Subject: Fix in lhetrack
Posted by StefanoSpataro on Thu, 10 Apr 2008 08:57:46 GMT
View Forum Message <> Reply to Message

Dear all,
I have found a problem in lhetrack code that gave me a crash while processing many events. In particular, the function PndTpcLheTrackCuts::Circle3pnts(...) which estimates from the first and the last point of the track candidate the centre and the radius of the circle crossing those two points and the origin (0,0).

You can see below the old version in PndTpcLheTrackCuts.cxx

Toggle Spoiler

```
void PndTpcLheTrackCuts::Circle3pnts(Double_t x[],Double_t y[], Double_t r[]) {
  // calc center and R of circle from 3 points

  Double_t m_a = (y[1] - y[0]) / (x[1] - x[0]);
  Double_t m_b = (y[2] - y[1]) / (x[2] - x[1]);

  if (m_a != m_b) {
    Double_t x_c = .5*(m_a* m_b*(y[0] - y[2]) +
        m_b*(x[0] + x[1]) -
        m_a*(x[1] + x[2])) / (m_b - m_a);

    Double_t y_c = - (x_c - .5*(x[0] + x[1])) / m_a +
      .5*(y[0] + y[1]);

    Double_t dely = y_c - y[0];
    Double_t delx = x_c - x[0];
    Double_t rad =TMath::Sqrt(delx*delx + dely*dely);

    r[0] = x_c;
    r[1] = y_c;
    r[2] = rad;
  }

}
```

As you can see, the function is not protected agains division by zero (so when x[1]-x[0]==0 or x[2]-x[1]==0). Moreover, the case m_a==m_b (the three points lay on a straight line) does not give results.

I have corrected it in the repository in the following way:

Toggle Spoiler

```
void PndTpcLheTrackCuts::Circle3pnts(Double_t x[],Double_t y[], Double_t r[]) {
  // calc center and R of circle from 3 points

  Double_t x_c = 0., y_c = 0.;

  if ( (y[1] - y[0])*(x[2] - x[1]) == (x[1] - x[0])*(y[2] - y[1]))
```

```
   {
     cout << " -W- PndTpcLheTrackCuts::Circle3pnts: The points lay on a straight line" << endl;
     x_c = 10000.;
     y_c = 10000.;
   }
 else
   {
     if ((x[1] - x[0])!=0 && (x[2] - x[1])!=0)
{
   Double_t m_a= (y[1] - y[0]) / (x[1] - x[0]);
   Double_t m_b = (y[2] - y[1]) / (x[2] - x[1]);

  x_c = .5*(m_a* m_b*(y[0] - y[2]) +
     m_b*(x[0] + x[1]) -
     m_a*(x[1] + x[2])) / (m_b - m_a);
  y_c = - (x_c - .5*(x[0] + x[1])) / m_a +
    .5*(y[0] + y[1]);
}
     else
{
   Double_t m_a = (x[1] - x[0]) / (y[1] - y[0]);
   Double_t m_b = (x[2] - x[1]) / (y[2] - y[1]);

  y_c = .5*(m_a* m_b*(x[0] - x[2]) +
     m_b*(y[0] + y[1]) -
     m_a*(x[1] + x[2])) / (m_b - m_a);
  x_c = - (y_c - .5*(y[0] + y[1])) / m_a +
    .5*(x[0] + x[1]);
}

   } // end of not collinear points condition

 Double_t dely = y_c - y[0];
 Double_t delx = x_c - x[0];

 r[0] = x_c;
 r[1] = y_c;
 r[2] = TMath::Sqrt(delx*delx + dely*dely);
}
```

If the three points lay on a straight line, the circle centre is set at (10000,10000) just to put a high number, probably the code will kill those tracks or be able to recover them.
If there is the division by zero, I have simply exchanged x->y y->x (in this system there is no division by zero anymore), calculated the centre coordinates and re-transformated them into the original system.

I hope I have not messed up things, however now I do not have the crash anymore (or at least there).

Comments and checks are welcome.