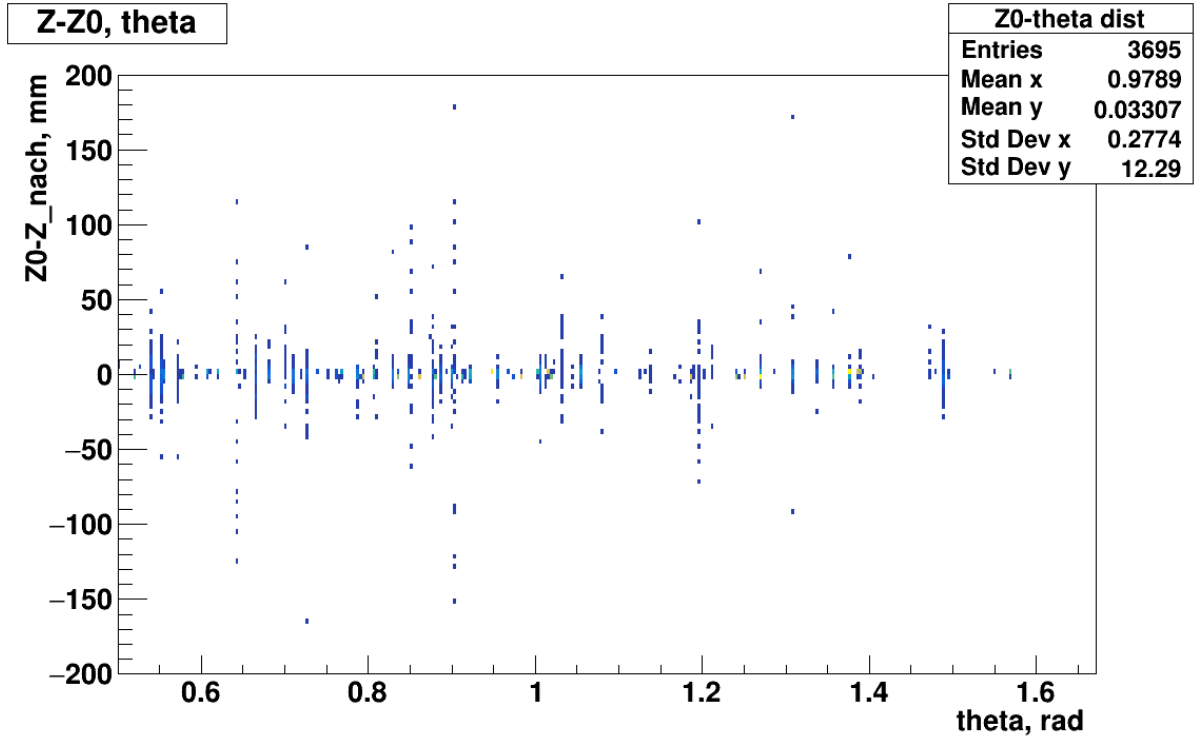
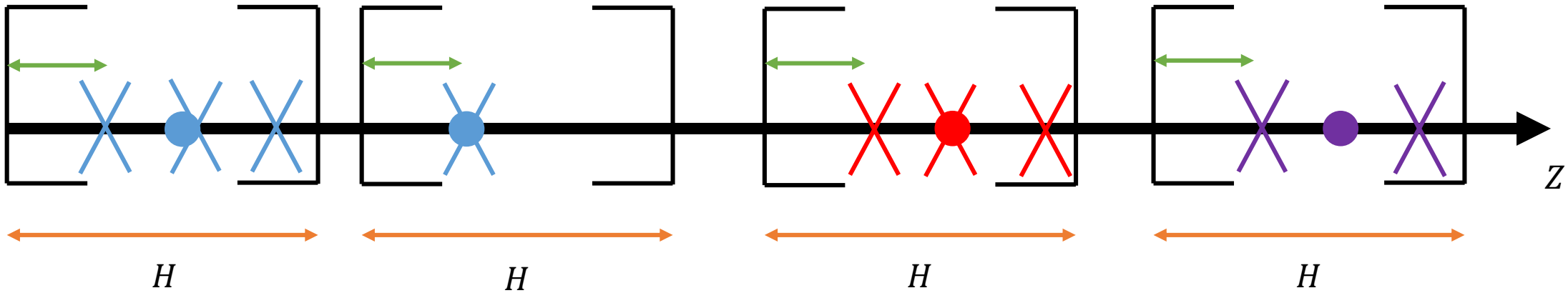


Fig.8. Distribution of the deviation of the reconstructed vertex from the true vertex along the polar angle θ muons (left), Pythia8 (right).

Reconstruction algorithm

- The Z_0 array is sorted in ascending order and divided into intervals
- The average Z_0 within the interval is the reconstructed vertex
- Search for a vertex, Z_{nach} coordinate of which belongs to the interval
- Matching unique TrackID lists

Result: $N_{true} = 3, N_{rec} = 4, N_{right} = 2$



Results

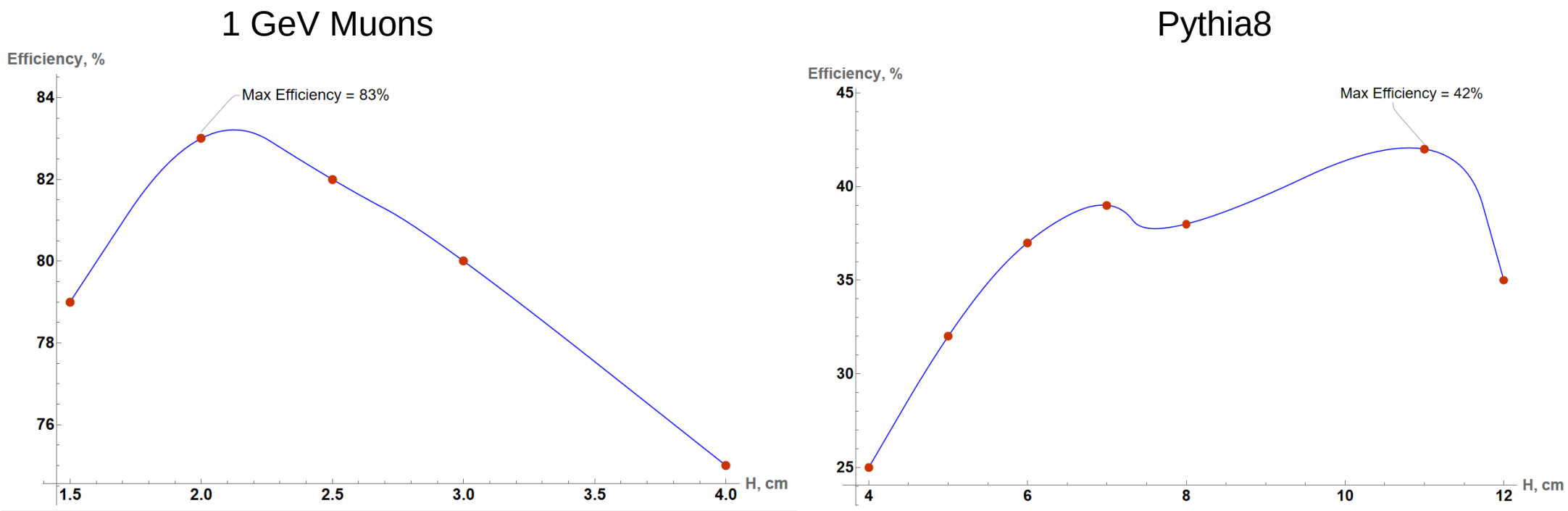


Fig.10. Track reconstruction efficiency as a function of the partitioning step H

- Maximum of reconstruction eff (muons): 83% (H = 2 cm)
- Maximum of reconstruction eff (Pythia8): 42% (H = 11 cm)

Conclusion and TODO

- The simplest and fastest possible algorithm for vertex recovery and clustering is tested
- The recovery efficiency was evaluated for muons and generation with Pythia8
- Kmeans algorithm was tried, but the clustering efficiency under the given conditions is not high enough for further application
- The next step is to improve the algorithm: complicating the rules of track selection, evaluating ‘bad’ tracks, improving the efficiency of clustering