Hit finder and track reconstruction algorithms in the Multi-Wire Proportional Chambers of BM@N experiment

S.P.Merts, **S.A.Nemnyugin**, V.A.Roudnev, M.M.Stepanova, D.P.Usov

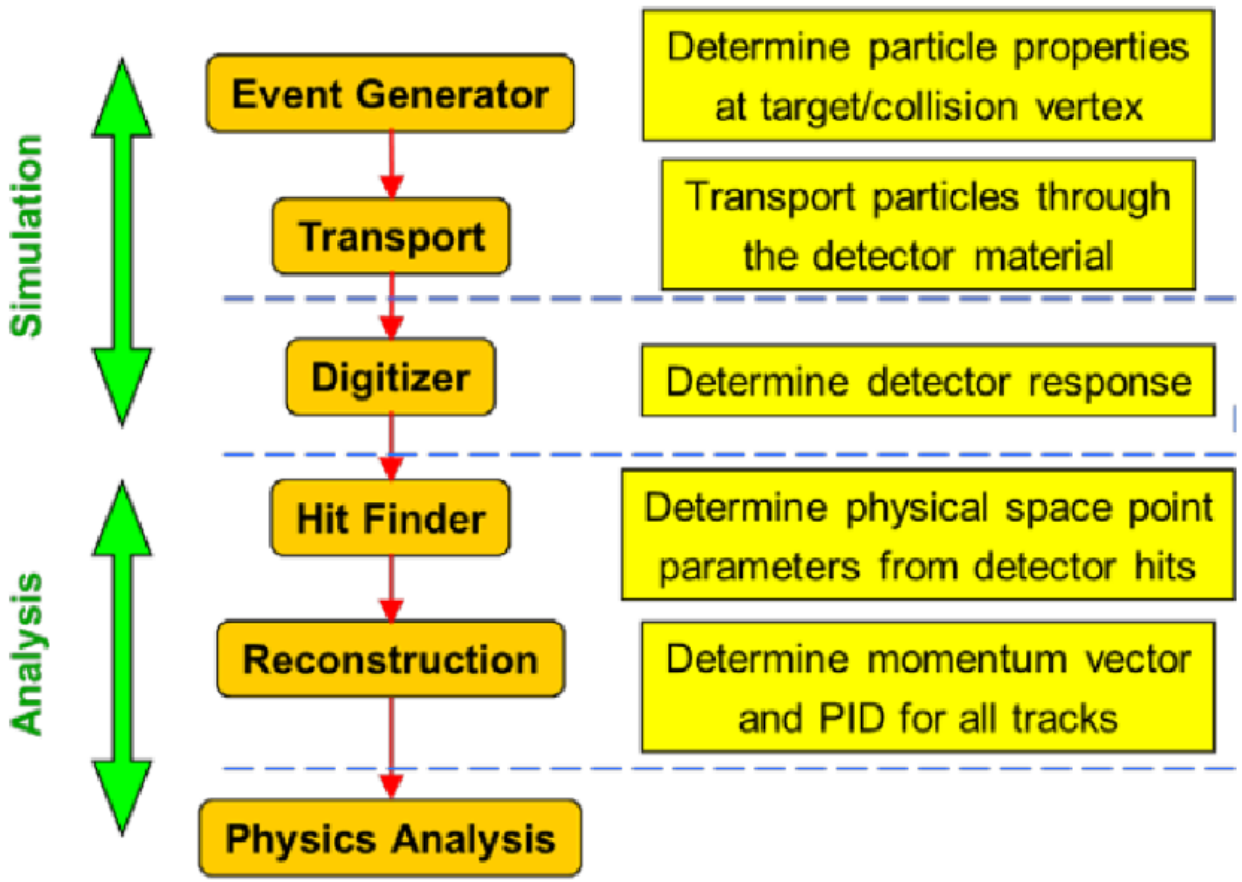
Saint-Petersburg State University

Joint Institute for Nuclear Research

27 Symposium on Nuclear Electronics and Computing 2019, Budva, Montenegro

Outline

- Realistic description of the MWPC detector geometry to describe particle interactions with material of the detector.
- Implementation of digitization – transformation of simulated data into detector signals.
- Implemetation of particle track reconstruction algorithms.



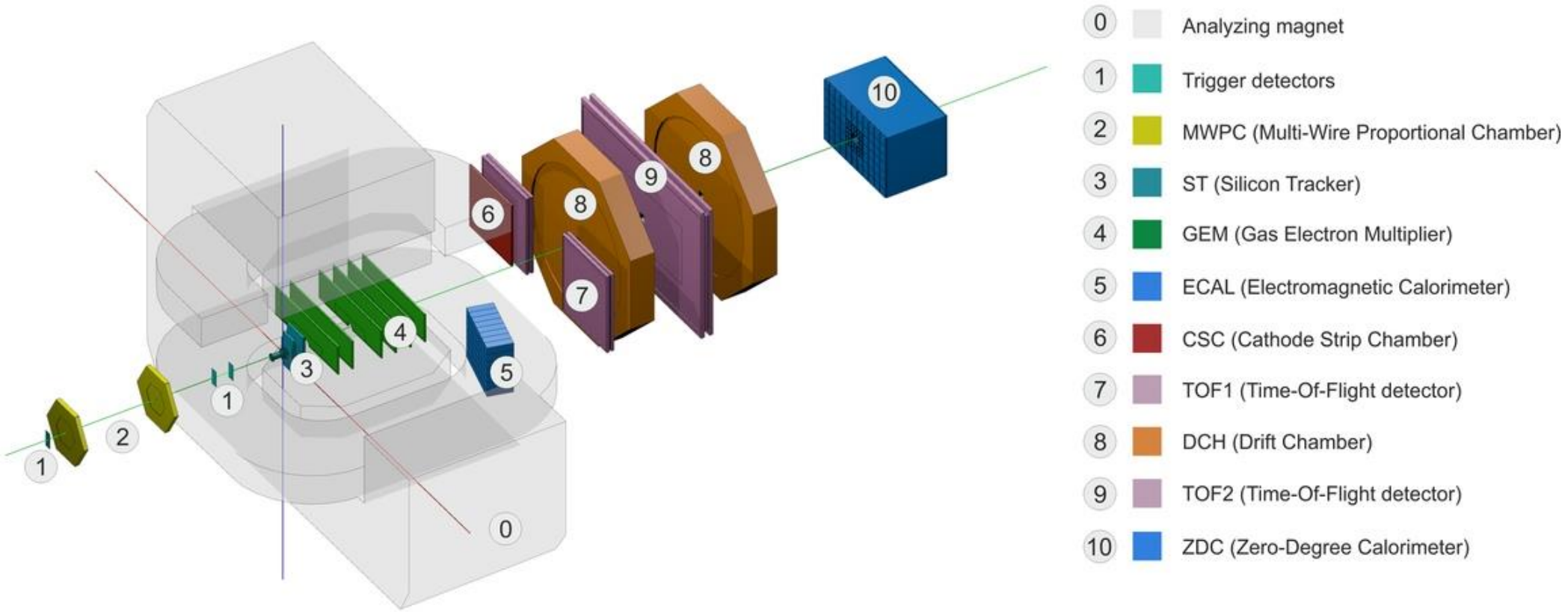
BmnRoot framework is used for the BM@N experiment processing - detector performance studies, event simulation, reconstruction and physics analysis of data.

BmnRoot is implemented in C++.
It is based on the **ROOT** environment and the object-oriented framework **FairRoot**.

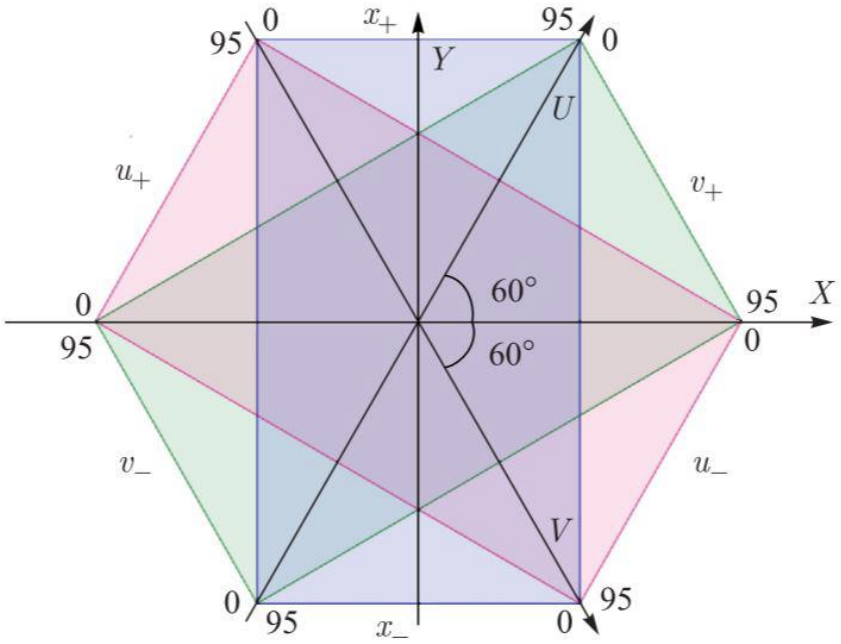
The detector inclusion in the simulation and reconstruction chain:

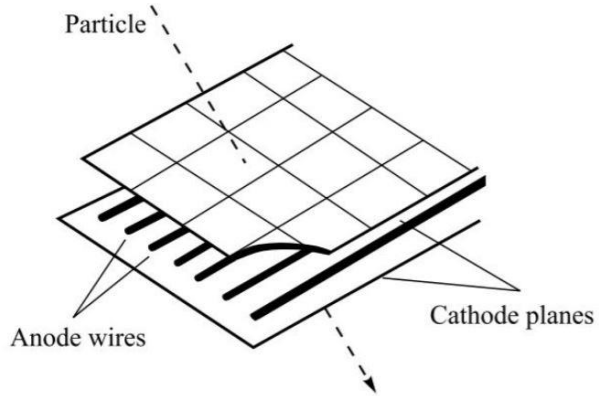
- Description geometry of detector to account particle interactions with it material.
- Digitization – transformation of simulated data into detector signals.
- Development of particle track reconstruction algorithms.

MWPC detector geometry



MWPC – MultiWire Proprotional Chamber

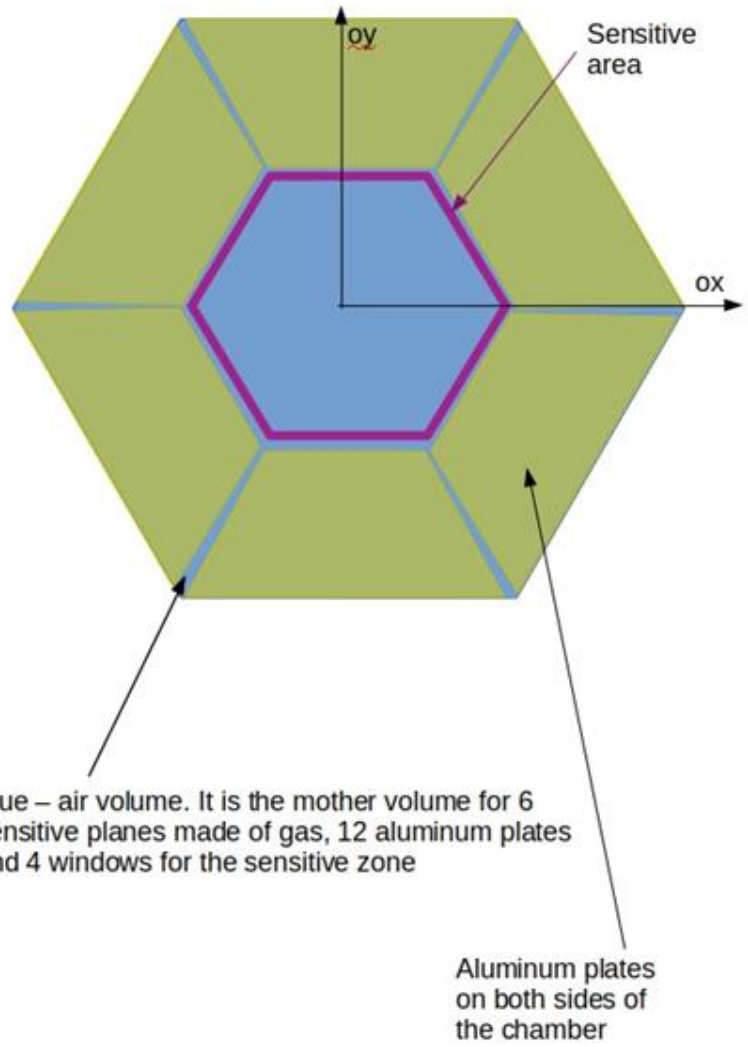




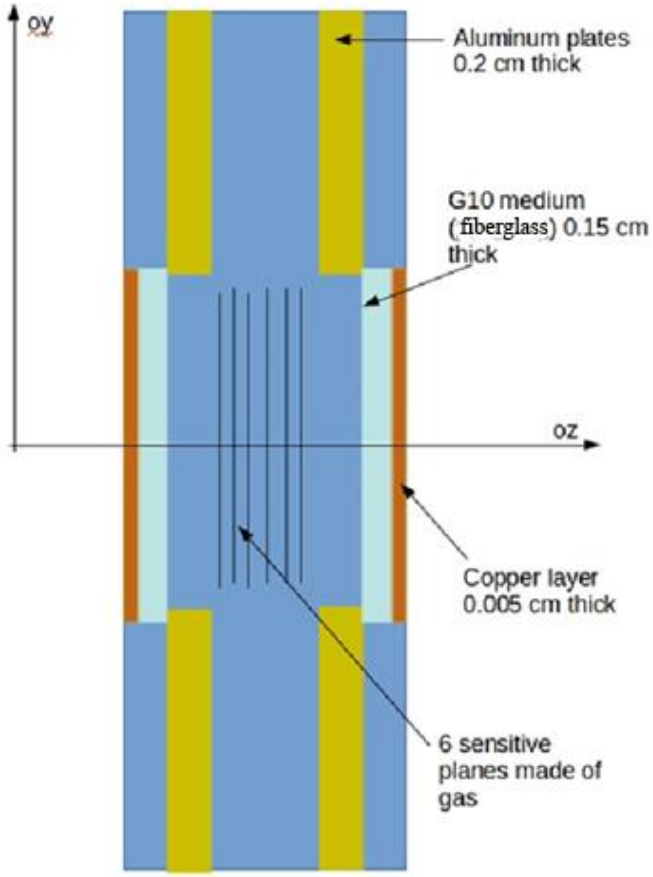
MWPC consists of 6 flat planes, each equipped with 96 wires.
Distance between wires in one plane dw = 2.5 mm.
Distance between planes in one chamber is 10 mm.
Each plane is rotated by an angle of 60° relative to the previous one.

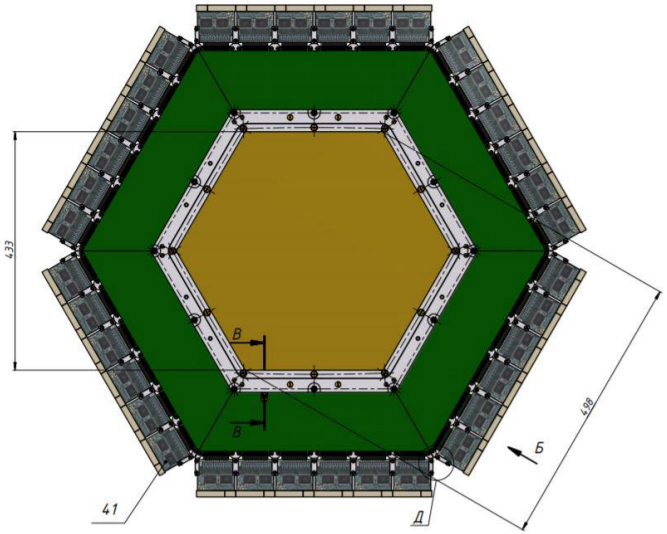
The detector is a chamber filled with a mixture of gases, in which anode wires are located between the cathode planes. Upon entry, a charged particle causes ionization of the gas, and generated electrons, under the action of the electric field of the chamber, trigger the wire closest to the particle's trajectory

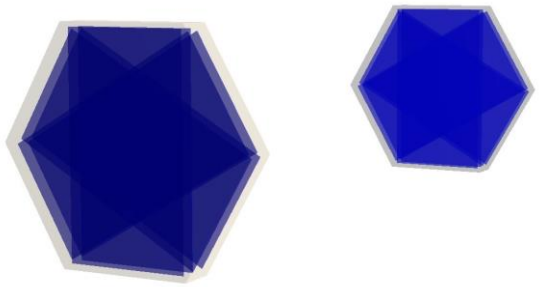
Schematic of the MWPC chamber structure.



Schematic of the MWPC chamber structure.

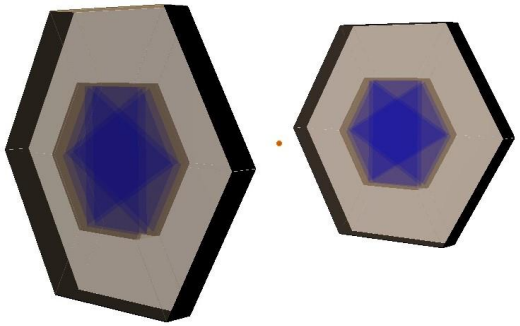




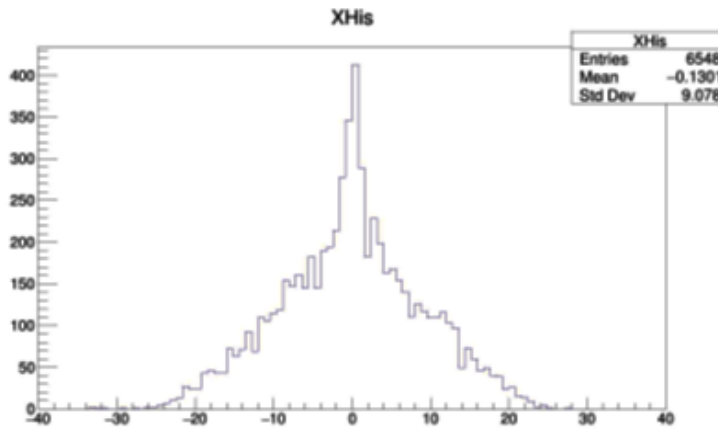


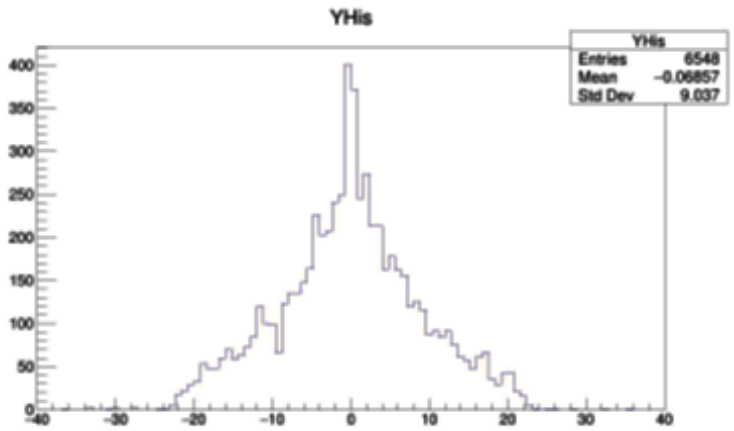
MWPC – schematic

Simulation requires the description of detector geometry, its proportions, materials and location. The **ROOT** geometry package is used for these purpose.

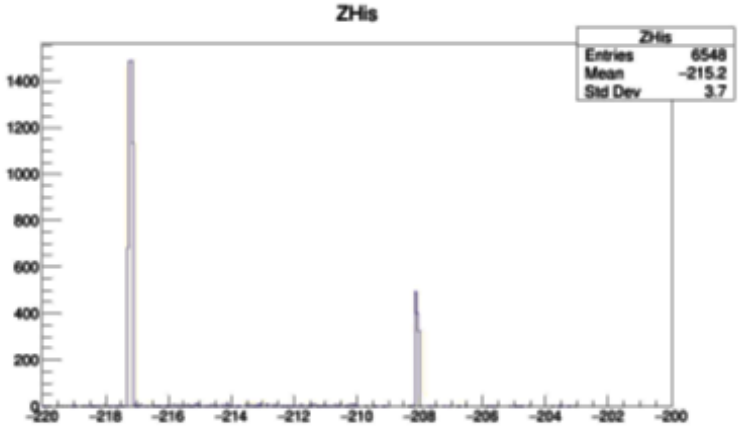


MWPC – realistic

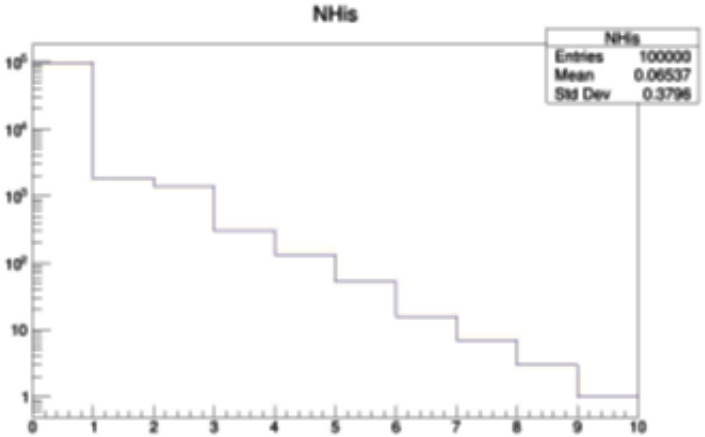




X and Y distributions of secondary particles



Z distribution of secondary particles



Number of events vs number of secondary tracks

Digitizer

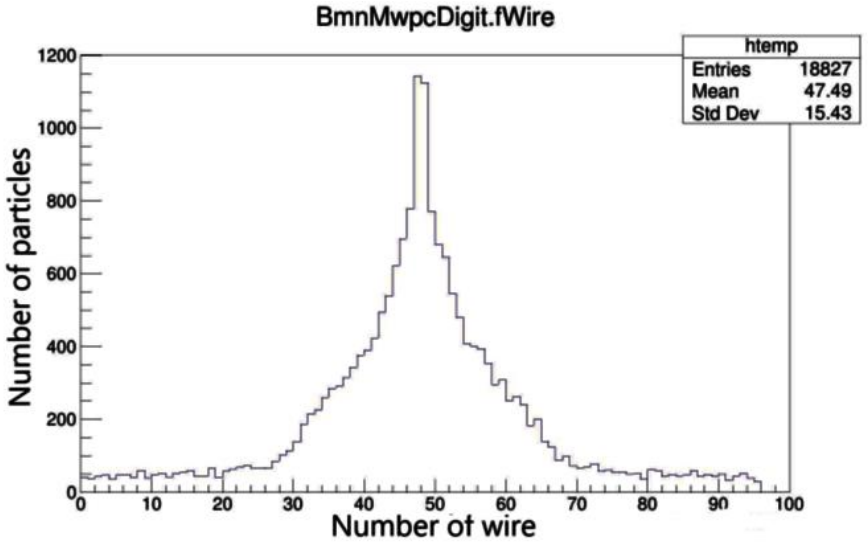
$$n_{x_+} = \frac{x}{d_w} + 47,5$$
$$n_{u_+} = \frac{u}{d_w} + 47,5$$
$$n_{v_-} = 47,5 - \frac{v}{d_w}$$
$$n_{x_-} = 47,5 - \frac{x}{d_w}$$
$$n_{u_-} = 47,5 - \frac{u}{d_w}$$
$$n_{v_+} = 47,5 + \frac{v}{d_w}$$

Digits

Primary goal of reconstruction is the processing of experimental data. However, for further tests of track reconstruction, the processing of simulated data is also important. Results of simulation should be brought in the same form that the experimental ones.

The coordinate data obtained during the simulation must be *digitized* - presented as a discrete set of wire numbers and detector planes. Based on the geometry of the detector, this can be done by rounding to integers the given expressions for different planes.

BmnMwpcDigitizer class is developed as inheritor of the FairTask class.
At each reconstruction step, the **ProcessPoints** method is called, during which the coordinates of the particles entering the detector are taken from the tree of simulated data, the numbers of wires and planes are calculated.
Then they are written to the **BmnMwpcDigits** branch of the output reconstruction tree.



Distribution of the number of secondary particles by wire numbers of digits.

Based on these graphs, it may be concluded that most of secondary particles are formed in the fiberglass protective layer on two sides of the chamber.

Hit finder

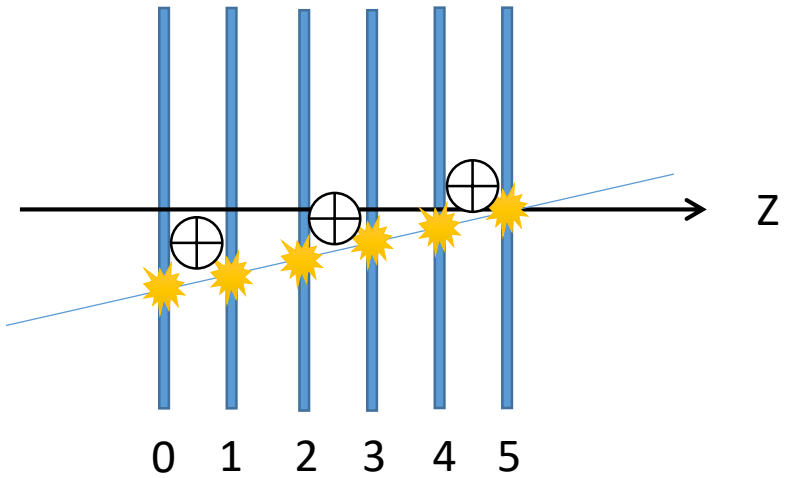
$$\left.\begin{aligned} x &= (n_{x_+} - 47,5)d_w \\ u &= (n_{u_+} - 47,5)d_w \\ y &= \frac{2u - x}{\sqrt{3}} \end{aligned}\right\} \text{Hit 1}$$

$$\left.\begin{aligned} v &= (47,5 - n_{v_-})d_w \\ x &= (47,5 - n_{x_-})d_w \\ y &= \frac{x - 2v}{\sqrt{3}} \end{aligned}\right\} \text{Hit 2}$$

$$\left.\begin{aligned} u &= (47,5 - n_{u_-})d_w \\ v &= (n_{v_+} - 47,5)d_w \\ x &= u + v \\ y &= \frac{2u - x}{\sqrt{3}} \end{aligned}\right\} \text{Hit 3}$$

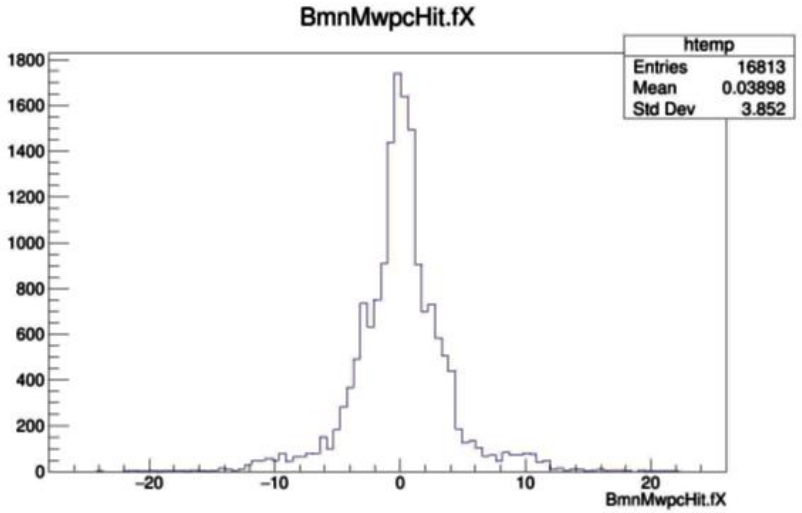
Next step is creating three-dimensional reconstructed points - *hits*, on which tracking will be built.
Tracks are built on 3 hits obtained from 6 values of the detector wire numbers, one from each plane.
The **BmnMwpcHitMaker** class for the BmnRoot framework was written.

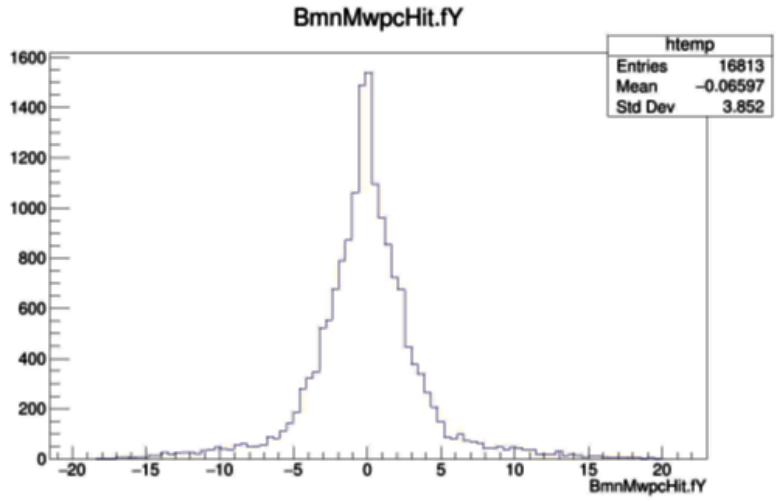
The intersections of wires on planes 0-1, 2-3 or 4-5 are converted into a hit, the z coordinate of which is taken as the average between the coordinates of the planes, and the remaining coordinates are calculated based on the detector geometry:



The proposed algorithm has drawbacks. Due to digits are taken pairwise from neighbor planes and hits are derived from wires intersections false hits may be obtained. “Dead zones” are also exists.

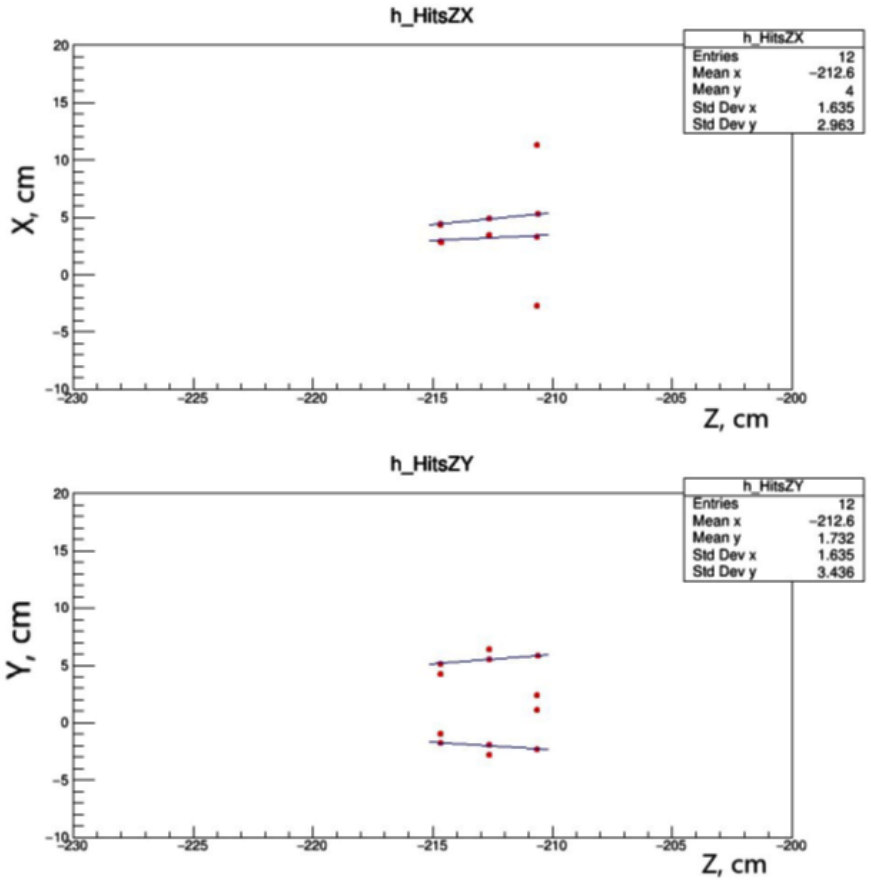
Result of **BmnMwpcDigitizer** and **BmnMwpcHitMaker** classes work.





Distribution of secondary particles by X and Y hit coordinates.

Track reconstruction



Example. Tracks reconstruction in projections XZ and YZ

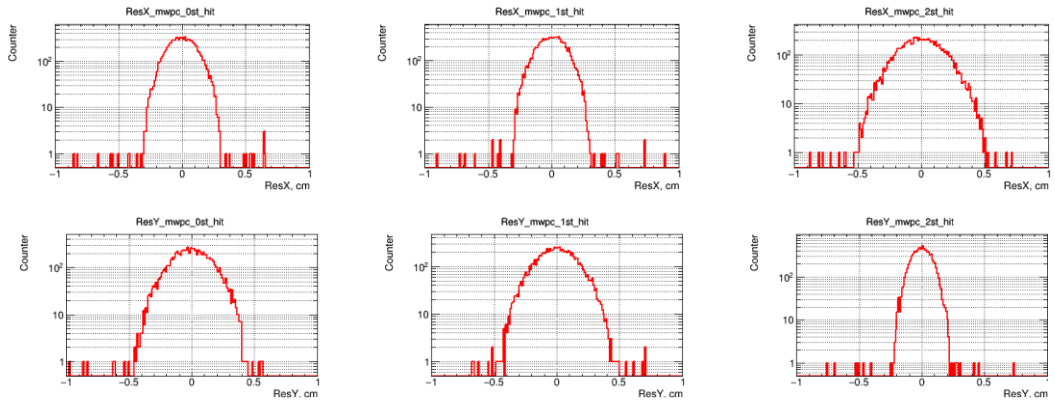
3 hits are used for the track reconstruction.

1. Data read from the **BmnMwpcHits** branch are sorted in three planes corresponding to different z coordinates.
2. All combinations of hits from different planes are approximated by straight lines using the least squares method.
3. The following functional is minimized:

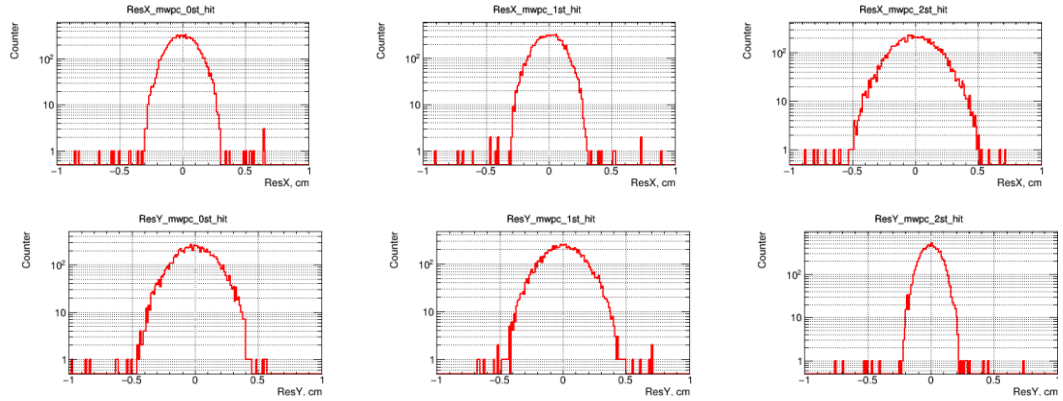
$$min \sum_i \frac{d_i^2}{\sigma^2}$$

d^2_i - distance between coordinates of hits and approximated coordinates in the i-th plane, σ^2 – standard deviation.

BmnMwpcTracking class was developed which implements the algorithm.

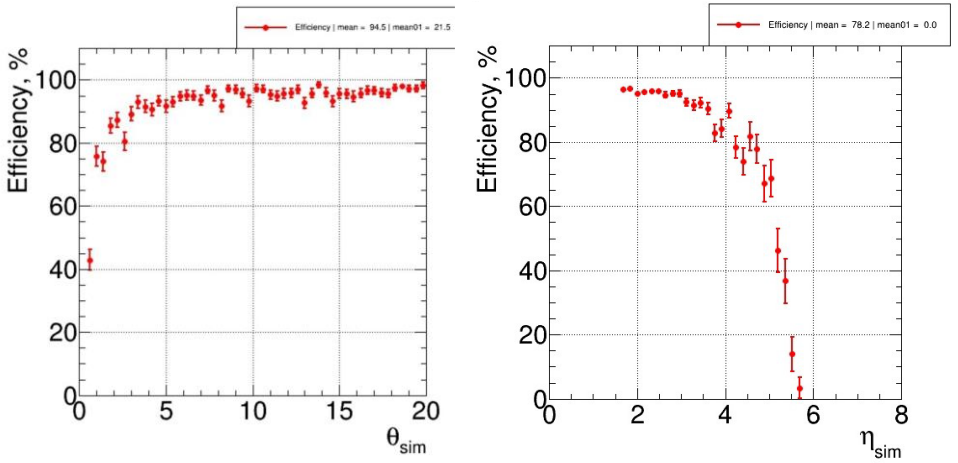


Residuals of point and hit coordinates for 1 particle in event.

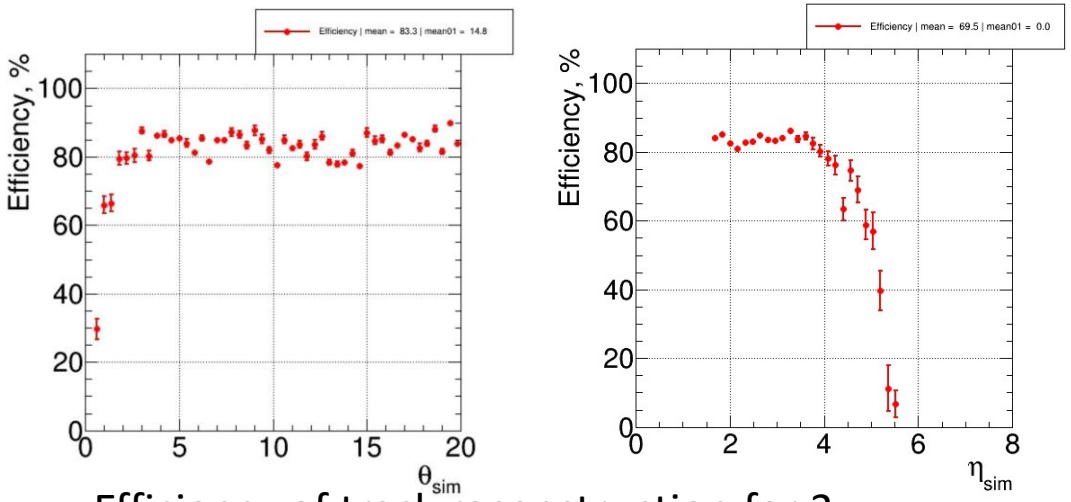


Residuals of point and hit coordinates for 3 particles in event.

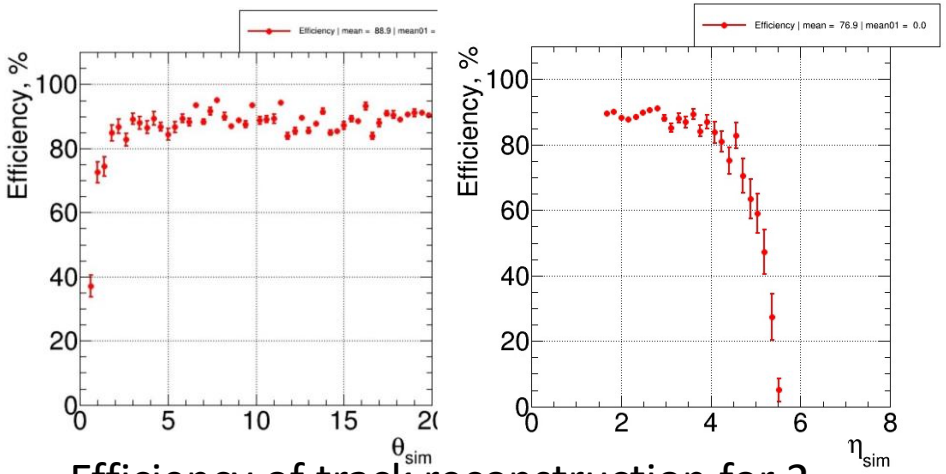
QA of tracking – comparison of hits and simulated points, residuals of point and hit coordinates.



Efficiency of track reconstruction for 1 particle in event.



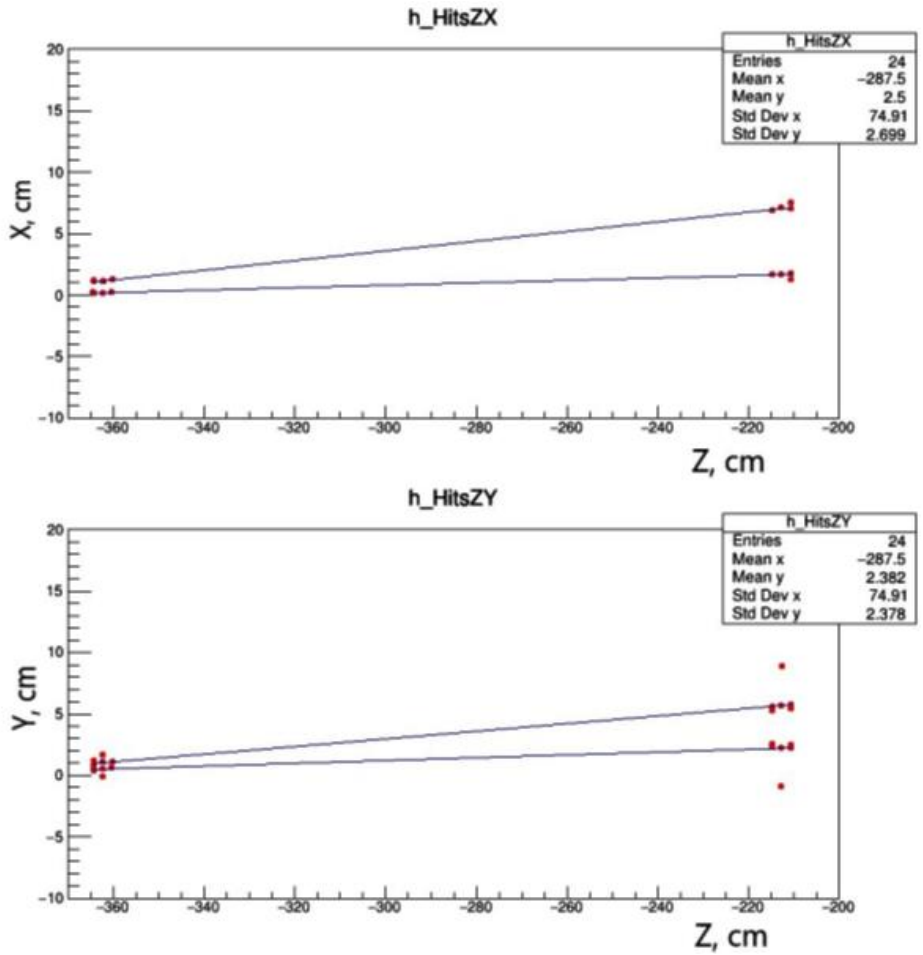
Efficiency of track reconstruction for 3 particles in event.



Efficiency of track reconstruction for 2 particles in event.

Criteria of efficiency:
If in a track more than 60% of hits refer to one simulated track, then reconstruction considered as correct.

MWPC is used for the beam monitoring so effects of multiplicity are not so important.



Example of merge between tracks in two chambers in two projections XZ and YZ.

Merger between tracks in system of chambers.

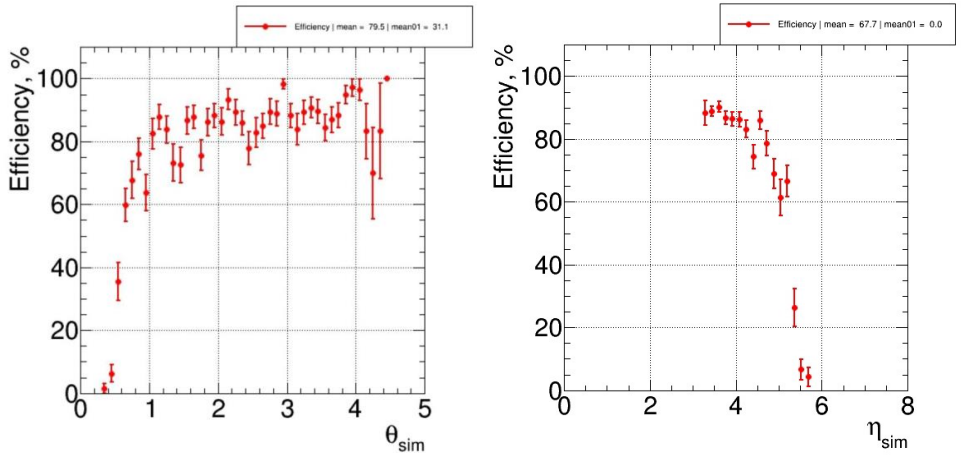
After building tracks in proportional chambers, it is necessary to combine them.

Tracks are extrapolated to two planes corresponding to

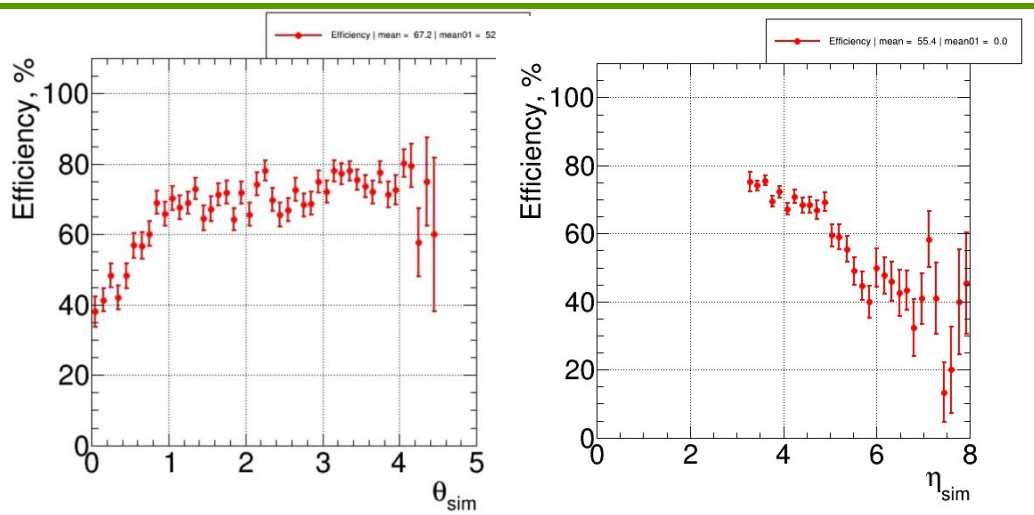
$$Z_1 = \frac{3Z_{ch1}+Z_{ch2}}{4} \text{ and } Z_2 = \frac{Z_{ch1}+3Z_{ch2}}{4}.$$

The best set of tracks is located at the minimum average distance between extrapolated coordinates with a small difference in angle.

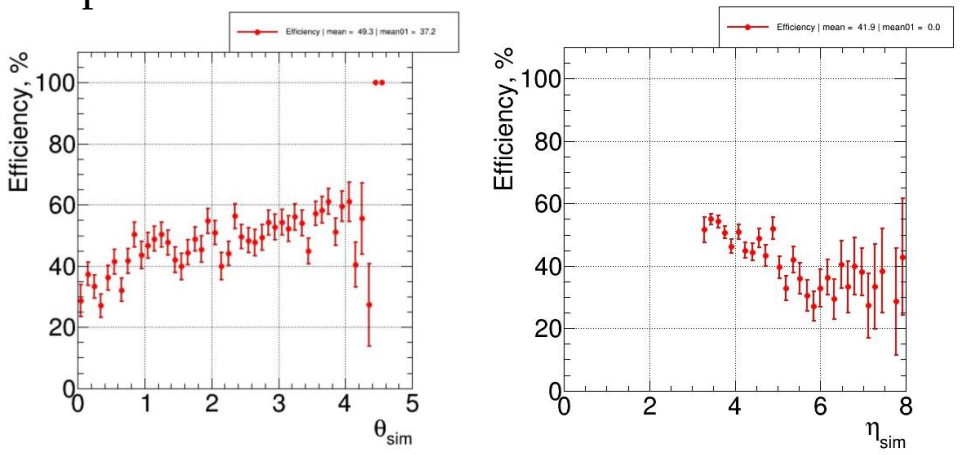
This set of two tracks is converted into a new one.



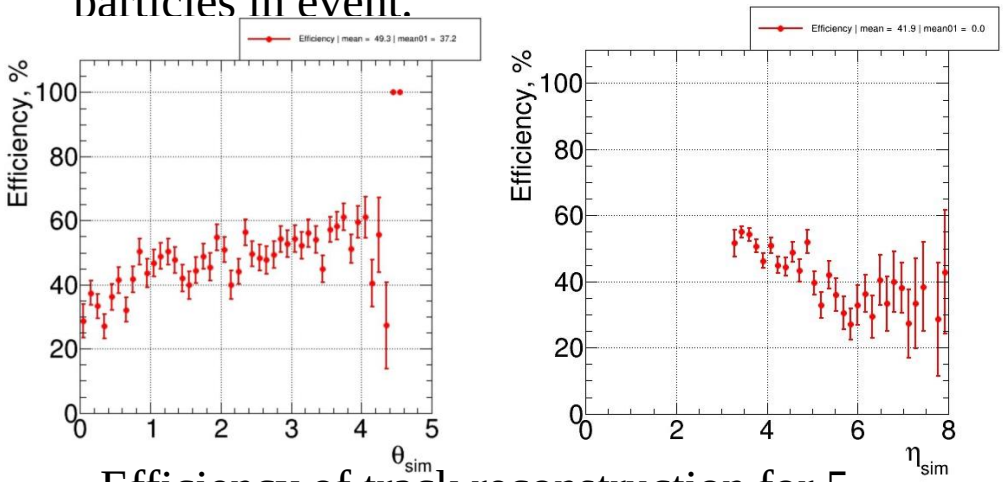
Efficiency of track reconstruction for 1 particle in event.



Efficiency of track reconstruction for 2 particles in event.



Efficiency of track reconstruction for 3 particles in event.



Efficiency of track reconstruction for 5 particles in event.

Conclusion

- Realistic description of the MWPC detector geometry is developed to account most of its components. Effect of materials around the camera on the appearance of secondary particles in it is evaluated.
- Algorithms for digitizing the simulated data and finding reconstructed hits, on which tracking is built, were developed and implemented as C++ classes for BmnRoot framework.
- Track reconstruction algorithm and an algorithm for merging tracks in a camera system have been developed and implemented as C++ classes for BmnRoot.
- Quality of reconstruction was studied.
- *We plan* to improve developed algorithms, minimizing “dead zones” and inreasing efficiency.
- *We plan* to optimize performance of implemented algorithms.
- *Sources will be uploaded in repository of the BM@N (BmnRoot).*

Presented results are part of the project supported by the RFBR grant #18-02-40104.