Exercise 4: A tracking detector

N. Berger (nberger@physi.uni-heidelberg.de)

31.10.2011

Please send your solutions to `nberger@physi.uni-heidelberg.de` until 7. 11. 2011, 12:00. Put your answers in an email (subject line *SMIPP:Exercise04*). Test macros and programs before sending them off...

1. **Simulating a tracking detector** In this exercise, we are going to simulate a simple particle tracking detector in 2D. Particles start at $x = 0$, $y = 0$ and propagate towards positive x. There are n thin layers of tracking stations at 2 cm intervals. Prepare a setup with ajustable n (start with $n = 5$) where you propagate the track from layer to layer and store the position of each intersection in a `TGraph` object (use the `SetPoint()` method. Also generate a detector resopnse (a *hit*) at the intersection and store it in a `TGraphErrors` (setting the error to 0 for now). After traversing all layers, draw the both the `TGraphErrors` (with option `"A*"`) and the `TGraph` (with option `"L"`), then fit the hits with a straight line (create a `TF1` object with `"[0]+[1]*x"` as the formula and call `TGraphErrors::Fit(TF1*)`. You can get the fit results via `TF1::GetParameter(int index)`. Make sure they are what you expect.

2. **Finite resolution** In a real detector, hit positions are always measured with a finite resolution. Take your code from above and smear every hit in the detector plane (y) with a normal distribution with a σ of 2 mm (you can use `TRandom3::Gaus(double mean, double sigma)` for this). Fill the results into the `TGraphError` with the σ of the normal distribution as the error in y (use `SetPointError()`. Fit the `TGraphError` as above.

3. **Tracker resolution** Now skip the drawing part of your code and generate 1000 tracks, and fill the fit parameters into a histogram. What do you get? How does this change if you increase n?

4. **Multiple scattering** A real detector will be made of a finite amount of material, leading to multiple scattering of particles in the detector. The precise simulation of multiple scattering is a very difficult problem, but the core of the angular distribution is well approximated by a normal distribution with a σ of

$$\sigma_{MS} = \frac{13.6\text{MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \ln \frac{x}{X_0}\right), \qquad (1)$$

where β is the fraction of the speed of light c of the particle speed, p the particle momentum, z the magnitude of its charge and $\frac{x}{X_0}$ the thickness of the detctor in radiation lenghts. Assume that for the particles and detector under consideration, σ_{MS} is 0.05 radian. Again propagate your track, this time also dicing a multiple scattering angle at every detector plane. What does this do to your resolution? To the n dependence of the resolution? (Attach one .C or .py file for the whole exercise (maybe you implement the detector as a class) and a few representative plots.)