Exercise 5: A tracking detector with particle ID

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

12.11.2011

Please send your solutions to nberger@physi.uni-heidelberg.de until 19. 11. 2011, 12:00. Put your answers (including plots, no (zip-) archives) in an email (subject line *SMIPP:Exercise05*). Test macros and programs before sending them off...

1. **Simulating a tracking detector - error propagation** Take the tracking detector class from exercise 4 (if you do not like your own, use the one from the solution of exercise 4, available on the website). Take a detector with $n = 5$ layers. We now want to extrapolate the particle to another detector located at $x = 30$ cm. First, expand the simulation to intersect the particle with a plane at $x = 30$ cm after scattering from the last detector plane. Then calculate the intersection with that plane from fit parameters, including an error. The slope and the intersect coming out of the fit are correlated, so you have to take their covariance into account. When you call the `Fit(fitfun,"S")` function (you have to use the `"S"` option) on the `TGraphErrors`, a `TFitResultPtr` is returned, which in turn gives you access to the covariance matrix via `TMatrixDSym cov = r->GetCovarianceMatrix();`. You can access the matrix elements via the bracket operators (`cov[i][j]`, where the diagonal elements are the squared variances and the off-diagonal elements the covariances for parameters i and j.

   With multiple scattering in the detector switched off, produce a resolution plot for the intersection. Then also produce a pull plot, where you divide the resolution by its error. What do you get? How important is the correlation term? What happens if you switch the multiple scattering on? (Attach a .C or .py file and suitable plots)

2. **Simulating a tracking detector - particle identification** We would now also like to simulate a detector for particle identification. The trick with particle ID is that the energy loss in material depends on the particle velocity β, so if the momentum p can be measured (by deflection in a magnetic field) separately, the particle mass can be determined. The mean energy loss in material (in MeV $\mathrm{g}^{-1}\mathrm{cm}^2$) is described by the Bethe (often also called Bethe-Bloch) formula

$$\frac{dE}{dx} = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \left(\frac{2m_e c^2 \beta^2 T_{max}}{I \cdot (1-\beta^2)} \right) - \beta^2 \right], \tag{1}$$

   where E is the particle energy, x the distance travelled, K is 0.307075 MeV g^{-1} cm^2, Z and A are the atomic number and atomic mass of the absorber, c is the speed of light, T_{max} is is the maximum kinetic energy which can

be imparted to a free electron in a single collision and I is the maximum ionisation energy (in eV). The distribution of the energy loss is described by the highly asymmetric Landau distribution (the most probable value is much smaller than the mean). The energy loss of a Kaon at a momentum of 500 MeV is roughly twice the energy loss of a Pion with the same momentum.

Assume that in your toy detector in suitable units the energy loss distribution of Pions is described by a Landau distribution of most probable value 1 and "sigma" 0.3 and the Kaon energy loss distribution is described by a most probable value of 2 and a "sigma" of 0.4. Use `TRandom3::Landau(mean,sigma)` to generate and plot these two distributions. How well can you separate Kaons from Pions? How does this change if you form the mean of the energy loss over $n = 5$ layers? How does this change if you throw away the measurement with the largest measured energy loss (this is called a truncated mean)? What happens if you also throw away the measurement with the smallest energy loss? What happens if your detector has a (Gaussian) resolution for the individual energy loss measurement of 0.1? Study the behaviour of these estimators for the RMS of the pion $\frac{dE}{dx}$ depending on the number of layers n.
(Attach a .C or .py file and suitable plots)