Status on SDT simulation work

Contents

- A first trial to see what happen for the IP resolution from the CEPCSW
- A simple circle fitting to X-Y plane of the hits are used, after some investigations.

- #Although I have separately started to test with the latest CEPCSW) the results in this slide was obtained from already existing simulated rootfiles.

- # only DCH hits are used, for confirmation of circle fitting

Confirmation of the CEPCSW circumstance

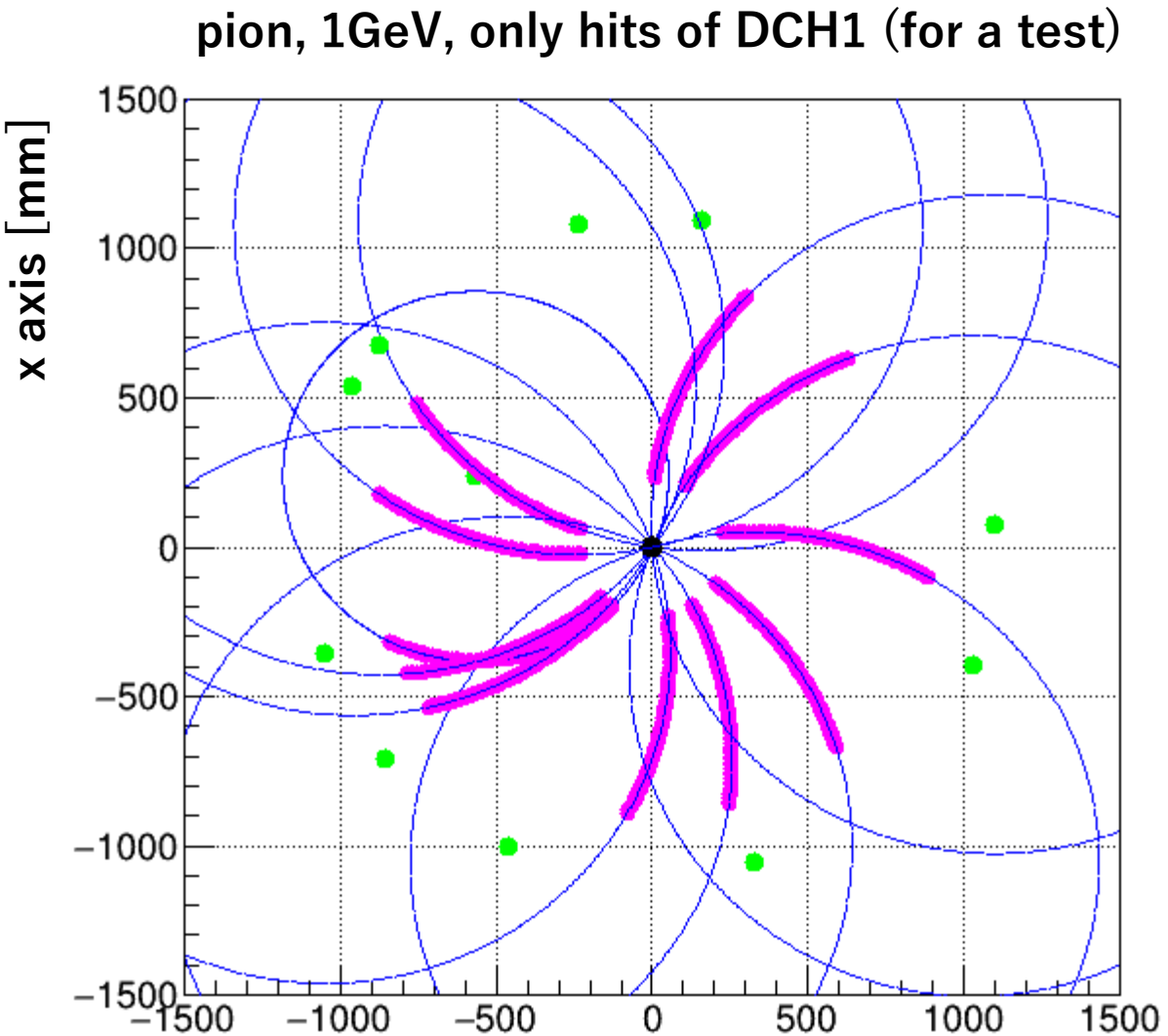
Fitting to the hits

-- Using MarlinTrk/HelixFit
(temporally modified “accordingly”)

-- To avoid multi-tracks, cuts on number of hits is applied on the events to select single track for fit
(therefore, ~ 1500 ? events of 2000events were fitted)

-- the weight is given as arbitrary set values, $w=1/\sigma^2$, where $\sigma=0.2\mu m$
(as long as one detector type, it might be no effect …)

-- input values are x, y hit positions. Z coordinate is not considered at this moment



(pink is showing hits, where as the blue curve is fit result)

Fitting to the hits: HelixFit routine

```
//  Created by Steve Aplin on 9/16/11.
//  DESY
//
//  C++ rewrite of the aleph Fortran routine TFITHL
//
//! Fast helix fit
//
//
//   Input:  NPT         Number of 3-D points to be fit
//           xf      Array of X-values of points to be fit
//           yf      Array of Y-values of points to be fit
//           zf      Array of Z-values of points to be fit
//           wf       Array of 1/(sig(rphi))**2 for each point
//           wzf      Array of 1/(sig(z))**2 for each point
//           iopt     < 3 : error matrix calculated
//                    = 3 : 3-dimensional iteration
//
//  OUTPUT: vv0    = Helix parameter in perigee form
//          ee0    = INVERSE OF ERROR MATRIX IN TRIANG. FORM
//          chi2ph = CHI SQUARED = SUM (PHI DEVIATIONS/ERRORS)**2
//          CH2Z   = CHI SQUARED = SUM (Z DEVIATIONS/ERRORS)**2
//  NOTE: DEGREES OF FREEDOM = 2*NPT-5
//-----------------------------------------------------------------
//    BASED ON  SUBROUTINE CIRCLE
//    REFERENCE:  COMPUTER PHYSICS COMMUNICATIONS VOL 33,P329
//
//   AUTHORS:  N. CHERNOV, G. OSOSKOV & M. POPPE
//   Modified by:  Fred Weber, 8 Jun 1989
//   Modified by:  M.Cattaneo, 27-Jan-1998
//              Protect against arg SIN > 1.0
//
```

MarlinTrk/HelixFit.h

Computer Physics Communications 33 (1984) 329–333
North-Holland, Amsterdam

EFFECTIVE ALGORITHMS FOR CIRCLE FITTING

N.I. CHERNOV, G.A. OSOSKOV

Joint Institute for Nuclear research, Head Post Office, P.O. Box 79, 101000 Moscow, USSR

Received 20 February 1984

After an investigation of known methods for circle fitting two new effective algorithms are proposed which meet all the requirements of mass data handling.

1. In many data handling problems there is a need to approximate data points by a circle. Some cases (pictures of lunar craters [1], eye cornea surface [2], etc.) are related to circle-like images, but more frequently a circular arc is used as an approximating curve that possesses some advantages when compared with other curves even allowing for their simplicity in fitting.

The circle is better for such a prediction than the parabola, for instance, because the circle retains its curvature. It is necessary to point out the extremely rigid requirements on the efficiency of the fitting algorithm in tracking problems. This follows from the very high rate at which the fitting procedure is called ($\approx 10^3$ times for each frame, with 10^5–10^6 frames for one experiment).

Ref:

Let us derive the corresponding formulae which will be needed later. To simplify the notation les ut denote by Gauss brackets expressions like

$$\sum_{i=1}^{n} x^p y^q = [x^p y^q]$$

and suppose that the origin of the coordinate system has been transferred to the centre of gravity of the point set (x_i, y_i); hence $[x] = [y] = 0$.

not sure many parts, such as taking average by weight .... (what values should be for the weights ?)

example

```
for (int i =0; i<npt; ++i) {
    xi = xf[i] - xm;
    yi = yf[i] - ym;
    xx = xi*xi;
    yy = yi*yi;
    x2 = x2 + xx*wf[i];
    y2 = y2 + yy*wf[i];
    xy = xy + xi*yi*wf[i];
    dd = xx + yy;
    xd = xd + xi*dd*wf[i];
    yd = yd + yi*dd*wf[i];
    d2 = d2 + dd*dd*wf[i];
  }
```

$$Fa + Hb - a\gamma = P,$$
$$Ha + Gb - b\gamma = Q, \qquad (9)$$
$$2Pa + 2Qb + \gamma^2 = T,$$

where as above $\gamma = R^2 - a^2 - b^2$. Using again the Gauss brackets we denote

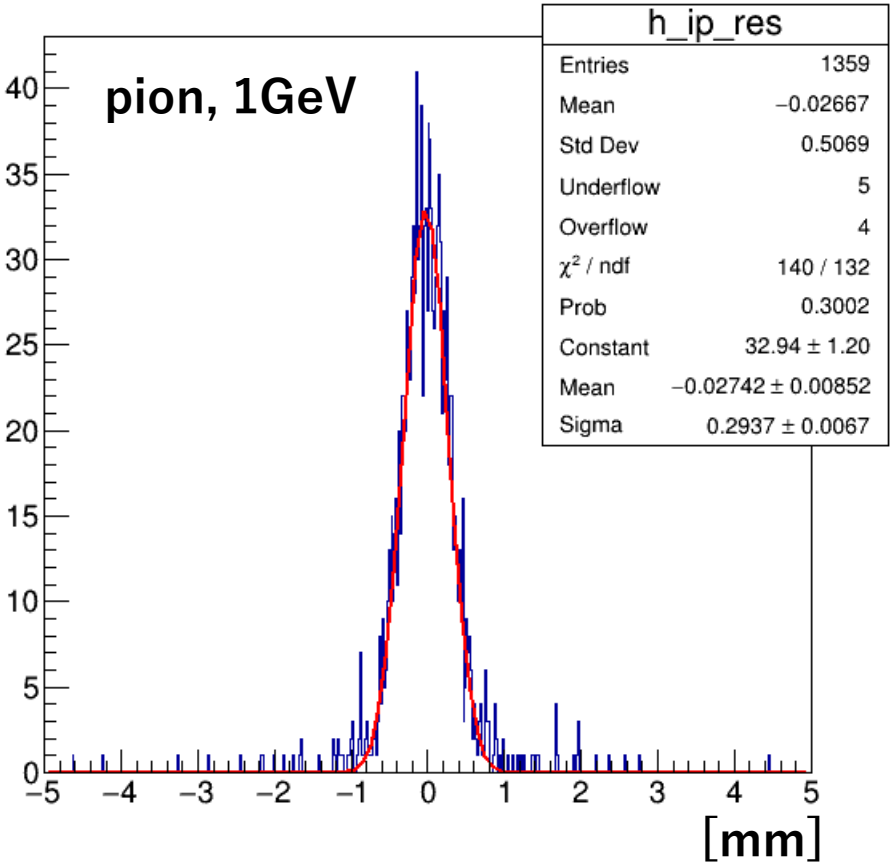
$$F = \frac{1}{n}[3x^2 + y^2], \quad G = \frac{1}{n}[x^2 + 3y^2],$$
$$H = \frac{2}{n}[xy], \quad P = \frac{1}{n}[x(x^2 + y^2)],$$
$$Q = \frac{1}{n}[y(x^2 + y^2)], \quad T = \frac{1}{n}[(x^2 + y^2)^2].$$

In the system (9) one can exclude any pair of unknown variables to get an equation of the 4th degree. The most suitable choice is to exclude a, b and thereby obtain the equation

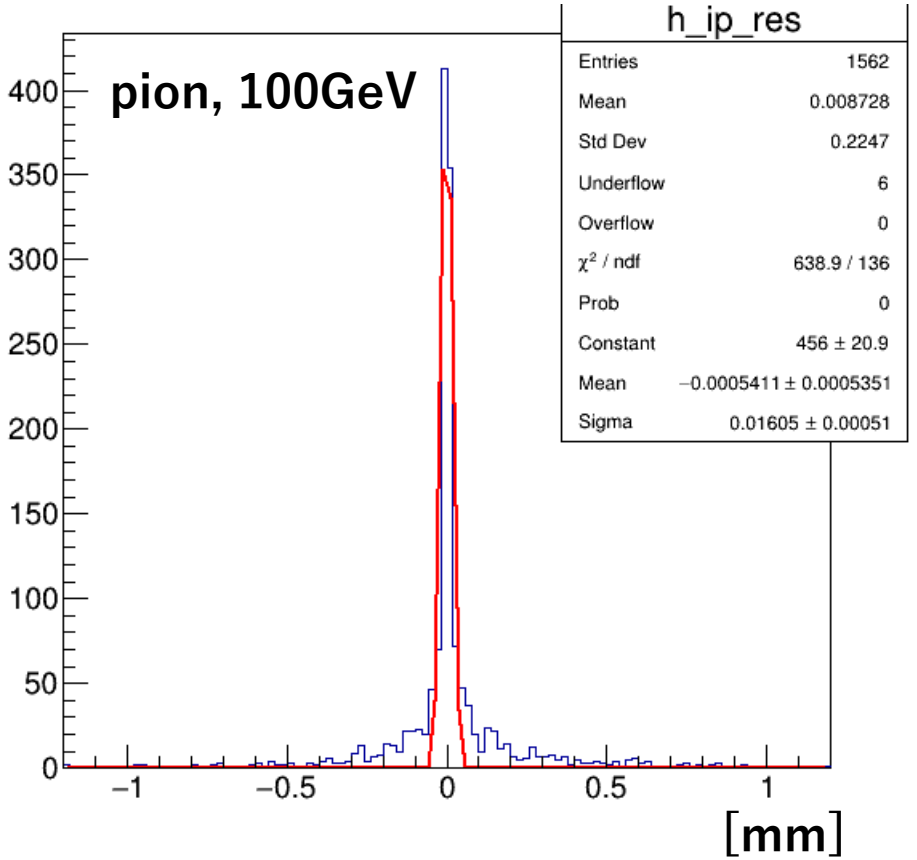
$$\gamma^4 + A\gamma^3 + B\gamma^2 + C\gamma + D = 0 \qquad (10)$$

d0 (=for $\sigma_{r\phi}$) distribution

・ Assuming that initial position of the injected particle is the origin = (0.0, 0.0, 0.0) so that obtained d0 is directly shown without any position subtraction

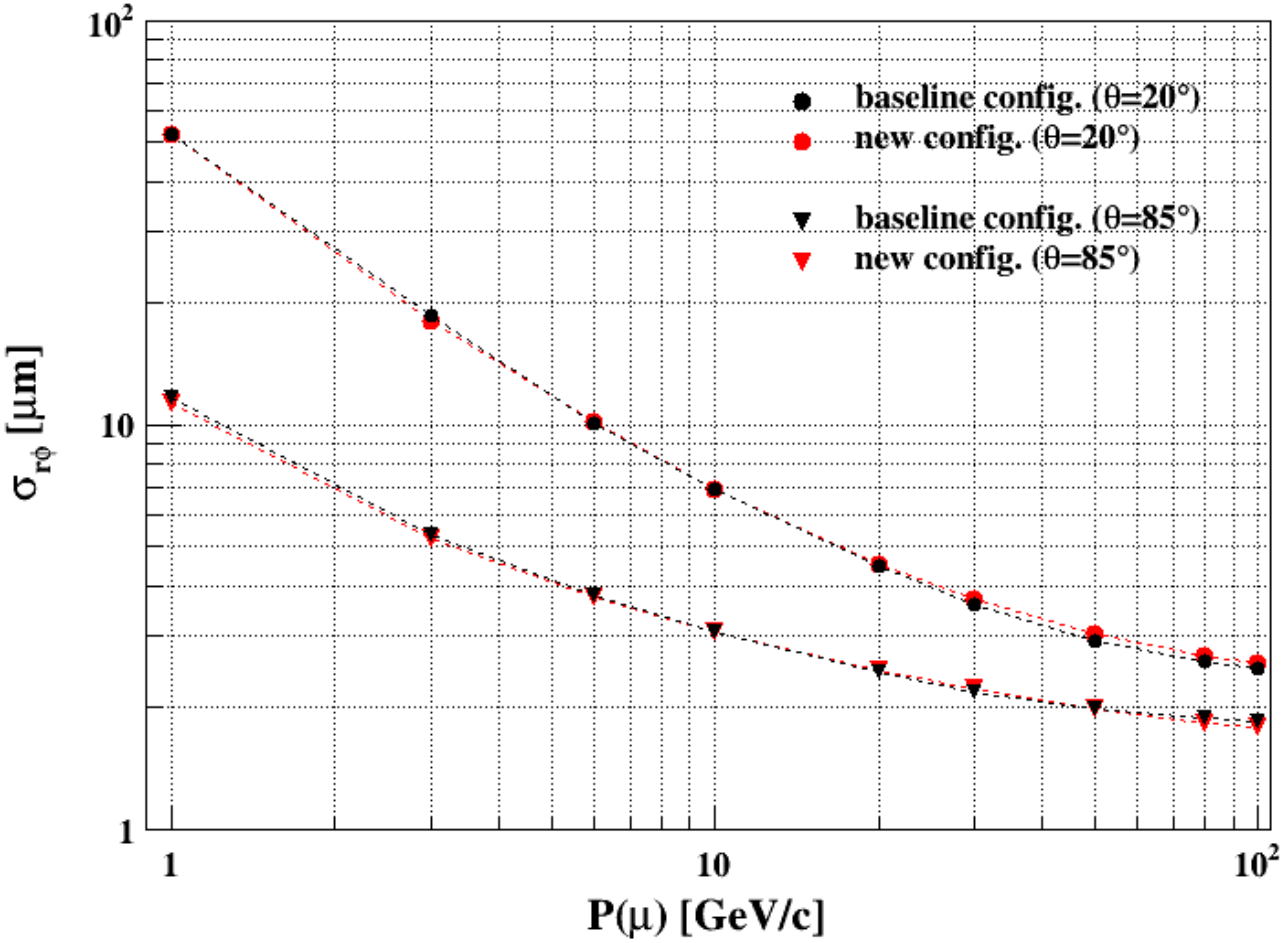


σ=294μm



σ=16μm

Ref: IP resolution from the LDT



The values in previous page are much worse. Need to include VTX hits etc. as well as tracking routine

Next steps

- Need further checks (actually going further to have numbers which can be compared with references…)
- Include VTX (& SIT/SET) hits which are also stored in the rootfiles
- Momentum resolution

  (Pt would be easy, need Z coordinate info. for P ?)

- At the same time, could I ask helps/suggestions about the tracking available at the CEPCSW ?

An update (just made it,,, need to check again)

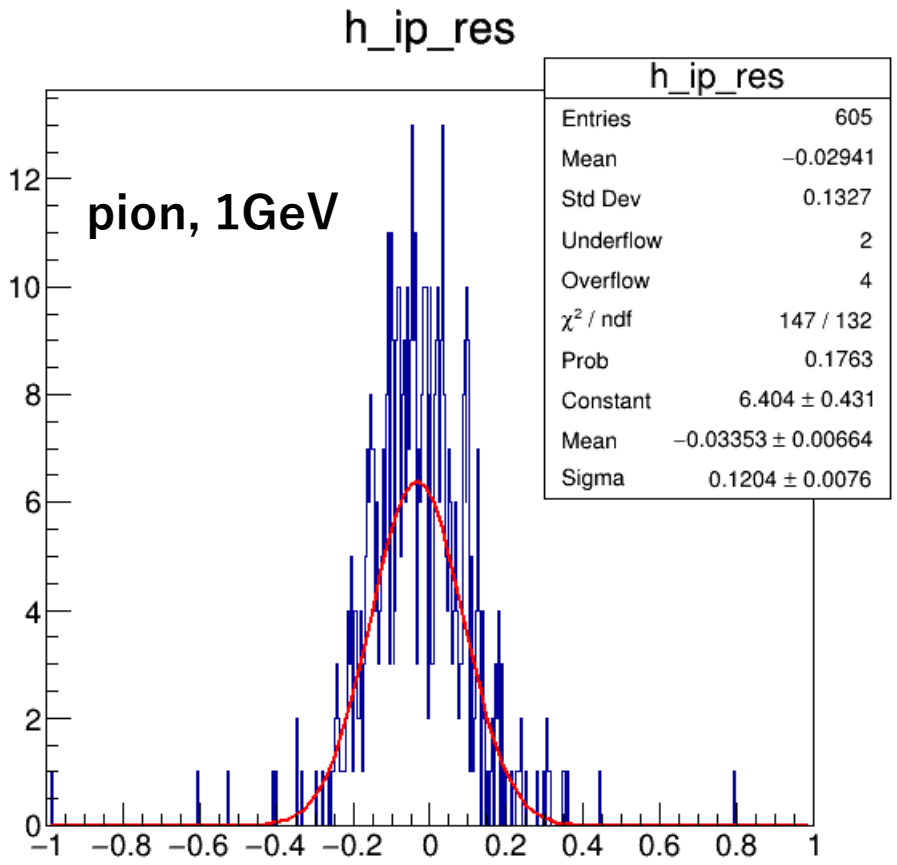
-- Add VXD/SIT/SET/DCH2 hits

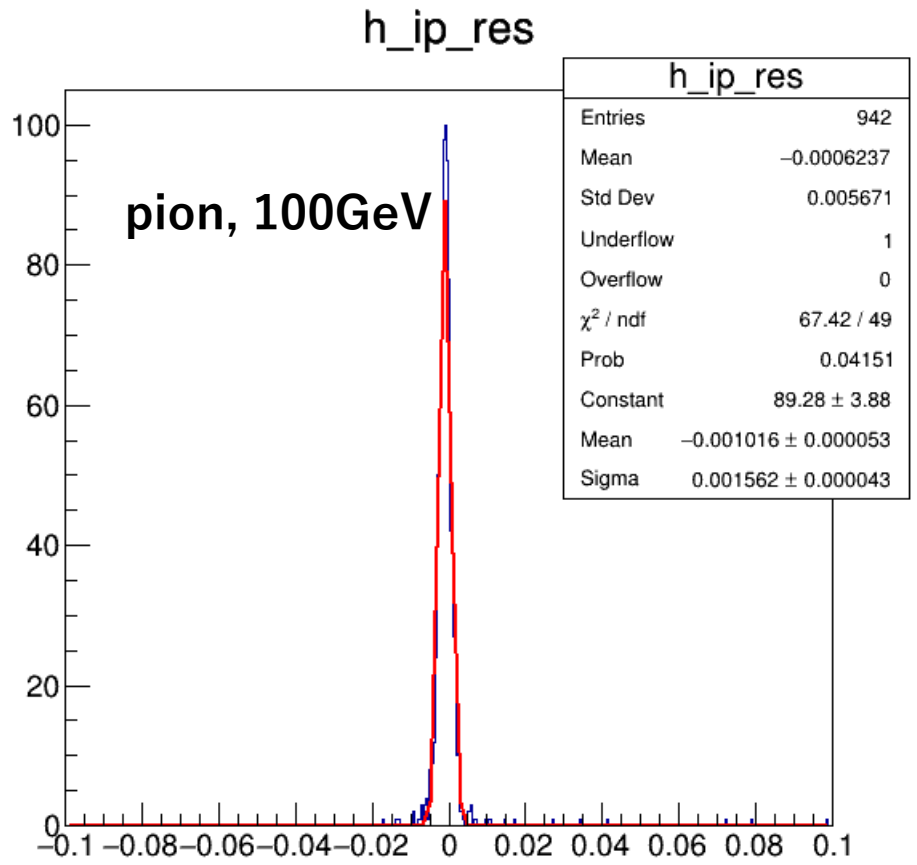
-- for the weight, w(VXD/SIT/SET) = $1/\sigma 2$, σ=10μm, w(DCH1/2) = $1/\sigma 2$, σ=0.2μm

-- events are selected as , Nhit(VXD)=6, Nhit(SIT)=4, Nhit(SET)=2, Nhit(DCH1/2) < 120

➔ ~50% (100GeV) -- 30%? (1GeV) was the efficiency of the events

d0 (=for $\sigma_{r\phi}$) distribution





σ=120μm

σ=1.6μm